## Evolving an efficient and effective off-the-shelf computing infrastructure for schools in rural areas of South Africa

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

of

Rhodes University

by

Ingrid Gisélle Siebörger

February 2017

## Abstract

Upliftment of rural areas and poverty alleviation are priorities for development in South Africa. Information and knowledge are key strategic resources for social and economic development and ICTs act as tools to support them, enabling innovative and more cost effective approaches. In order for ICT interventions to be possible, infrastructure has to be deployed. For the deployment to be effective and sustainable, the local community needs to be involved in shaping and supporting it.

This study describes the technical work done in the Siyakhula Living Lab (SLL), a long-term ICT4D experiment in the Mbashe Municipality, with a focus on the deployment of ICT infrastructure in schools, for teaching and learning but also for use by the communities surrounding the schools. As a result of this work, computing infrastructure was deployed, in various phases, in 17 schools in the area and a "broadband island" connecting them was created.

The dissertation reports on the initial deployment phases, discussing theoretical underpinnings and policies for using technology in education as well various computing and networking technologies and associated policies available and appropriate for use in rural South African schools. This information forms the backdrop of a survey conducted with teachers from six schools in the SLL, together with experimental work towards the provision of an evolved, efficient and effective off-the-shelf computing infrastructure in selected schools, in order to attempt to address the shortcomings of the computing infrastructure deployed initially in the SLL. The result of the study is the proposal of an evolved computing infrastructure model for use in rural South African schools.

# Contents

1	Setti	ing the sce	ne	1
	1.1	Backgrou	nd	1
	1.2	The Siyal	shula Living Lab	3
	1.3	The probl	em statement	4
	1.4	Research	design	4
		1.4.1 R	esearch aims and outcomes	4
		1.4.2 R	esearch questions	5
		1.4.3 R	esearch Methodology	5
		1.4.4 D	elimitation of Scope	6
	1.5	Structure	of thesis	7
2	Rese	earch meth	nodology and design	9
	2.1	Introducti	ion	9
	2.2	Ontologic	cal Framework	9
	2.3	Methodol	ogical Approach: The mixed methodology	10
		2.3.1 A	ction research	1
		2.3.2 T	he living lab methodology	12
		2.3.3 L	ong-term experiments	14
		2.3.4 C	ase studies	15
	2.4	Research	goals and questions	16
		2.4.1 R	esearch aims	16
		2.4.2 R	esearch questions	16
	2.5	Research	environment	17
		2.5.1 R	esearch site: Dwesa, Mbashe Municipality	17
		2.5.2 S	outh African school classifications	20
		2.5.3 R	esearch participants	22
	2.6	Research	tools	23
		2.6.1 St	urveys	24
		2.6.2 C	omputer laboratory experimentation	27
		2.6.3 O	n-site deployment and long-term follow-up (field experiments)	27
	2.7	Research	activities	28
		2.7.1 Q	uestionnaire	28
		2.7.2 F	ocus groups and Interviews	29
		2.7.3 L	ab experiments	30
		2.7.4 C	computing infrastructure deployments	30
	2.8	Ethical is:	sues	31
	2.9	Data anal	ysis	32

		2.9.1 Qualitative data and analysis					
		2.9.2 Quantitative data and analysis					
	2.10	Conclusion					
3	Com	puting infrastructure deployment: the initial phases 39					
	3.1	Introduction					
	3.2	The Siyakhula Living Lab 39					
		3.2.1 Mbashe Municipality					
		3.2.2 ICT infrastructure					
		3.2.3 SLL network: The broadband island					
		3.2.4 Computing infrastructure in the SLL					
	3.3	Conclusion					
4	Und	erstanding ICTs in Education 54					
	4.1	Introduction					
	4.2	Use of technology in Education					
	4.3	ICTs for education in developing countries					
	4.4	Affordances					
	4.5	ICT Integration, Pedagogical Practices and Policy					
	4.6	TPACK					
	4.7	Conclusion					
5	ICT infrastructure for Education in rural South Africa 76						
	5.1	Introduction					
	5.2	Hardware					
		5.2.1 Computing clients					
		5.2.2 The digital divide and the possible use of refurbished computers					
		5.2.3 Grid computing					
		5.2.4 Cluster computing					
		5.2.5 Cloud computing					
		5.2.6 Desktop Virtualization (Virtual Desktop Infrastructure, VDI)					
	5.3	Software					
		5.3.1 Edubuntu					
		5.3.2 Linux Terminal Server Project					
		5.3.3 Network File Systems					
		5.3.4 LDAP					
		5.3.5 Domain Name System					
		5.3.6 Sikuli					
	5.4	Connectivity					
		5.4.1 SA Connect: The SA Broadband Policy					
		5.4.2 Broadband Islands					
		5.4.3 IEEE 802.11 (WiFi)					
		5.4.4 IEEE 802.16 (WiMAX) 105					
		5.4.5 LTE					
		5.4.6 Satellite Communications: VSAT					
	5.5	Conclusion					

6	Finc	dings from the Siyakhula Living Lab survey 11	4
	6.1	Introduction	4
	6.2	Questionnaire results	4
		6.2.1 Demographics	4
		6.2.2 ICT facilities	5
		6.2.3 ICT use	9
		6.2.4 ICT maintenance, management and utilization	2
		6.2.5 Summary	2
	6.3	Interviews and Focus groups results	4
		6.3.1 In Vivo coding	4
		6.3.2 Infrastructure	6
		6.3.3 TPACK coding	1
		6.3.4 Affordances	5
	6.4	Discussion	6
	6.5	Conclusion	8
-	C		•
/		Introduction 12	9 0
	7.1	Introduction	9
	7.2 7.2	Current computing infrastructure model in the SLL. A feedp	9 0
	7.5	7.2.1 DNS Bound Bokin	1
		7.5.1 DINS Round Robin	1 2
		7.5.2 Obuntu LISP Cluster	2 1
		7.5.5 Single System Image	4
		7.5.4 Virtual Desktop	07
	7 4	7.5.5 Edubuntu Fat chents	7
	7.4 7.5	Sikuli Scripts	1 0
	1.5	Cluster server vs standard thin client server laboratory testing	0
		7.5.1 Single cluster node as thick client	0
		7.5.2 Single node with one client connected	1
		7.5.3 Single node, heavily loaded $\dots$ 15	1
		7.5.4 Pentium 4 cluster with 20 clients spread across them (DNS Round Robin) 15	2
		7.5.5 Quad core single LISP server with 12-14 clients	2
	7.6	Cluster server laboratory testing 15	3
		7.6.1 Standard Thin Client Configuration	3
		7.6.2 Fat Client Configuration	5
		7.6.3 DNS Round Robin (Dual core atom cluster)	7
	7.7		9
	7.8	Conclusion	0
8	Con	nputing infrastructure deployment: evolved model 16	1
	8.1	Introduction	1
	8.2	DNS Round Robin deployment to FH5	1
		8.2.1 Configuration	1
		8.2.2 MRTG results	4
	8.3	Edubuntu Fat/Hybrid client deployment to FH1	0
		8.3.1 Configuration	0
		8.3.2 MRTG results	3

	8.4	Standa	rd thin client deployment to FH2	175
		8.4.1	Configuration	175
		8.4.2	MRTG Results	177
	8.5	Discus	sion: lessons learned from the case studies	179
		8.5.1	DNS Round Robin solution	179
		8.5.2	LTSP Fat/Hybrid client solution	180
		8.5.3	Standard Edubuntu thin client solution	181
	8.6	Compu	ting infrastructure for rural South African schools: a revised model	181
	8.7	Conclu	sion	185
9	Con	clusion	and recommendations	186
	9.1	Summa	ary	186
	9.2	Resear	ch questions revisited	188
		9.2.1	What is the currently deployed computing infrastructure and computer networks in the schools in the SLL?	188
		922	What current and future computing infrastructures (that are robust easy to main-	100
		1.2.2	tain financially sustainable and nedagogically appropriate) are available for use in	
			South African rural schools?	102
		023	What are the technological based issues that the school teachers face in introducing	192
		9.2.3	the use of technology into their teaching practices in the SLL?	102
		024	What are the pedagogical issues that the school teachers face in introducing the use	172
		Э.2.т	of technology into their teaching practices in the SLL?	103
		025	How do clustered LTSP servers and fat/hybrid client configurations compare (in	195
		9.2.5	terms of performance, reliability, cost and ease of maintenance) with the standard	
			thin client configuration in the school environment in the SLL?	194
	9.3	Conclu	sion and recommendation	196
	9.4	Resear	ch contribution	197
	9.5	Future	work	197
	9.6	Parting	thoughts	198
Li	st of H	Referenc	ces	199
A	SLL	Survey	: Permission letter to District office	217
B	SLL	Survey	: Permission letter to school Principal	221
С	SLL	Survey	: Consent forms	225
	C.1	School	Principal consent form	225
	C.2	Teache	rs' consent form	226
D	SLL	Survey	: Questionnaire	228
-	<del>~ -</del> -	, ,		<b>.</b>
Ε	SLL	Survey	: Interview Schedule	235
F	Stan	dard E	dubuntu LTSP Thin client configuration	238
	F.1	Installa	tion and configuration	238
	F.2	How Ľ	TSP thin clients work in practice in Edubuntu	239
G	Edu	buntu L	TSP Fat (hybrid) client configuration	241

Η	DNS	Round Robin Cluster configuration	243
	H.1	The NFS installation/configuration, server-side	243
	Н.2	The NFS installation/configuration, client-side	243
	Н.3	OpenLDAP Installation/Configuration, server-side	244
	H.4	OpenLDAP Server GUI applications	247
		H.4.1 Jxplorer	248
		H.4.2 $GOsa^2$	248
		H.4.3 phpLDAPadmin	249
		H.4.4 Webmin	249
	H.5	OpenLDAP installation/configuration, client-side	250
	H.6	Bind installation/configuration	251
	H.7	Putting it all together	255
I	Ubu	ntu LTSP Cluster configuration	257
	I.1	Ubuntu LTSP Cluster installation and configuration	257
		I.1.1 Root server for LTSP-cluster	257
		I.1.2 Application server(s) for LTSP-cluster	262
		I.1.3 LTSP-cluster and NAT	263
		I.1.4 LTSP-cluster and OpenLDAP	263
		I.1.5 LTSP-cluster and NFS	266
	I.2	The difference between the DNS Round Robin and Ubuntu LTSP-cluster solutions	267
J	Sing	le System Image using Kerrighed configuration	269
	J.1	Setting up NFS	269
	J.2	Setting up Kerrighed	271
K	Virtu	ual Desktop using Edubuntu and VirtualBox configuration	276
	<b>K</b> .1	Desktop virtualization: Installation and configuration	276
L	Acco	ompanying dissertation DVD	281

# **List of Figures**

1.1	Structure of this thesis	8
2.1 2.2	Key components of a Living Lab as described by Bergvall-Kåreborn and Ståhlbröst [1] Maps of Dwesa in relation to the rest of South Africa (a) A Google Earth map with the Dwesa area marked inside the red box (b) A map of the Wild Coast with the Dwesa nature	13
	reserve marked	19
2.3 2.4	Organogram of the South African Education System during the Apartheid era, taken from [2] Screen capture of the transcript of Interview 1, showing the data classifications and the	21
	colour coded TPACK classifications	35
3.1	Topographical map of SLL area around Ngwane and Mpume. The three additional schools that were added to the network later are also marked, these are, Nondobo, Mthokwane and	
	Nqabara	43
3.2 3.3	The original SLL network diagram configuration	44
	connected to either Ngwane or Badi	46
3.4	Logical network diagram of the SLL network as of August 2016	49
3.5	The original design of a generic computing infrastructure (typically as a computer lab) in a	50
3.6	MRTG graphs of the load experienced by a single central server, serving 20 thin clients, at Ngwane over the period of one month (this server was in place from 2009 to early 2013). (a) CPU utilization over four weeks (b) network interface utilization over four weeks (c)	52
	memory utilization over four weeks	52
4.1 4.2	TPACK Framework as described by Harris, Mishra and Koehler in [3]	70
	from Mishra and Koehler [4]	71
5.1	Thick client computers; some of which connected to a central server for data backup and	
	retrieval others stand-alone	78
5.2 5.3	Thin client computers, all of which connected to the central server in order to operate Overview of the system architecture for Wikipedia as of December 2010, as sourced from	78
	Wikipedia [5]	85
5.4 5.5	Results from a DNS server query requesting the A-records for www.google.co.za One of Google's geographically dispersed clusters of commodity computers, sourced from	87
	[6]	87
5.6	The relationships among middleware modules in SSI Systems [7]	89

5.7	: Cloud ontology from [8], the "layered figure represents the inter-dependency and com-	
	posability between the different layers in the cloud" [8]	92
5.8	Cloud Computing Conceptual Stack, taken from [9]	92
5.9	Desktop virtualization breaks traditional bindings between layers, taken from [10]	94
5.10	How the different desktop virtualization options provide different benefits, taken from [10]	95
5.11	Typical NFS client and server arrangement and communication, adapted from [11]	98
5.12	File system illustration taken from [12]	99
5.13	EPS architecture, taken from [13]	109
5.14	Distribution of the functionality within an LTE network, taken from [13]	110
6.1	Number of Teachers per age group	115
6.2	Number of years teaching by participants	116
6.3	Software available to and used by teachers interviewed in the SLL	118
6.4	Provision of ICT support at the schools	119
6.5	How teachers are making use of their ICT facilities at school	120
6.6	Breakdown of when learners have access to ICT facilities at school	121
6.7	Ease of ICT maintenance according to the teacher participants	122
6.8	Reported levels of ICT utilization at schools by teachers and learners	123
6.9	Mind map of the categories and sub-categories identified through In Vivo coding of the	
	interviews and focus groups with the SLL teachers	125
6.10	Mind map of the Infrastructure category with its sub-categories and sub-sub-categories .	126
6.11	Spreadsheet of interview data categories vs the TPACK framework knowledge types and	
	context	132
6.12	Total number of comments per TPACK knowledge types and context	132
6.13	Pie chart of the Technological Knowledge (TK) classification of the in vivo categories;	
	representing the non-zero categories percentage as part of the whole	134
6.14	Affordances data analysis results from interviews and focus groups with teachers in the six	
	selected SLL schools	136
7.1	Diagrammatic representation of the LTSP DNS Round Robin cluster	142
7.2	LTSP Cluster implementation [14]	144
7.3	SSI Cluster with Ubuntu NFS server (with file system and storage) and Edubuntu Kerrighed	
	Linux cluster nodes, serving thin client computers	145
7.4	Sikuli Firefox test results on a standard thin client configuration with single central server	
	(a) "Cleaned" data set (b) Full data set including times of crashed clients	154
7.5	Hardware resources utilization on the standard thin client server (a) CPU utilization (b)	
	Free Memory available	155
7.6	Sikuli LibreOffice test results on a Fat client configuration with a single central server (a)	
	"Cleaned" data set (b) Full data set including the times of the crashed clients	156
7.7	Hardware resource utilization on the central server serving in the fat client configuration (a)	
	CPU utilization (b) Free Memory available	157
7.8	Hardware utilization on single fat/hybrid client image (a) CPU utilization (b) Free memory	
	available	157
7.9	Hardware utilization on the cluster's File, LDAP and DHCP server (a) CPU utilization (b)	
	Free memory available	158
7.10	Hardware utilization on the DNSNode of the LTSP cluster server (a) CPU utilization (b)	
	Free memory available	158

7.11	Hardware utilization on GenNode1 of the LTSP cluster server (a) CPU utilization (b) Free memory available	158
7.12	Hardware utilization on the GenNode2 of the LTSP cluster server (a) CPU utilization (b)	156
7 12	Free memory available	159
7.15	Free memory available	159
8.1	Computing and networking infrastructure configuration at FH5	163
8.2	MRTG data for Node1 reflecting the CPU load, Network traffic, Free Memory (load on the	
	memory) and the number of processes on Node1	165
8.3	MRTG data for Node2 reflecting the CPU load, Network traffic, Free Memory (load on the	
	memory) and the number of processes on Node2	166
8.4	MRTG data for Node3 reflecting the CPU load, Network traffic, Free Memory (load on the	
	memory) and the number of processes on Node3	167
8.5	MRTG data for Node4 reflecting the CPU load, Network traffic, Free Memory (load on the	
	memory) and the number of processes on Node4	168
8.6	MRTG data for Node5 reflecting the CPU load, Network traffic, Free Memory (load on the	
	memory) and the number of processes on Node5	169
8.7	MRTG data of (a) CPU loads and (b) Free Memory on the Five Cluster Nodes	171
8.8	Computing and networking infrastructure configuration at FH1	172
8.9	MRTG data for FH1's server; reflecting the CPU load, Network traffic, Free Memory (load	
	on the memory) and the number of processes running	174
8.10	Memory utilization on the Fat/Hybrid client server at FH1 over three days at the start of the	
	2017 school year	175
8.11	Computing and networking infrastructure configuration at FH2	176
8.12	MRTG data for FH2's server; reflecting the CPU load, Network traffic, Free Memory (load	
	on the memory) and the number of processes running	178
9.1	The original design of a generic computing infrastructure (typically as a computer lab) in a	
	DAN at a school in the SLL	189
<b>K</b> .1	Virtualbox Manager Window	277
K.2	Virtualbox Settings Window	278

# **List of Tables**

2.1	School and teacher participant codes	23
3.1	Alvarion BreezeMAX 3500 µbs Modulation Schemes	43
3.2	Alvarion BMAX OD MICRO 3.5 µbs 4Motion Series Modulation Schemes	45
4.1 4.2	SA Broadband Policy targets (as per SA Connect [15]) Extract from Harris, Mishra and Koehler [23, p408], activity types and their associated	66
	technologies	74
5.1	Features of a SSI cluster systems (from [16, 7])	90
6.1	Number of teachers using computers for teaching purposes per school	120
6.2	Technological Knowledge classified comments per in vivo category	133
7.1	Similarities and differences between Fat and Thin clients [17]	148
7.2	Timing results of Firefox and OpenOffice.org across the various hardware configurations .	152
7.3	Timing values of Quad core single LTSP server	153
7.4	Sikuli Firefox and Sikuli LibreOffice tests average times and standard deviation per cluster	
	node	157
8.1	Average maximum Free Memory available on the cluster Nodes (in the year 2015) as well as the average maximum Free Memory available on the single central server (in the first	
	three months of 2013) (all deployed at FH5)	179
9.1	Computing infrastructure of the six SLL participant schools in this study	191

## **List of Abbreviations**

μbs	Micro Base Station
AC	Access Concentrator
ACE	Advanced Certificate in Education
AMPS	Advanced Mobile Phone System
AP	Access Point
ASN	Access Service Network
BIND	Berkeley Internet Name Domain
BPSK	Binary Phase Shift Keying
BS	Base Station
BSS	Basic Service Set
BWA	Broadband Wireless Access
Caas	Communication as a Service
CAP	Community Access Point
CAT	Computer Application Technology
CCK	Complementary Code Keying
CDMA	Code Division Multiple Access
СК	Content Knowledge
CoE	Centre of Excellence
CPE	Customer Premises Equipment
CPU	Central Processing Unit
CSAIL	Computer Science and Artificial Intelligence Laboratory

- CSAR Computer Services for Academic Research
- CSMA/CA Carrier Sense Multiple Access protocol with Collision Avoidance
- Daas Database as a Service

- DAN Digital Access Node
- DFT-S-OFDM Discrete Fourier Transform-Spread-OFDM
- DHCP Dynamic Host Configuration Protocol
- DIT Directory Information Tree
- DNS Domain Name Service
- DoBE Department of Basic Education
- DoC Department of Communications
- DRBL Diskless Remote Boot in Linux
- DSL Digital Subscriber Line
- DSSS Direct Sequence Spread Spectrum
- DTN Delay Tolerant Networks
- DTPS Department of Telecommunications and Postal Services
- EDGE Enhanced Data Rates for GSM Evolution
- EDO Education District Officer
- EPS Evolved Packet System
- ESS Extended Service Set
- ETSI European Telecommunications Standards Institute
- EV-DO Evolution-Data Optimised
- FBWA Fixed Broadband Wireless Access
- FDD Frequency Division Duplex
- FDET Former Department of Education and Training
- FDM Frequency Division Multiplexing
- FDMA Frequency Division Multiple Access
- FET Further Education and Training
- FFT Fast Fourier Tranform
- FH Former Homeland Schools
- FHOA Former House of Assembly
- FQDN Fully Qualified Domain Name
- GEO Geostationary Earth Orbiting
- GET General Education and Training

- GGF Global Grid Forum
- GIG Grid Infrastructure Group
- GPRS General Packet Radio Service
- GSM Global System for Mobile communications
- GUI Graphical User Interface
- HaaS Hardware as a Service
- HPC High Performance Computing
- HPF High Performance Fortran
- HSDPA High Speed Downlink Packet Access
- HSS Home Subscriber Server
- HSUPA High Speed Uplink Packet Access
- HTTP Hyper Text Transfer Protocol
- I-WAY Information wide-area year
- IaaS Infrastructure as a Service
- ICT Information Communication Technology
- ICT4D Information Communication Technology for Development
- IETF Internet Engineering Task Force
- IFFT Inverse Fast Fourier Tranform
- IP Internet Protocol
- IPC Inter-process Communication
- ISDN Integrated Services Digital Network
- ISM Industrial, Scientific and Medical band
- ISO International Organisation for Standardisation
- ISPs Internet Service Providers
- IT Information Technology
- ITU International Telecommunications Union
- JSS Junior Secondary School
- LAN Local Area Network
- LCT Ligitimate Code Theory
- LDAP Lightweight Directory Access Protocol

- LDIF LDAP Data Interchange Format
- LDM LTSP Display Manager
- LEO Low Earth Orbiting
- LLC Logical Link Control
- LOS Line of sight
- LTE Long-Term Evolution
- LTSP Linux Terminal Services Project
- MAC Medium Access Control
- MAN Metropolitan Area Network
- MEO Medium Earth Orbiting
- MIMO Multiple input, multiple output
- MME Mobility Management Entity
- MP MultiPoint
- MPI Message Passing Interface
- MRTG Multi Router Traffic Grapher
- NAS Network Attached Storage
- NBD Network Block Device
- NDP National Development Plan
- NEPI National Education Policy Investigation
- NFS Network File System
- NLOS Non Line of Sight
- NSF National Science Foundation
- OECD Organisation for Economic Co-operation and Development
- OFDM Orthogonal Frequency-Division Multiplexing
- OFDMA Orthogonal Frequency-Division Multiple Access
- OGSA Open Grid Services Architecture
- OLPC One Laptop Per Child
- OS Operating System
- OSI Open Systems Interconnection
- OSS Open Source Software

- P-GW Packet Date Gateway PaaS Platform as a Service PAR Participatory Action Research PBCC Packet Binary Convolutional Coding PC Personal Computer PCK Pedagogic Content Knowledge PHY Physical Layer PK Pedagogical Knowledge POTS Plain Old Telephone System PPP Point to Point Protocol PPPoE Point to Point Protocol over Ethernet PTMP Point to MultiPoint PVM Parallel Virtual Machine PXE Preboot execution environment QAM Quadrature Amplitude Modulation QPSK Quadrature Phase Shift Keying RAM Random Access Memory RF Radio Frequency RFCs Request for Comment **RNCS** Revised National Curriculum Statement RPC Remote Procedure Call RR Resource Record RS **Repeater Station** RTT Round Trip Time S-GW Serving Gateway SaaS Software as a Service SAE Service Architecture Evolution SAIDE South African Institute for Distance Education
- SAN Storage Area Network
- SASAMS South African School Administration and Management System

SC-FDMA Single Carrier - Frequency Division Multiple Access

- SLL Siyakhula Living Lab
- SMB Server Message Block
- SNR Signal to Noice Ratio
- SOA Service Orientated Architecture
- SOFDMA Scalable Orthogonal Frequency Division Multiple Access
- SPS Senior Primary School
- SRI Stanford Research Institute
- SS Secondary School
- SS Subscriber Station
- SSI Single System Image
- std dev Standard Deviation
- TACS Total Access Communication System
- TCK Technological Content Knowledge
- TDD Time Division Duplex
- TDM Time Division Multiplexing
- TDMA Time Division Multiple Access
- TFTP Trivial File Transfer Protocol
- TK Technological Knowledge
- TPACK Technological Pedagogical and Content Knowledge Framework
- TPK Technological Pedagogical Knowledge
- UE User Equipment
- UMB Ultra-Mobile Broadband
- UMTS Universal Mobile Telecommunication System
- VDI Virtual Desktop Infrastructure
- VoIP Voice over Internet Protocol
- VSAT Very Small Aperature Terminal
- W-CDMA Wideband Code Division Multiple Access
- WiMAX Worldwide Interoperability for Microwave Access
- WLAN Wireless Local Area Network
- WSDL Web Service Definition Language
- WWW World Wide Web

## Acknowledgements

Heartfelt thanks go to:

- to my supervisors, Prof Alfredo Terzoli (from the Department of Computer Science) and Prof Cheryl Hodgkinson-Williams (from the Centre for Innovation in Learning and Teaching at UCT) who guided, encouraged and assisted me throughout this study;
- David Siebörger, Kevin Duff, Barry Irwin, Sibukele Gumbo, Mitchell Kavhai, Hannah Thinyane and the numerous students from Rhodes and Fort Hare (over the years) for their assistance in deploying the SLL infrastructure;
- Hannah Thinyane and David Siebörger, for proof-reading this thesis;
- Prof Pat Terry, for always having faith in me;
- my family, friends and colleagues, for all their love, support and encouragement and the Lord for His provision;
- David, without your constant love and support I don't know that I would have made it this far;
- Tim and Becca, for reminding about what is really important in life; and
- the Centre of Excellence in Distributed Multimedia and its sponsors, who provided equipment and funding.

## **Chapter 1**

### Setting the scene

#### 1.1 Background

Worldwide, computers are being used in schools for, amongst other things, developing the knowledge and skills required for citizens to be able to operate within the 21st Century Information Age. It is argued that the role of education is to develop higher-order thinking, inquiry, collaborative problem solving and transmit culture, values and lessons of the past while preparing learners for the world in which they live; education is seen as the principal means of creating a productive and sustainable society. It is further asserted that nations need to be internationally competitive and thus require a well-educated work force; requiring knowledge workers who will compete in a global market [18, 19, 20]. In order to address this educational goal many governments around the world have put specific policies in place to encourage the use of Information Communication Technology (ICT) in education [21, 22, 23]. The use of ICTs in education "is now a political orthodoxy, seen by many politicians and educators as a ready means of widening participation to those social groups traditionally excluded from learning" [24, p258] and South Africa is no different [25].

Investment in ICTs in education may also be seen as an attempt to address the so-called "digital divide" [24]. With reference to developing countries, the World Bank suggested that "[t]his new technology greatly facilitates the acquisition and absorption of knowledge, offering developing countries unprecedented opportunities to enhance educational systems, improve policy formation and execution, and widen the range of opportunities for business and the poor" [26]. Plainly the use of computers has had a profound impact on our society; the digitisation of text, images, sound and video allows for easy access to multimedia content as well as easy communication with others from all over the world and easy access to information from anywhere in the world. As a result of these changes, Cornu proposes that two ideas emerge: that of the networked society and the networked collective intelligence, both impacting how we teach and learn in our information society [27].

In the past society has been organised in a hierarchical or tree structure – a pyramidal structure [27]. It is argued however that our society is changing, becoming a networked structure as a result of the development of networks and the structure of networks within society [27]. In this networked society, people and information are connected in a networked form, which results in a change in the relationships between people and between people and information [27]. This kind of society means that people are able to communicate with each other whether they know each other or not. Institutions accustomed to a tree-structured hierarchy will have to change, "they will have to evolve, adapt and take a networked form [...]. The case of education is particularly important. Since society is more and more networked, schools must prepare the learners for this networked society" [27, p42]. Consequently, the world is witnessing "the emergence and the development of a more collective form of intelligence" [27, p42]. Traditionally, intelligence has been

thought of as an individual and personal quality. Cornu [27] argues that schools have aimed at developing the individual intelligence of its learners. However, he suggests that society needs a form of collective intelligence, which he defines as not only the addition of intelligence from various people, but also the juxtaposition of those intelligences [27]. It is through the development of a network society that the idea of a collective intelligence emerges. Through the traditional tree structure hierarchies people are only able to form part of a collective intelligence through co-operation, transmission and hierarchy. But ICTs make new networks and new ways of communicating with people and with groups available. They create the possibility of a new form of collective intelligence, which Cornu believes "may in turn substantially increase the collective capacity and competence of human beings" [27, p43]. Cornu [27] feels that "preparing students for collective intelligence and related competencies will be a necessity in the networked society" [27, p44]. The development of collective intelligence in the information society would require the inclusion of computing infrastructure and computer networks in schools. The South African government has proposed the inclusion of computing infrastructure through the e-Education white paper [25] and the South African Broadband Policy, typically referred to as SA Connect [15].

According to the South African e-Education White paper every "South African learner in the general and further education and training bands will be ICT capable (that is, use ICTs confidently and creatively to help develop the skills and knowledge they need to achieve personal goals and to be full participants in the global community) by 2013" [25, p10]. Furthermore, "South Africa Connect, the national broadband policy and the associated strategy and plan, gives expression to South Africa's vision in the National Development Plan (NDP) of 'a seamless information infrastructure by 2030 that will underpin a dynamic and connected vibrant information society and a knowledge economy that is more inclusive, equitable and prosperous'. As envisaged in the NDP, at the core of this will be 'a widespread communication system that will be universally accessible across the country at a cost and quality that meets the communication [needs] of citizens, business and the public sector and provides access to the creation and consumption of a wide range of converged applications and services required for effective economic and social participation. This ecosystem of digital networks, services, applications, content and devices, will be firmly integrated into the economic and social fabric of the country. Together, it is hoped that these broadband elements provide an enabling platform for economic enterprise, active citizenship and social engagement and innovation. It [the proposed national network] will connect public administration to the active citizen; promote economic growth, development and competitiveness; drive the creation of decent work; underpin nation-building and strengthen social cohesion; and support local, national and regional integration.' (NDP 2012)" [15, p2]. Realising the potential of providing access to networks and the network society, the South African broadband plan proposes to prioritise schools and sets the target that 50% of schools will be connected by the end of 2016 at 10 Mbps, and all schools to be connected at 1 Gbps by 2030 [15].

Rural development is seen as a priority in South Africa; information and knowledge are key strategic resources for social and economic development. ICTs are seen as important tools in development, assisting in enabling change through economic development [28]. Policy makers in South Africa are concerned with bridging the digital divide [29] that exists between urban (high socio-economic areas) and rural (low socio-economic areas) regions of the country [30]. Organizations such as the CSIR [30] and other academic institutions, for example some of the universities in South Africa (Rhodes University, the University of Fort Hare [28, 31], the University of the Western Cape [32], and the University of Cape Town [33], to name a few) have been investigating sustainable methods of bridging the digital divide between rural and urban communities of South Africa.

#### **1.2** The Siyakhula Living Lab

The Siyakhula Living Lab (SLL) is a long term experiment in connecting the unconnected and has been conducting research into providing sustainable, off-the-shelf and appropriate computing infrastructure in a rural community (targeting schools particularly) of South Africa (amongst other things); locating the computing infrastructure within local schools.

Many development projects using ICTs in the developing world mimic projects in the developed world. Researchers have found that this approach to development is not necessarily appropriate in the contexts of developing nations and argue that development projects need to thoroughly investigate their contexts in order to make decisions about appropriate technologies [34]. It can be argued that the investigation and use of appropriate technologies that are grounded in the local context and support the needs of the local communities is key to the success of ICT-based developmental projects in developing nations [34]. In order to ground the research properly in the local context, SLL project employs the "living lab methodology" to research and development, meaning "an approach that deals with user driven innovation of products and services that are introduced, tested and validated in real life environments" [35, p8]. The SLL is a quadruple helix partnership of academia, industry, government and community. It was initiated in 2005 to conduct applied ICT for Development (ICT4D) postgraduate research work in the two departments of Computer Science at Rhodes University and the University of Fort Hare, through the support of the Telkom Centres of Excellence (CoEs) programme located in those departments. The SLL is a multidisciplinary initiative incorporating researchers from Computer Science, Information Systems, Education, Linguistics, Communication, Anthropology, among others. The SLL believes that ICTs in low income and marginalised areas (of which rural communities are an example) can facilitate:

- Poverty alleviation;
- Development of local economies;
- The achievement of basic standards of health, education, access to governmental services and other developmental infrastructure and services;
- The encouragement of people (though empowering them) to invest in themselves and their communities; and
- Cultural regeneration, including the development and integration of indigenous knowledge systems into a community's "ways of doing and learning".

Seventeen schools in the Dwesa-Cweba region of the Mbashe Municipality operate as points of presence for SLL facilities within the communities. These facilities are used by the learners, teachers and members of the the surrounding communities. Each school makes their facilities available to the local communities. This access supports local community members by allowing them to access on-line and locally-available resources, such as searching for work and accessing on-line government resources that support rural life activities such as subsistence farming or making use of local content and services.

The configuration of the computing infrastructure has been to build thin client computing facilities using a powerful central server together with cheaper, low-end thin client computers. The software employed was Edubuntu, which incorporates the Linux Terminal Services Project (LTSP) and much of the Free and Open Source educational software currently available. In terms of networking infrastructure, the SLL deployed a school wide network through both wired and wireless networking technologies so that learners and teachers can access the Internet connection, and store and retrieve work and other available services from anywhere within the school property.

#### **1.3** The problem statement

The National Broadband Policy [15], the Eastern Cape Broadband Master Plan [36] and the e-Education White paper [25] do not specify the IT infrastructure that should be available in schools in order to access and unlock the potential of the proposed newly available network. Contrary to popular belief, IT infrastructure needed in schools is different from that required by the business world [23]. Furthermore, schools operate on a much smaller and tighter budget, placing a strong emphasis on cost effective, sustainable solutions.

Over the course of the SLL project, the following requirements have been identified as key in providing computing infrastructure within schools:

- First line maintenance must be possible by non IT specialists (either teachers or local, not necessarily well educated, community members);
- IT infrastructure, should be easily healed, as well as robust (more like a mobile phone than the classical desktop); and
- The price of each end-user device must be as low as possible in order to allow a reasonable number of stations to be available (30 workstations in a school of 800 learners will a lot less effective than say 100 workstations).

The thin client computing model that has been employed in the SLL fails to fully meet all these requirements. As a result, this project aims to investigate alternative computing infrastructure configurations better able to be easily healed; improving uptime and system performance as well easing maintenance of the computing infrastructure such that the schools and communities are able to meaningfully make use of the available facilities. Furthermore, the alternative computing infrastructure should offer a scalable solution in order to readily support a growing number of clients but without significantly increasing the costs.

#### 1.4 Research design

This section briefly introduces the research aims and outcomes, questions, methodology and delimitation of scope (greater detail of the research design and methodology is covered in Chapter 2 of this dissertation).

#### 1.4.1 Research aims and outcomes

The main research aim of this study is to develop a refined model of computing infrastructure for use in rural schools in South Africa; focusing on design choices, cluster and grid technologies, the use of Edubuntu and free software, ease of maintenance and financial sustainability. I seek to improve the reliability and functionality of typical school hardware installations in rural South African schools (what I refer to as computing infrastructure throughout this dissertation) such that maintenance is minimised and simplistic, facility reliability is improved and more cost effective hardware configurations are suggested. There are five broad outcomes of this research study, namely to:

- Provide a detailed history of the SLL;
- Identify suitable (robust, easy to maintain, pedagogically appropriate and financially cost effective/sustainable) current and future computing infrastructure available for teaching and learning;
- Compare and contrast the use of cluster LTSP servers vs hybrid Edubuntu labs and standard thin client computing;

- Understand how teachers are using ICTs for teaching and learning and the technological (to a greater extent) and pedagogical (to a lesser extent) difficulties that they face doing so; and
- Propose and refine a model of computing infrastructure for use in rural South African schools.

#### 1.4.2 Research questions

As a result of the research context and the desired goals of the study the main research question posed in this study is:

What efficient and effective, off-the-shelf computing infrastructure, that is easy to maintain, robust, pedagogically appropriate and financially sustainable, can support rural schools in South Africa in implementing their vision for integrating ICT use into teaching, learning and administration?

The related sub-questions that stem from this main research question are:

- What is the currently deployed computing infrastructure and computer networks in the schools in the SLL?
- What current and future computing infrastructures (that are robust, easy to maintain, financially sustainable and pedagogically appropriate) are available for use in South African rural schools?
- What are the technological-based issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?
- What are the pedagogical issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?
- How do clustered LTSP servers and fat/hybrid client configurations compare (in terms of performance, reliability, cost and ease of maintenance) with the standard thin client configuration in the school environment in the SLL?

#### 1.4.3 Research Methodology

The research work undertaken in the SLL, and specifically this project within the broader SLL, draw from a wide range of methodological approaches, namely: Action Research and more specifically Participatory Action Research, the Living Lab methodology, long-term experiments, and case studies. The purpose of mixed methods research is to draw from the strengths of qualitative and quantitative research methods and thereby minimize the weaknesses of both when utilized together in a single research study [37]. It can be seen as inherently incorporating the idea of triangulating data, by combining different methodologies in the study of the same phenomenon to corroborate or refute findings and gain a deeper understanding of the phenomenon under investigation [38].

The SLL utilizes both action research, specifically participatory action research and the living lab research methodology. Living labs are, by definition and very nature, research initiatives that require researchers from universities (or research institutions) to partner with local communities in order to effect the desired changes or the creation of desired "products". As such, the living lab methodology is very closely aligned with participatory action research, where the researchers from an institution work as partners with the communities that they seek to assist in either developing or producing products or services that are needed by those communities. Furthermore, each project or development endeavour undertaken by the researchers and the community can each be seen as a case study – an empirical investigation in a real world setting. Finally, because of the long term nature of the Siyakhula Living Lab -11 years and counting - it is also a long-term empirical experiment. Even this project, within the broader SLL project, has been a long-term intervention of four years.

Consequently, I used the following research tools to inform the design and description of a computing infrastructure model for rural South African schools:

- a historical look at the SLL project;
- desktop surveys of understanding and implementing ICTs in Education;
- a survey of 6 schools and twelve teachers in the SLL;
- laboratory experimentation, implementing and testing various computing infrastructure configurations; and
- real world case studies deploying computing infrastructure configurations (based on results from the laboratory testing) to schools within the SLL.

#### **1.4.4 Delimitation of Scope**

This research operates within the much broader project of the SLL, whose primary objective was to develop and field-test a distributed, multi-functional community communication platform (called Teleweaver), in order to be deployed in low income and marginalized communities in South Africa, where a large proportion of the South African population live (37 % of South Africans live in rural areas [39], while 28.9% of South Africans live in townships<sup>1</sup> and 5.8% live in informal settlements [40]). Over time, the objectives of the SLL have become more refined and include:

- providing state of the art, long distance wireless telecommunications infrastructure that forms a "broadband island" and provides Internet connectivity;
- the provision of appropriate and sustainable computing infrastructure in order to support the various development initiatives of the project as well the the computing needs of the communities;
- support grass-root activities through harnessing indigenous knowledge for proof of concept of eService applications such as e-commerce, e-government, e-health, e-education, micro-tourism, etc;
- · continuous participatory and multi-disciplinary action research; and
- replication of results and seeking sustainability in the creation of a software factory Reed House Systems.

However, this project only contributes to the computing infrastructure design and development with a focus on education (as schools are the points of presence within the communities) and supporting teaching and learning. Naturally, investigating the use of ICTs in education is a multi-faceted problem, requiring research work into technology, pedagogic strategies, teacher competencies and the broader context in which they operate. My work contributes to one aspect of this complex research area, namely, the computing infrastructure of schools in low income and marginalised areas.

<sup>&</sup>lt;sup>1</sup>I am referring to the South African definition of "township" which refers to an under-developed (and typically under-served) suburb or peri-urban area, where "non-whites" were forced to live during Apartheid. Although segregation is no longer enforced, townships are still primarily inhabited by community members from low socio-economic strata, who are typically non-white by race.

#### **1.5** Structure of thesis

This thesis describes the work done in order to answer the research questions posed in section 1.4.2; the structure of the thesis is graphically depicted in Figure 1.1 and described in the paragraphs below.

*Chapter 2* details the research design of the project. The chapter describes the research orientation in which I operate and describes the research methodology employed in the study. It further details the research goals and questions. It then describes the research environment and participants. Next it describes the research tools employed as well as the research activities, in order to generate the requisite data and information in order to better understand the area of study. Lastly, the chapter covers the ethical considerations made during the course of the research work, and the data analysis methods and techniques employed in order to better understand and interpret the data generated throughout the study.

*Chapter 3* describes the history and evolution of the SLL. Describing the initial interventions through to the gradual expansion of the SLL. The chapter details the deployed computing infrastructure within schools as well as the growth of the network; describing work conducted from 2006 through to 2013 and highlighting shortcomings of the computing infrastructure that need to be addressed.

*Chapter 4* is the first of the two literature review chapters and focuses on the use of ICTs in education, together with the underlying theories that help us to understand that use and improve upon it. In addition, these theories help inform the appropriate choice of technological interventions within schools and therefore contribute to the design of an appropriate model for the computing infrastructure.

*Chapter 5* is the second literature review chapter and describes some of the necessary computing infrastructure and networking technologies that could be used in order to provide computing infrastructure solutions to schools. Furthermore, it touches on ICT policies in South Africa, such as the South African Broadband policy and their impact.

*Chapter 6* details the findings from the SLL schools' survey. It describes results from the questionnaire conducted with participants and the follow-up interviews or focus groups done at the six schools. Further, the chapter describes some analysis of those results in light of the theories discussed in Chapter 4, in order to better understand the findings from the survey. This analysis and discussion contribute to informing the design and deployment of two alternative proof of concept computing infrastructure configurations within two of the six schools in the SLL.

*Chapter* 7 discusses the proposed model of computing infrastructure; recapping the current SLL computing infrastructure design and highlighting its shortcomings, with a view to experimenting with server clusters and grids as a means of addressing the model's flaws. The chapter describes the cluster and grid techniques investigated and describes laboratory experiments conducted in order to investigate the levels of performance of the various solutions within an ideal environment.

*Chapter 8* describes the deployment of the DNS cluster to school FH5, the Edubuntu Fat (hybrid) client (grid-like) configuration to FH1 and the standard thin client configuration to FH2. The chapter presents the performance data gathered from the deployments to the three schools and compares their respective performance to each other, in order to tease out lessons learned. Finally, the chapter discusses a revised model, taking the initial model as described in Chapters 3 and 7 and the lessons learned from the deployments and subsequent changes, discussing implications and a way forward.

*Chapter 9* concludes the thesis, by detailing a summary of the research project followed by a revisit of the research questions and attempting to answer them. Finally, a look to the future of where work in this area could go.



Figure 1.1: Structure of this thesis

### **Chapter 2**

## **Research methodology and design**

#### 2.1 Introduction

This chapter describes the research design of this study, describing the ontological framework, research methodologies, tools and activities employed in order to understand in greater depth the context of using computing infrastructure for education in rural South African schools. Furthermore, the chapter describes the ethics adhered to and the data analysis techniques employed.

#### 2.2 Ontological Framework

While a Computer Science initiative, this research project works at the interface of science and social science, specifically, Computer Science and Education. The computing infrastructure of the SLL has always been placed in schools within the community in which the SLL lives and operates. The schools are often community focal points within rural areas of South Africa, but also, they are by nature places of learning, and one of the tenets of the SLL is that ICTs become part of everyday life within rural communities. As a result the project is conducted within a critical realist ontological framework.

Ontology here refers here to the philosophical definition which is the study of existence and the nature of being, rather than the Computer Science definition of ontology which refers to a data model that represents a set of concepts within a domain, usually hierarchical, and the relationships between the concepts. All research tends to be conducted within an ontological framework, whether explicitly stated or not. The principal tenet of realism is "that there is a world existing independently of our knowledge of it" [41, p8]; thus a critical realist believes that there is a reality independent of our thinking about it that science can study. Thus there is a distinction between our understanding or knowledge of the world and the world as it exists [42]. Critical realists recognise that all observations are fallible and that theories can be revised. As a result, a critical realist is critical of one's ability to know reality with certainty [43]. Trochim says it succinctly: "where the positivist believed that the goal of science was to uncover the truth, the postpositivist critical realist believes that the goal of science is to hold steadfastly to the goal of getting it right about reality, even though we can never achieve that goal!" [43, para 6].

According to Patomäki and Wight [44], every theory of knowledge logically presupposes a theory of what the world is like in order that knowledge is possible, in other words, "[t]he question is not of whether to be a realist, but what kind of realist to be" [44, p223]. Critical realism holds that the world is not only composed of events, states of affairs, experiences, impressions and discourses, but also of underlying structures and powers that exist, whether or not they are or have been detected or known through experience or discourse. Thus, for critical realists the underlying reality provides conditions of possibility for actual events, perceived and experienced phenomena [44]. The critical realist also acknowledges that although the

underlying reality may possess certain powers or tendencies, these are not always manifest in experience or even realised. An example can be the potential of nuclear arsenal: it poses vast destructive power, even if that power is never actualised [44]. Thus the world consists of more than just events, experiences or the discourse about them and science is not a deductive process that attempts to find constant event conjunctions, but rather aims at identifying and illuminating the structures and powers of the underlying reality [44]. This work then seeks to highlight the underlying affordances (structures) of technology in order to shed some light on how these technologies might better be utilised in the classroom environment. More specifically, how to provide that technology in such a way that it is of value, use and meaning to schools the communities in which they reside. Patomäki and Wight [44] succinctly summarise critical realism as being "committed to ontological realism (that there is a reality, which is differentiated, structured, and layered, and independent of mind), epistemological relativism (that all beliefs are socially produced and hence potentially fallible), and judgmental realism (that despite epistemological relativism, it is still possible, in principle, to provide justifiable grounds for preferring one theory over another)" [44, p224].

In scientific research, positivism is generally the ontological framework of choice [44]. Because positivism is dominated by reasoning, rationality and scientific thinking, it tends to underestimate the effects of social factors within the research domain [45]. This research study will, for example, evaluate the opinions of teachers with regards to ICT computing infrastructure for educational purposes and as such must take into account the social factors and realise that the results are not necessarily generalisable. However a pure interpretivist approach would not be appropriate either, because of the experimental work that was done during this project in order to configure, deploy and test various computing infrastructure possibilities in schools. During these experimental stages, reasoning, rationality and scientific thinking were invaluable to finding computing infrastructure solutions that can best support the needs of schools, within the very real constraints in which they operate. As a result, the ontological framework needs to capture both scientific reasoning and the subjective nature of working with people.

#### 2.3 Methodological Approach: The mixed methodology

The research work undertaken in the SLL and specifically this project within the broader SLL draw from a wide range of methodological approaches, namely: Action Research and more specifically Participatory Action Research; the Living Lab methodology; long-term experiments; and case studies. The purpose of mixed methods research is to draw from the strengths of qualitative and quantitative research methods and thereby minimize the weaknesses of both when utilized together in single research studies [37]. It can be seen as inherently incorporating the idea of triangulating data, by combining different methodologies in the study of the same phenomenon to corroborate or refute findings and gain a deeper understanding of the phenomenon under investigation [38]. Findings from multiple sources can be corroborated and used to understand a single conclusion; if the multiple sources support each other or if they conflict with each other, the researcher has a greater knowledge of the research area and can modify interpretations and conclusions accordingly [37]. Johnson and Onwuegbuzie [37] further argue that the goal of the mixed methods research is not necessarily corroboration but rather to expand the knowledge of the researchers. Johnson and Onwuegbuzie [37] define mixed methods research as "the class or research where the researcher mixes or combines quantitative and qualitative research techniques, methods, approaches concepts and language into a single study" [37, p17]. Johnson and Onwuegbuzie [37] and Johnson, Onwuegbuzie, and Turner [38] argue that the mixed methods approach is the third research paradigm (where qualitative and quantitative research are the first two paradigms), which offers researchers a logical and practical alternative to the first two research paradigms. They claim that mixed methods research uses the pragmatic method and system of philosophy, allowing it to include the use of induction, deduction and abduction, to discover patterns,

test theories and offer the best set of explanations of one's results, respectively [37, 38]. Mixed methods research attempts to legitimate the use of multiple methods and approaches to answering research questions; an expansive and creative form or research, which is inclusive, pluralistic and complementary and encourages researchers to take an eclectic approach to selecting methods and conducting research [37].

The four research methods mentioned above (action research, the living lab methodology, long-term experiments and case studies), generating and utilizing both qualitative and quantitative data, allow for a much deeper understanding of the phenomenon of ICTs for development, and more specifically in this dissertation, the use of computing infrastructure in the school environment to support both education endeavours in the schools as well as contribute to the general development of the communities in which the SLL operates.

Within this section I will discuss the fundamentals of each of the four research methodologies that are utilized together in order to provide a deeper understanding of the needs of schools and the implementation and deployment of computing infrastructure in schools in the SLL. In Section 2.9 I will present on the data analysis (both qualitative and quantitative) undertaken in this research work.

#### 2.3.1 Action research

Walsham and Sahay [46] argue for the importance of using action research methodologies in ICT research for developing countries (such as South Africa), as the methodology readily lends itself to effecting change (development) and shows relevance in contexts where resources are scarce. There are numerous definitions of action research in literature, as there are many subsets of the methodology within the broader area of action research [47]. A basic definition of action research is that it is a form of inquiry that is initiated to solve a problem typically through iterative reflective problem solving, where researchers and research participants work together to undertake the research work and problem solving, thereby contributing to the growing theoretical knowledge surrounding the area (problem) of inquiry [47, 48]. It is generally agreed upon by the broader research community that action research is about two things, namely, taking action to improve practice and undertaking research in order to find new understandings or creating new knowledge [47]. Lewin, to whom the origins of Action Research are attributed, described the process of action research as a cycle of steps which include observing, reflecting, acting, evaluating and modifying [49, 47]. These cycles can repeat, becoming additional cycles, allowing a project to span out over time and continue to further refine the changes made and the lessons learned [49], as is the case with the SLL. Action research in information systems/technology research has been growing in popularity since the late 1990s; the method produces relevant research results because it is grounded in practical action in order to attempt to solve real-world problems while contributing to ongoing theory formulation [50].

The two main types of action research are interpretive action research and "living theory" action research [47]. Interpretive action research believes that the proper way to undertake action research is for the researcher to be external to the practices or environment under study, while the "living theory" action research believes that the practitioners are capable of self-study, interpretation and theorising with the aim to effect change [47]. The SLL project could be seen as both a "living theory" action research as the concept of the living lab has the researchers and the community members as partners and co-innovators; and a sub-class of interpretive action research, specifically second-person action research where "the practitioner can 'inquire face-to-face with others into issues of mutual concern …", which is classed as "participatory" action research [47, p12]. Participatory Action Research (PAR) typically takes place within communities and emphasizes both participation and action, seeking to understand the environment in which they are operating and try to effect collaborative change and then reflect on those actions and learn from them [51, 49]. PAR is typically a type of qualitative research that integrates methods and techniques for observing, documenting, analysing and interpreting the human phenomena understudy; the purpose being to describe and understand what is happening rather than predict and control [49]. I would argue that it is not limited to qualitative research only and quantitative data can be employed to further enhance the qualitative findings of interactions with the participant researchers. PAR has been defined as "systematic inquiry with the collaboration of those affected by the issue being studied for purposes of education and taking action or effecting social change" [52, p475]. The aim of PAR is to afford growth in capacity and development (of the local community), empowering individuals and communities, providing access (in our case specifically to technology and information through Internet access), social justice and affording participation [49].

I would argue that the SLL and this study within the bigger SLL research project (which adopt the living lab methodology, see more details below), are participatory action research activities. Bergvall-Kåreborn and Ståhlbröst [1] argue that the action research methodology is particularly well suited to living lab interventions as both emphasize interaction between theory and practice, the need for constant reflection post intervention, and the importance of all stakeholders playing active roles in the research process. According to Cornwall and Jewkes [53], there are four "modes of participation" in research, namely [53]:

- · contractual: participants are contracted into research projects to assist in inquiries or experiments;
- consultative: participants are asked to provide their opinions to researchers before interventions are undertaken;
- collaborative: researchers and participants work together on the research project but the overall control still lies with the researchers; and
- collegiate: researchers and participants work together as colleagues, each bringing different skills to the research project, for mutual learning and where both have control in order to govern the direction of the research project.

Because the SLL is by its very nature a living lab and thus uses the living lab methodology (as discussed in the next subsection), the SLL would be characterised as a collegiate participation research project, making it a PAR project. Participants in PAR are not classified as "subjects of research" but rather co-researchers who participate at all levels of the research process [49]. Cornwall and Jewkes [53] argue that achieving pure collegiate PAR is challenging, as often participants feel out of their depth, and/or researchers tend to drive the direction, analysis and representation of the findings rather than being a wholly joint effort from both researchers and participants [53]. In the SLL, this is a reality. While the project strives to be as collegial as possible in having the two groups work together, often the analysis and meaning-making exercises that are derived from the successive interventions (iterations) of the SLL are heavily driven by the researchers. However, the SLL continues to attempt to provide greater control (in such striving towards a collegiate PAR project) to the communities living within its footprint. One such example, was the creation of a SLL committee, consisting predominantly of community members living in the area in which the SLL operates, whose mandate is to guide the SLL initiatives and research operations in conjunction with the universities.

#### 2.3.2 The living lab methodology

There are numerous definitions of a living lab in research literature. Dell'Era and Landoni [54] report 13 definitions used in research work from various researchers and projects. However, they note that despite the lack of consensus on the definition of a living lab, all the various definitions share two common points, namely, that a living lab is set in a real-life test and experimentation environment and that users of the living lab are actively involved in the innovation process [54]. Dell'Era and Landoni define the living lab as "a design research methodology aimed at co-creating innovation through the involvement of aware users in a

real-life setting" [54, p139]. The "Living Labs for user-driven open innovation, an overview of the living labs methodology, activities and achievements" [55] defines the living lab as follows,

"A Living Lab is a user-driven open innovation ecosystem based on a business – citizens – government partnership which enables users to take an active part in the research, development and innovation process:

- bringing the users early into the creative process in order to better discover new and emerging behaviours and user patterns;
- bridging the innovation gap between technology development and the uptake of new products and services involving all relevant players of the value network via partnerships between business, citizens, and government;
- allowing for early assessment of the socio-economic implications of new technological solutions by demonstrating the validity of innovative services and business models. [55, p7]"

The SLL's operation and method tend to lend itself more to this definition, but would include that academia is part of the partnership within the living lab methodology, which is echoed in other research work, like that reported by Bergvall-Kåreborn and Ståhlbröst [1]. Living labs are said to be composed of five key components, namely, users, application environment, technology and infrastructure, organisation and methods, and partners (community, academia, government and industry/business), as depicted in Figure 2.1 (taken from [1]).



Figure 2.1: Key components of a Living Lab as described by Bergvall-Kåreborn and Ståhlbröst [1]

Living labs emerged at the beginning of 2000 with an initial focus on testing new technologies in realworld environments [1]. Since then the concept has grown to be defined as a methodology, an organization, a system, an arena, an environment, and/or a systemic innovation approach, but it remains throughout all of these definitions that the activities of a living lab are situated within a real-world context [56, 1]. Bergvall-Kåreborn and Ståhlbröst [1] argue that these multiple definitions of the living lab are not contradictory but rather complementary perspectives of a very powerful concept. They further argue that the various views of a living lab allow the various aspects of the living lab to be brought to the forefront. For example, a more environment-centric definition brings to the forefront the focus of the living lab, such as data transfers and methods for user involvement, to the fore [1]. Bergvall-Kåreborn and Ståhlbröst [1] further argue that there are five key principals within the living lab methodology. There are:

- continuity: the long-term experimental nature of living labs by default lend themselves towards continuity, which Bergvall-Kåreborn and Ståhlbröst [1] argue strengthens creativity and innovation, and builds trust between the researchers and the community members;
- openness: this principal is key to supporting user-driven innovation. The users are co-creators in the innovation process and so these processes are open and scrutinizable by all parties;
- real-world setting: in order to generate real-world results, living labs are set in real-world environments, in real-world situations, making the application development more readily applicable to the end users;
- empowering of users: the engagement of the end user is fundamental to the development process, giving the end user power to control the process and contribute to the development of innovations that can be really beneficial to them; and
- spontaneity: this principal ensures that the researchers are also willing and able to respond to user driven demands in the development process, which can change rapidly.

Of these five principals, Bergvall-Kåreborn and Ståhlbröst [1] argue that three are important and represent the core of the living lab methodology, namely, empowering users; openness; and the real-world setting. These three principals are core to the SLL in the Dwesa region. The project's aims are to empower low income, marginalised, rural communities in order to have them move, over time, from a second economy to a first economy. In addition, the SLL aims to be as open as possible – end users have much say in the processes of the SLL, with a community committee driving a lot of the research and innovation – and obviously, the SLL is situated in a very real rural South African environment.

#### 2.3.3 Long-term experiments

A long-term experiment is an experimental procedure that takes place over a long period of time in order to observe a phenomenon that takes place at a slow rate [57]; to quote Brian Owens, "[t]he world's longest running experiments remind us that science is a marathon, not a sprint" [58]. The choice between a short-term and a long-term study depends on whether or not the research aims to learn something about a single point in time, or learn about changes over a long period of time [59]. The Siyakhula Living Lab is a long-term experiment and has been running for 10 years, investigating the introduction of technology into a rural, marginalised community in South Africa. This particular intervention described in this dissertation, namely, the experimentation conducted with the computing infrastructure in the schools was over a shorter period, 2013-2016, it remains, that it is part of the bigger SLL experiment, which is long-term. Arguably, the three year nature of this particular portion of the SLL intervention could also be construed as long-term.

The value of long-term experiments is well recognised and are often used in ecological, botanical, agricultural, microbiological, medicinal, psychological and physics experiments [57, 60]; however, long-term studies of communities are rare [60]. Pelton and van Manen postulate that the reason for this is that long-term studies are expensive and require greater institutional commitment; they define a long-term study as a continuous study that focuses on a specific community (or species or ecosystem, in cases of zoological studies) over a long duration of time [60]. Some of the advantages of long-term experiments noted by researchers are: that the experiments could adapt and change and learn from past work; that new hypotheses and ideas could be tested in short-term experiments within the long-term study; and that long-term studies could facilitate in planning for similar studies in different areas; the ability to detect

developments or changes in characteristics of the target population; and the ability to establish the sequence of events that result in observed changes [60, 59]. The main shortcomings of long-term experiments are: it is difficult to secure funding for long-term research studies, funding is typically sourced in short-term intervals (between one and three years); and it can be difficult to maintain consistency and continuity, especially when considering the high turnover of researchers (including post graduate students) working on long-term projects [60, 61].

Pelton and van Manen describe the following four criteria that indicate the necessity for long-term experiments [60]:

- 1. Slow processes;
- 2. Rare events or episodic phenomena;
- 3. Processes with high variability; and
- 4. Subtle processes and complex phenomena

Arguably, the SLL is both a slow process in that change as a result of the introduction of technology to rural, marginalised communities will take place over time, as well as a subtle and complex phenomena, as the changes involve the lives and practices of the people living in the communities in and round Dwesa (the site of the SLL). Similarly, introduction of technology to the processes of school education, teaching and learning, can see slow changes to overall processes, as well as subtle changes within the complex phenomena of teaching and learning.

#### 2.3.4 Case studies

This research project can be broadly construed as a case study, where the various and successive ICT infrastructure deployments at the schools are different case studies. Robson defines case study methodology as "a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon in its real life context using multiple sources of evidence" [62, p52]. Similarly, Yin describes the case study as "[a]n empirical inquiry about a contemporary phenomenon (the "case"), set within its real-world context" [63]. Case study research does have its drawbacks in that it cannot be easily generalized, but if it is "contextualised and carefully described [. . . ] then others can consider its usefulness in other contexts and examples" [64, p191]. Other methods of study were employed within the bigger case study method of this project, namely, a history, survey and experiments, in order to gain a deeper understanding (through the interrogation and triangulation of multiple data sources [65]) of the use of computing infrastructure for the school environment in rural South Africa. There are typically three types of case studies; descriptive case studies, explanatory case studies and exploratory case studies [63]. This work reported in this thesis is an exploratory case study; the goal being to develop a pertinent hypothesis and proposition (specifically, a model of computing infrastructure for rural South African schools) for further inquiry [63].

Social realist research explicitly holds that "[n]o individual-level intervention works for everyone. No institution-level intervention works everywhere. The net effect of any particular programme is thus made up of the balance of successes and failures of individual subjects and locations. What this points to is the need for a careful look at subject and contextual difference in terms of who succeeds and who fails within any programme" [66, p30-31]. This dissertation therefore can not claim to generalise, but rather attempts to highlight key issues so that other schooling environments might gain from these experiences. As Pawson explains: "At the extreme, we can still learn from a negative net effect of a single evaluation study, since the application of an initiative to the wrong subjects and in the wrong circumstances can leave behind vital clues about what might be the right combination" [66, p31].

#### 2.4 Research goals and questions

In this section I describe the overall research aim together with the explicit deliverables that were sought to be achieved by the end of the project. Furthermore, I articulate the overall or main research question that this study sought to answer, as well as the related sub-questions that contributed to answering the main research question. These questions and the answers that were derived as a result of this study will be revisited in the final chapter of this dissertation.

#### 2.4.1 Research aims

The overall aim of this study is to pave the way to assisting schools, governments, NGOs and teachers in selecting technologies that are appropriate to a school's educational needs and ICT vision (specifically able to support activity types within teaching practices) based on the affordances of the technology. In order to do this I propose a refined model of computing infrastructure for use in rural schools in South Africa; focusing on design choices, cluster and grid technologies, the use of Edubuntu and free software, ease of maintenance and financial sustainability. I seek to improve the reliability and functionality of typical school hardware and software installations within the SLL (what I refer to simply as computing infrastructure throughout this dissertation) such that maintenance is minimised and simplistic, facility reliability is improved and overall cost reduced, while catering for the educational needs as specified by the teachers. There are five broad sub-goals of this research study, namely to:

- provide a detailed history of the SLL; a long term experiment in connecting the unconnected;
- identify suitable (pedagogically, for ease of maintenance, robust, cost effective and sustainable) current and future computing infrastructure available for teaching and learning;
- compare and contrast the use of cluster LTSP servers and hybrid Edubuntu labs with standard thin client computing;
- understand how teachers are using ICTs for teaching and learning and the technological (to a greater extent) and pedagogical (to a lesser extent) difficulties that they face doing so; and
- propose and refine a model of computing infrastructure for use in rural South African schools.

The main intended deliverable of this project is to produce:

• a refined model of computing infrastructure for use in rural schools in South Africa; focusing on design choices, robust configurations, the use of Edubuntu and free software, ease of maintenance and financial sustainability. And most importantly improve the ease with which the facilities can be healed while providing increased computing resources to the clients in order to readily support a growing number of clients.

#### 2.4.2 Research questions

Based on the research context and the desired goals of the study, the main research question posed is:

What efficient and effective, off-the-shelf computing infrastructure, that is easy to maintain, robust, pedagogically appropriate and financially sustainable, can support rural schools in South Africa in implementing their vision for integrating ICT use into teaching, learning and administration? The research work undertaken has been conducted within the experimental space of the SLL; a space that has been instrumented to facilitate appropriate analysis and derive real world solutions. The SLL is a good representative of rural and deep rural areas of South Africa. Consequently, the research sub-questions focus on the SLL, as a model environment in which to test and experiment. The related sub-questions that stem from this main research question are:

- What is the currently deployed computing infrastructure and computer networks in the schools in the SLL?
- What current and future computing infrastructures (that are robust, easy to maintain, financially sustainable and pedagogically appropriate) are available for use in South African rural schools?
- What are the technological-based issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?
- What are the pedagogical issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?
- How do clustered LTSP servers and fat/hybrid client configurations compare (in terms of performance, reliability, cost and ease of maintenance) with the standard thin client configuration in the school environment in the SLL?

#### 2.5 Research environment

The research environment presents the physical place in which the research takes place, as well as the research participants – those people who live or work in the research environment and are key participants and co-researchers within the project (co-researchers because this project uses both the living lab methodology as well as participant action research). In this section I start by introducing the research site (the physical location) of the SLL, namely, Dwesa in the Mbashe Municipality. This is followed by an overview of the South African school classification system, largely influenced by the Apartheid system. This is necessary in order to better understand the various school environments and their history in South Africa. The history of a school plays a significant role in its current circumstances and how it operates. Lastly, I introduce the research participants, namely, the twelve teachers and six schools that undertook various aspects of this research work with me.

#### 2.5.1 Research site: Dwesa, Mbashe Municipality

Dwesa is a marginalised, low income, rural community in the former Transkei homeland of the Eastern Cape (specifically the Mbashe Municipality). It lies directly along the Wild Coast of South Africa, approximately 50 kilometres from Idutywa and (see Figure 2.2). The inhabitants of Dwesa are traditionally subsistence farmers with an estimated population of 15000 people living in 2000 households. Dwesa was the second successful land restitution claim in South Africa and the first in the Eastern Cape [67]. The villages targeted in the SLL are adjacent to the Dwesa-Cwebe area, which comprises the nature reserve and adjacent communities which are extended over a land area of approximately 153 square kilometres [67]. Both the reserve and the local communities have become involved in development initiatives as the natural environment, consisting of the nature reserve and the coast, are assets for the communities. The nature reserve serves to provide income generating activities to support rural development in the area [67]. The unspoiled natural beauty and wild beaches have the potential to draw significant ecotourism to the region.

Furthermore, the high levels of rainfall and rich soil in a typically dry area of the country have potential for controlled agricultural intensification and commercial forestry [67].


Figure 2.2: Maps of Dwesa in relation to the rest of South Africa (a) A Google Earth map with the Dwesa area marked inside the red box (b) A map of the Wild Coast with the Dwesa nature reserve marked

The Mbashe Municipality and specifically Dwesa, is also faced with socio-economic challenges associated with poverty and poor development. The region is characterized by a lack of infrastructure, namely, roads, electricity, piped/plumbed water, piped sanitation, fixed line telephone access (GPRS cellular phone access is available in most parts of the community, with 3G in isolated areas) and services. In addition, there is widespread poverty and the community is isolated. This isolation is not just physical but also in terms of knowledge and information [68]. Consequently, this area has been the target for developmental projects over the years. ICTs can support rural development by enabling communication which can further enable the effective operation of developmental activities.

There are a number of schools in the Dwesa community and currently the Siyakhula Living Lab project [69, 70] works with seventeen of those schools [71]. These schools are located in areas of (in some cases) extreme poverty where cost effective and pedagogically sound ICT architectures would be of benefit to the schools and surrounding communities. The schools typically also have strong principal leadership and enthusiastic teaching staff, which make them ideal test beds for understanding the needs of rural education and testing different ICT solutions.

### 2.5.2 South African school classifications

The National Education Policy Investigation (NEPI) report on the Apartheid Governance and Administration of the Education system in South Africa (published in 1992) described the Apartheid Education system as "one of extraordinary complexity" [2, p6]. In line with the National Party's policy of Apartheid, the objectives of the education system in South Africa was "segregated, differentiated, and unequal education for different racial groups" [2, p6]. According to Apartheid there were four racial groups, namely, Whites, Coloured, Indians and Blacks [2]. As such, there were four basic departments which governed the different school classifications according to race within South Africa, as well as a number of Homeland departments. The control of education in South Africa was therefore also fragmented: there were four separate education systems for the "independent homelands" (Transkei, Bopthuthatswana, Venda and Ciskei); six education systems for the so-called "self governing territories" (these were the remaining Homelands that were not granted independence); one education system for White education (House of Assembly); one education system for Coloured education (House of Representatives); one education system for Indian education (House of Delegates); and lastly one education system for Black education happening within South Africa and not a Homeland (the Department of Education and Training) [72, 2]. This resulted in a total of 19 different education systems/departments, that were operating within 14 different cabinets, each one implementing their own regulations, governed by 12 different Education Acts and 17 different employment authorities (employing teachers) [2] (see Figure taken from [2]).



Figure 2.3: Organogram of the South African Education System during the Apartheid era, taken from [2]

In 1990, prior to the democratisation of South Africa and the fall of Apartheid, the House of Assembly announced a series of new governance models for the schools under its control. The proposal gave the schools four options: they could retain the status quo, or by a majority vote of the parents elect to adopt one of the three new governance models, A, B or C [2]. Model A provided for the conversion of the school into an independent school, and required the community to purchase the school buildings from the state [2]. Model B provided for the continuation of the school as a state school, but with the right given to a management committee, elected by the parents, to determine the admission policy, subject to certain limitations which would ensure that the majority of children were white [2]. Model C provided for the conversion of the school into a state-aided school under the control of a governing body elected by the parents. The major condition of this model was that the state would only cover the costs of salaries, and other recurrent costs such as textbooks and educational materials, maintenance and building insurance, rates and services [2]. All capital costs would have to be met by fund-raising activities of the parent body [2]. The increased autonomy of the Model C schools also allowed the schools to change their admissions policies and allow children of other races to attend their school. However, in February 1992 there was a further development when an announcement was made that unless parents specifically voted against it, all status quo and Model B schools would be converted to Model C schools [2].

The Homelands that had been granted "independence" were considered by the Apartheid government to be "foreign states" but wholly contained in South Africa, historically a part of South Africa, and considered by the rest of the world and the people living there as part of South Africa [2, 73]. Today they are once again recognised as part of South Africa. As independent states their ostensibly had control over how they operated and thus over the education system operating within its borders. However, the reality of the Homelands were that they served as means to further impoverish Black population of South Africa and subject them to dysfunctional systems of government, poor education and schooling and no opportunities for employment [74]. As such the schools in the Homelands were, like in former Department of Education and Training schools in South Africa, were subjected to the so-called "Bantu Education" system [75]. The purpose of

the Bantu Education system was to provide Black children with an inferior education as compared to their White counterparts, in order to prevent them from being a part of "white collar" sector of the workforce and thus provide a constant supply of cheap labour ("blue collar workers") for the sectors of agriculture, mining and domestic services [74, 75]. Hendrik Verwoerd, at the time that he was Minister of Native Affairs in South Africa, was famously quoted as saying "*There is no place for [the Bantu] in the European community above the level of certain forms of labour … What is the use of teaching the Bantu child mathematics when it cannot use it in practice? That is quite absurd. Education must train people in accordance with their opportunities in life, according to the sphere in which they live.*" Consequently, Homeland schools and Department of Training and Education schools were severely under-resourced; lacking in physical facilities, text-books, quality teachers and quality curriculum [75, 74].

When South African became a democratic nation these state school classifications largely fell away. The disparities of the past, however, still have a prominent impact on these schools today: former Department of Education and Training (FDET) and former Homeland (FH) schools tend to be more disadvantaged than former House of Assembly (FHOA) schools. For this reason we retain the former terms in order to differentiate between the historically most poorly resourced FDET and Homeland schools and the mostly well-resourced FHOA schools [72].

The Dwesa community (home of the SLL) is found in the former Transkei Homeland; a former independent homeland in the Eastern Cape of South Africa. As such all schools in the SLL are former Homeland schools. In order to preserve anonymity in this study the various schools are referred to as former Homeland schools without identifying the individual school names. This is achieved by numbering each school and producing an associated code for the school (see section 2.5.3).

### 2.5.3 Research participants

In this section of the methodology I present the research participants; those individuals (teachers) and organisations (schools) who took part in this particular research activity to investigate appropriate computing infrastructure in order to support the teaching and learning activities of rural South African schools.

In order to understand what knowledge teachers possess about ICTs and how they were making use of ICTs in teaching and learning in the classroom in the SLL, I conducted questionnaires and semi-structured interviews with two teachers from six of the schools in the SSL. The SLL operates in 17 schools within the Dwesa region, all of varying ICT experience and years active within the SLL. Of the six schools that I had chosen to survey, three had long term exposure to ICTs, ICT trained teachers and a long history of activity within the SLL. The remaining three were new to the SLL project with limited teacher training and ICT exposure in the classroom. Furthermore, the schools were a mix of high schools and junior secondary schools. Of the participating schools, two are secondary schools (from Grade 8 to Grade 12), and the other four are junior secondary schools (from Grade R to Grade 9). All of the participating schools are former homeland schools (see Table 2.1).

In order to do this I began by approaching the Department of Basic Education District office in Idutywa to request permission to conduct questionnaires and interviews with some of the teachers. Once their approval had been granted I approached the principals of the six chosen schools in the SLL and asked their permission to conduct a questionnaire and interview with two members of their teaching staff. Each of the principals agreed and nominated or suggested two staff members whom they felt would be the most interested in the project and were enthusiastic about using technology for teaching.

School	School Classification	School Type	School Code	Respondent (Teacher) code
School 1	Former homeland	Secondary School	FH1	Teacher_A
				Teacher_B
School 2	Former homeland	Junior Secondary school	FH2	Teacher_C
				Teacher_D
School 3	Former homeland	Junior Secondary school	FH3	Teacher_E
				Teacher_F
School 4	Former homeland	Junior Secondary school	FH4	Teacher_G
				Teacher_H
School 5	Former homeland	Junior Secondary school	FH5	Teacher_I
				Teacher_J
School 6	Former homeland	Secondary School	FH6	Teacher_K
				Teacher_L

Table 2.1: School and teacher participant codes

Of the 12 teachers, 11 filled in the questionnaire, the 12th teacher (from FH1) lost their questionnaire but was happy to participate in the interview process none the less. However, of the 12 teachers, only 11 were interviewed as one of the teachers (from FH3) was not prepared to be interviewed even though the teacher had completed the questionnaire. As such the remaining teacher from FH3 was interviewed alone in a one on one interview. In addition, the teachers at FH4 had to be interviewed separately as they were not available to be interviewed together in a small focus group. At all the other schools (FH1, FH2, FH5 and FH6) the interviews were conducted with both teachers at the same time in a small focus group setting.

Three slightly different computer lab solutions were implemented at three of the six schools participating in the study. Unfortunately not all schools could host lab installations due to time constraints, limited human resources and financial constraints. Labs were placed at FH1, FH2 and FH5. FH1 was configured as a fat/hybrid client computer lab, with a central server for file storage, authentication and serving the fat/hybrid client image to the fat/hybrid client computers in the lab. FH2 was configured as a standard thin client computer lab, with a central server serving standard thin client computers in the lab. Lastly, FH5 was configured as a cluster of central servers (5 servers in the cluster) serving standard thin client computers in the lab.

### 2.6 Research tools

In this section I describe the research tools employed to better understand how teachers use computing infrastructure for teaching and learning and hopefully identify the issues surrounding computing infrastructure and ICT integration into their teaching processes. These are questionnaires and interviews or focus groups together with laboratory testing of computing infrastructure and live field testing of computing infrastructure solutions.

Six of the seventeen schools in the SLL project were invited to participate in the survey portion of the research study. Surveys tend to be most effective when the aim is to reach a large proportion of a population. They tend to be used to extrapolate and generalize from a typically small sample of a population to the whole population [76]. I used a survey in order to understand which computing infrastructure might best support the educational (pedagogical) needs/vision of teachers in South African rural schools, while being constrained by costs and requiring ease of maintenance. The survey employed the use of two data collection tools, namely, a questionnaire and semi-structured focus groups/interviews. The questionnaire (see Appendix D for a copy of the questionnaire used) was employed to attempt to provide insight into the current configuration of computing infrastructure within the participating schools, while highlighting

some of the uses of those facilities by teachers. In addition, I hoped to identify those factors which both negatively and positively impact on the use of computing infrastructure for the purposes of teaching and learning.

The follow up interviews/focus groups were based on the findings of the questionnaires in order to allow respondents to clarify points and explore deeper issues raised in their questionnaire responses. If possible each interview at each of the schools was conducted as a mini focus group with both teachers. However, not all the teachers were either available to be in a focus group at the same time, or elected not to be interviewed (more details in section 2.7.2). "Interviewing is a two-person conversation initiated by the interviewer for the specific purpose of obtaining research-relevant information, and focused by him on content specified by research objectives of systematic description, prediction or explanation" [77, p1]. For each interview there was an interview schedule (see Appendix E for a copy of the interview schedule employed, and section 2.6.1 for an explanation of the design of the interview schedule) of questions that I covered. These questions were based on the responses that the teachers provided in the questionnaire. The interviews provided data that is richer than that of the questionnaire and it also allowed for more focused questioning in order to clarify answers/responses from the questionnaire. The interviews gave me a deeper understanding of what the needs of each school teacher was and what problems they are facing. The interviews provided information about software, teacher training, cost issues, infrastructural issues, pedagogical strategies or a lack thereof and many other pertinent areas that affect the use of technology for teaching and learning and will be discussed further in chapter 6. More details about the design of both the questionnaire and the interview schedule are provided in sections 2.6.1, 2.7.1 and 2.7.2.

In addition to the survey (questionnaires and interview schedules), I undertook laboratory testing experiments to better understand the computing infrastructure solutions available for schools, especially, cluster computing. Furthermore, the laboratory testing provided me with a deeper understanding of the LTSP intricacies and nuances, while also providing insight into which computing resources LTSP has the biggest impact on or makes the most use of. More details about the computing infrastructure laboratory experiments can be found in section 2.6.2 and 2.7.3.

Lastly, the final research tool employed during the course of this study was the deployment of computing infrastructure solutions to real world rural school environments in the SLL. Arguably the best way in which to understand the environment in which computing infrastructure is to be deployed and to learn about the complexities, issues, short-comings, and positive traits of the computing infrastructure within a particular environment or setting is to do a real world deployment and record the results. Of the six schools that took part in the questionnaires and interviews/focus groups, three schools were chosen for the real world computing infrastructure deployments. More details about the computing infrastructure deployments within the SLL can be found in section 2.6.3 and 2.7.4.

### 2.6.1 Surveys

The purpose of a survey is to collect information from people or organisations in which you are interested in or are related to your field of research [78]. Questionnaires are one of the tools typically employed in a survey and are used to collect the desired information from your research participants [78]. A well designed questionnaire is considered to be structured to allow the same information to be collected from a number of people in the same way allowing the data to be analysed quantitatively and systematically [78]. Another tool typically used in the course of a survey to obtain data, is that of the interview or the focus group. I will discuss the design of both these research tools employed during the source of the SLL survey in this section.

### Questionnaires design

The first step in designing a questionnaire is to decide on the aim of the questionnaire [79]. In the case of this research study, the aim of the questionnaire was to obtain a better understanding of: the use of ICTs by the participant teachers; the available computing infrastructure and networking technologies (ICTs) at the schools of the participant teachers and schools; and to gain an understanding of what teachers understand about the ICT infrastructure available at their schools. Once the aim of the questionnaire is decided upon, the next step is to typically choose the population and sample size [79]. The population in this study was the teachers working in the schools that are part of the SLL in Dwesa. The sample size of the chosen population for this study was 12 (n=12 teachers). This was to allow two teachers from each of the six schools to contribute to the understanding about ICT use, ICT needs, ICT facilities available at their school and ICT knowledge of participating teachers. A population size of 12 does not allow one to generalise to all teachers working in rural areas in South Africa, but it does provide for a deeper understanding of the nuances of teachers and their experiences in teaching in rural areas and using ICTs while teaching.

The questionnaires employed were self-administered questionnaires; the questionnaires were delivered to the teachers, and then the teachers could take their time in filling out the questionnaire. After a period of three weeks, the questionnaires were collected from each of the schools.

According to Burgess [79], the design of a questionnaire can be split into three areas: determining the questions to be asked; selecting the question type for each question and the appropriate wording of the question; and lastly, design the question sequence and overall questionnaire layout. In the case of this questionnaire, I first decided on the overall layout of the questionnaire. The questionnaire was designed to have four sections: personal information about the teacher answering the questionnaire; the current ICT infrastructure available at the school of the teacher (answering the questionnaire); the computer use taking place within their school; and lastly, the IT maintenance that takes place (or doesn't take place) at their school. Once these sections were in place, decided upon in order to get as well rounded an understanding as possible about ICT use within each participant school and by each participant teacher, the questions per section were decided upon. Burgess [79] stresses the importance of ensuring that the "right questions" are asked in the questionnaire. By the "right questions" he means that one needs to ask the questions that will provide you with the answers you seek in terms of your overall research questions and aims [79]. If you do not ask the "right questions" then you could find that you have a hole in your results and are not able to adequately answer your research questions. In addition, it is important to decide upon the types of questions to be used, open or closed. Typically, closed questions are used when attempting to answer something quite specifically so that you are able to produce a picture of your population [79], for example, "What type of Internet Access do you have?" "(a) DSL (b) Fibre (c) WiMAX (d) I don't know". This allows you to get very coherent and simple answers. Open questions allow you to ask questions where you get answers that give you insight into how your respondents think and what they personally feel about issues. Answers can vary dramatically (in length, articulation and opinions) [79]. Examples of an open ended question would be, "What do you think are the benefits and drawbacks of using ICTs for teaching and learning?" In my questionnaire, I decided not to use any open ended questions, and only closed ended, and chose to use open ended questions in the follow-up interviews or focus groups with the teachers.

The questionnaire started with an explanation for the teachers about the purpose of the questionnaire, namely that the questionnaire sought to understand which computing infrastructure would best support the teachers educational needs and vision within South African schools, while suggesting computer infrastructure that is financially sustainable and easy to maintain. This was followed by a paragraph to remind the teacher participants that their involvement was completely voluntary and that they could withdraw from the questionnaire or follow-up interviews at any stage. A copy of the questionnaire answered by the participating teachers can be seen in Appendix D.

#### Interview schedule design

There are many ways in which to structure an interview; you can use closed ended questions in order to illicit a fixed set of responses from participants; or open ended questions to allow participants to express their views in their own words; an interview can also be structured, semi-structured or unstructured, where the questions to be answered vary from fixed with no deviation through to discussing whatever topics are raised by participants [80].

The structured interview, also known as the formal interview, has a fixed set of questions (in an interview schedule) to be asked in a fixed order, from which the interviewer will not deviate. These interviews are rigid (not flexible) and the interviewer can not probe the interviewee beyond the set of predefined questions on the interview schedule [80]. This type of interview is easy to replicate as closed questions are typically used which are quantifiable (quantitative data) [80]. Furthermore, according to McLeod [80], this type of interview is quick to do and thus more readily lends itself to being conducted with large population sizes, making the findings more readily representative and able to be generalized to a population. However, because the interviewer is not able to ask more probing questions and adapt to the answers being provided during the interview, the process is rigid and answers can lack in depth and detail [80].

The alternative to the structured interview is the unstructured interview, sometimes referred to "discovery interviews" or "guided conversations" [80]. In the unstructured interview there is no need for an interview schedule or if the interviewer wishes they might have one with open-ended questions which they can ask in any order depending on the flow of conversation. In addition, they might choose to add new questions to the interview, or just skip some of their suggested questions from their interview schedule altogether [80, 81]. The advantages of the unstructured interview is that its very responsive to the interview taking place and can be adapted. In addition, the open-ended questions support an in-depth conversation (affording the interviewer the ability to probe the interviewee for a deeper understanding) with the possibility of generating rich data [80, 81]. However, the unstructured interview can be time consuming to conduct and to analyse, and can be more difficult to conduct for novice researchers [80, 81].

The middle ground between the structured and the unstructured interview, is the semi-structured interview; the interview is not highly structured, nor is it unstructured [82, 81]. A semi-structured interview utilizes an interview schedule that offers topics and open-ended questions to the interviewer in order to probe the interviewee about their thoughts and opinions on the topics of interest; without leading the interviewee or imposing any meaning during the course of the interview [82]. The strength of the semi-structured interview is that the interviews are all able to cover similar questions or areas of topic while still allowing the interviewer the freedom to ask probing questions in order to adapt to the answers provided by the interviewee [81]. This approach provides more focus than the unstructured interview while supporting flexibility in order to adapt to the answers provided during the interview [81].

The alternative to one-on-one interviews is to do interviews with a group of participants; where there is more than one participant being interviewed, at the same time [80]. These group interviews are typically referred to as focus groups. Focus groups are said to offer increased validity over one on one interviews because there is more than one respondent commenting on a particular issue – meaning that a deeper sense of what is really happening or opinions are presented, as participants can interact with one another and gauge one another on their responses [80]. Furthermore, within the focus group, the power relation that the respondents might have felt in a one on one situation is mitigated as there are more respondents than interviewers. However, because typically open-ended questions are used it can be easy for respondents to deviate off the intended course, resulting in the different focus groups all producing different responses or data [80]. Furthermore, anonymity is hard to guarantee as other respondents from the focus group could potentially disclose the responses of their fellow respondents with others [80].

During the course of this research study, I used both semi-structured interviews and focus groups in order to understand how teachers were or weren't integrating the use of ICTs into their teaching practices in the SLL, as well as what problems they were experiencing in using or integrating the use of their facilities at their schools. A copy of the interview schedule used in each instance can be seen in Appendix E, while greater detail pertaining to the exact implementation of events around the interviews/focus groups can be read in section 2.7.

### 2.6.2 Computer laboratory experimentation

The primary method of inquiry in science is that of the experiment [83]. A sequence of events, or procedure, used to support, refute or validate a hypothesis is referred to as an experiment [84]. Experiments highlight the relationship between cause and effect [84]; if one changes particular variables in a sequence of events, what will the resulting outcome be? According to McLeod [83], there are three types of experiments: laboratory/controlled experiments, field experiments and natural experiments. In this section, we are referring to the laboratory/controlled experiments, and the next section (section 2.6.3) will discuss the field experiments. McLeod [83] defines a laboratory/controlled experiment as one which takes place in a controlled environment (which might not necessarily be a laboratory). As a result of the laboratory/controlled experiment's controlled venue more accurate measurements are supposedly possible as compared with the other types of experimentation [83]. The advantages of the controlled experiment include, that they are easier to replicate while allowing for precise control which afford the ability to test cause and effect relationships [83]. Limitations include: the unnatural environment of the setting (namely the laboratory) can cause results that are not congruent with those observed in natural or real life settings meaning that results are not necessarily generalizable to real world environments [83]. Furthermore, results can be affected by any bias that might be present in the experimenter [83].

The aim of experimentation is to gain a deeper understanding about the phenomenon under study. In the case of this research study, the reason for conducting laboratory experiments was two-fold. Firstly, it allowed me to experiment with a number of computing infrastructure configurations in order to make appropriate and informed decisions about which computing infrastructure configurations were best suited for deployment in the SLL for long-term real world experimentation. Secondly, it allowed me to gain a deeper understanding about what kind of problem thin-client desktop computing is (CPU or memory intensive) as well as a deeper understanding of how the thin-client (specifically LTSP) works. More details of the computing infrastructure experimentation done in the computer laboratory can be found later in this chapter, in section 2.7.3.

### 2.6.3 On-site deployment and long-term follow-up (field experiments)

As discussed in section 2.6.2, experiments consist of a sequence of events designed to test a particular hypothesis, in order to either refute it or prove it. In addition, in section 2.6.2, three types of experimentation were described, laboratory, field and natural. This section (together with section 2.7.4) discuss the field experiments done during the course of this research work, in deploying selected computing infrastructure configurations to three schools in the SLL.

A field experiment is an experiment that is done in a real world setting [83]. The experimenter still has some control over the variables within the experiment, but there is also an element over which he/she has no or little control, as this is a real life environment [83]. However, the setting and associated lack of complete control, often result in more realistic results as compared to the clean and sterile environment of the laboratory; what McLeod [83] refers to as a "higher ecological validity" [83]. The real world setting

can, however, be a cause for limitation, in that the results in one setting might not be replicable to another, as different extraneous circumstances will likely exist in different real world environments [83].

In this study, I undertook three different computing infrastructure deployments to three of the schools within the SLL. The three schools, were three of the six that were chosen to be surveyed. (Only three schools could be used for testing computing infrastructure configurations due to time and financial constraints.) The purpose of the deployments was to investigate the performance of the computing infrastructure solutions in a real world setting, in order to ascertain if any or all of the deployments were useful in supporting the ICT needs and vision of the schools, while also contributing to the overall goals of the bigger SLL research project. More detail about the exact sequence of events in the deployment process are discussed in section 2.7.4

# 2.7 Research activities

In this subsection I will describe the specific research activities undertaken in order to understand how teachers are using ICTs for teaching and learning and gain insight into how they think about using ICTs for teaching and learning. The intention is that through these processes I will be able to highlight the issues surrounding ICT integration.

Prior to commencing any of the research activities (for both the survey and the ICT deployments) ethics clearance was sought from Rhodes University. I provided copies of the letters to the Department of Basic Education (DoBE) (see Appendix A), letters to the school Principals (see Appendix B), consent forms (see Appendix C), the questionnaire (see Appendix D), and interview schedule (see Appendix E) to the Computer Science Department's ethics committee. Ethics clearance was granted (CS12-01) and the research work could commence.

As previous explained in Section 2.6, six of the seventeen schools in the SLL project were invited to participate in the survey portion of the research study. Permission from the DoBE was first sought by visiting the district office at Idutywa, and then each school Principal was approached. After we had their permission, the school principal suggested the appropriate teachers that they would recommend for inclusion in the survey. Each of those teachers was approached and asked if they would be willing to complete a questionnaire and submit to an interview process (specifically a focus group involving the two teachers together). Teachers were also informed that they could withdraw from participating at any stage throughout the process, which one of the teachers at FH3 did (meaning only one teacher at FH3 was interviewed).

### 2.7.1 Questionnaire

As previous described, a letter was sent to the Education District Officer (EDO) for the SLL schools in the DoBE District office in Idutywa to obtain permission from the DoBE to do the survey at the six schools in Dwesa. Accompanying the letter of request and explanation for the DoBE was a copy of the letter that would be sent to each of the school principals, a copy of the questionnaire that the teachers would be asked to complete, a copy of the interview questions that would mostly likely be asked (given that the questions were to be informed by the results of the questionnaire) and a copy of the consent forms (copies of all of these documents are available in appendices as specified above). A follow up meeting was then held with the EDO at the DoBE offices in Idutywa, where a verbal explanation of the project was offered and any questions were answered. At this point permission was granted by the EDO and the six schools could be approached.

Letters were then sent to each of the school principals at each of the six identified schools, seeking the principal's permission to conduct the research at their school. Once permission was granted by the school

principal (by signing a consent form for their school), they were asked to nominate or suggest two teachers who would be prepared to take part in the survey. Once these two teachers were known, questionnaires and consent forms were sent to each of them for completion. After a period of about two to three weeks (in order to not rush anyone) the questionnaires and consent forms were collected from the schools (personally). I was then able to enter the results of the questionnaires into a spreadsheet and begin to do some basic quantitative data analysis and statistical investigation. The results of the questionnaire helped to finalise the interview schedules and specifically the interview questions that were listed for inclusion. The results of the questionnaire can be seen in Chapter 6, while more details of the interview process are captured in the subsequent section below.

### 2.7.2 Focus groups and Interviews

After the questionnaires had been completed, collected and the data captured with some basic data analysis, the same 12 teachers at the six schools were approached about participating in follow up focus groups. I arranged appointments with the teachers in order to hold the focus groups/interviews at each school. Teachers were under no obligation to participate in the focus groups and Teacher\_F elected to not be a part of the focus group or be interviewed. As such, Teacher\_E was interviewed on her own at FH3. In addition, the teachers from FH4 were not available at the same time in order to run the interview process as a focus group. As a result Teacher\_G and Teacher\_H were each interviewed on a one on one basis.

At the start of each interview I explained to the teachers that their participation was completely voluntary and that they could refuse to participate, or to answer questions along the way. In addition, I informed them again that their response would be completely confidential and that no personal information that could identify them or their school would be made available in any reports. Each teacher was also asked to sign a consent form for the interview or focus group taking place.

The interview schedule consisted of 17 questions for those teachers who were not identified as the ICT teacher. For those teachers who were identified as the ICT teachers (not necessarily employed as such, but had been assigned that role at their schools because they had the most computer related knowledge and experience) they were asked a further four questions about computer related maintenance issues as they would be in the best position to answer them.

In order to record the interviews/focus groups I made use of a Olympus Digital Recorder VN-8500PC. This device produces MP3 files which can be copied to a computer via USB. I used Audacity to play back the recordings while I transcribed the interviews/focus groups. The reason for my choice of Audacity to play the audio back is that I was able to follow the recording visually as well as audibly, which made it easier to stop and start the audio in the correct places. During the transcription process I followed a sequence of listening to a couple of seconds of the recording, then pausing it to type that portion and then re-listening to the portion in order to make sure that I had typed it correctly. When I was satisfied that I had transcribed those few seconds correctly I moved on to listening to the next portion of the recording. I followed this process until I had completed the entire interview. Once the interview was fully transcribed I ran each one through a spell checker and sent off a copy to my supervisors for their perusal. Then a copy was printed out for each of the participants involved in that interview or focus group and I personally delivered each one and requested that the interviewee read through it and make sure that it was accurate; adding or deleting content as they saw fit. This process is known as member-checking [85]. When they were finished with the transcript they were asked to either return it to me or have me collect it from them should they wish me to make any changes. None of the participants asked to return an amended transcript and so it was assumed that I had accurately captured the proceedings of the interviews/focus groups. The validated interview transcriptions and questionnaire data were then used to analyse (more about the analysis

is discussed in section 2.9.1) how teachers are using ICTs for teaching and learning and identifying problem areas that they are experiencing. The results of the interviews/focus groups can be read in Chapter 6.

### 2.7.3 Lab experiments

In order to determine what would be suitable and sustainable ICT architectures that support teaching and learning and are robust and relatively cost effective, I needed to investigate as many solutions as possible. To this end a "desktop survey" was undertaken of a number of ICT architectures that might be suitable in school settings. In addition to this desktop survey, I was able to implement a number of the possible architectures on virtual hosts in order to better understand each solution, their strengths and weaknesses and thus their suitability in a schooling environment.

The various architectures surveyed focused on solutions that offered high availability, while hopefully lowering costs in terms of capital expenditure and easing maintenance. As such, I focused on solutions that would improve upon the standard thin client computer model of Edubuntu and LTSP. Thin clients ease maintenance because only the servers need to be maintained as all data is stored on the servers. This also can result in a lower overall cost as the end points of the system, namely the thin clients, can be cheaper, low-end or refurbished computers. Improving availability and fault tolerance was investigated through using cluster server solutions in order to attempt to make the computing infrastructure more easily healed. Each of the investigated solutions was then tested in a virtual environment, by setting up a virtual system either using VMWare Server or VirtualBox. I tested the following configurations in virtual environments:

- DNS round robin
- Single System image
- Ubuntu LTSP cluster
- Custom virtual desktop
- Edubuntu Thick thin clients

Setting up and configuring each of these solutions afforded me the opportunity to become familiar with the strengths and weaknesses of each solution and gain a deeper understanding into which might be viable solutions in a school environment (the configuration together with the the results of their simulations are discussed in greater detail in Chapter 7).

During the course of the experimentation with the various computing infrastructure configurations, I utilised Sikuli to simulate user interactions in order to test that the solutions were able to adequately support school or classroom interactions (discussed in Chapter 7) and specifically meet the specified requirements of the SLL, as detailed previously in Chapter 1.

### 2.7.4 Computing infrastructure deployments

As a proof of concept, three different architectures were implemented in three of the six schools surveyed in order to assess how each architecture performed in the real world school environment. I used Multi Router Traffic Grapher (MRTG) software to monitor each of the architecture's performance, in order to have a better understanding of how well the architecture was able to support the needs of the school. MRTG was used to monitor the performance of the computing infrastructure in terms of CPU load, memory load, network interface traffic graphs, number of processes and hard disk utilization. These results could be compared with the initial MRTG results gathered from the standard thin client computing infrastructure configuration that had been deployed at FH5, which are described in Chapter 5 of this dissertation.

I implemented a DNS round robin solution in FH5, an Edubuntu Thick thin client solution at FH1 and a standard thin client computing infrastructure configuration with a high-end server at FH2. Greater detail regarding the design and implementation of the three computing infrastructure deployments is available in Chapter 7 of this dissertation.

### 2.8 Ethical issues

During the study I wished to maintain a standard of ethics throughout the entire process, and therefore attempted to conform to accepted standards of social and professional behaviour. Thus this project avoids scientific misconduct and research fraud. Scientific misconduct occurs when a researcher falsifies or distorts the data or the methods of data collection, while research fraud occurs when a researcher fakes or invents data that were not really collected or falsely reports how research was conducted [85].

Semi-structured interviews require a great deal of trust between the researcher and the participants, and a good understanding of what it is that the researcher wishes to do with the research. For these reasons, participants were clearly briefed on what the research was about and what it would hopefully amount to. Thus I was very careful not to create false expectations. Particular attention was paid to the fact that I was not promising to bring free technology to the schools with a working solution. Although I hoped that this would be the long term goal, there were no guarantees that my research would succeed. Thus I was able to avoid deception by making my intended goal known to all [85]. Deception can increase distrust by people who are frequently studied [85], as is the case with people living and working in the SLL. That said, I do acknowledge that it is still possible that the participants could have misunderstood me, but I did attempt to be as clear as possible regarding the research project, process and the intended outcomes.

Ethical research requires balancing the value of advancing knowledge against the value of non-interference in the lives of others. Thus we sought voluntary inclusion by all participants [85]. It was made clear to all participants that they could withdraw from the research at any time or stage should they no longer wish to be a part of the research project. In addition, they were under no obligation to take part in the research from the beginning. All schools decided to participate, but one teacher did not wish to be interviewed but was happy to fill in the questionnaire. A fundamental ethical principle that we followed in the study was: never coerce anyone into participating; participation must be voluntary [85]. It is not enough to just get participants' permission to have them participate, they must also be informed of what the research will entail so that they can make informed decisions [85].

After each of the interviews had taken place all transcripts from interviews were sent to the interviewees so that they could verify our transcripts and that I could ensure that I was not misrepresenting them in any way. This is known as member checking. In addition I have protected the identity of each of the schools and teachers that participated in the study. In order to protect the privacy of the individuals, participants remain anonymous or nameless [85]. In my research I provided each school and teacher with a code name. School codes were based on their former Apartheid school classification and a number created in numerical order. Teacher codes were then assigned in order of the schools by assigning teachers an alphabet letter in order.

When I offered a SLL funded Internet connection and ICT facilities (i.e. servers and/or thin clients) to schools for the purposes of experimentation it was made clear to those schools that the facilities were not guaranteed to work reliably all the time, as the network and infrastructure deployments were not being run as a business, but rather as an experiment and thus those maintaining it would not always able to do so at all times. It was important for schools to understand that the research team (myself and other members of the Siyakhula Living Lab research group) were not able to travel out to the schools at any time of day or night in order to fix any ICT infrastructural problems; typically the research team travelled to the SLL once

a month and stayed for a period of one week. I explained that the research team was not able to provide money for computers or software. I emphasized that the role of the research team was to collect data in order to make known the facts to those who can help schools, such as the DoBE. This clarity allowed me to avoid further misconception or even deception and thus maintain good working relationships with all the schools involved in the research project [85].

# 2.9 Data analysis

Data analysis [86] permeates all stages of a study. Concern with analysis should begin during the design of a study, continue as detailed plans are made to collect data in different forms, become the focus of attention after data is collected, and be completed only during the report writing and reviewing stages [86].

There are commonly two types of data analysis, qualitative data analysis and quantitative data analysis. In the Social Sciences, qualitative research is a broad term that describes research that focuses on how individuals and groups view and understand the world and construct meaning out of their experiences; it is essentially narrative-oriented. Qualitative data can be described numerically using frequencies, or counts, cumulative frequencies and contingency tables or graphically using bar graphs, pie charts and categorised bar graphs [87]. Contingency and multi-way tables are used to describe relationships between qualitative variables. Qualitative is often referred to as categorical variables [87].

In Social Science, quantitative analysis refers to the use of numerical and statistical techniques rather than the analysis of verbal material [88]. In Computer Science, quantitative analysis involves the measurements of quantities [88]; more specifically to this study, the measurement of variables within computing infrastructure. Quantitative analysis involves the use of statistical techniques to understand quantitative data and to identify relationships between and among variables, often referred to as measurable variables [87]. From the questionnaire data and the computing infrastructure experiments and deployments data we were able to perform quantitative data analysis, which involves statistics of some kind from the (typically closed-ended) questions (from the initial questionnaire) or results from computing infrastructure experimentation. In addition, from the open-ended questions asked during the interviews, we were able to do qualitative and to a certain degree quantitative data in order to better understand the community in the SLL and their developmental needs and endeavours. In the case of this study, to understand how and why teachers are currently using or not using computing infrastructure for teaching and learning and what constraints they face in doing so.

### 2.9.1 Qualitative data and analysis

Qualitative data analysis includes the following steps [89, 90]:

- · Capture data
- Code data
- Categorise data
- · Conceptualise data
- Create theory

Data capturing can be done through interviews or observations (and notes are made by the researchers). The interviews are typically recorded and then transcribed which allow researchers to work with data and

begin coding. The coding of data can be done in two ways, through top-down analysis or through bottom up/in vivo coding [89]. Top-down coding is when data is approached with preconceived codes, in other words, pre-emptively expecting to find certain kinds of information in the data and so the coding process involves looking specifically for those kinds of data. This is also referred to as analytic induction or analytic research [89]. The codes that are used are either determined by theory or through a hypothesis that has been proposed. This kind of coding is often criticized by some scholars who believe that instead of the data speaking with its own voice, the ideas of the researcher are expressed, which creates interpretations that uphold the researcher's preconceptions [89]. When using top-down coding, one needs to be very careful to explain the codes used so that work will be better accepted.

In bottom-up coding, or otherwise known as in vivo coding, one works from the data (bottom) up. Upon reading through the text (e.g. interview or observation notes) the issues or areas that need to be coded will emerge, so the data will speak to the reader [89]. However, in this kind of coding, where it is said that there is less of a bias when compared with top-down coding, it is not completely open. No matter how hard one tries, one will bring one's own ideas into the data analysis process and thus there will still be a certain amount of bias in the coding. Codes are usually attached to "chunks" of data of varying size - words, phrases, sentences or whole paragraphs, connected or unconnected to a specific setting. They can take the form of a straightforward category label or a more complex one, for example, a metaphor [90]. Various codes that are slightly similar can then be grouped together into categories [90]. These codes and categories can then be compared to what is known from literature and theory and concepts can be based on them [89].

I used semi-structured interviews or two person focus groups with teachers from each of the six participating schools in order to capture qualitative data about the current use of computing infrastructure for teaching as well as the teachers' perceptions of computing infrastructure for teaching and learning. Once the transcripts had been completed and had undergone the member checking process, I was able to begin the next phase of the qualitative data analysis, namely to begin the coding process.

This involved identifying the issues that were raised during the interviews/focus groups by each of the teacher participants. I began the process by looking for trends or themes in the data. I could see similarities, differences, patterns and structures within the seven interview/focus group transcripts. From this I identified 12 broad categories, these were:

- Infrastructure (I)
- Support (Su)
- Training (Tr)
- Content (Ct)
- ICT Use (IU)
- ICT Integration (II)
- Barrier to use (BU)
- Facilitate use (FU)
- Teachers (Te)
- Learners (Le)
- Community members (CM)
- ICT benefits (IB)

Each of the broad categories was then also further sub-divided in order to note specific themes that were coming through. For example, the Infrastructure category was further divided into:

- Hardware (H)
- Software (So)
- Networking/Internet (N/I)
- Physical (P)
- Finance (F)

Furthermore, some of these subcategories were again divided to better capture and articulate the themes of the data. For example, Hardware was further subdivided into:

- Age (A)
- Limited resources (LR)
- Placement (P)
- Security (Sec)
- Data projectors (DP)
- General comments (G)
- Requirements (Req)
- Longer life span (LLS)
- Reliability (Rel)
- Speed (Sp)

A detailed diagram of the categories and subcategories of the data as it was coded is available in Chapter 6.

Once I had completed coding the interviews in terms of the categories and subcategories that presented in the data, I then went back and analysed the categories through the lens of TPACK (Technological Pedagogical Content Knowledge framework; see section 4.6 for further details). I assigned a colour to each of the TPACK categories, based on the colours of Figure 3.1, namely, Technological Knowledge was assigned yellow; Content Knowledge, blue; Pedagogical Knowledge, red; Pedagogical Content Knowledge, purple; Technical Pedagogical Knowledge, orange; Technical Content Knowledge, green; Technological Pedagogical Content Knowledge, black; and Context, Pink. I then coloured each of the codes assigned to the data according to which of the TPACK knowledge classifications or TPACK context that applied. For an example of this see the screen capture of a portion of the spreadsheet for Interview 1 in Figure 2.4.

### CHAPTER 2. RESEARCH METHODOLOGY AND DESIGN



Figure 2.4: Screen capture of the transcript of Interview 1, showing the data classifications and the colour coded TPACK classifications

Once every one of the in vivo produced classifications (or codes) had been colour coded according to the TPACK framework, I performed a final iteration through the data to look for instances of the use of (explicitly and implicitly) affordances by teachers. I used the Affordance classifications described by Bower [91], as well as three additional affordances, that presented within the data but had no current classification by Bower, namely Efficiency affordances, Communication affordances and Social affordances.

Efficiency affordances refer to how technology improves ones levels of efficiency or help to reduce workload. Communication affordances refer to when technology affords opportunities for communication with other people and places. Social affordances refers to when technology affords users opportunities to engage with others socially; forging or supporting relationships. By coding the data according to these various methods and frameworks it allowed me to identify pertinent issues that teachers face in using technology in teaching practices as well as allows me to analyse and better understand what teachers think and personally understand about the use of technology in teaching in the light of frameworks such as TPACK and Affordances.

This process generated an immense volume of data and more than is possible to include in a thesis dissertation. As such the thesis will focus on the reporting of the data generated around infrastructure. Thus it will predominantly focus on the Infrastructure category from the in vivo coding process, as well as the the Technical Knowledge data generated through the lens of TPACK. I also give some attention to the affordances of the technology as understood by the teachers. The rest of the data analysis will be written up in separate papers and only features when it offers supporting arguments to understanding the data around the computing infrastructure. The results of the questionnaires, interviews/focus groups and the associated data analysis is discussed in Chapter 6.

### 2.9.2 Quantitative data and analysis

Quantitative data analysis was performed in questionnaires, interviews/focus groups, laboratory experiments and the real world field experiments. Generally, quantitative data analysis begins with some theory. Theories can vary between abstract general approaches to fairly low-level theories to explain specific phenomena [92]. In the case of this project, there is no specific theory being tested, but rather a hypothesis that computing infrastructure (ICTs) in education can be beneficial to teachers in helping them teach learners in schools and that more robust computing infrastructure can be employed in order to maximize use of resources and improve reliability. Thus the project looked to support the SLL schools in their use of computing infrastructure. In quantitative data analysis, hypotheses relate to a limited facet of the theory and are used to test the theory [92].

In order to test a hypothesis it is necessary to decide what variables are needed to be known in order to answer the hypothesis. This process is known as operationalization. There are several methods of performing operationalization, three of which were used in this study, namely questionnaires, interviews and experimentation (both in the laboratory and in the field) [92].

During the school survey a questionnaire was sent out to each of the targeted schools regarding their current computing infrastructure, use, reliability and the level of training of teachers in the use of technology. Their responses to the questions provide the necessary variables in order for me to better understand what facilities were available at each school, how those facilities were being used and begin to identify where problems lay with regard to the available facilities. The closed questions nature of the questionnaire made it more easily quantifiable and thus lent itself more readily to quantitative analysis (the results of which are available in Chapter 6). The interviews and/focus groups are not as obviously quantifiable. However, once categorized (through in vivo coding) one can begin to see a pattern of repetition of categories as they emerge from the data. This lends itself to being counted, which is quantifiable, and thus an element of quantitative analysis is possible (and was used) with the interview data, in order to assess which issues were considered more important in the minds of the teachers (results also available in Chapter 6).

In terms of quantitative analysis with regard to computer systems, analytical models describe system behaviour and performance characteristics of the system [93]. During the computing infrastructure testing, experiments were conducted and results thereof recorded in order that the reliability, load and performance of the deployed computing infrastructure could be assessed in order to determine if the reliability and com-

puting needs of the school was being met. Quantitative computer performance analysis consists of discovering and ascertaining the efficiency of a computer system [93]. The findings of a quantitative performance study may be used to guide decisions relating to system design, the allocation of resources, acquisition of additional facilities or the fine tuning of existing configurations [93].

Once all data had been collected it was then analysed and from it findings began to emerge. These findings provided insight into how teachers are using or not using computing infrastructure and what prevents and/or promotes use, together with data about which computing infrastructure might best be suited for use in rural schools in South Africa, such that they support teaching and learning objectives, while being easy to maintain and financially viable given the context in which they are to be used. Details of these findings are elaborated upon in chapters 6 and 7 of this dissertation. From the questionnaires answered by the teachers, together with the MRTG graphs of the hardware performance I was able to gain an understanding into both the teachers, their use of computing infrastructure and their understanding about computing infrastructure, together with how the proposed computing infrastructure "improvements" (the computing infrastructure solutions tested in the three schools) performed in the school environment.

The results of the questionnaires were entered into a spreadsheet for further analysis. This data was used to support and sometimes refute findings from the interviews conducted with the teachers. Furthermore, it allowed me to use numerical data to support or highlight various issues that the teachers had discussed in their interviews.

Results from the laboratory tests were also collected in spreadsheets for further analysis, and graphs created in order to better understand the performance of the proposed changes in computing infrastructure vs the standard thin client computer configuration. Furthermore, the data could be graphed to assess whether thin client desktop computing is a memory bounded or a CPU bounded problem; which provides further insight about what computing resources are more important in this particular type of configuration, in order to get the best performance and thus experience for the end user.

The MRTG records of the performance of the hardware of the various computing infrastructure configurations allowed me to graph the performance of the hardware and assess whether or not the hardware had adequately provided ICT services to the users. For example, MRTG collected data on the performance of the network card, CPUs, hard disks and memory of the systems, describing how the computing infrastructure was or was not providing adequate ICT services or meeting the computing infrastructure demands of each school. This allowed me to determine if each of the computing infrastructure was a viable option in a school environment and if they were able to address any or all of the issues highlighted by teachers during their interviews.

The results of both sets of quantitative data and their analysis can be seen in Chapters 6 and 7.

# 2.10 Conclusion

The post-positivist critical realist research orientation enables this study to bridge the Science and Social Science disciplines in which the study was conducted. Furthermore, it paves the way for the use of mixed methods in order to make a serious attempt to fully understand how best to provide computing infrastructure for the school environment in rural South African schools. This project utilizes both qualitative and quantitative methodologies, data and data analysis techniques in order to attempt to provide a holistic explanation of how computing infrastructure is being used in rural schools, and what variables either hinder or help. It also seeks to begin to understand what teachers think about, or understand about the integration of technology into teaching practices, and how they envisage the use of technology for teaching and learning. As such the research orientation, methodologies, tools and activities employed are all with a view to gaining

as much understanding about the use or lack thereof of computing infrastructure in education practices in rural schools in South Africa.

The next chapter introduces the SLL in-depth and provides a history of the SLL project. It covers the location of the SLL, together with a detailed description of the network and the computing infrastructure. During the description I highlight the shortcomings of the current SLL infrastructure as this project seeks to investigate alternative solutions to fix the problems we have encountered with our computing infrastructure up to this point.

# **Chapter 3**

# **Computing infrastructure deployment: the initial phases**

## 3.1 Introduction

Internet penetration in Africa is low compared to other parts of the World [94]. Despite its strong economy and first-World infrastructure in metropolitan areas, South Africa only averages an Internet penetration of 35.2% (combining access at home, via cellphone, work and elsewhere), as per the 2011 Census [95]. Similarly, only 19.4% of households reported having access to a computer in the 2011 Census [96]. This is largely due to poor ICT infrastructure in rural areas where a significant portion of the population (approximately 37% according to the 2011 Census [39]) live. The fixed-line teledensity in rural areas is less than 5% [97] and the high cost of deploying and maintaining copper and fiber plants make it risky for telecommunication operators to roll out traditional wired technology into rural areas [31]. For these reasons wireless technologies appear to be a more viable solution [34]. Access to the Internet has recently been recognized by the United Nations as a human right [98]. Upliftment of rural areas and poverty alleviation are a priority for development in South Africa as information and knowledge are key strategic resources for social and economic development. In order for ICT interventions to be effective and sustainable, the local community of prospective users needs to be involved in shaping and supporting the intervention [28]. In this chapter I describe a holistic ICT-for-development project, which involves developing and field-testing a distributed, multi-functional community communication platform. Such a platform is specifically designed for marginalized communities in South Africa and its distinctive feature is localization through innovation.

# 3.2 The Siyakhula Living Lab

The SLL (a quadruple helix partnership of academia, industry, government and the community) was initiated in 2005 to conduct applied multi-disciplinary ICT4D research work in the two universities of Rhodes and Fort Hare. The initiative was led by the Telkom Centres of Excellence (CoEs) programme hosted by the departments of Computer Science at both universities. The CoEs are industry sponsored research units within Computer Science and Engineering departments across the universities of South Africa. The Siyakhula Living Lab incorporates researchers from Computer Science, Information Systems, Education, Linguistics, Communications, Anthropology, African Languages and Journalism from both Rhodes University and the University of Fort Hare.

The primary objective of the SLL was to develop and field-test a distributed, multi-functional community communication platform (now called Teleweaver), to deploy in low income, marginalized communities in South Africa. These communities, by sheer size and because of current political dynamics, could represent a strategic emergent market. The second objective of this project was to build technically skilled human resources in ICT, particularly, but by no means only, in the context of marginalized and semi-marginalized communities. Over time the objectives of the SLL have become more refined and include:

- Providing state of the art, long distance wireless telecommunications infrastructure that forms a "broadband island" and provides Internet connectivity;
- Provide appropriate and sustainable computing infrastructure in order to support the various development initiatives of the project as well the the computing needs of the communities;
- Support grass-root activities through harnessing indigenous knowledge for proof of concept of eService applications such as e-commerce, e-government, e-health, e-education, micro-tourism, etc;
- · Continuous participatory and multi-disciplinary action research; and
- Replicate results and seek sustainability in the creation of a software factory Reed House Systems.

The research work from the two universities exist as sub-projects to support the main project objectives [68]. Masters and PhD research has developed throughout the years, to build (from lessons learned and applied) and pilot ICT diffusion in a rural environment. My contribution to the SLL, since its inception in 2006, has been to implement, test and deploy the evolving computing and networking infrastructure (as the primary researcher in this element of the project) within the various communities and schools that make up the SLL.

Siyakhula means "we are growing" in Xhosa, the African language spoken where the platform is currently being built and tested. The name of the project was chosen by the local community and encompasses some of the values underlying the project. The project adopted the Living Lab approach in 2008, hence the name, Siyakhula Living Lab. The user driven approach, central to a living lab, is essential to understand the local requirements and constraints associated with the rural environment, as well to activate the local innovation capabilities. In the SLL, local schools in the area have been targeted as venues for the ICT deployment. The schools were chosen as the points of presence within the community for a number of reasons:

- they have the necessary infrastructure for housing computer labs, such as electricity and appropriate venues to contain computer labs;
- schools are educational centres and are thus in a position to be able to train both local learners and local community members; and
- the schools are open to all community members, allowing access to all.

### 3.2.1 Mbashe Municipality

The Mbashe Municipality is a deep rural area situated along the Wild Coast of the Eastern Cape province of South Africa. The villages targeted in the SLL are adjacent to the Dwesa-Cwebe area, which comprises the nature reserve and adjacent communities which are extended over a land area of approximately 153 square kilometers [99]. The Dwesa nature reserve was one of the first successful land restitution claims in post-apartheid South Africa. The process has been documented through extensive ethnographic research [99]. Both the reserve and the local communities have become involved in a development initiative as the natural environment, consisting of the nature reserve and the coast are assets for the communities. The nature reserve serves to provide income generating activities to support rural development in the area [99]. The unspoiled natural beauty and wild beaches have the potential to significantly promote ecotourism in

the region. Furthermore, the high levels of rainfall and rich soil in a typically dry area of the country, has potential for controlled agricultural intensification and commercial forestry. Furthermore, the area is in the heartland of Xhosa culture and a centre for the production of traditional Xhosa art, music and dance. As such, it has the potential to foster small entrepreneurship and attract cultural tourism.

The Mbashe municipality is faced with socioeconomic challenges associated with poverty and poor development. Some characteristics of poor development in the area include: severe local limitations of the road network, limited access to government services and delivery, inadequate education and health care facilities, constrained access to national grid electricity, and poor/vandalized telecommunication (land-line telephones) infrastructure in some areas. Consequently, this area has over the years been the target for developmental projects. ICTs can support rural development by enabling communication which can facilitate development activities. The importance of developing ICT skills and their potential contribution to rural development as well as the preservation and promotion of indigenous knowledge and languages are recognized by the South African Government [25] and are perfectly aligned with the activities of the Siyakhula Living Lab.

### 3.2.2 ICT infrastructure

My role, as a member of the SLL research team and the project coordinator of the CoE at Rhodes University, was to design, test, deploy and maintain the ICT infrastructure (both computing and networking infrastructure) within the SLL. As mentioned, the primary objective of the SLL is to develop and field-test a distributed, multi-functional community communication platform, as discussed in Section 3.2. As such a reliable local loop access network was required; we decided to make use of wireless technologies as there was no fixed line infrastructure in the area which we could use. WiMAX, more specifically the Alvarion BreezeMAX technology, was chosen to build a wireless local access loop. The communities access the SLL infrastructure and communications platform via the Digital Access Nodes (DANs) at the schools where the DANs are hosted.

The next three sections of this chapter will detail the networking infrastructure, the computing infrastructure and the hardware and software choices utilized in the creation of a community-wide ICT (computing) broadband island.

### 3.2.3 SLL network: The broadband island

The Siyakhula Living Lab network can be described as a community-wide broadband network that connects a number of DANs, housed on school properties, within the various communities the SLL operates. The network spans approximately 20 square kilometers and uses WiMAX technologies from Alvarion to share Internet access (via VSAT solutions initially from Telkom, then Telkom and Internet Solutions and currently YahClick/Vox and SAAB Grintek) as well as other services such as VoIP and shared local web content (for the purposes of education mainly, but with a view to increasing local services to other areas too, such as health and governance). The choice of VSAT was because there is no fixed-line alternative in the area of the SLL in the Mbashe Municipality and VSAT offers greater through-put than the isolated 3G connectivity in the area.

The research team, of which I was a part, had good experience in building wireless networks over long distances in order to connect schools. My own masters work was focused on building WiFi networks for providing Internet access to previously disadvantaged schools in peri-urban areas [100]. From our past experience we knew the potential shortfalls of using WiFi over long distances. As a result we wanted to provide a network solution that was more robust. At the time, early 2006, there was a lot of promise about what was then a relatively new, technology called WiMAX; described in both academic literature and to

the consumer as "WiFi on steroids" [101, 102]. The promise of WiMAX was that it offered solutions to the pitfalls of WiFi; namely, it could be used over longer distances, without clear line of sight, supported higher data throughput and was less susceptible to interference (greater details about WiMAX are discussed in Section 5.4.4 of Chapter 5). As a result, it was decided to use WiMAX technology to build a local loop network that spanned the targeted communities; the broadband island. Fortunately, the CoE at the University of Fort Hare had an industry sponsor, SAAB Grintek, who specialized in and resold WiMAX equipment. SAAB Grintek are suppliers of Alvarion WiMAX technologies. Through their support and expertise we were able to acquire Alvarion WiMAX micro base stations and customer premises equipment (CPEs). Furthermore, SAAB Grintek provided us with training and support materials in order to configure and deploy WiMAX networks within previously disadvantaged communities and more specifically within the communities participating in the SLL.

### The initial network

The first school to join the SLL project was Mpume Junior Secondary School (JSS). As a result and because the school was one of few with electricity at the time (late 2005), it was chosen as the site for housing the first VSAT installation, provided by Telkom, in early 2006. This link from Telkom provided a 512 kbps down-link and a 256 kbps up-link with a 3 GB cap. Later, during the same year (2006), Ngwane JSS joined the SLL project. At the time of the first site visit to Ngwane it was soon realized that the school was geographically higher than Mpume (see the topographical map of the area with Ngwane and Mpume marked on it in Figure 3.1). This meant that Ngwane was a better site for housing the WiMAX base station that would be used to connect identified schools to one another and the shared Internet access. WiMAX technologies do not require a direct line of sight (LOS) like WiFi, however large obstructions will still affect the signal path and either disrupt or prevent communication. Thus a high site is still required so that the best possible path is available for wireless communication between the base station and the CPEs. Despite Ngwane being chosen as the location for the base station it was decided to leave the VSAT unit at Mpume (where it had originally been installed) in order to ensure that no one school was responsible for all the network facilities. Rather, the schools and communities needed to work together; pooling their resources and collective capacities to jointly run and operate the network for the benefit of all.

The initial Alvarion WiMAX base station was the BreezeMAX 3500 Micro base station ( $\mu$ bs), running software version 1.0.2. Together with the  $\mu$ bs, the SLL sourced five vertically polarized BreezeMAX CPEs (BMAX-CPE-ODU-AV-3.5) in order to connect up to five schools to the  $\mu$ bs. The  $\mu$ bs was configured and installed at the Ngwane school. The BreezeMAX 3500 Micro base station used 256 OFDM carriers; used the modulation schemes listed in Table 3.1; supported bandwidth of up to 14 MHz; operated in the 3.5 GHz band, the up-link operating in 3.3995 to 3.4535 GHz and the down-link in 3.4995 to 3.5535 GHz; and used Frequency Division Duplex (FDD) [103]. In South Africa this frequency requires a license. Amatole Telecommunications, a USAL in South Africa and a partner in the CoE project owns the frequency license for that spectrum in the Mbashe Municipality and we were given permission to operate the network in collaboration with them. We mounted a 13 dBi omni-directional antenna and the  $\mu$ bs's radio interface on a 3m pole attached to the wall of the school, at the apex of the roof. This resulted in the omni being at an elevation of approximately 6m above the ground. The LMR cable from the radio unit then ran into the building, to the classroom that housed the school's computer lab, where the indoor unit of the  $\mu$ bs was housed in a rack.

Three additional schools around Ngwane were then selected to join the SLL network. These schools were Mthokwane JSS, Nondobo JSS and Nqabara Secondary School (SS). At each of these schools, to-gether with Mpume JSS, we installed a CPE and a router in order to connect them to the WiMAX µbs at Ngwane. Each school is typically referred to as a DAN, and are the vehicles through which the commu-



Figure 3.1: Topographical map of SLL area around Ngwane and Mpume. The three additional schools that were added to the network later are also marked, these are, Nondobo, Mthokwane and Nqabara

Modulation Schemes
BPSK 1/2
BPSK 3/4
QPSK 1/2
QPSK 3/4
QAM16 1/2
QAM16 3/4
QAM64 2/3
QAM64 3/4

Table 3.1: Alvarion BreezeMAX 3500 µbs Modulation Schemes



Figure 3.2: The original SLL network diagram configuration

nity members (and schools) have access to the SLL infrastructure, including the Internet. Each DAN was equipped with a thin client computer lab running Edubuntu Linux and approximately 5 to 20 thin clients, depending on the size of the school's computer labs and the level of security available. At each DAN there was a community access point (CAP) which provided access to the local loop WiMAX network for all the clients at each site. The CAP was a FreeBSD router that was configured to manage and monitor each DAN through a number of services such as SMTP and Netflow. The CAP acted as a gateway between the local area network (LAN) within each school and the bigger local loop access network (the WiMAX network). The CAP ran a Point-to-Point Protocol (PPP) client, specifically PPP over Ethernet (PPPoE), which contained the school's user-name and password for authenticating with the access concentrator housed at Mpume. Once authenticated and the link has been established the CAP would route all outgoing traffic, intended for one of the other schools (such as local VoIP traffic) or the Internet, onto the next hop which was the access concentrator. The Access Concentrator (AC) thus terminated all incoming PPPoE connections from the other four school sites in the SLL local loop access network. Traffic from the other schools was switched at the µbs and sent to the Mpume school where the AC terminated the PPPoE sessions. The four connecting schools (Ngwane, Mthokwane, Nondobo and Nqabara) were reliant on the AC being up in order to reach the Internet, while Mpume was reliant on access to the base station in order to make use of other local services such as VoIP and access to shared resources (off-line content) that were housed at the other schools. A diagram of the network as it was originally configured and deployed from August of 2006 to August of 2011 is available in Figure 3.2.

### The expanded network

In 2011 SAAB Grintek offered to fund the expansion of the SLL network. The expansion would see the old fixed or nomadic µbs replaced with a new mobile WiMAX base station and the current connected schools'

Modulation Schemes
QPSK 1/2
QPSK 3/4
16QAM 1/2
16QAM 3/4
64QAM 1/2
64QAM 2/3
64QAM 3/4
64QAM 5/6

Table 3.2: Alvarion BMAX OD MICRO 3.5 µbs 4Motion Series Modulation Schemes

CPEs upgraded to the mobile WiMAX CPEs. Furthermore, the remote schools that connected to Ngwane would be increased to a total of 10 schools (six new schools would be added to the network). The change from the older WiMAX equipment to the new mobile WiMAX equipment would free the older µbs to be deployed elsewhere and begin a second network "cell". The new mobile WiMAX base station was a MAX OD MICRO 3.5, able to support up to two antennas (to enable MIMO), together with the Mini-Centralised Access Service Network (ASN) gateway in the 4Motion series from Alvarion. The CPE for the new mobile µbs were the 16e BreezeMAX PRO Outdoor Radio units operating in the 3400-3600 MHz range (BMAX-CPE-ODU-PRO-SA-3.5) from Alvarion. The new mobile µbs also supported OFDM modulation and used the modulation schemes listed in Table 3.2; used OFDMA as its access method; supported bandwidth channels of 5, 7 or 10 MHz; operated in the 3.5 GHz band; and used Time Division Duplex (TDD) as its operation mode [104].

The upgrade of the WiMAX technology at Ngwane from a fixed WiMAX base station to the mobile 802.16e WiMAX base station allowed the SLL to experiment with the use of nomadic stations through the use of USB WiMAX dongles together with laptop computers or WiMAX enabled mobile handsets that community members could use from their homes. Badi SS, a secondary school approximately 10 km from Ngwane, was chosen as the host of the new second "cell" because the school infrastructure had recently been renovated as part of the Dinaledi initiative to create centres of excellence in Mathematics and Science Education and had a secure and fully furnished 30-seater computer lab. The school also had a water tower, which could be used to mount antennas onto in order to ensure links with remote schools. The second "cell" allowed the network to expand to more sites.

In order to identify potential schools for inclusion in the expansion of the network, two methods were employed: a desktop survey and a site visit. The desktop survey produced a visual representation of potential schools on a topographical map, using Radio Mobile and Google Earth, based on a comprehensive list of the possible schools in the area and their GPS coordinates (provided by the Eastern Cape Department of Basic Education). The set of potential schools had to be narrowed down to 12 due to limitations in funding and prospective technical support. Numerous factors played a role in selecting schools to be included in the newly expanded SLL. One of the key technical considerations was the location of the schools; we needed to connect schools that were within a 20 km radius of Ngwane as, firstly, 20 km was the maximum distance at which we had reliably connected remote sites to a µbs. Secondly, the 20 km radius limitation eased the logistical constraints of moving around and supporting the numerous schools. Another technical consideration was that the area around the Dwesa-Cwebe nature reserve is mountainous and there are many areas hidden in the shadows of higher mountain tops (radio dead zones). We were able to make use of Radio Mobile to determine the likelihood of connecting schools to either Ngwane or Badi. Even when considering schools within a 20km radius and not located in dead zones, we still needed to shorten the list. At this point we considered the schools with at least one educator enrolled in the ACE ICT course (teach-



Figure 3.3: Topographical map of SLL area showing all the old and new schools in the SLL network, connected to either Ngwane or Badi

ers from the communities in and surrounding the SLL network had enrolled in an Advanced Certificate in Education course, specializing in ICTs) at Rhodes University. This yielded a list of ten schools: Lurwayizo SS; Zwelidumile SS; Lurwayizo JSS; Ngoma JSS; Ntubeni JSS; Ngqeza JSS; Lower Nduku JSS; Lower Ngwane JSS and Kunene Senior Primary School (SPS). While the desktop survey had its merits in narrowing down the list of possible schools, we still needed to conduct a physical site survey in order to confirm that the schools could in fact connect to the base stations at either of the two schools (Badi or Ngwane) and to ensure that each school had no physical constraints, such as a lack of power, preventing their inclusion in the expansion. Our site survey revealed that one of the ten schools had no electricity with no time frame for connection. A further three schools (Lurwayizo SS, Lurwayizo JSS and Ngoma, JSS) were soon to be connected to the power grid (end of July 2011). In addition, the school principal of Ngwane, a key driver within the adoption of ICTs in the community, suggested a further three new schools (Mevana JSS, Nquba JSS and Hlabizulu JSS) all of which are connected to the power grid. Another significant finding during the site survey was the need for a 12 m mast at Ngwane and Badi in order to connect some of the new school sites and in order to connect with each other. Combining the findings from the desktop and site surveys we found that the following six new schools could be connected to Ngwane: Lurwayizo JSS, Lurwayizo SS, Ngoma JSS, Ntubeni JSS, Lower Ngwane JSS and Ngqeza JSS. While, the following five schools could be connected to Badi: Kunene SPS, Zwelidumile SS, Nguba JSS, Hlabizulu JSS and Mevana JSS (see Figure 3.3 for a graphical representation of the expansion network, created using Radio Mobile). Within the group of new schools we had a combination of schools that were closer to the original communities (expanding the facilities there) as well as some schools much further away, such as Zwelidumile SS and Lurwayizo SS, allowing new communities to have access to technologies. In addition, the schools that were further away allowed the project to test to the capabilities of the WiMAX equipment over greater distances (up to 16km hops).

The new mobile WiMAX µbs's CPEs were able to act as routers, where the older generation fixed WiMAX CPE were wireless bridges. For this reason, we did not need additional router devices at the schools that connected via the mobile WiMAX µbs. It was decided that for the older CPE, that would be used in the new cell to connect to Badi, which did require additional router devices to use Mikrotik RouterBOARD RB750G. The decision to stop using old computers running FreeBSD (referred to as CAPs in subsection 3.2.3) was because the Mikrotik routers would consume less power, be more easily stowed away (small form factor), and are less complicated to use and maintain, while still providing the same level of functionality needed in the satellite schools.

The network expansion was then undertaken in three phases:

- Phase 1: upgrade the existing network and deploy the new cell. At Ngwane this entailed having
  the new mast installed and replacing the old base station with the new mobile WiMAX unit. At
  Mpume, Mthokwane, Nondobo and Nqabara we replaced the CPEs with the new mobile WiMAX
  units. While at Badi a second mast was installed together with the old base station. Link Badi to
  Ngwane, such that should the VSAT at one of them become unavailable traffic bound for the Internet
  could be rerouted to the alternative VSAT link.
- Phase 2: connect seven new schools. Ntubeni and Ngqeza received the new Mobile WiMAX CPE. While Zwelidumile, Kunene, Nquba, Hlabizulu and Mevana all received the old fixed WiMAX CPE.
- Phase 3: connect the remaining schools (Lurwayizo JSS, Lurwayizo SS and Ngoma) to the mobile WiMAX base station when electricity became available.

Phase 1 was fully completed by the end of 2012 and Phase 3 by the end of 2014. Phase 2 was only partially completed. The portion of Phase 2 that saw the new schools connected to Badi is still to be completed due to limited human resources and available funds. At the time of the expansion, we also procured a second VSAT connection (one for each of the root nodes of our two network cells). The original Telkom VSAT link was moved from Mpume to Badi and a new VSAT link from Internet Solutions (IS) was procured for installation and use at Ngwane. The new VSAT link from IS provided a 1 mbps down-link and a 256 kbps up-link.

### The current network

The VSAT links to the Dwesa network were upgraded again at the end of 2014, in order to provide connections that offered faster speeds with higher caps for less money. A single service provider, YahClick from Vox Telecoms replaced both the IS and Telkom VSAT packages. Each connection, still housed at Ngwane and Badi, provided a 10 mbps down-link, 2 mbps up-link and have 30 GB monthly caps (month data usage limit). In addition, in March of 2015, SAAB Grintek agreed to provide a third VSAT link (run by themselves) at another school within the SLL network. Mpume was chosen, as it was also the site for a newly refurbished computer lab, also supplied by SAAB Grintek. This link provides 2 mbps down-link and a 512 kbps up-link, but is uncapped. These three VSAT links are the current (at the time of writing) methods of traversing to the Internet from the SLL network. All the schools in the network share these three Internet connections - depending on which root school they connect to, either Ngwane or Badi (currently no satellite schools connect to Mpume). All the units have a public IP address and this allows us to reach the core routers from outside of the SLL network, particularly from our university campuses. What this means is that we can reach either router and perform maintenance to the core routers or the Edubuntu servers or other servers running within the network.

The mobile WiMAX  $\mu$ bs was installed at the beginning of 2012, but over the next four years (up to 2016) provided very erratic service. We found the equipment more susceptible to faults (as compared with the

original fixed-wireless  $\mu$ bs). In addition, the complexity of the new mobile  $\mu$ bs made it difficult to remotely manage and configure, and the knowledge of the SLL team across the two universities with respect to the mobile  $\mu$ bs was very limited. Indeed, the knowledge of the mobile  $\mu$ bs amongst the technicians from SAAB Grintek was also limited. As a result maintenance became impossible and the network was often rendered unusable, sometimes for months at a time. For this reason, the SLL team made the strategic decision to replace the  $\mu$ bs with an old fixed-wireless unit. Fortunately, we had spare fixed-wireless  $\mu$ bs units that SAAB Grintek had donated to us and we were able to deploy one of those spare units to Ngwane and remove the mobile  $\mu$ bs. In addition, we were also in possession of enough CPE units for the fixed-wireless WiMAX technology, and so all the satellite schools that connected to Ngwane could have their mobile WiMAX CPE replaced with fixed WiMAX CPE. This replacement process was undertaken in the middle of 2016.

Thus, at the time of writing, the two WiMAX base stations at Ngwane and Badi are both fixed WiMAX µbs units (802.16d). Neither are capable of mobility (which really never became a requirement), however, nomadic users are possible; the units are suited to fixed wireless installations and are easy to configure and maintain and can be done via a telnet session making remote configuration possible. The two WiMAX base stations work in a Point to Multi-Point configuration with omni-directional antennas radiating signal uniformly through 360 degrees. The satellite schools connect to their nearest base station using the fixed-wireless (WiMAX) CPE. As mentioned, the old base station CPE are wireless bridges and require the installation of separate routers in order to allow the local LAN computers to communicate beyond the local LAN. Again, we use Mikrotik RouterBOARD RG750G as the routers at each school.

In August 2016 there are two core routers in the SLL network, one at Ngwane and one at Badi. These two routers are responsible for routing all the traffic generated in each of the wireless cells of the larger broadband island. The router at Ngwane is a Mikrotik RouterBOARD RG750G, while the router at Badi is still an older rack mount Atom processor server (purchased and installed in 2010) running FreeBSD 9.3. Each of these devices is responsible for routing all traffic on the two network cells in the SLL network. Both devices are configured with static routes. The router at Badi also provides a proxy service (using Squid) to the clients at Badi and the school networks that connect to Badi. The Squid service at Ngwane, however, runs on the Edubuntu server at the school. The logically network, as it is currently configured for the SLL network (August 2016), is depicted in Figure 3.4.

Before concluding this section on the networking infrastructure present in the SLL, it is worth noting that the continued use of WiMAX as the technology used to provision the broadband island is limited. Few WiMAX products ever entered the market and as such are still rather expensive when compared with WiFi technologies. Furthermore, there are currently few telecommunication companies operating WiMAX technology on a commercial basis in South Africa, and given that it would seem that WiMAX has "lost" its battle to LTE (many networks in the USA which employed WiMAX have shut them down [105, 106]), there seems very little chance that WiMAX technologies will be deployed on a large scale anywhere in the world. Large scale deployments of LTE, fixed LTE and WiFi networks are more likely.

### **3.2.4** Computing infrastructure in the SLL

From the very beginning of the project the SLL has used a thin client topology as the primary means of computing infrastructure that should be deployed in schools for access by the schools and the community members. Furthermore, we have chosen to use free and open source software solutions (operating systems and user applications) as they are cost effective, robust and promote access to knowledge (i.e. users are able to see the source code, understand how software works and modify it (or get members of the SLL team to modify it) in order to better meet their needs). The reasons for providing computing infrastructure via thin clients over thick clients is that of cost effectiveness and sustainability. The reasons for which we have



Figure 3.4: Logical network diagram of the SLL network as of August 2016

chosen the computing infrastructure to take on the form of "standard PC" over something more trendy, such as a tablet, are three-fold. Firstly, there is a historical aspect to our choice of the "standard PC" form; when the SLL began in 2006 the use of smart mobile phones was limited (widespread adoption beginning in the late 2000s), while widespread consumer uptake of tablet PCs began in 2010. Secondly, smart phones and tablet computers are personal devices, while a more generic desktop computer (even if a thin client) can be more readily thought of as a shared commodity. They they more readily lend themselves to being shared by teachers, learners and community members. Thirdly, and finally, we believe that users in rural communities will derive much more benefit from being producers of information and not just consumers of information. High end cell phones and tablets are devices that essentially promote the consumption of information but do not readily afford production of information – for example, it would incredibly tedious to attempt writing this thesis on a tablet rather than at a computer with a keyboard, monitor and mouse. As producers of information members of the school and local communities can actively contribute and take part in the digital networked society. It is not to say that this will not change in the future; as technology advances manufacturers might provision new methods in order to support mobile devices being more readily useful in the production of knowledge or artefacts.

It is worth noting that while the underlying focus of computing infrastructure in the SLL schools has been on a "computer lab", as in a specific area of the school where a number of computers are located for teaching specific classes, the architecture we have proposed, implemented and deployed in the schools supports single computers in the classroom, for typical class work such as presentation, demonstrations or Internet surfing. Also, terminals can be located in the secretary's and Principal's office for more administrative types of work.

### 3.2.4.1 SLL Hardware

A thin client computer is one which depends primarily on a central server for all its processing activities and are used to collect input from the user (to the applications running on the server) and convey the output (from the application on the server) to the users. The thin clients are diskless, which means that they have no software installed on them nor are they able to suffer from hard-disk failures. Using the Preboot eXecution Environment (PXE) [107] enabled network cards the thin clients are able to request the necessary software for booting from the central server. All the software, user files and profiles are stored centrally on the server, making it easier to upgrade or add software.

Because the thin clients don't need hard disks and they are not required to do any processing, older and cheaper refurbished computers can be used to act as the thin clients and only the central server requires a significant financial investment. As a result the investment required in setting up a computer lab is reduced when compared to thick client computer laboratory configurations, where thick clients (which are more expensive) and a central server are purchased, although this can be omitted naturally.

### 3.2.4.2 SLL Software

The choice of software used in the schools is coupled with the choice of running thin client computer labs. We use Edubuntu Linux. Firstly, Edubuntu is an operating system aimed at education and the computer labs are in the local schools. Edubuntu is based on the Ubuntu operating system but includes educational applications, tools, content, and themes. Edubuntu was developed in collaboration with teachers and technologists from multiple countries. The educational applications are aimed at users aged 6 to 18 [108]. Secondly, Edubuntu ships with the LTSP as standard which allows thin clients computers to boot off a central server running the LTSP service [108]. LTSP is a free and open source terminal server for Linux-based operating systems. Thirdly, Edubuntu Linux is open source software, which means that source code of the

software is available to the user; affording them the opportunity to study and change the software and then redistribute to others [108]. This means that the software directly supports the tenets of education, allowing users (both school learners and and local community members) to learn how software works, make changes or additions, and thereby create their own locally useful versions of various software packages. This, the SLL research team believes, can only help to foster the development of human capital within communities. Finally, the Edubuntu Linux operating system, together with a number of other user applications, such as the LibreOffice Suite and the Firefox web browser, are available in some of the South African local languages, such as isiXhosa, the dominant language in the Mbashe municipality. More details about Edubuntu and LTSP are described in Chapter 5.

### 3.2.4.3 SLL Computing infrastructure configuration

The original computing infrastructure configuration can be seen in Figure 3.5. From the diagram it can be inferred that a wired LAN is deployed within the school campus, to which the central server and thin clients are connected. Figure 3.5 suggests that the clients are present in a laboratory configuration, but the clients can, in fact, be available through out the school as well as within a traditional computer lab, as said earlier; thin clients can also be placed within administration offices (such as the school Principal or school clerk's offices) and also distributed throughout the classrooms if required.

Access to the network is extended to mobile devices, such as laptops, cellphones and tablet PCs, by the addition of one or more WiFi access points through the school. This will further facilitate teachers and learners (as well as local community members making use of the facilities) accessing the network and its resources from anywhere on the school property. These devices were, however, not provided by the SLL project and would be the private devices of the teachers, learners and community members.

The LAN was in turn connected to the wider community network (and its shared resources; VoIP, local web services, etc) and the Internet via the network connection present at the school using the WiMAX network described earlier, but could be done using any number of alternative networking technologies that might be available in other communities.

The advantages to this configuration have already been touched on section 3.2.4.1, however, I will elaborate on those here. The advantages to thin client computing are numerous and include: they are easy to deploy as they require no software installation; provide single sign-on, so any workstation can be used to access ones account and files; afford the opportunity to use older PCs as thin clients or low-cost minimalist hardware both of which reduce the cost of the computing facilities; offer reduced security threats; and they are easier to maintain as only the server needs to be maintained - if a client PC fails there is no loss of data as everything is stored on the server and the client can be replaced with another client. The disadvantages of this model are that the server has to work very hard as it carries a large computational load, specifically loading to its CPU, memory and network interface in order to support all the clients, and most significantly is a single point of failure within the configuration. This means that if the server fails for whatever reason, the computing infrastructure within the DAN is rendered useless as none of the clients can work without access to the central server.

Figure 3.6 reflects the load on the server at Ngwane that was in service from 2009 to 2013. Each of the images depicts one month's worth of load on the server. Figure 3.6 (a) shows the load on the CPU, Figure 3.6 (b) shows the load on the network interface, and lastly Figure 3.6 (c) shows the amount of free memory available on the central server over the period of one month.



Figure 3.5: The original design of a generic computing infrastructure (typically as a computer lab) in a DAN at a school in the SLL



Figure 3.6: MRTG graphs of the load experienced by a single central server, serving 20 thin clients, at Ngwane over the period of one month (this server was in place from 2009 to early 2013). (a) CPU utilization over four weeks (b) network interface utilization over four weeks (c) memory utilization over four weeks

From the graphs it can be seen that that during peek times when the lab was full and the majority of the clients were being utilized the CPU (Figure 3.6 (a) ) of the central server was being utilized at 100%, which would have resulted in a very high CPU load during these times. Furthermore, during the same peek

time slots, the amount of free memory (Figure 3.6 (c)) available was minimal, indicating that the server's memory was also being heavily utilized. While Figure 3.6 (b) shows that during those same peek time slots there was increased network activity on the network interface, which is to be expected as there would have been a lot of network traffic during those times as the clients were communicating with the central server and the server sending data back to the clients. From the graphs though it could be seen that the biggest effect of the increased load was to the CPU and the available memory.

As such there are two problems that have been identified with the thin client computing infrastructure topology used in the SLL, namely, the single point of failure of the server, together with the high demand on the CPU and memory resources of the server when simultaneously servicing multiple clients. A possible solution to provisioning more computing power (in terms of CPU and memory resources) in the server would be to investigate the usefulness of both cluster and grid like solutions. Furthermore, both of these computing infrastructure paradigms could be possible solutions to the overall ability of the computing infrastructure to be more easily healed. Furthermore, it is possible to build cluster and grids out of older, refurbished computers or low-end computers, reducing the overall cost implications.

The rest of this thesis will focus on my endeavours to test the notion of providing cluster and grid based configurations for thin client LTSP computing infrastructure in schools as a model solution to the computing needs of schools and community members in previously disadvantaged, rural areas of South Africa.

# 3.3 Conclusion

This chapter presented a comprehensive discussion of the SLL; focusing on the networking infrastructure evolution together with the first phase of the computing infrastructure model. In the chapter I highlighted the two pitfalls of the thin client computing infrastructure topology that need to be addressed, namely, the single point of failure and the large demand on computational resources, specifically CPU and memory.

The thin client computing infrastructure model in use in the SLL in Dwesa makes the most sense as the primary computing solution for schools and communities in rural and marginalized areas in order to provide functional computing infrastructure that supports users as both producers and consumers of information. This model can easily be supplemented by use of more trendy mobile devices such as smart cellphones and tablet PCs. However, the single point of failure of the central server, as well as the large load that the server experiences in servicing a large number of clients needs to be addressed. In this thesis I explore technical solutions to add more computational resources while containing the costs, as well as methods for improving the ease with which the computing infrastructure can be healed. Furthermore, the thesis describes an investigation into what teachers perceive as important for computing infrastructure in order to support teaching and learning activities within rural schools in the SLL network. From the exploration of technical solutions for the provision of increased computing resources which are more easily healed together with the perceived requirements of computing infrastructure as explained by the local SLL teachers I propose a computing infrastructure model for use in the SLL. This model was then deployed and tested and from this a revised model was derived that was tested in real world environment case studies. The next two chapters discuss the necessary literature required to understand the use of computers and ICTs in education as well as the various technologies available for use and deployment within schooling environments.

# **Chapter 4**

# **Understanding ICTs in Education**

## 4.1 Introduction

This chapter begins with a discussion around the use of technology, and more specifically computing infrastructure, within education, attempting to focus more specifically on school level education and appropriate technologies for inclusion within teaching and learning spheres. This is followed by a discussion of the concept of affordances. Understanding the affordances provided by computing infrastructure begins to facilitate a conversation about how one might adequately begin to integrate the use of computing infrastructure into teaching and learning processes such that they support those activities directly. The fourth section of the chapter then discusses ICT integration, and the pedagogical practices and policies that help to support the integration of ICTs into teaching and learning. Following from this, I look more closely at TPACK, which is a framework specifically geared to support the integration of technology into teaching and learning practices and from which one can gain a deeper understanding of how teachers might best achieve successful ICT integration. Finally, this is followed by some concluding remarks for the chapter.

# 4.2 Use of technology in Education

In order to provide a framework for identifying appropriate ICTs for integration in teaching and learning, I begin with an investigation into the past, current and possible future uses of technology in teaching and learning. The use of computers in education is not a new endeavour, early computers were designed and built at American universities during the 1940s and were used to perform complex mathematical calculations [109]. The concept of large, centralised computers (mainframes) dominated the field of technology use for 30 years after the creation of the first computer in the 1940s [109, 18]. Universities, along with governments and corporations were the customers of the computer manufacturers, and within this context of mainframe computer use, the first educational computer applications were developed [109]. Applications included programs like PLATO [110] and TICCIT, which were designed to take advantage of the mainframe's strengths [109]. Educators used these systems to develop text-based tutorials and tests which learners could access outside of normal classroom hours [109, 18]. This use was generally targeted at large entry-level courses where knowledge acquisition was the key learning objective [109].

Following the mainframe computer, in the 1970s and 1980s, was the personal computer (PC), which has had a "tremendous, yet mixed impact on education" [109, p382]. Increasingly, computers were introduced to schools, as specialised computer laboratories (or labs) and in individual classrooms [109, 18]. The result of the PC was to lower the barrier of entry to access technology, especially to schools. However, while educational applications do exist, they often employ limited pedagogies such as drill and practice [109, 18]. In addition, as a result of the limited choices of software packages and in some cases the costs involved,
often the primary use of computers in education involved work with productivity tools such as Microsoft Office, purchased not because of their potential educational benefit but because they represented the state of the consumer market [109].

Today, 2017, this same reasoning is often used to purchase software for teaching and learning. As such many applications are not specifically designed for educational purposes; for example, Microsoft Office was designed for use in business, while web technologies like blogs and podcasts were designed for entertainment, communication and social networking. Watson [20] argues that the continued emphasis on using applications that were designed for business (together with policies that focus on workforce training) contribute to the minimal use of technologies for teaching and learning [20]. As a result, it is important that teachers have the required knowledge and skills that enable them to appropriate technology for pedagogical purposes [3].

Following on from the use of the stand alone PC, the next technological development to have an impact on the world was the creation of the Internet. One of the most successful Internet experiments was the World Wide Web (WWW). The Web was originally conceived as a way of sharing and linking documents (files such as graphics, audio, video, etc) amongst a group of researchers [109]. Educators have taken advantage of the Web as a tool for communication. Learners and teachers can converse, collaborate and debate with a large and diverse audience through this medium [109].

The use of technology has also resulted in a rapid growth and change of knowledge in numerous fields of study, resulting in changes to those fields of study and learning, and introducing new theories and concepts as a result of new representations and manipulations afforded by new technological developments [3, 18, 20]. New understandings have emerged from the development of new technologies that afford the representation and manipulation of information and ideas in new ways, prompting fundamental changes in the nature of the disciplines [3]. For example, in medicine the introduction of x-rays resulted in changes to diagnostic processes and medical knowledge, while similarly carbon-14 dating revolutionised the field of archaeology [3].

Typically, in many schools in both the developing and the developed world access to technology in schools is via computer laboratories [109, 19, 111, 112, 113, 20] primarily, and secondly through access to a few computers in the classroom, sometimes coupled with an interactive whiteboard and data projector, or just a data projector [111]. Computer labs provide a shared resource to the school with equal access and support, while circumventing space and cost constraints [109, 19, 111, 112]. Computer labs consist of anywhere between one and forty computers which are either stand-alone PCs or networked together as thin or thick clients [19, 112, 113]. In addition, some schools will also place computers in staff rooms for use by teachers. Some schools provide combinations of computer labs, computers in the classroom and computers in the staff room [111]. The Enlaces project [111], is a Chilean ICT project that placed ICTs in schools around Chile, a developing country in South America. Research described in [111] presents some of the key findings from the project which found three typical approaches to ICT deployments: the centralised technology lab; the technology in the classroom; and the mixed approach.

Centralised computer labs are easier to deploy, maintain and are less expensive, but according to the Enlaces project [111] are less effective in supporting teacher involvement. Researchers in the Enlaces project [111] found that the teachers seemed to find the use of a computer lab intimidating and frustrating as the use of the computer became the entire purpose of the lesson, instead of being integrated into the lesson along with other activities. According to [111], technology in the classroom tends to be a favoured solution as it places the technology within the normal learning environment. However, there are space-and time-constraints experienced with technology in the classroom, forcing teachers to employ group work which, according to the researchers involved in the Enlaces project, teachers do not always appreciate [111]. The mixed approach (both a computer lab and computers per classroom) was the recommended

choice of the Enlaces project. It was acknowledged that often budget constraints prevent this and thus they suggested that schools employ a phased approach of technology deployment, beginning with the computer lab and later placing computers in classrooms and the staff rooms as money becomes available. However, researchers did acknowledge that even with a phased approach this solution might not be feasible for some schools, especially in rural areas where they operate in harsh conditions and factors such as temperature, humidity and dust have profound effects on the life span of technology [111]. Similar constraints exist in other developing nations, like South Africa, where it becomes too costly to equip many venues throughout the school with the necessary environmental controls, such as air-conditioning [111] and required security measures such as burglar bars and alarms. The Enlaces project further highlighted that it "is often the case that budgetary restrictions impose a single solution for every school, i.e. a fixed number of computers (typically 6 - 20) and a few printers clustered in a special room. [...] This solution is far from optimal, and several other factors should be taken. [... A] one size fits all solution is rarely optimal for any one school" [111, p38].

The computer lab is also the dominant form of computer access in schools in South Africa. Isaacs, looking at ICT in education in South Africa [113], found that, in 2007, the "predominant large scale technology access model has been the computer lab containing 10 to 20 PCs, networked and sometimes connected to the Internet. [...] While useful, the computer lab model has been also proven to be spatially- and timebounded. Alternative solutions using mobile devices such as laptops [...] and cellular phones [...] have not as yet reached roll-out on a significant scale" [113, p10-11].

Gall tends to agree with Isaacs and also argues that there are disadvantages (like Isaacs' spatially- and time-bounded constraints) to the standard computer facilities provided in schools. Firstly, typical desktop computers are physically constrained (immobile) because of their size and also as a result of security constraints and connectivity requirements [109]. Gall does acknowledge that laptop computers could be used instead of desktop computers in order to introduce mobility, but argues that laptop computers of comparative hardware specifications are more expensive than their desktop counterparts [109]. However, Gall's argument against laptops would have to be thoroughly re-investigated, because, with the introduction of netbook computers (small form factor laptops) the price of entry level laptops has decreased and are now comparable with entry level desktops, if not a little cheaper. According to Weiss [114], laptop sales are out-pacing desktops and mobile devices are becoming more common place.

The second disadvantage to the current computer lab model in schools, according to Gall [109] and James [115], is that advancements in technology spur constant upgrading of hardware and software, which result in a continuous stream of spending on newer, faster and better technologies. Continuous spending is clearly not possible for many schools who are burdened by financial constraints, as a result schools would possibly resort to upgrading in batches which make ICT management more challenging as different hardware would support different software images. This would result in additional work for school technical staff to create the possibility of multiple software images, appropriate to varying hardware.

The third and final disadvantage Gall mentions is the need for connectivity and to pool resources [109]. Gall suggests an alternative model to that of the school "computer lab" as the model for technology integration in schools. He argues that instead of "focusing on networked, personal computers, [that] the future of educational technology may lie in the use of specialised, economical computing devices built around open software solutions" [109, p379]. Gall argues that the majority of computers in school computer labs are used by teachers and learners to access email, browse the Web and create and edit word processing files; typically desktop computing which does not generally require high-end computing hardware. He suggests that these desktops could be replaced with compact, less expensive and easier to maintain computing devices that are geared to specific educational purposes, such as PDAs, e-books, Web-enabled cellular phones and MP3 players. Researchers in [111] also support the use of specialised equipment such as those listed in the preceding paragraph, however they do not feel that they would replace the desktop as they see the desktop as a valuable resource to teachers.

Current computing/technology trends facilitate ubiquitous access to applications and data [116, 114, 9, 117]. Computation is seen as moving into the "Cloud" making it possible for users to connect to resources from any device anywhere [118, 9]; "Powerful services and applications are being integrated and packaged on the Web in what the industry now calls 'Cloud Computing'" [114, p1]. This is often facilitated through multi-functional small devices, such as laptops, mobile phones and tablet PCs that support numerous functions and connect to services, such as email, browsing, video playback and other online services that provide access to data files [116, 114, 9]. These online services are fast becoming supported by groups of servers working together in either clusters or grids [118, 114, 9]. It is important to note that access to some Cloud services require high bandwidth data connections to the end devices, which in the context of the developing world and especially South Africa, are expensive. The use of grids, clusters and clouds also have potential value in educational settings, providing high availability and increased computing power [119, 9]. In addition, they facilitate resource sharing and can readily be built using commodity, off-the-shelf equipment, which have cost reduction implications [119, 9, 117].

# 4.3 ICTs for education in developing countries

A number of developing countries are concerned with bridging the digital divide that exists between themselves and the developed world. There is unfortunately no generic method for overcoming the digital divide and the greatest difficulty is in providing physical access to technology [120]. The use of refurbished and low end computers has also been suggested, together with the use of open source operating systems such as Linux to overcome cost constraints and increase the life span of technologies [111, 112, 115]. Naturally, it needs to be born in mind that the initial outlay for technology is not the only cost involved, but the technology also needs to be maintained and appropriately disposed of at the end of its life span [120]. There are numerous projects which make use of refurbished computers together with the Linux Operating system and Open Source applications (OSS). The best-known in South Africa is the tuXlab Project. The primary objective of the project was to develop an ICT model that could be used in schools and actively promote the use of open source software as a computer lab solution for schools. The project began in the Western Cape in 2003 and used Linux thin client technology to establish a viable economical solution for South African schools [121]. Most tuXlabs use a combination of a new server, networking equipment and diskless refurbished computers. The decision to use thin client technology is purely cost driven, but also recognises the benefits of centralised software, data and support [121]. Projects involve both the teachers' and the learners' participation in establishing the labs, which range from initial infrastructure planning to installing networking cables and setting up of the computer hardware and the open source software [121]. Part of the involvement of the teachers and learners is to foster a sense of ownership amongst them, which the tuXlab project believed assisted in the sustainability of the facilities as highlighted in other studies [28]. Sustainability refers to development that "meets the needs of the present without compromising the ability of future generations to meet their own needs" [120, p3501]. In addition to playing a role in prolonging the lifespan of computers, OSS supports language localisation so that learners and teachers can use computers in their mother tongue [115]. Language localisation of ICTs provides access to the use of technology by people who are not fluent in English, in South Africa that means extending access to marginalised people [122]. Some researchers [123] have found that the lack of availability of software in local languages is an important contributing factor to the slow adoption of ICTs in developing countries [123]. OSS by its nature supports language localisation in that it allows for software adaptation by others as part of its model [124].

In a study conducted by the Swiss Federal institute of Technology in Zürich, investigating the sustainability of refurbished computers for use in schools in Colombia, researchers found that while locally refurbished computers were more expensive than internationally refurbished computers or new cheap hardware, locally refurbished computers were still a better choice because of the positive benefits they had on the economy [120]. These benefits included increased local job creation and an increased positive impact on the local economy [120]. It was therefore the recommendation of the researchers that the Colombian government provide locally refurbished computers to schools primarily and only rely on internationally refurbished or cheap new hardware when the domestic supply of second hand computers could not meet the demand [120]. By contrast a study conducted in Mozambique recommended that second-hand computers not be used in schools because they require more maintenance and schools typically do not have skilled individuals who can repair and maintain the computers [125]. Other studies have found that often refurbished or low end computers are poorly received as they are perceived to be of an inferior quality [126].

A study [126] conducted with rural Indian schools, compared three models of computer usage for education, namely, (a) Single ownership (that is where each child has their own computer/laptop, like the One Laptop per Child (OLPC) project), (b) Single user per classroom computer/terminal, and (c) Multiple user per shared classroom computer. They found that the shared classroom computer (option c) was the most economical and also fostered greater learning through group work amongst the learners [126]. Alternatively, Power [127] advocates the use of mobile devices such as laptops and PDAs over refurbished computers.

Other alternatives to refurbished computers for use in education include the OLPC project [112, 128], or alternatively the use of cheap netbook computers together with Linux and other open source software [128]. This has had varied success with some laptop projects at some schools still continuing the initiative, while others have scaled these down.

With so many choices regarding technologies for teaching and learning there seems to be a clear need for "rigorous but innovative investigations of new paradigms of educational technology for development that are specifically orientated to the settings, needs and experiences of rural personnel and the communities they serve" [127, p8]. In this thesis specific attention will be given to the use of thin client computing models, together with the use of Linux operating systems in order to reduce costs; support localisation and local needs; prolong the lifespan of technology where possible; maximise resource utilization in order to get the best return on investment; and support user defined educational needs. In the next section I discuss the concept of affordances, in order to create an understanding and awareness of how computing infrastructure capabilities can be mapped to classroom practices in order to take the first steps towards supporting appropriate integration of computing infrastructure into the learning processes.

# 4.4 Affordances

A key concept in relating the plethora of technologies to teaching and learning is that of "affordances", which can be used as a theoretical perspective to better understand and explain the use of ICTs and how they relate to teaching and learning. Researchers have noted that technologies tend to be underutilized or used in ways that don't fully realise the potential that computing infrastructure offers to the teaching and learning environment. Little is understood by teachers about the affordances of different technologies and as such how these properties can be exploited in particular teaching and learning contexts [129]. A better understanding of the affordances of technologies to the requirements of teaching and learning would allow users to better match the affordances of technologies to the requirements of teaching and learning [91, 129]. It would seem that many teachers are still unclear about how to use technology appropriately in the classroom, and so the use of ICTs is often based on common sense rather than being theoretically

informed by pedagogical theory. As such the authors of [129] argue that "there is a need to explore a theoretical basis for the use of ICTs and present a conceptual framework as an attempt to further explore the notions of affordances in this respect" [129, p114]. Bower [91] also describes a need for a framework that provides tools for teachers to structure and co-ordinate the activities or exercises in the classroom. He describes a methodology of supporting technology selection by identifying technological "affordances" of e-learning tools as well as the "affordance" requirements of e-learning tasks to scaffold the learning design process; matching learning tasks to technologies [91].

# **Definitions of Affordance**

The term affordance was first coined by J.J. Gibson, a perceptual psychologist, and is defined as,

"The affordances of the environment are what it offers the animal, what it provides or furnishes, either for good or ill. The verb to afford is found in the dictionary, but the noun affordance is not. I have made it up. I mean by it something that refers to both the environment and the animal in a way that no existing term does. It implies the complementarity of the animal and the environment [130, p127]."

Simply stated then an affordance, in accordance with Gibson's definition, is a relation between an organism and an object where the object is perceived in relation to the needs of the organism. However, it is important to remember that the object may also possess other affordances, outside of those that are perceived by the organism – direct visual perception is not a necessary condition for an affordance to exist, only for an affordance to be made use of. McGrenere and Ho [131] clarify Gibson's definition further by explaining that there are three fundamental properties of an affordance, namely,

- 1. An affordance exists relative to the action capabilities of a particular actor;
- 2. The existence of an affordance is independent of the actor's ability to perceive it; and
- 3. An affordance does not change as the needs and goals of the actor change.

Gibson's definition of affordances introduces the idea of the actor and the environment as an inseparable pair [131]. He maintains that the existence of the affordance and the information that specifies the affordance are independent and as such there are cases when there is no information to specify the affordance; direct visual perception is not possible [131]. For example, if one considers a door handle, it provides both a physical affordance as well as a visual cue, so that we are able to perceive the physical affordance, which is that the door allows us passage through. However, no matter the visual cue (for example, a door with a plate on it suggesting pushing) the door still affords passage through. Thus, even if we are not able to perceive the affordance of the door, it exists independently of the visual cues.

A colleague of Gibson's, Donald Norman, reworked the notion of affordances for his work in the design field [132]. Norman describes affordances as design aspects of objects which allude to how they should be used, which is almost in opposition to Gibson's notion of affordance. Norman defines affordance as,

"the term affordance refers to the perceived and actual properties of the thing, primarily fundamental properties that determine just how the thing could possibly be used. A chair affords ('is for') support and, therefore, affords sitting [133, p9]."

Norman's definition highlights both the perceived and actual properties of an object and implies that a perceived property may not be an actual property of a "thing" but is nonetheless an affordance. This deviates from Gibson's definition because in Norman's view perception by an individual is involved in characterizing the existence of an affordance. Thus Gibson focuses on the fundamental characteristics of

the object in relation to the user; "the utility of the object". While Norman places greater emphasis on how an object is perceived by the user, which relates to usability [91, 131]. According to [131], designing affordances and designing information about the affordances should not be confused or conflated; the utility of an object is related to but still separate from the usability of an object. A useful object is one where the design contains the right functions, such that a user can perform their jobs efficiently and accomplish their goals; the usefulness of a design is determined by what the design affords and whether those affordances match the goals of the user. Usable designs (usability), however, refer to a clear design of the perceptual information that specifies the affordances [131]; those properties that "provide strong clues to the operations of things [133, p9]."

The difficulty in understanding the two authors, Gibson and Norman, stems from the confusion between what Norman describes as real affordances and perceived affordances [134]. According to Norman, the unqualified term affordance refers to the real or physical affordance, which are the physical characteristics of a device or interface that allow its operation, as described by Gibson's original definition. A perceived affordance, according to Norman, refers to the characteristics in the appearance of a device that give clues to its proper operation. Another and perhaps easier way of understanding perceived affordances is to use the term cognitive affordances, helping the user understand how a device is best used [134]. Hartson [134] describes a "cognitive affordance as a design feature that helps, aid, supports, facilitates, or enables thinking and/or knowing about something" [134, p319] (emphasis in original text). A physical affordance on the other hand is described as "a design feature that helps, aids, supports, facilitates, or enables physically doing something" [134, p319] (emphasis in original text). As an example, a door handle is both a cognitive and a physical affordance. It not only provides the user with visual cues as to how one opens the door but also physically allows one to open a door. Hartson [134] decomposes the concept of affordance further and instead of just real or physical affordances and perceived or cognitive affordances he also argues that there are sensory affordances which help users with sensory actions and functional affordances which allow users to complete a function or task (making them useful). A sensory affordance is a "design feature that helps, aids, supports, facilitates, or enables the user in sensing (e.g. seeing, hearing, feeling) something" [134, p322], while a functional affordance is as a result of a physical affordance giving rise to functionality.

Gaver [135] describes affordances as a powerful approach to thinking about technology because the effectiveness of an affordance depends on the attributes of both the user and the device, allowing one to focus on the strengths and weaknesses of technologies with respect to the possibilities that they offer users. The concept of affordances is described as a mechanism for focusing on links between the user, the technology and how best to make use of the technology [134]. Hammond [136] explains affordances as 'point[ing] both ways' to the object and the organism. He describes affordances as an emergent property of an object that is always there but needs to be perceived in order to be realised [136]. He further believes that the strength of the idea is that it offers an alternative to the realist perspective while still rejecting "the radical nominalist position that properties are perceptual and exist only so far as they identified" [136]; avoiding the idea that there are "thingless properties" or "propertyless things" [136].

According to Gibson, affordances are perceived directly [130, 136] however direct perception need not rule out the importance of experience, memory or context (they do not undermine direct perception) as these clearly affect the ways in which we know different environments; knowing is an active process [136]. That said, it is argued that Gibson underplays the distinction between natural and cultural objects and that direct perception primarily contributes to understanding how humans meet their immediate needs. Cultural tools are however made with intended or specific affordances which are nested in systems of societal praxis; there are different types of perception, some more immediate and others requiring knowledge of properties [136].

### Affordances in ICTs

The concept of affordances has been appropriated/used to understand the use of ICTs in education [91, 129, 137]. The affordance of ICTs refers to those functional properties of ICTs that determine the possible (both perceived and real) utility of ICTs. Researchers have begun to investigate how one might determine the actual and perceived affordances of technology in order to better utilize them for teaching and learning [91, 129, 137]. There has been a significant growth in the number of technologies available to teaching and learning but even so there is evidence of slow uptake of these technologies in education [129]. Britain and Liber [138], quoted in [129] conclude that the amongst the factors which affect this slow up take is a lack of a coherent framework which can be used to evaluate the pedagogical benefits and the organisational changes that are required in order to effectively make use of ICTs in teaching and learning [129]. Researchers have noted that technologies tend to be underutilized or used in ways that don't fully realise the potential that they offer in the teaching and learning environment. Little is understood about the affordances of different technologies and as such how these properties can be exploited in particular learning and teaching contexts [129]. A better understanding of the affordances of technologies and the requirements of teaching and learning would allow users to more accurately match the affordances of technologies to the requirements of teaching and learning [91, 129]. It is clear that many teachers are still uncertain about how to appropriately use technology in the classroom, and so the use of ICTs are often based on common sense rather than being theoretically informed by pedagogical theory.

In applying the concept of affordances to the use of technologies in teaching and learning, Conole and Dyke [129] focus on three main research questions:

- How might the affordances of technologies be articulated? ;
- How can the articulated affordances be developed into a taxonomy for ICT? ; and
- How can a critical knowledge of the affordances of technologies be used to support teaching and learning?

In order to answer these questions Conole and Dyke [129] developed a taxonomy of ICT affordances. Their research led them to create a list of 11 ICTs affordances that users could exploit in teaching and learning, listed here:

- Accessibility: ease through which one can access vast amounts of information through multiple channels and mechanisms;
- Speed of change: as a result of the ease with which one can access information, information is rapidly updated and changed as events unfold;
- Diversity: ease of access together with the vast amount of information available on the WWW means that the available information is often diverse. In addition, it allows for access to any and all experts who are contactable via either the WWW or other Internet services such as email;
- Communication and collaboration: the Internet and the services that it offers facilitate communication and collaboration through tools such as email, video conferencing and online wikis;
- Reflection: through the use of asynchronous tools, like forums, learners are encouraged to critically evaluate and engage with concepts and discussions taking place, by slowing the pace at which debates and engagement occur. As a result of the slower pace they are encouraged to think more deeply about the concepts that they are studying;

- Multimodal and non-linear: the non-linearity of the WWW creates the potential for different routes through or different forms of learning. This creates the potential for the learner to move beyond the linear pathways of learning and to adopt more individualised strategies and pathways;
- Risk, fragility and uncertainty: the risks to technology include that technologies are sometimes used in ways that were not intended and people can be exposed to harmful situations. Furthermore, they are vulnerable to abuse through things like SPAM or network attacks. In addition, with the increased access to a plethora of information and data, plagiarism has become a more prevalent issue;
- Immediacy: the speed through which users can access or exchange information and communicate via the web has led to an increased expectation in response times and intensification of working patterns;
- Monopolization: there is a danger when technologies are not interoperable that a particular technology might monopolize and dominate the market. This tends to make items very expensive and can have negative effects on sustainability and scalability of technologies within education; and
- Surveillance: concerns regarding the potential to infringe on the privacy of others through the use of digital media.

#### Affordances in ICTs to support teaching and learning

There are both positive and negative aspects to each of these affordances, and if used unwisely may hinder the teaching and learning experience rather than enable it. Conole and Dyke [129] propose that a taxonomy of affordances such as that listed above could be used in a number of ways, for example, to provide a clearer understanding of affordances to practitioners in order to help them make informed use of technology for teaching and learning. For example, helping practitioner's better match pedagogical design and support learning activities; identify limitations and inappropriate uses of technologies; or as a checklist for practitioners to understand costs and benefits of different technologies. Often affordances of technologies are known intuitively but the development of an explicit taxonomy would provide a basis for discussion, critique and refinement [129]. An understanding of the affordances of technologies would scaffold the decision making process when designing learning activities rather than prescribing to teachers what they should do [91].

Bower [91] also describes the need for a taxonomy or framework of affordances which can be mapped to technologies to assist teachers in designing meaningful lesson activities that incorporate the use of technology to appropriately support the learning objectives or outcomes. In his research he categorizes affordances based on their physical characteristics in order to emphasize their functionality, which he argues is in accordance with Gibson's original definition of affordance. This, Bower claims, allows him to undertake a more fundamental level of analysis [91].

Bower [91] describes the following affordance classification, defining terms within categories as abilities:

- Media affordances; abilities include, "read-ability", "write-ability", "view-ability", "draw-ability", "listen-ability", "speak-ability"
- Spatial affordances; "resize-ability" and "move-ability"
- Temporal affordances; accessibility, "record-ability", "playback-ability", "synchronicity"
- Navigation affordances; "browse-ability", "link-ability", "search-ability", "data-manipulation-ability"
- Emphasis affordance; "highlight-ability", "focus-ability"

- Synthesis affordance; "combine-ability", "integrate-ability"
- Access control affordances; "permission-ability", "share-ability"
- Technical affordances; refers to the cross platform nature of many software packages, as well as the ability to adapt to various technological setups such as varying bandwidth and abilities of computer hardware
- Usability; the intuitiveness of software and the ease with which users are able to interact
- Aesthetics; refers to the appeal of the design and how well the user's attention is held and their level of satisfaction
- Reliability; refers to the robustness and reliability of the platform

The last four categories in the list from Bower [91] are what he describes as non-functional from an educational point of view in that they relate to the quality of the learning experience but not to what can be accomplished in terms of educational design of lessons; they relate to the usability of the system rather than the educational potential or utility. Bower does not believe this list to be exhaustive but rather offers as an example of fundamental, pragmatic and functional levels at which affordances can be identified in order to be useful in matching with the requirements of various learning activities. The focus of Bower's framework is to match technologies with the cognitive and collaborative requirements of activities in the classroom. By focusing on the affordances of the underlying technologies, Bower believes that the framework is adaptable to changes in technology over time [91]. By way of example, Bower assesses the activities of a course that students need to complete, and identifies affordances that are required in order to complete those activities. He then identifies three possible technologies that he could use and notes their affordances. He can then compare the required affordances of the learning activity to those offered by the three technologies and choose appropriately which one best meets his needs. Bower argues that the successful use and integration of technology into teaching and learning is dependent on teachers being able to identify the affordances that are required for activities and identify the affordances of a particular technology and then appropriately match tasks and technologies [91].

While acknowledging the importance of proper design, so that the use of technology is clear (usability), I employ Gibson's more fundamental definition of affordances in this project, namely that they are the properties of an object that determine its utility. The work of Conole and Dyke [129] and Bower [91] highlight that the utility and the usability of technology is not obvious to teachers – there are few if no cognitive affordances (as described by Hartson [134]) to hint at the physical affordances offered. Furthermore, it would seem that there is no cultural societal praxis (stemming from the work of Hammond [136]) through which teachers know and understand technology, further limiting their understanding of how ICTs might be integrated into teaching and learning. According to Bowers [139], computers are not culturally neutral but rather carry the cultural view points of their creators, which typically are slanted towards dominant western cultures which teachers in South Africa might not understand and hence struggle to interpret how technology could be meaningfully used in their classrooms.

The scope of this dissertation does not include the production of a reference guide to assist teachers in matching technology to teaching activities, but is the first step in reaching that outcome (it is one of the suggested future works stemming from this dissertation and takes the first steps in achieving that output). The focus in this dissertation is to investigate technologies that would best lend themselves to integration in teaching practices by attempting to insure that the computing infrastructure implicitly affords the desired teaching and learning outcomes of teachers who participated in the research project. To this end, I will next discuss ICT integration, pedagogical practices and policy, in order to better understand the difficulties around ICT integration and how pedagogy and policy can either support or detract from the process.

# 4.5 ICT Integration, Pedagogical Practices and Policy

In the Enlaces project in Chile, researchers found that teachers needed to change their pedagogical culture and beliefs about ICTs in order to successfully integrate the use of technology into their classroom practices [111]. The use of ICTs in education is heavily influenced by teachers' pedagogies and pedagogical reasoning; according to Cox, Webb, Abbot, Blakeley, Beauchamp and Rhodes [140, p3], "the teacher's own pedagogical beliefs and values play an important part in shaping technology-mediated learning opportunities". In order to select appropriate technologies and employ them in the learning process teachers will need knowledge about ICTs and may need to adopt new pedagogies [140]. However, few teachers have extensive knowledge about the wide variety of ICTs available for use in education or the affordances that they offer. In addition, many teachers fail to recognise that a change in pedagogy is needed when integrating the use of ICTs into their teaching and learning and rather use ICTs to add to or enhance their existing practices [3, 140]. There has been much research surrounding the association between technology based practices in teaching and changes to curriculum and pedagogy and there seems to be a consensus that teachers' pedagogies do need to adapt if they are going to successfully employ the use of ICTs in their teaching practices as their ideas, beliefs and values tend to influence their practices [91, 140, 141, 142]. However, according to Watson [20], teachers feel threatened by change and do not approve of change that appears to focus on the technology rather than on learning [20]. Thus there needs to be a pedagogical understanding of ICTs in order to integrate their use meaningfully [142, 143, 144]. It is important to thus realise that even with perfectly working and supported ICT facilities in place, if teachers are not properly trained, do not have proper support and are not able to adequately adjust their pedagogical practices, sustainable ICTs in teaching and learning may not be realised [141]. Successful ICT integration does not only involve good technical solutions but incorporates a host of other facets such as the teacher's pedagogical view, the task structure and the organisational and educational mindset [145].

Watson [20], quoting the National Inspection Agency, Ofsted, in the UK, found that teachers rarely employed pedagogic expertise to make the most effective use of ICTs in the lessons. Bowers [139] states that teachers may not understand the difference between analogic and digital thinking and as a result may be unable to make intellectual judgments about the deeper implications of the educational technology they choose [139]. Watson [20], again quoting Ofsted, "Many teachers still have difficulty in deciding when, and when not, to use computers" [20, p258]. As such, there is a need to highlight and identify the affordances of various technologies such that they might become more readily apparent to teachers, thus helping teachers identify which technologies to use and how to use them in achieving learning outcomes. According to Watson [20] "the few teachers who do use computers … can clearly relate the use of technology to their pedagogic strategy" [20, p259].

The concept of affordances can also be used to understand the benefits of one ICT architecture over another. The benefits and constraints of various ICT configurations can be understood using the concept of affordances, which allow us to understand the technology in terms of its usefulness to teaching and learning needs of teachers and schools in South Africa. However, identifying the affordances of technologies is only one side of the problem. The other involves understanding the classroom practices or pedagogical practices that the technology affordances are to support. In order to understand how technology is expected to be used in the classroom, we need to understand the policy documents that provide the guidelines for teachers regarding ICT integration in the curriculum. In South Africa the National DoBE has developed a number of policy documents that govern the use of computers and other ICTs in teaching and learning. These policy statements provide the framework within which schools, teachers and learners are expected to operate. These documents include the White Paper on e-Education [25], the Revised National Curriculum Statement documents for Grades R-9 for the General Education and Training (GET) band [146], the National Curriculum Statement for Grades 10-12 (schools): Computer Application Technology [147] and the National Curriculum Statement for Grades 10-12 (schools): Information Technology [148]. Grades 10-12 fall within the Further Education and Training (FET) band.

At the level of national curriculum policy, the Revised National Curriculum Statement (RNCS) stipulates the inclusion of computer use for grades R - 9 in a new learning area called Technology. Technology is a stand-alone subject in the senior phase of GET (grades 7 - 9) while in the intermediate phase (grades 4 - 6) it is combined with Science. There is no direct inclusion of Technology in the foundation phase of R - 3. ICT comprises one of the three strands in the Technology learning area and the RNCS states that the computer skills developed in Technology will be applicable to other learning areas as well [149].

The National Curriculum statement for the FET phase for grades 10 - 12 provides details of the framework for the two computer-related subjects: Computer Applications Technology (CAT) and Information Technology (IT). CAT is a relatively new subject within the FET phase and is a hybrid of the old Typing curriculum and the old Computer Science (Standard Grade) curriculum. The aim or outcome of this learning area is to allow learners to develop medium- to high-level end-user computer skills in order that they may be able to work in areas of office administration, industry, business and commerce, amongst others, or that they may be able to create employment for themselves and for others [149]. The new Information Technology curriculum is in essence the old Computer Science (higher grade) curriculum. The aim of this learning area is to enable learners to use information and communication technologies, more specifically computers, in social and economic applications, systems analysis, problem-solving, logical thinking, information management and communication [149]. The FET phase does not specify how computers should be integrated across other learning areas [149].

The White Paper on e-Education of 2004 [25], is another framework document that governs computer or ICT integration in schools in South Africa. It provides a framework for collaboration between the DoBE and the private sector in providing ICTs in education [149, 150]. The eagerness of the South African government to adopt ICTs in education is captured in the White Paper on e-Education, "As in other spheres of social and economic development, ICTs have the potential to improve the quality of education and training. It is for these reasons that Government has been quick to seize the opportunity presented by the practical benefits of ICTs to support teaching and learning in the twenty-first century" [25, p1]. To this end the primary goal of the White Paper is that "Every South African learner in the general and further education and training bands will be ICT capable (that is, use ICTs confidently and creatively to help develop the skills and knowledge they need to achieve personal goals and to be full participants in the global community) by 2013" [25, p40]. The White Paper provides a framework which details the DoBE's vision, however vague, of ICT implementation in schools. The document describes background information positing the DoBE's beliefs regarding the benefits of ICTs to society at large and education in particular. In addition, the framework specifies funding and resourcing, strategic objectives and implementation strategies that would hopefully see its primary goal delivered [25]. Recently, the South African government has announced its broadband policy, referred to as SA Connect (more about SA Connect is in section 5.4.1). According to the SA Connect policy the South African government has realised the potential of providing access to networks and the network society, and plans to prioritise schools setting the ambitious target that 50% of schools will be connected by 2016 at 10 Mbps and all schools to 1 Gbps connections by 2030 (see Table 4.1 for all school targets) [15].

Despite the policy documents stipulating the use and integration of ICTs in teaching and learning there is very little in terms of direct guidance to teachers regarding how to integrate technology in the classroom or which technologies best support which classroom activities or pedagogical practices. The closest set of guidelines to the use pertaining to the integration of ICTs in teaching and learning is "Managing ICTs in South African Schools: A Guide for school Principals". The purpose of the guideline is to "give principals".

Target	Penetration measure	Baseline (2013)	By 2016	By 2020	By 2030
Schools	% of Schools	25% connected	50% at 10 Mbps	100% at 10 Mbps; 80% at 100 Mbps	100% at 1 Gbps

Table 4.1: SA Broadband Policy targets (as per SA Connect [15])

and senior management information on using and managing ICT resources so that they can provide leadership in their schools" [151, p6]. This document was produced in response to research that was undertaken by South African Institute for Distance Education (SAIDE) that established that school principals are often not properly informed regarding the affordances of technology and that this can have a negative impact on the success of technology integration in schools [151]; hampering their ability to manage the introduction of ICTs into their schools. These findings confirmed a need to support the principals and school management with the integration of computers and related resources into teaching and learning activities in their schools [151]. While useful in attempting to assist school management to support the implementation of technology in teaching and learning, the Principal's Guide [151] lacks the level of detail to wholly achieve its objectives [100]. Furthermore, there is no guidance regarding use of technology in the classroom or for teaching and learning specifically. Thus there remains a need to support teachers in the classroom; a need for a guideline or framework that supports teacher's integration of technology in their pedagogical practices (or classroom activities). According to Dabbagh [152], teachers employ pedagogical models derived from the learning theories (the basis of which is knowledge acquisition models or views about cognition and knowledge) that they employ to understand the curriculum [152]. Therefore, pedagogical models are mechanisms which allow teachers to link theory to practice. These models lead to instructional strategies, which are described as the "plans and techniques that the instructor/instructional designer uses to engage the learner and facilitate learning [152, p32-33]"; they are the plan, method or series of activities designed to facilitate learning. Within these instructional strategies, teachers employ specific learning activities, like those described by Harris, Mishra and Koehler [3]. In order to match ICT affordances to the learning objectives of teachers in South African schools we also need to understand what the learning objectives and activities of the classroom are. According to Harris, Mishra and Koehler [3] many approaches to technology integration in teaching and learning are "technocentric" and don't consider the dynamic and complex relationships between content, technology, pedagogy and context. These technocentric approaches tend to begin with the affordances and constraints of the technologies and the skills required to operate them and only later attempt to discern how they can be integrated into learning at different levels. Harris, Mishra and Koehler [3], note that educational technologies are often pedagogically unsophisticated and not well integrated into curriculum-based teaching and learning. They note that there tends to be a mismatch between educational technologies and how most teacher practitioners use digital tools in the classroom; teachers tend to focus on use that enhances existing practice [3]. Furthermore, approaches to use of technologies in the classroom tend to focus on the technologies being used rather than on the learning needs of students relative to the curriculum content [3]. Content and pedagogy are the two key domains that are often short changed in ICT in education projects. Both, however, are important, as each learning area has different content knowledge and different pedagogical strategies appropriate for teaching. Therefore it is necessary to understand that the introduction of technology to teaching and learning impacts the nature of the contentarea learning as well as the pedagogical approaches that teachers can select [3]. Harris, Mishra and Koehler, introduce the use of the technology, pedagogy and content knowledge (TPACK) framework as an effective method of approaching technology integration; TPACK recognises technology, pedagogy, content and context as interdependent aspects of a teacher's knowledge that are necessary to effectively teach content-based curricula using educational technologies [3].

# 4.6 TPACK

Traditional learning technologies, such as pencils, microscopes, chalkboards, etc., are said to be characterised by specificity, stability and transparency of function. Over time they have become transparent to the users and so commonplace that often they are not even considered as technologies [153]. Digital technologies, by contrast, are considered to be adjustable (and therefore used in various different ways), unstable and opaque. This very nature presents challenges to teachers who are struggling to integrate the use of technologies into their teaching practices [153] because technologies can be used in so many different circumstances and ways, are constantly changing and improving and are not always easy to understand the inner-workings thereof. Furthermore, technologies are not considered to be neutral or unbiased. But rather, different technologies "have their own propensities, potentials, affordances and constraints [153, p61]" that make them more or less suitable for some tasks as compared with others [153]. An example of this that Koehler and Mishra [153] give is a comparison of email vs instant messaging. Email affords asynchronous communication and easy storage of messages. However, if synchronous communication is required then instant messaging would be a better technology to utilize instead of email. They note that understanding the affordances and constraints of different technologies and how they influence the teaching process of a teacher is not a straight forward or obvious process [4]. Koehler and Mishra [153] further note that often social and institutional contexts are not supportive of teachers' efforts to integrate the use of technology into the teaching process. As a result, teachers then have inadequate or inappropriate experience with using ICTs for teaching and learning.

Harris, Mishra and Koehler [3] note that generally studies investigating the instructional application of educational technologies show many teachers' use to be pedagogically unsophisticated. They clarify this as limited in breadth, variety and depth and not well integrated into curriculum-based teaching and learning [3]. They further note that there is a clear mismatch between the vision for technology integration, projected by educational technology leaders, and how teachers are using technology in practice. They note that while researchers emphasize the use of ICTs to support inquiry, collaboration and reformed practice, teachers tend to focus on using presentation software, Internet based research and digital management tools (e.g. spreadsheets for management of class marks) to enhance existing practices [3]. Even when properly trained in the use of technology, Koehler and Mishra [153] argue that teachers will not make use of ICTs in their teaching unless they can conceive of uses for ICTs that are consistent with their existing pedagogical beliefs. Furthermore, training for teachers often takes the form of a "one-size-fits-all" model to technology integration, rather than realising that teachers operate in diverse contexts of teaching and learning. Therefore integration efforts should be designed or structured in accordance with particular subject areas in specific classroom contexts [153]. Howard and Maton [154] similarly argue, using Legitimate Code Theory (LCT), that if the use of technology does not "code match" with the subject matter/area that the teacher teaches they will not effectively integrate the use of technology into their teaching process. Harris, Mishra and Koehler [3] argue that the discrepancy between the vision of ICTs as used for transformation and the prevalent use of ICTs rather for efficiency and extension of current practices stems from how ICTs are conceptualised and supported. They describe five approaches which have to date dominated technology integration efforts:

- Software-focused initiatives;
- · Demonstrations of sample resources, lessons and projects;

- Technology-based educational reform efforts;
- · Structured/standardised professional development workshops or courses; and
- Technology-focused teacher education courses.

Harris et al [3] argue that these approaches tend to organise their efforts according to the educational technologies being used rather than students' learning needs in relation to the curriculum; professional development of teachers tends to emphasize technology affordances and constraints. They further describe these approaches at "technocentric" (quoting Papert [155]) because they begin with the affordances and constraints of ICTs together with the skills needed to successfully use said ICTs and then only later attempt to understand how they can be successfully integrated into content-based learning [3]. As such, these typical ICT integration approaches tend to ignore the other two key domains in teaching, namely, pedagogy and content [3].

According to Koehler and Mishra [153], citing Glaser [156], Putman and Borko [157] and Shulman [158, 159], "effective teaching depends on flexible access to rich, well-organised and integrated knowledge from different domains", which includes knowledge about how learners think and learn, knowledge about the subject domains in which they teach and increasingly, knowledge about technology or ICTs [153]. TPACK is a conceptual framework for educational technology that builds on the framework by Shulman – Pedagogic Content Knowledge (PCK) – and extends it to include the phenomenon of teachers integrating technology into their pedagogy. The authors argue that "thoughtful pedagogical uses of technology require the development of a complex, situated form of knowledge that [they] call Technological Pedagogical Content Knowledge (TPCK/TPACK) [4, p1017]". They argue that a conceptually based theoretical framework about the relationship between technology and teaching can transform the ways in which teachers conceptualise the practice of teaching [4]. Furthermore, that the development of TPACK by teachers is essential for effective teaching with computers and other ICTs [3, 153]. In addition, this framework acknowledges that effective pedagogical use of ICTs is influenced by the content domains in which they are situated [160]; similarly noted by Howard and Maton [154] who argued that technology will only be used effectively when technology "code matches" with the content area in which the teachers are operating.

The TPACK framework highlights the connections between technology, curriculum content, and pedagogical approaches. This demonstrates the teacher's understanding of technology, pedagogy and content and that they interact with one another when integrating the use of technology effectively in teaching and learning. The TPACK framework (Figure 4.1) consists of three interdependent components of a teacher's knowledge, namely, Content Knowledge (CK), Pedagogical Knowledge (PK) and Technological knowledge (TK), all of which are framed within and influenced by contextual knowledge [3]. CK is knowledge about the subject matter that needs to be learned or taught. CK includes knowledge of concepts, theories, ideas, organisational frameworks, methods of proof and established practices and approaches within a discipline and tend to vary across different disciplines [3]. PK is knowledge about the process and practices of teaching and learning. It encompass educational purposes, goals, values and strategies, including knowledge about techniques or methods used in the classroom, the nature of the learners' needs and preferences and strategies for assessing students' understanding. PK requires that the teacher understand the cognitive, social and developmental theories of learning and how to apply them in the classroom environment [3]. According to Harris, Mishra and Koehler [3], TK is constantly in a state of flux and thus difficult to define and acquire. However, they believe that there are ways of thinking about working with technology that apply to all technology that goes beyond traditional computer literacy and requires that people understand information technology broadly enough to apply it productively at work and in their everyday lives; showing a mastery in information processing, communication and problem solving.

Equally important within the framework is how the three components interact with one another. Components interact with one other component as well as all three interacting, producing the following interactions, PCK, Technological Content Knowledge (TCK), Technological Pedagogical Knowledge (TPK) and TPACK [3]. PCK is the intersection and interaction of pedagogy and content knowledge. It covers knowledge regarding teaching and learning of content-based curricula as well as assessment and reporting. Expressions of PCK include, an awareness of students' prior knowledge, alternative teaching strategies, common content-related misconceptions and how to forge links between different content-based ideas [3]. TPK is an understanding of how teaching and learning change when particular technologies are used. TPK includes knowing the pedagogical affordances (both enabling and constraining) of technological tools and resources and how they relate to appropriate pedagogical designs and practices. This requires that the teacher builds an understanding of the potential benefits and limitations of particular technologies [3]. Flexible use of technological tools is essential when one considers that popularly used software programs are often not designed for educational purposes, such as Microsoft Office Suite, discussed earlier [3]. According to Harris, Mishra and Koehler [3], teachers must therefore possess knowledge and skills that allow them to appropriate technologies for pedagogical purposes, in order that technologies are used not for their own sake but for the advancing of student learning and understanding. TCK involves an understanding of the ways in which technology and content influence and constrain one another. For example, Harris, Mishra and Koehler [3] refer to the new understandings in various disciplines that have emerged as a result of the development of new technologies that afford the representation and manipulation of information and ideas in novel and fruitful ways [3].

The TPACK framework is the interaction of all three components with each other, which goes beyond the individual components, and according to Harris, Mishra and Koehler is what underlies effective and highly skilled teaching with technology. According to [3] the TPACK framework "encompasses understanding and communicating representations of concepts using technologies; pedagogical techniques that apply technologies appropriately to reach content in differentiated ways according to students' learning needs; knowledge of what makes concepts difficult or easy to learn and how technology can help redress conceptual challenges; knowledge of students' prior content-related understanding and epistemological assumptions, along with related technological expertise or lack thereof; and knowledge of how technologies can be used to build on existing understanding to help students develop new epistemologies or strengthen old ones" [3, p401]. The authors of [3] aptly state that learning about technology is vastly different to learning with technology and that teaching technological skills in isolation does not help teachers to develop learners knowledge about how to use technology, its relationship to disciplinary content, or how to use technology appropriately to help students meet curriculum content standards [3]. Therefore, the development of a teacher's TPACK knowledge requires "flexibility and fluency" in curriculum content, pedagogy, technology and context, as all four influence each other in pervasive ways [3]. Furthermore, they argue that TPACK is the basis of good teaching practices with technology and is different from a knowledge of all three components individually, requiring an understanding of following [4, 153]:

- Representations of concepts using technologies;
- Pedagogical techniques that use technologies in constructive ways to teach content;
- Knowledge of what makes concepts difficult or easy to learn and how technology can be used to try and help redress some of the problems faced by learners;
- Knowledge of students prior knowledge and theories of epistemology;
- And how knowledge of technologies can be used to build on existing knowledge and develop new epistemologies and strengthen old ones.

The interactions between the three key knowledge sources (pedagogy, content and technology) are bidirectional – deciding on a particular technological tool will constrain the possible representations that can be developed and the course content that could be covered, which in turn would affect the pedagogical process. Their model of technology integration thus argues that developing good content requires a thoughtful interweaving and navigation of all three of the fundamental sources of knowledge of pedagogy, content and technology. Furthermore, that quality teaching involves developing a nuanced understanding of the relationships between the three fundamental sources of knowledge and using this understanding to develop appropriate and context specific strategies and representations [4, 153].



Figure 4.1: TPACK Framework as described by Harris, Mishra and Koehler in [3]

TPACK has its roots in Shulman's PCK. Prior to Shulman's work teachers knowledge regarding content and pedagogy were treated as two mutually exclusive domains. The practical consequences of this was that teacher education programmes tended to focus on either pedagogy OR content. To address this he proposed considering the necessary relationship between the two and introduced the notion of PCK. PCK exists at the intersection of PK and CK, moving beyond the simple consideration of content and pedagogy in isolation of one another. This blending on content and pedagogy allows for a deeper understanding of how particular aspects of a subject are organised, adapted and represented during teaching. Shulman argued that while possessing content knowledge and general pedagogic strategies is important, it is not sufficient for understanding the knowledge of good teaching practices. His argument was that PCK supports the understanding of how content knowledge relates to the teaching process in terms of representing and formulating the subject knowledge such that it is comprehensible to others, namely the students [4]. Therefore "at the heart of PCK is the manner in which subject matter is transformed for teaching" [4, p1021]. This takes place when teachers are able to interpret and transform the subject matter in such a way that makes it comprehensible to their learners.

Though Shulman's approach to teaching still holds true, what has changed since is that "technologies have come to the forefront of educational discourse" [4, p1023] as a result of the new digital technologies available (such as computers, laptops, mobile phones and all the associated software that go with them, together with the Internet) and the growing requirements for their inclusion within the teaching and learning



Figure 4.2: Three circles representing Pedagogy, Content and Technology, where Pedagogy and Content inter-relate (Shulman's PCK) and Technology is treated as being separate. As taken from Mishra and Koehler [4]

processes [4]. According to Mishra and Koehler [4], technologies are able to afford and constrain a "range of representations, analogies, examples, explanations, and demonstrations that can help make subject matter more accessible to the learner" [4, p1023]. They further argue that this new context has foregrounded technology in ways never imagined and as a result the knowledge of technology has become an important aspect of teacher knowledge. Mishra and Koehler [4] note however, that similarly to Shulman's assessments of pedagogy and content, technological knowledge is often considered as separate from the knowledge of pedagogy and content. They represent this idea as three circles, where two overlap, namely pedagogy and content (as per Shulman's framework) and the third, technology, is completely separate (see Figure 4.2). As such technology is viewed as constituting a separate body of knowledge and skills that must be learned, where the relationship between these skills and the basis of teaching, namely PCK, is non-existent and the skills considered to be trivial to acquire and implement in the teaching environment. Indicative of this is that workshops and teacher training programmes tend to promote learning specific hardware and software skills as sufficient for teachers in order that they can integrate the use of technology in the classroom [4].

Mishra and Koehler [4] note that this is not a reflection of the truth but that rather the relationships between Content, Pedagogy and Technology are complex and nuanced. They note how technology is able to constrain and afford representations of content to be taught and that this in turn can constrain instructional decisions and other pedagogical choices. And thus they argue that it is inappropriate to see the knowledge of technology as isolated from the knowledge of pedagogy and content. Instead, their proposed framework "emphasises the connections, interactions, affordances and constraints between and among content, pedagogy and technology" [4, p1025]. They therefore argue that knowledge about content, pedagogy and how they inter-relate is essential for good teaching practices.

Typically, teachers have been expected to acquire an understanding of these complex relationships between technology, pedagogy and content through workshops and training programmes that teach teachers how to use technology; ICT literacy courses. The suggestion or implication is that teachers will be able to successfully integrate technology into their classrooms if they are able to demonstrate their proficiency with hardware and software; what Mishra and Koehler refer to as a leap of faith [4]. The assumption being that knowledge of hardware and software tools leads to good teaching practices with technology. However, approaches that teach skills only are insufficient; learning about technology is not the same as learning with technology [3]. As noted by Harris et al [3], teaching technology skills in isolation (TK) does little to help teachers understand how to use technology to teach more effectively (TPK), nor understand the relationship between technology and subject area content (TCK), nor help students meet particular curriculum content standards through the appropriate use of technology in learning (TPACK).

Mishra and Koehler [4] list some of the reasons why this form of competency knowledge is not sufficient for producing good teaching practices with technology. These include:

- The rapid rate of change of technology;
- The inappropriate design of technology noting that converting general tools for classroom teaching is never trivial or obvious; requiring teachers to engage with the affordances and constraints of particular technologies in order to creatively re-purpose them to meet the specific pedagogical goals of content areas;
- The situated nature of learning technology use is context bound, namely, that it is dependent on the subject matter, grade level, student background and the kinds of technologies available to the teacher; and
- An emphasis on what and not how arguing that workshops that teach specific hardware or software
  applications lead to the accumulation of inert facts and not to knowledge of integration or application
  within teaching environments; merely knowing how to use technology is not the same as knowing
  how to teach with it.

According to Mishra and Koehler [4] the use of the TPACK framework provides an analytic framework and categorisation schemes for the analysis of teacher knowledge. This was exactly what Graham, Borup and Smith [160] did; they used the TPACK framework as a lens to understand how teacher candidates made decisions about the use of ICTs in their teaching. Harris and Hofer [161, 162] cited in Graham, Borup and Smith [160] found that teachers typically first focused on the content to be taught and then choose appropriate activities to support students learning the chosen content. However, in technology integration programmes, often the ICTs affordances and constraints are the starting point. Harris and Hofer [161, 162], cited in Graham, Borup and Smith [160], rather argue that the integration of ICTs should be consistent with the teachers' planning processes and thus teachers would first determine the curriculum (content), then choose the activities. This shifts the focus of the approach away from the affordances and constraints of the technology and rather to content-based pedagogy and student learning needs while still making appropriate decisions about technology in order to support the learning outcomes [160].

Graham, Borup and Smith [160] cite a study by Manfra and Hammond [163] which found that teachers were driven by pedagogy and not technology when planning lessons. The same cited paper discusses a teacher who used technology just to present to the learners, for them to absorb information via it and then reproduce that information later. Rather than using the computers as tools to support/challenge thinking, or inquiry [160]. Graham, Borup and Smith [160], acknowledge that while technology may not be the teachers' focus during lesson planning, they do note that teachers recognise that technology can enhance the pedagogy and improve student content understanding. In fact, "Shulman contended that teaching begins with a process of cognitive reasoning that cumulates in the act of imparting knowledge to others" [160, p534], and also imparting knowledge to oneself. However, cognitive reasoning does not seem to be

encouraged within some government schools in SA, particularly previously disadvantaged schools, where the emphasis rather seems to lie with learners accepting and absorbing facts (no reasoning required or encouraged) for later regurgitation.

Graham, Borup and Smith [160] noted that while initially educational technology research tended to focus predominantly on the technology (TK) as independent from the other two knowledge groups (PK and CK), that has changed to focus more on technology integration. This new focus emphasised knowledge of the use of ICTs within the teaching context; primarily TPK focused as a result of courses being taught by generalists with no content-specific methods [160]. As a result of this, Graham et al [160] found in their study that teacher candidate rationales for choosing particular technologies in their lessons were typically based on general pedagogical practices rather than content-specific pedagogical practices. Furthermore, while they did note that there had been an increase in TPACK orientated rationales from the teacher candidates in their post-course assessment, they still found that most were heavily reliant on knowledge orientated toward TPK. For example, many of the rationales that they received focused on technology as "motivating" or "engaging" the students rather than linking ICTs to any content-specific pedagogies. Graham et al [160] continued, noting that there were very few codings in their study that focused on learner's prior knowledge or learner's misconceptions. In addition, most teacher candidates claimed that the use of technology could improve content understanding but never provided a rationale for how this was accomplished, other than linking to general pedagogical practices or learner characteristics. They therefore noted the need to expose teacher candidates to more content-specific technology integration examples.

Harris, Mishra and Koehler [3] describe an alternative approach to supporting teachers successfully integrate the use of technology into teaching and learning, namely, TPACK-based learning activity types. In their paper they describe how they match the affordances of technologies to the various teaching activities that teachers employ in the classroom in various settings [3]. In order to help teachers make use of the TPACK framework, Harris, Mishra and Koehler [3] began by highlighting some of the possible learning activities within content areas. They called these learning activities "activity types" which they derived from the term "activity structures" from the work of Shulman [158], Polman [164], Windschill [165] and Lemke [166] all cited in [3]. Harris, Mishra and Koehler's preferred term, "activity type" is the "friendlier interpretation of 'activity structures' [3, p404]." "Activity structures" according to Windschitl [165] (cited in [3]) are "a set of classroom activities and interactions that have characteristic roles of participants, rules, patterns of behaviour, and recognisable material and discursive practices associated with them. 'Taking attendance', 'having a discussion' and 'doing an experiment' could all be considered activity structures" [3, p404]. While, Polman [164], also cited in [3], asserts that when a paradigmatically new teaching approach is attempted there is not an "obvious set of well-established cultural tools to structure ... interaction" [3, p404] and thus results in confusion for teachers and a resistance to adoption of the new paradigm (e.g. the adoption of technologies in teaching and learning). Thus, Harris, Mishra and Koehler [3] advocate the conscious identification of new or revised technologically enhanced activity types.

In their 2009 paper [3], Harris, Mishra and Koehler, identify a group of "activity types" that can be used in the teaching of social sciences to middle school aged learners. They argued that while technology affordances create opportunities for enhancing existing activity types and creating new ones, effective teaching requires knowledge of both the activity types and the manners in which technology can be utilized as part of the lesson, and during the course of the paper align "content-driven pedagogical strategies with the specific and compatible technologies" [3, p407]. Not every technology is appropriate for each activity type, and thus particular applications/technologies, based on their affordances, should be selected to match the specific activity type to be used [3]. An example of how Harris, Mishra and Koehler [3] tease out the learning activity type and the corresponding technologies can be seen in Table 4.2. Harris, Mishra and Koehler [3], not only list compatible digital technologies but non-digital as well. Most importantly, the process through

Activity	Activity description	Compatible technologies	
Read text	Students extract information from textbooks,	Books, Web browsers, CD-ROM,	
	historical documents, census data, etc	document viewers	
Group discussion	In small to large groups, students engage in	Discussion forums, blogs, wikis,	
	dialogue with their peers	chatrooms	
Debate	Students discuss opposing viewpoints with	Discussion forums, e-mail, chat	
	their peers		

Table 4.2: Extract from Harris, Mishra and Koehler [23, p408], activity types and their associated technologies

which the authors derived their list of appropriate technologies for activity types, namely the identification of the affordances is not made explicit here, which doesn't provide teachers with a clear understanding of affordances in order that they might be able to make their own decisions in the classroom about which technologies provide the required affordances for their activity types. For example, if one considers the two activities, "Group discussion" and "Debate", both activities involve discussion among students and yet, the "Debate" activity does not include the all the same technologies as the "Group discussion" activity, presumably because the of the different affordances of each technology. For example, blogs and wikis, which are not available in the "Debate" activity's list of technologies, do not capture the names of the contributors in the discussion. In a debate, knowing who said what is important, as there are two opposing sides and all comments or statements need to be attributed to one side or the other and thus the use of a blog or a wiki would not afford the teacher or learners with immediate identification of who said what. Providing a list of the specific affordances would allow teachers to better understand the technologies and thus make more informed decisions regarding what technologies to use in their activity types.

This framework allowed Harris, Mishra and Koehler [3] to acknowledge that different combinations of pedagogical strategies and technologies (each with differing affordances, both enabling and constraining), are appropriate for different disciplines, learning needs/preferences and contexts [3]. However, it still requires that the affordances of those technologies, such as the work begun by Bower [91] and Conole and Dyke [129], be specifically identified so that teachers can make more informed decisions when matching activity types to appropriate technologies. Harris et al [3] commented that a mapping of affordances to activity types should not be seen as the total process in instructional planning, but rather that effective use of technology in teaching required teachers to possess expertise in pedagogical, content and technological knowledge and with respect to their interactions.

As part of selecting appropriate technologies teachers or researchers need to investigate a new way, what Power [127] refers to as new concepts of educational technology [127, p8], for development that is specific to the settings, needs and experiences of rural South African schools. To this end, the SLL has been investigating the appropriate technologies that best suit the needs and experiences of rural South African schools. This dissertation documents that investigation, describing what interventions have been undertaken, lessons learned and providing an understanding of what computing infrastructure needs teachers in the SLL identify as being necessary for teaching and learning in their schools (including an investigation into how teachers are explicitly or implicitly employing teaching frameworks like TPACK and what understanding of the affordances of technologies they have). The specific focus of this dissertation investigates the computing infrastructure that could be best suited to rural schools, in order that, it is financially sustainable, while attempting to provide affordances that will meet the teaching and learning requirements (activity types) in the classroom; with a view to contributing to future work in South Africa that would culminate in the production of an ICT integration guide for teachers that clearly matched technological affordances with teaching activity types.

# 4.7 Conclusion

It is clear that the use of computers has had a profound impact on our society, however teachers still struggle with its influence both inside and outside the classroom [109]. For some that struggle is through providing access to appropriate technologies, while others focus on how to integrate the use of technology in the classroom environment [109]. In the case of rural schools in South Africa the struggle incorporates both the provision of technology as well as the integration of its use into teaching practices. This chapter has described how technologies can afford teachers many opportunities for supporting the teaching and learning activities of the classroom, but has also highlighted that when teachers do not have the necessary theoretical knowledge and approach (such as TPACK) that they are not best placed in order to unlock the affordances of technologies and utilize them in an effective manner, regardless of the policies in place in order to support or force teachers to use available computing infrastructure. Furthermore, in the context of South Africa, a largely developing nation, fundamental access to technology and financially sustainable technology remain core issues at the heart of providing ICTs to schools. The next chapter will focus on various technologies - hardware, software and connectivity - as well as the various policies that govern technological expansion within South Africa, that could have a bearing on providing/sourcing financially sustainable ICTs (specifically computing infrastructure) that best meet the requirements of teaching and learning in rural South African schools.

# Chapter 5

# ICT infrastructure for Education in rural South Africa

# 5.1 Introduction

In this chapter I discuss various hardware, software and connectivity options available for utilization within school environments. In South Africa the first major stumbling block to using ICTs in teaching and learning begins with the often complete lack of access to technologies for the purposes of teaching and learning. The e-Education White paper of 2004 [25] stated that of the schools in the Eastern Cape (5863 schools according to the Eastern Cape Department of Education in 2011) only 8.8% had computers, while only 4.5% had computers for the use in teaching and learning. Nationally, the white paper stated that 39.2% of schools had computers and 26.5% of schools had computers for teaching purposes [25]. More recent data (2013) indicates that of the 24793 public schools in South Africa, 19037 schools do not have computers or a computer centre available for the purposes of teaching and learning, while a further 3267 schools have a classroom (computer centre) designated as a computer centre but without computers [167]. This means that nationally, only 10.03% of public schools in South Africa have access to computing infrastructure for the purposes of teaching and learning, a significant decrease (from 2003 to 2013) in the number of schools with computers available for teaching and learning. Furthermore, in November of 2013 the then Minister of Education, published the "Regulations relating to minimum uniform norms and standards for public school infrastructure" [168], making it law that every public school must have water, electricity, Internet, working toilets, safe classrooms with a maximum of 40 learners, security, libraries, laboratories and sporting facilities [167]. While the figures pertaining to computing infrastructure quoted above are not up to date (but are currently all that is available) it can be seen that there is a dearth of access to technology in South African public schools. Addressing access is the first step in integrating the use of ICTs into teaching and learning (but certainly not the last or only step, as I have discussed in Chapter 4).

This chapter focuses on technologies that are relevant for financially sustainable ICTs deployment in public schools in South Africa, focusing on rural South African public schools. This chapter elaborates on numerous hardware, software and connectivity choices available that could be utilized in providing financially sustainable ICT infrastructure for schools, supporting teaching and learning activities. The chapter also discusses ICT related policies within South Africa, such as the South African broadband policy, SA Connect, and their impact on providing financially sustainable, teaching and learning appropriate ICT infrastructure. The chapter ends highlighting salient points which contribute to an understanding of an appropriate computing infrastructure model for rural South African schools, which is refined in subsequent chapters.

# 5.2 Hardware

As described in Chapter 3, the SLL values the use of conventional PC access (as opposed to deploying mobile end user hardware only) in order to support users that are both consumers and producers of information and thus affording the ability for their ideas and voices to be more readily heard in the digital world. Furthermore, the SLL values the idea of maximizing resource use in order to promote financially sustainable computing infrastructure, leaning toward resource sharing. In this section I look at various technology hardware solutions that can be employed in providing financially sustainable ICTs in schools in South Africa. This section covers the types of clients (thin clients being the type of client predominantly deployed in the SLL up to 2013) that can be deployed in schools, as well as investigates the use of refurbished computers for use in schools and how limited resources drives the need to extend the lifespan of technology. I also investigate and describe different server configurations, such as grids and clusters, that might be employed in order to provide cost effective, and powerful servers that are more easily healed. It is important to note that while the focus is providing a more "standard" desktop computing experience in terms of school computing infrastructure, the systems discussed here (and in subsequent chapters) do not preclude the use of mobile computing infrastructure such as tablet PCs or mobile phones. These devices are perfectly integrable into the overall computing infrastructure model proposed by the SLL and this research work.

# 5.2.1 Computing clients

There are typically three types of computer clients: thick clients, thin clients and fat/hybrid clients.

#### 5.2.1.1 Thick client computers

A thick client computer is a computer that typically provides rich functionality independent of any central servers. Thick clients can be stand-alone computers or networked in such a way that they are able to communicate with a central server typically to provide file storage or access to other centralized services but without which the computer is still able to function independently [169]. Thick clients tend to communicate with the central server only to store or retrieve previously stored data, thus communication between the thick client and the server is not continuous. Thick clients thus rely on and utilize all the local hardware resources of the client computer in order to provide a usable system to the end user. The advantages of thick clients are: lower server requirements but higher capacity - thick clients do not require the same level of performance from the server as thin clients do and thus are able to support more clients; ability to work "off-line" - don't need to be constantly connected to the central server; better multimedia performance - thick clients can better support multimedia rich services that would be bandwidth intensive if handled by the server; and local file and application storage [170]. The disadvantages to thick clients are they often more expensive than thin clients to deploy and require more work to deploy; they require more resources than thin clients; they present greater security considerations; and they are harder to maintain and configure, particularly if the hardware is non-uniform [171].

# 5.2.1.2 Thin client computers

Thin clients are typically not capable of performing any computations and rely on their access to a central server to perform all necessary computations for them [169]. Thus without access to the central server the thin client can not function in any meaningful way. Thin clients thus merely take input from the end user and display output to the end user and all computation is done on the central server, utilizing predominantly the hardware resources of the central server [169] (obviously graphics, I/O and networking hardware resources are utilized on the thin client). Typically thin clients are simpler PCs in that they don't have a hard disk

(just motherboard, graphics, networking, memory, monitor, keyboard and mouse) and thus require constant communication with the central server in order to send input from the user and display the output from the server to the user via their monitor [169]. Thin clients can typically be classified into three groups: Ultrathin/zero client, Run time environment client, and Web thin client. An Ultra-thin or zero client is a thin client which doesn't run a full operating system. Rather the kernel merely initializes the network interfaces and handles the display of the server's output. An example of this type of client is an Edubuntu LTSP thin client. The Run time environment client contains task specific applications like Firefox for Internet browsing and a minimal underlying operating system in order to run those applications. The Web thin client only provides a web browser and relies on web applications to provide general purpose computing functionality [172]. The advantages of thin client computers are: they are easy to deploy as they require no software installation; single sign-on, so any workstation can be used to access ones account and files; opportunity to use older PCs as thin clients; reduced security threats; and they are easier to maintain as only the server needs to be maintained - if a client PC fails there is no loss of data as everything is stored on the server. The main disadvantage of thin clients is that if the server fails, all clients are rendered useless; the server is not easily healed and presents a single point of failure that can result in a large amount of downtime [172, 171, 169].

Figures 5.1 and 5.2 display thick client and a thin client computers. For an explanation of the configuration of a Edubuntu Standard thin client configuration see Appendix F.



Figure 5.1: Thick client computers; some of which connected to a central server for data backup and retrieval others stand-alone



Figure 5.2: Thin client computers, all of which connected to the central server in order to operate

#### 5.2.1.3 Fat/Hybrid client computers

A fat/hybrid client is a mixture of the other two client types, think and thin. Fat/Hybrid clients, like thick clients, are able to run applications locally on their hardware but fat/hybrid clients, like thin clients, rely on

the server for persistent storage of data and files, including their operating system and user authentication [173, 174, 175]. Fat/hybrid client offer advantages from both the thick and thin clients. For example, multimedia support and high performance like a thick client, together with easy management and flexibility like a thin client [173, 174]. Unfortunately, because the OS (and root file system) of the fat/hybrid clients, together with all the user data is still stored centrally on the server, the server is not easily healed and represents a potential single point of failure for the computing infrastructure where fat/hybrid clients are employed [174].

Both fat/hybrid and thin clients, can be realized through the use of refurbished computers (naturally, one could also use refurbished computers as thick clients, but obviously they will be less powerful than newer versions). In the next section I discuss the use of refurbished computers for use as end point devices for users in school environments.

#### 5.2.2 The digital divide and the possible use of refurbished computers

The 'digital divide' commonly refers to the gap between those who have access to technology and ICTs (i.e. Computers and Internet access) and those who do not [176]. The digital divide is defined by the Organization for Economic Co-operation and Development (OECD) [177] as "the gap between individuals, households, business and geographic areas at different socio-economic levels with regard both to their opportunities to access information technologies and to their use of the Internet for a wide variety of activities" (as quoted in [176, p1]). A typical approach to bridging the digital divide amongst or within countries is through the provision of technology and Internet resources to schools [176], indeed as the SLL is doing.

Studies conducted into bridging the digital divide seem to suggest that a crucial element to bridging the gap focuses around the issue of the cost of technology and Internet access. James [178], states that "the costs of computers used in developing countries can be reduced in two main ways, the first of which is by extending the lives of existing models and the second of which is by designing entirely new products" [178, p25]. An example of a new design would be that of the OLPC project; whose vision is to produce laptops designed for children that are affordable to the governments of developing countries in huge quantities [176]. The programme has however been fraught with problems [179, 128], which is the risk in new product development. Furthermore, the very design of the laptops is for young children making their use by other sectors of the communities (which would go some way to sustainability) during non-school times less feasible. The alternative, as James [178] stated, is that we extend the life span of already existing technologies. One way of doing this is through refurbishment [176].

Besides reducing the overall costs of technology procurement, refurbishment is an environmentally sustainable strategy for bridging the digital divide [176]; e-waste contains hazards such as lead, mercury, arsenic and flame retardants, which can cause damage to the environment [176]. Increasing the life span of technology will thus hopefully lessen the demand of new technologies and slow the pace at which we have to dispose of technologies.

In addition to the cost of technology other factors influencing the growing gap include lack of telecommunications infrastructure and the high prices in developing countries, human capital, and regulatory quality and rules [176]. According to Marthaler [176] there is no academic consensus regarding appropriate policies to implement and achieve higher IT penetration rates in developing countries; some argue that the market alone will take care of the disparities; while other argue that there should be government intervention.

Marthaler [176] undertook a study to investigate sustainable methods of providing technologies (namely computer resources) to schools in Colombia (considered to be a developing nation). He found that of the scenarios he investigated the most sustainable method of providing computers to schools in Colombia was through refurbishment of computers donated by local Colombian industries and refurbished locally in

Colombia [176]. The second most sustainable solution was through the refurbishment of computers from local Colombian sources together with computers provided by overseas companies through international refurbishment companies like Computer Aid [176]. Additional findings included that computers that were refurbished recycled PCs rather than just recycled PCs had a 16.8% better environmental performance. Furthermore, Marthaler found that computers that were properly maintained lasted a further two years (on average) in the schooling system [176]. These two findings highlight the importance of computers undergoing some refurbishment prior to deployment in the schools and the necessity (or what Marthaler [176] refers to as the "indisputable benefit" of PC maintenance) for ongoing maintenance. In addition, Marthaler reported on previous research work that found that of the energy consumption of a computer throughout its lifespan, 81 % of it was used in the production phase to make the computer and only 19% in running the computer [176]; it takes a lot more energy to produce computers than to run computers, thus its more energy wise to have computers run for longer periods than shorter periods.

It is argued by researchers in [180] that PC refurbishment is one of the most effective mechanisms to

- · create additional value to the computers being refurbished; and
- prevent an accelerated flow rate of computers through the whole system being production, sale, use, and disposal.

Streicher-Portem, Widmer, Jain, Bader, Scheidegger and Kytzia [180] further argue that the incentive to utilize refurbished computers is twofold, first, there is an increasing need for low cost personal computers, and second, the refurbishment process creates opportunities for the creation of businesses in a second hand computer industry while decreasing the overall load of e-waste [180].

These findings in Colombia might be applicable to other developing countries. The sustainability of the Colombian refurbishment project was largely attributed to the growing demand of new or reused technologies from the private and public sectors in combination with the low labour costs, both of which are commonly noted in developing nations [176], which would include South Africa.

Cossa and Cronjé [125] however argue that the use of second hand computers should be avoided in schools as they require more maintenance and schools do not have the skills to cope. However it can be reasonably argued that maintenance is a key factor to the provision of technology in all spheres (be that education or business, etc). Whether computers are new or refurbished/recycled they are going to require maintenance. ICT implementation projects that do not consider or plan for maintenance are therefore by their very nature not sustainable, regardless of the age of the technologies deployed. In addition, school children do not necessarily require high end computers in order to achieve the learning outcomes set in their curriculum [176]. Furthermore, electronic waste or e-waste can be considered as dangerous to humans and the environment [180].

# 5.2.3 Grid computing

Foster and Kesselman define the grid as "a system that co-ordinates distributed resources using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of service" [181, p46]. While [182] defines the grid more thoroughly as "a system that co-ordinates resources which are not subject to centralised control, using standard, open, general-purpose protocols and interfaces to deliver nontrivial qualities of services" combining resources from different organisations towards a common goal [182, p53]. The hardware resources that underlie a grid include computers, networks, data archives, scientific instruments, visualisation devices and so on. These devices are distributed and heterogeneous with differing performance profiles. The resources are also often dynamic, in that old resources are removed and newer ones added and there are varying performance profiles for each resource available to the multiple users of

the grid [183]. Another key element to the grid is the software that allows the users to access the underlying resources. A popular example of such grid management software is the Globus Toolkit [183]. Grid computing allows administrators to "virtualise" or pool IT resources (computers, storage and applications) into a single virtual system whose resources can be managed from a single administration console and allocated dynamically based on demand [184].

The core purpose of grid computing, maximising resource utilization is not a new idea. In the early days of computing, technicians would often look to make use of spare machine cycles in order to get the most out of the hardware [184]. However, the physical development of grid computers was a follow on from parallel computing in the 1980s and 1990s [183, 184]. Researchers from multiple disciplines were coming together in attempts to solve "Grand Challenge Problems" [183]. These are large problems in science and engineering that require large scale computational infrastructure to solve, such as aerospace simulation, circuit design and human DNA sequencing [183, 184]. As application developers built software that pushed the boundaries of parallel machines, they began to consider distribution beyond the boundaries of the physical machines as methods for solving large scale problems [183]. Software developed for parallel computers focused on communication management between processors and development and execution environments for parallel machines. This led to the development of some well known parallel and grid software such as Parallel Virtual Machine (PVM), Message Passing Interface (MPI), High Performance Fortran (HPF) and Open MP to support the communication for scalable applications [183]. Successful application paradigms were thus developed in order to leverage off the potential of shared and distributed memory architectures. It was initially thought that the grid would be most useful in extending parallel computing (as this was what initially led to its development) from tightly coupled clusters to geographically dispersed systems. However, the grid has rather been utilised as a platform for the integration of loosely coupled applications and for linking separate resources such as storage and computation. The management of the grid resources that are geographically separate is what distinguishes grid computing from parallel computing [183].

The first modern grid is generally considered to be the information wide-area year (I-WAY), which was developed as a demonstration at the Supercomputing conference in 1995 (SC95). During the conference researchers worked together to connect a national distributed testbed of 17 sites. During the course of the conference over 60 applications were developed including a rudimentary grid management application that provided access, security and co-ordinated resource allocations. The I-WAY grid spring-boarded development of grid software. Applications such as Globus provide basic system level grid information; the Condor project, which provides high-throughput scheduling; and Mars, which provides high-performance scheduling, resulted from the I-WAY experiment and have provided the foundations of much of today's grid software, understanding and research [183, 184].

In the late 1990s grid researchers came together and formed the Grid Forum, which later became known as the Global Grid Forum (GGF). This body is developing much of the grid research into standards for future grids. The GGF is developing key Grid-specific standards such as the Open Grid Services Architecture (OGSA), which is emerging as the de facto standard for grid infrastructures [183]. The lack of standards hampered grid development; the standards developed are helping to establish a common vocabulary and systemization of concepts that makes grid research and development easier [184]. Today the grid is global, with worldwide collaboration, involving government agencies, academic researchers, universities, national centres and laboratories, with research in related areas such as networking, digital libraries and peer-to-peer computing providing additional relevant ideas to grid research [183].

At the centre of any grid is the network that connects the geographically distributed resources together allowing them to be used collectively to support the execution of applications. The development and increase in high capacity networks will in turn increase the capacity of grids to support both parallel and distributed applications [183]. As the volumes of data increase (in the range of petabytes) and users require faster access to resources, a key enabler of grid technologies is high speed networks [184]. In the future, wired networks will be further enhanced by the improvements of wireless technologies and networks, which will drive the integration of smaller and smaller devices into grids. Users will also be able to access grids from handheld devices such as cellular phones which will promote grids as platforms for e-Science, e-Commerce and e-Education [183]. To manage the changing technological landscape, grids utilize a common infrastructure to provide a virtual representation of the grid to developers and users while allowing for the ease of incorporation of new technologies. Thus the development of standards that allow the complexity of the grid to be managed by developers and users is critical to the success of the grid; standards such as the Web Services Definition Language (WSDL) and OGSA. Furthermore, the use of common infrastructure and standards promote interoperability and re-usability. The use of open source software and standardisation are changing the way in which software is written and systems are designed, impacting the evolution of grid technologies.

There are many grid projects and implementations available around the world today. Projects are making use of computers of all shapes and sizes as the nodes within the grids, from the ASCI program and TeraGrid Project in the USA to the e-Science project in the UK, which also includes supercomputers that house many hundreds of processors for parallel processing. One of the fastest growing application areas in grid computing is the "Life Sciences", which include areas like computation biology, bioinformatics, genomics and computational neuroscience. It is said that data is "emerging as the 'killer application' of the Grid [183, p34]". Data can be collected in many ways, such as scientific instruments, experiments, sensors and sensornets. It is noted by [183, p34] that grids will be used to "collect, store and analyse data and information" and from which generate new knowledge.

The UK e-Science grid includes access to a 7 Teraflop High Performance Computing (HPC) machine located at Daresbury Laboratory, the HPC Computer Services for Academic Research (CSAR) and other high performance clusters available via the UK Grid Certification Authority. The UK e-Science grid is intended to provide a model for a production grid that can be utilised by both academic researchers and industry for evaluation [183]. The TeraGrid in the USA is an initiative of the National Science Foundation (NSF) to build and deploy the world's largest distributed infrastructure for open scientific research. The TeraGrid resources included more than a petaflop of computing capacity and more than 30 petabytes of on-line archival data storage linked via high performance networks. Researchers can also access over 100 discipline specific databases. This distributed cyber-infrastructure brings together a number of collaborators from a number of institutions across the USA through the Grid Infrastructure Group (GIG) at the University of Chicago, working in partnership with the resource provider sites, namely, Indiana University, the Louisiana Optical Network Initiative, National Center for Supercomputing Applications, the National Institute for Computational Sciences, Oak Ridge National Laboratory, Pittsburgh Supercomputing Center, Purdue University, San Diego Supercomputer Center, Texas Advanced Computing Center, University of Chicago/Argonne National Laboratory, and the National Center for Atmospheric Research [183, 185].

Another well known example of grid technologies is the SETI@Home project. It is a highly distributed data-mining application whose goal is to determine patterns of extraterrestrial intelligence from the large volumes of data received from the Arecibo radio telescope located near the city of Arecibo in Puerto Rico [183]. Through SETI@Home millions of computer owners worldwide are contributing to the search for extraterrestrial intelligence by allowing their computers to analyse SETI data while the computer is idle. Participants download a client program which is only activated when the computer is idle. The client program fetches data from the SETI@Home server and begins to process it, sending back results when complete then retrieving more data. This continues until the computer user wishes to use their computer again and the program is halted [186].

Grid technologies are also seen as possible technologies that can be used to support education. Grids offer access to resources such as large data stores of information and knowledge, large processing capacity for data manipulation, interpretation or simulation and other resources that could be used in education [183]. Access to these resources and technologies potentially afford learners a richer technological environment.

#### Benefits of grid computing

The benefits of grid technologies include cost effective, computing intensive, resource rich solutions. Grids are cost effective because often pooling many smaller resources together is cheaper than the purchase of a single supercomputer with similar capabilities. In addition, pooling resources into a distributed collection and network of technologies results in massive computing power, massive storage facilities, collaboration and access to technologies not locally available [184]. Minoli [184] describes grid computing as a promising technology for three reasons:

- 1. Grids are a more cost effective use of a given number of computing resources;
- 2. Grids are a method for solving problems that require significant computing power; and
- Grids afford the cooperative and synergistic harnessing and management of resources for a common objective.

Minoli [184] argues that often there is not a need for more computing hardware or resources but rather better utilization of what is already available; mainframes, servers, desktops and workstations all "harbour enormous, untapped processing and storage power" [184, p34]. The Grid allows one to connect all the underutilized resources together and manage them while providing secure and reliable access. In industry, companies have reported that when moving to grid technologies their processor utilization rates have grown by as much as 80% while costs have dropped by as much as 90%. Some firms have found that using Intelbased Linux servers can be 1% to 10% of the cost of mainframes or high-end UNIX servers [184].

Additional Grid benefits listed in [184] include:

- access to a plethora of IT resources;
- better utilization of underused resources;
- improved availability of computing;
- increased reliability of computing;
- parallelization of processing;
- resource balancing;
- · simplified management of IT resources; and
- virtual resources and virtual organisations for collaboration.

Grid computing relies on the problem or task at hand being parallelisable. This means that if you are not able to break down the problem into a series of smaller tasks that can be run in parallel then grid computing does not provide the improved performance that one expects. The greater number of smaller parts that it can be divided into the better the scalability.

Grid developers and researcher believe that the development in this field is only beginning and that in the future grids will expand existing technologies and include new technologies; more resources will be linked by more and better networks. They postulate that smaller devices, such as tablet PCs or cellphones, will be able to access and join grids producing completely heterogeneous grids of varying performance capabilities [183]. The future is described as a time in which computing, storage and software are no longer seen as objects that are owned by the individual but rather as services or utilities to which we subscribe.

#### 5.2.4 Cluster computing

A cluster consists of a number of independent computers connected to one another through a switch working together as though they were one machine [187, 188]; aggregations of processors in tightly coupled parallel-based configurations [184]. The difference between a cluster and a grid relates to the way in which resources are managed. In the case of clusters, the resource allocation is performed by a central resource manager and scheduler. In addition, nodes work together as a single unified system. In the case of grids, however, each node has its own resource manager and does not aim to provide a single system image [184]. Cluster and Grid computing differ significantly – Grid computing can be regarded as the evolution of the cluster [9]. The main difference between the two technologies is that grids allow for the management of geographically disturbed computational resources which support different hardware configurations and operating system versions [9]. The birth and evolution of large scale search engines resulted in the need for increased computing resources to support the highly parallel search queries. Major search engines, like Google, are based on the cluster-based computing paradigm. This was the most suitable technology for parallel computing – alternatives included supercomputers and mainframes which for economic reasons and the need for highly personalised systems were rejected [9].

The development of switches as part of LAN technologies meant that it was no longer essential that people relied on custom machines and networks to build parallel machines. With LAN technologies available as off-the-shelf components, together with commodity PCs such as desktop computers and storage devices, an inexpensive computing infrastructure can be created to tackle large computing problems. Because clusters are made of a number of components they are easy to scale; for more complex problems we just add more resources. There are many applications which lend themselves to running on coupled machines (such as clusters), namely, databases, file servers, web servers, simulations, etc. Often these applications need to be highly available with some form of fault tolerance which clusters offer because there are a number of machines working together to offer the service, so if one fails the other machines will continue functioning normally [187], albeit at a performance loss to the overall service. The dependable and scalable nature of clusters is one of their greatest advantages. Because they consist of independent machines working together, when one breaks or needs to be taken down for repair or replacement the others are not affected. The cluster software runs on top of each computer in the cluster making it easier to disconnect and replace broken machines. This highly available nature of clusters makes them attractive to service providers on the world wide web. In addition, Internet applications such as search engines are amenable to more loosely coupled computers because service provision requires the completion of millions of independent tasks making them highly parallelizable. Accordingly, companies like Amazon, Yahoo!, Google, Wikipedia, rely on clusters of PCs or workstations to provide services on the Internet that are used by millions of people each day. Another advantage of clusters is that they are competitively priced. They are typically made from commodity off-the-shelf components which are cheaper than specialty hardware. They are also therefore easier to source and replace when components fail. Considering Figure 5.3, one can see the complexity of Wikipedia's resources as of December 2010, in order to facilitate the 25000 - 60000 page requests per second during the course of a day [5]. In order to facilitate such high volumes of data requests Wikipedia runs on a dedicated cluster of Linux servers, mainly utilizing the Ubuntu OS [5].

There are a few drawbacks to clusters. Firstly, clusters are usually connected using the I/O bus of the computer whereas multiprocessor systems are connected via the memory bus. The memory bus has higher bandwidth and a lower latency which allows the multiprocessors to drive the network connection at a higher



Figure 5.3: Overview of the system architecture for Wikipedia as of December 2010, as sourced from Wikipedia [5]

rate and have fewer conflicts with I/O traffic. In addition, this means that clusters use software based communications while multiprocessors use hardware based methods of communication. The second drawback is the division of memory in the system. A cluster of N machines will have N separate memories and N separate copies of the operating systems. However, multiprocessor systems have one copy of the operating system and so allow applications to access almost all the memory. This means that a sequential program on a cluster has access to 1/Nth the memory that it would on a shared-memory multiprocessor. However, as the price of memory drops this becomes less of an issue as individual computers in the cluster are able to have more memory at an affordable rate [187]. Another drawback to clusters is the cost of administrating the machines. According to Hennessy and Patterson [187], the cost of administrating a cluster of N machines is the same as administrating N independent machines, while the cost of administrating a single machine with multiple processors (N processors) is roughly the same as administrating one machine. A suggestion from the authors of [187] to lower administration costs is to keep the cluster data outside of the cluster, in, for example, a storage area network (SAN). This would allow the individual computers in the cluster to be treated as clones of one another. Thus if one of the nodes fails it could simply be discarded (instead of spending time and money repairing) and later replaced with another computer.

Hennessy and Patterson [187] describe four examples of cluster design, highlighting the advantages and disadvantages of each and their associated costs. Interesting points to note regarding their findings were that there is a cost to performance ratio that needs to be carefully balanced when purchasing the PCs to use in the cluster. In addition, smaller computers are generally cheaper and faster for given functions compared to larger computers. Furthermore, the cost of purchase of the components needed to build a cluster is less than half the cost of ownership, thus cheaper hardware solutions don't necessarily conserve as much money as one would expect, and sometimes expensive purchases, such as a SAN could save you more money in the long run in terms of operation costs in the future. Finally, the amount of space and the power that the cluster requires are important to consider as they can significantly contribute to the overall cost of operations and thus ownership [187].

#### The example of Google

A highly successful example of cluster PCs is that owned and run by Google. Between 1994 and 1997 the number of pages on the Internet grew by a factor of 1000. Generally, consumers of web searches are only interested in the top 10 results for their search stream. Thus, search engines need to not only search through many millions of pages, but they also need to rank them in order of most appropriate to the user's request. From the outset Google was designed to be a search engine that could scale at the growth rate of the Internet. In addition to designing a system that would keep up with the growth rate of the Internet, Google also improved the relevance of the top queries produced so that users would more likely get the results they desired [187]. Google employs the use of Domain Name Service (DNS) round robin in order to load balance search requests across its cluster of PCs (see Figure 5.4). A DNS round robin cluster employs DNS round robin in order to load balance incoming requests (and thus server load) across the entire pool of servers listed in the DNS round robin. DNS round robin is a method for load distribution or balancing, or fault-tolerance by providing multiple, redundant Internet Protocol service hosts through controlling the DNS responses to address requests from client computers [189]. In its simplest implementation DNS round robin works by responding to DNS requests with a list of Internet Protocol (IP) addresses of several servers that host identical services. With each DNS response, the IP address sequence in the list is permuted, which is the basis for the use of the term round robin. Typically IP clients attempt connections with the first address returned from a DNS query so that on subsequent connection attempts clients would receive responses from different servers, thus distributing the overall load across the servers [189].

ings@tribble:~\$host www.google.co.za www.google.co.za is an alias for www.google.com. www.google.com is an alias for www.l.google.com. www.l.google.com has address 66.249.93.104 www.l.google.com has address 66.249.93.147 www.l.google.com has address 66.249.93.99

Figure 5.4: Results from a DNS server query requesting the A-records for www.google.co.za



Figure 5.5: One of Google's geographically dispersed clusters of commodity computers, sourced from [6]

Search engines require more computational power than most other web based services; on average a single Google query searches hundreds of megabytes of data and uses tens of billions of processor cycles. In order for Google to support the peak query rates, which is in the order of thousands per second, they require infrastructure which is comparable with the largest supercomputers available [190]. At the time of the last publication in 2006, Google reportedly achieved this using an estimated 450000 commodity-class computers with fault tolerant software to create a solution which they believe to be more cost effective than comparable systems built from fewer high-end servers [190, 191]. Fortunately Google's searching services are easily parallelisable and different queries can run on different processors, in fact, their searching methods are designed that even a single query can be further broken down and run on multiple processors [190]. The 450000 computers making up this supercomputer are not housed in one location, but are scattered across the world in smaller clusters consisting of thousands of computers (see Figure 5.5). When a user requests information from Google the user's browser, via their computer, will perform a DNS lookup to map the www.google.com domain name to a specific IP address. A DNS-based load-balancing system will select a cluster to redirect the user's request to, based on their geographic location and the current load of the various clusters. This load balancing system minimises the latency or round-trip time experienced by the end user as well as ensuring no one cluster is more heavily loaded than the others [190]. It is Google's goal that no search should take longer than 0.5 seconds to complete, including network delays [187]. The user's browser will then send the query to the chosen cluster, after which all processing for that query will be handled by that cluster only [190].

At each of the clusters a hardware load balancer monitors the available set of Google web servers and performs local load balancing of the queries that the cluster receives. Once a Google web server has received a query it co-ordinates the query execution. Query execution consists of a number of phases, the most important of which are searching through the index servers and searching through the document servers. The search through the index servers produces a list of matching documents that are related to the query. The index servers determine the relevance of the document list by finding an intersection using all the query words and then computing a relevance score for each document. That relevance score determines the order of results presented on the output page. Once the list has been composed the document servers retrieve a copy of each document to provide information such as the exact title and document summary that will be presented to the end user. Additional phases include checking the spelling of a query and locating relevant advertisements (if available) that should be returned with the results. Once all phases are complete, the Google web server generates the HTML page and returns it to the end user's browser. The scale of the cluster(s) and the various levels of load balancing within the Google system make it one of the biggest and most complex content delivery networks and crucial to their business [190]. In addition to needing a large number of PCs and hence processors to compute the search requests from its users, Google also has a reliability requirement because people all over the world are making use of Google throughout the entire day. Google must therefore be continuously available. Google's cluster configuration provides the high availability and reliability that Google need in order to reliably provide their services [187].

## The example of the Single System Image

Another example of a cluster implementation is a Single System Image (SSI). A SSI is a cluster of computers that appear as though they are a single unified computing resource, achieved by hiding the heterogeneous and distributed nature of the available resources [16, 7, 192, 188, 193]. SSI can be achieved either via hardware or software (at either the operating system, middleware or application layers) mechanisms. SSI provides a globalized view of the available resources to the users, regardless of which node in the cluster the user is connected to. In addition, SSI can ensure that the system continues to operate after individual node failures (high availability) as well as can ensure that the system is evenly load balanced and provides communal multiprocessing (resource management and scheduling) [7, 192]. Typically SSI design goals are transparency, system availability and scalable performance [7]. The concept of SSI clustering is not new; the Lotus Distributed System provided an SSI environment from the 1980s, while Sun Microsystems had an SSI project in the mid 1990s and SSI concepts were being discussed in textbooks by the late 1990s [192].

Depending on their intended use, SSI systems can provide a subset or all of the features listed in Table 5.1, while the functional relationship amongst the various middleware packages can be seen in Figure 5.6 (taken from [7]). There are a number of advantages provided by SSI systems [7]:

- they provide a simple, straightforward view of all the system resources and activities from any node in the cluster;
- they free the end user from having to know where on the cluster their applications are running;
- they afford the transparent use of resources regardless of the physical location of the resources;
- they facilitate the use of familiar environments (interface and commands) and allow administrators to manage the entire cluster as a single entity;
- they offer the same command syntax as other systems and thereby reduce the risk of operator errors and improving performance, reliability and high availability;
- they support centralised or decentralised system management and control to minimize the need of skilled system administrators;
- they simplify system management and thereby reduce cost of ownership;
- they provide location-independent message communication;



Figure 5.6: The relationships among middleware modules in SSI Systems [7]

- they reduce the time, effort and knowledge requirements of system programmers and support staff in handling large and complex systems; and
- they promote the development of standard tools and utilities.

An SSI system can be implemented in one or more of the following levels:

- hardware [192];
- operating system, e.g. GLUnix, Mosix [192];
- middleware, e.g. CODINE and JESSICA [192, 188]; and
- application. e.g. Linux Virtual Server [192].

In a good SSI there is usually co-operation between all these levels as a lower level can simplify the implementation of a higher one. Some examples of SSI clustering systems include: Genesis, Inferno, Kerrighed, LinuxPMI, LOCUS, MOSIX, OpenMosix, OpenSSI, JESSICA and VMSCluster [16, 188, 193]. Today (2017), however, there is not as much enthusiasm or hype around the idea of SSI systems or for owning SSI systems in order to provide cluster servers. Kusnetzky [194] argues that this is the case because IT solutions seldom build or deploy monolithic blocks of code as a computing solution and as such environments that are designed to virtualise the functions of those systems are no longer necessary. Instead, Kusnetzky [194] argues that large code blocks are implemented as network services and are broken up and hosted on multiple machines; arguing that the SSI approach has been replaced with Service Orientated Architectures (SOA) as the approach of choice for major application design, typically serviced to the end user via Cloud computing services.

# 5.2.5 Cloud computing

Cloud computing is a technology which is the result of the convergence of several computing trends: service-oriented architectures, (distributed and parallel) grid computing, cluster computing, peer2peer computing and virtualisation [8, 182]. Weiss [114] describes cloud computing as web-based applications for end users that rival thin client computing, or utility computing [195, 117, 9] that is metered out to users and companies, or large scale distributed and parallel computing – cloud computing takes on many shapes and forms [114]. Armburst, Fox, Griffith, Katz, Konwinski, Lee, Patterson, Rabkin, Stoica and Zaharia [117]

SSI Feature	Feature description		
Process migration	A process can be started on one node but later moved to another node in order to		
	balance the load across all servers. As processes are moved from one node to		
	another other associated resources can also be moved with them (E.g.		
	Inter-Process Communication (IPC) resources)		
Process check-pointing	Process check-pointing allows the current state of a process to be saved and		
	reloaded later. Often used in conjunction with process migration; process is first		
	checkpointed, then migrated and then restarted on the new node		
Single process space	When the SSI system gives the illusion that all processes are running on the same		
	machine. All processes, regardless of which node they reside on, have a unique		
	cluster-wide process ID and they are able to create child processes on the same or		
	different nodes and communicate with other processes regardless of which nodes		
	they are running on. SSI systems should support globalized process management		
	and allow the management and control of processes as if they were running on		
	the same local machines		
Single root/Single file	When the SSI system provides a single view of the file system, either via NFS,		
hierarchy	AFS, xFS, shared disk devices or file replication		
Single I/O space	When SSI systems allow all nodes to access the I/O devices of other nodes		
Single IPC space	When SSI systems allow processes on different nodes to communicate using IPC		
	mechanisms as if they were running on the same machine. Sometimes this can		
	include shared memory		
Cluster IP address	When SSI systems provide a cluster-wide IP address; a single IP address visible		
	from outside the cluster that can be used to communicate with the cluster as if it		
	were a single computer		
Single user interface	When SSI systems provide a single GUI thought which the user is able to		
	interact with the cluster		
Single memory space	When the SSI system creates the illusion of a big, centralized main memory,		
	which in reality is a set of distributed local memories		
Single virtual networking	When any node in the SSI can access any network connection throughout the		
	cluster domain even if the network is not physically connected to all nodes in the		
	cluster		
Single job management	When a user job can be submitted from any node to request any number of host		
system	nodes to execute it		
Single control point and	When the entire cluster and its nodes can be configured, monitored, tested and		
management	controlled from a single application using GUI tools		

Table 5.1: Features of a SSI cluster systems (from [16, 7])
describe cloud computing as both the applications delivered as a service over the Internet and the hardware and software in the datacentres that provide the services. They [117] describe the Cloud, which when made available to users in a pay-as-you-go manner, as a Public Cloud – the service being sold is utility computing [117]. Alternatively, Private Clouds are datacentres that are not made available to public users but are only used within the organisation or business which own them [117].

Another description of cloud computing is "computing over the Internet"; a computing paradigm that exploits the resources provided by computer networks, where the cloud architecture assumes that the hardware and software services are stored on the web servers, the "Cloud", rather than hosted by single computers connected via the Internet [9]. As long as the user's end device (be it PC, laptop, cellphone, etc) has a web browser installed and can access the Internet the user will be able to access the "correct" cloud and the services that they require [9]. Cloud computing affords users access to software, data and other services from anywhere in the world [9].

The numerous competing definitions of what cloud computing has left researchers and industry in a state of confusion [182]. Furthermore, Vaquero, Rodero-Merino, Caceres and Linder [182] argue that the lack of a coherent definition hinders further research and wider scale adoption. For this reasons they have proposed the following, comprehensive definition of cloud computing:

"Clouds are a large pool of easily usable and accessible virtualized resources (such as hardware, development platforms and/or services). These resources can be dynamically reconfigured to adjust to a variable load (scale), allowing also for an optimum resource utilization. In fact, if the systems require human management to allocate processes to resources, then the system is not a Cloud [9]. This pool of resources is typically exploited by a pay-per-use model in which guarantees are offered by the Infrastructure Provider by means of customized SLAs [182, p51]."

The elasticity of the resources (accessible resources can grow and shrink on demand) at reasonable costs is unprecedented in the history of IT [117]. The elastic nature of the Cloud is provided through the use of virtualisation, which allows server resources to appear as if they can dynamically grow and shrink as needed [117]. Cloud computing has increased the attractiveness of software as a service and removed the necessity for developers to make large capital outlays of hardware in order to provide new innovative Internet services [117].

Arguably the concept of cloud computing is not entirely new [195]. Hayes [195] likens it to the earlier times in computing history when service bureaus and time-sharing systems were created to provide access to computing machinery for those users who did not own their own mainframe in a room down the corridor [195]. However, with the invention of the personal computer in the 1980s users could own individual computers and the idea of a shared resource beyond the typical client server model of central storage faded [195]. Developments in Internet access and a greater need for high performance computing has once again pushed the computation outwards from the local computer to data centres somewhere on the Internet [195]. However, this development is not a return to the earlier access of dumb terminals to a mainframe computer, because not only are the workstations of the user not dumb but they are also able to access multiple data centres, which in turn can also communicate amongst themselves [195]. Armburst et al [117] identify three new aspects from cloud computing from the perspective of hardware: firstly, the illusion of infinite computing resources available on demand; secondly, the elimination of up-front commitment by Cloud users (they can start small and increase hardware resources over time, or as needed); and finally, the ability to pay for use of computing resources on a short-term basis as needed (e.g. processors by the hour or storage per day) [117]. This outsourcing of computation to the Internet provides the end user with many benefits, such as access to their data and applications from anywhere, vast computing resources and the ability to be mobile and collaborate [195].

#### **Cloud computing structure**

Youseff, Butrico and Da Silver [8] postulate that there are five layers to cloud computing: applications, software environments, software infrastructure, software kernel, and hardware. The hardware layer is at the bottom of the stack and consists of the physical components of the system. The next layer is the software kernel layer, followed by the software infrastructure (which is composed of Computational resources, Storage and Communications), then software environments. Finally, the top of the stack is the cloud application layer, which is the interface to cloud users through web browsers and thin client terminals [8].

Each of these layers provides a different set of services to potential customers; the hardware layer provides Hardware as a Service (HaaS) to end users who would like total control over their infrastructure; the Cloud Software infrastructure layer provides Infrastructure as a Service (IaaS), Databases as a Service (Daas) and Communication as a Service (Caas); the Cloud Software Environment provides a Platform as a Service (PaaS); and finally, the Cloud Application layer provides Software as a Service (SaaS) [8, 182, 9]. For an in-depth explanation of each of the layers see [8]. A pictorial representation of the cloud ontology can be seen in Figure 5.7, while the Cloud Computing Conceptual stack (which is an abstraction of the cloud ontology) can be seen in Figure 5.8.



Figure 5.7: Cloud ontology from [8], the "layered figure represents the inter-dependency and composability between the different layers in the cloud" [8].



Figure 5.8: Cloud Computing Conceptual Stack, taken from [9]

The advantages of Cloud computing can be classified according to four categories, namely, technical advantages, user advantages, architectural advantages and advantages for companies [9]. Examples of each category are:

• Technical advantages [9]:

- Easily handle peak load situations and prevent resource underutilization;

- Support the use of processing power to perform tasks that traditional productivity applications can not; and
- Supports the use of commodity PCs which are more cost effective and easier to support.
- User advantages [9]:
  - Any device able to access the Internet and able to run cloud-based applications; and
  - Fewer maintenance issues, no worry regarding storage capacity, and latest software version.
- Architectural advantages [9]:
  - It enables innovation by allowing developers to develop, test and innovate without having to worry about resources; and
  - Allows for the efficient use of IT hardware and software investments improving resource utilization.
- Advantages to companies [9]:
  - Provide affordable and yet very powerful resources for companies for whom proper IT access is critical; and
  - Cost savings, remote access, ease of availability and real-time collaboration capabilities.

To conclude, researchers in [182, 195] note that sometimes a source of confusion around cloud computing is the relation to Grid computing; both share similar visions, namely, reducing computing costs and increasing flexibility and reliability by using third-party operated hardware [182] (definition of the grid can be read in section 5.2.3 on Grid computing). Vaquero et al [182] seem to see grids and clouds as distinct, while Hayes [195] and Aymerich, Fenu and Survis [9] describe the cloud as the natural evolution of the grid. Vaquero et al [182] describes potential convergence of grid and cloud computing, as they postulate that grid computing would benefit from virtualisation technologies, which they feel is the key enabler of cloud computing because it facilitates features such as on demand sharing of resources and security by isolation [182].

Weiss [114] believes that we will one day be able to utilize ultra-light input devices with a screen and all processing and data storage will be handled by the Cloud. Similarly, Armburst et al [117] predict that cloud computing will continue to grow in the future. Server virtualization is now standard practice and there is a move to now also virtualize the desktop.

In South Africa, a realisation of bulk computing on the cloud will require a significant price drop in Internet access in order to make it feasible for users to make use of resources in the cloud.

#### 5.2.6 Desktop Virtualization (Virtual Desktop Infrastructure, VDI)

Desktop virtualization separates the personal computer desktop environment and applications from the underlying physical machine using the client–server model of computing [196]. A typical desktop computer consists of a PC on which a computer operating system and individual applications are executed and the graphical user interface (GUI) is displayed on the monitor directly attached to the PC [10]. By introducing virtualization, the direct connection between physical hardware, operating system, application and display is broken (see Figure 5.9) [10].



Figure 5.9: Desktop virtualization breaks traditional bindings between layers, taken from [10]

There are six different options to desktop virtualization, namely, user state virtualization, application virtualization, session virtualization, virtual desktop infrastructure, blade-based virtual desktops and singledesktop virtual machine. User state virtualization is the simplest of the options where the user's configuration settings and data are linked to the user's desktop but stored centrally on a server. This is effectively known as "Single sign-on", where the user can log on from any client on the network and have access to their profile and all their files. If something goes wrong with their current desktop machine they can make use of another one on the network and continue working as normal [10]. Traditionally, applications are installed on a particular desktop and tied to that desktop. In this virtualization model the individual applications are bundled up with the resources they need to run. The applications aren't installed directly onto the desktop from which they are being accessed, which removes potential conflicts with other applications and minimizes changes to the underlying operating system [10].

Session virtualization is the thin client architecture model, like that of the LTSP Edubuntu configuration. In this configuration multiple clients log onto a single instance of an operating system running on a single server; thus all the applications run on the same server and display the graphics to the desktop client. The client needs not do any processing at all, all the processing is handled by the central server [10]. Virtual desktop infrastructure (VDI) is an extension of the thin client idea. The difference being that instead of all the clients connecting to the same instance of the operating system (with multiple user accounts) each client connects to its own operating system (with unique accounts) running on a virtual server as a virtual machine [10].

In the blade-based virtual desktops, each blade runs a single operating system which is allocated to an individual user at runtime. Finally single-desktop virtualization is when a virtual machine is run on desktop computer (with a type 2 hypervisor). This is typically done by developers who need access to additional operating systems but do not have the hardware on which to install the operating systems. They can run a type 2 hypervisor on their current desktops and install multiple other virtual machines with the guest operating systems of those that they require in order to develop and test their applications and services [10].

Benefits of the various desktop virtualization options were tabulated in [10] and can been seen in Figure 5.10. However, this table does not include data for the key consideration of cost. The benefits of desktop virtualization are similar to the benefits of virtualization, in that the overall hardware needs and costs are reduced and the desktops are centrally stored making them easier to manage, control and backup [10]. In addition, desktop virtualization eases access to applications and data for the users, so that they can access their desktops from anywhere: home, work, Internet kiosk (depending on the virtualization configuration and the security measures implemented on the network). This allows for a higher availability of their desktops, which can help in supporting users to be more productive. There are also a number of security

Option Type	Flexibility	Control	Security	Availability	Offline Access	Performance
User state virtualization			0			
Application virtualization						
Session virtualization				•	0	
Virtual desktop infrastructure					0	
Blade-based virtual desktops					0	
Single-desktop virtual machine						
Key: • Fully applicable						

Figure 5.10: How the different desktop virtualization options provide different benefits, taken from [10]

benefits to virtualizing the desktop, such as increased control over the users' desktops to keep them to date as well as prescribing the applications that each user has access to. Some desktop virtualization variants also support encryption of both the data and communication on the network, while centrally storing the data for enhanced protection [10].

While still a relatively new technology, VDI is being tested and used in education initiatives in India [197, 198], and could be good technology choice for supporting ICT education needs in rural schools in South Africa in the future.

## 5.3 Software

In this section I look at various software technologies that we employed in providing financially sustainable ICTs in schools in the Eastern Cape. I focus on software solutions that facilitate the use of technology in education and could support the creation and use of cluster solutions in schools.

The focus is on OSS because it affords the users access to the source code which can facilitate users learning about the software in greater depth and ultimately changing software in order to better meet their local needs (or at the very least working with software engineers to localise the software). Furthermore, OSS is freely available and distributable to all, meaning that users can share software with others freely without violating any licensing agreements, and typically, OSS is financially free too, meaning no costs are incurred in obtaining and using the software. As discussed in Chapter 3, Section 3.2.4.2, these are important issues within the SLL, that the SLL would like to promote.

As said, some of the software has to support sustainable cluster solutions for the SLL computing infrastructure, in order to provide access to increased resources at a reduced cost, while remaining easy to maintain. To this end, I have investigated software applications that could be employed in developing and deploying cluster solutions within the SLL school computing infrastructure. Thus, following the descriptions of Edubuntu and LTSP, is a description of NFS, LDAP and DNS. The section concludes with a brief description of Sikuli, a software product that facilitates the simulation of user interactions in order to test the functionality of the proposed computing infrastructure environments.

#### 5.3.1 Edubuntu

Edubuntu is an official derivative of the Ubuntu operating system designed for use in classrooms. According to [199], Edubuntu has been developed in collaboration with teachers and technologists in multiple countries. It is built on top of the Ubuntu base and incorporates the LTSP thin client architecture and several education-specific applications, targeting grades R to 12. The distribution environment is thus aimed at users aged 6 to 18 and is designed for easy installation and ongoing system maintenance [199].

The stated goal of Edubuntu is to enable teachers/educators with limited ICT knowledge and skills to be able to configure a computer lab or e-learning environment in a short time and then easily be able to administer the computing facilities [199]. The design goals of Edubuntu are to allow for centralized management of configuration, users and processes, along with providing facilities for working collaboratively in a classroom environment. In addition, it aims to provide access to as much of the best available free software and digital materials for use in Education. Furthermore, it also aims to support low income schools to maximise their available older hardware for continued use as thin client computers [199, 108]. Edubuntu was chosen for use in the SLL because it is free and open source software and targeted for use in schools. Furthermore, the research team from the two universities, working in the SLL, had breadth and depth of knowledge in the Ubuntu OS. This together being packaged with the LTSP project, supporting the use of thin client computers, made it an ideal choice for use in the SLL. In the next subsection I will describe LTSP in greater detail.

#### 5.3.2 Linux Terminal Server Project

The Linux Terminal Server Project (LTSP) is a collection of free and open source software that turns any Linux server into a terminal server and allows multiple thin clients to simultaneously connect to and use the same server to run applications and access files [200, 201, 169]. All the applications run on the central server while the thin clients handle input and output. Generally, thin clients are low-powered, low-cost, lack a hard disk and are quieter and considered to be more reliable than desktop computers as they do not have any moving parts [200, 201, 202].

Around the world and particularly in developing nations, LTSP is becoming increasingly popular in schools. The motivation for use in schools is that LTSP allows schools to provide learners with access to computers without having to purchase or upgrade expensive desktop computers. As a result, access to computing infrastructure can become less costly as the thin client computers can be older computers that are no longer suitable for running as full desktop computers. This results in extending the useful life of computers and cuts costs, as even slower CPUs with 128 MB of RAM can deliver adequate performance as thin clients [200, 169, 202] (further benefits to extending PC lifespan and the use of older/refurbished computers were discussed in section 5.2.2). In addition, the use of a central server make its easier to increase and upgrade performance for less money by only having to upgrade the server and not all the end user computers. Additional benefits include that only the server needs to be maintained (as there is no software image stored on the clients) and the user profiles follow the user regardless of which thin client they connect to [202, 169]. Furthermore, in education institutions, converting existing computers to thin clients can also result in a gain in control over how students are using computing resources as all the user sessions can be monitored on the central server [200].

Several Linux distributions integrate LTSP either into their mainline operating system such as Ubuntu and Debian, or as a separate product, such as Edubuntu (Ubuntu), K12LTSP (CentOS) and Skolelinux (Debian) and KIWI-LTSP (SUSE) [200].

LTSP is different from other terminal server offerings in that the clients do not require any client-side software to boot and only require a PXE capable network card, which typically thin clients and desktop

PCs have built-in. (It is important to note that wireless network cards typically do not support PXE.) As a result no physical media storage, such as a hard-disk, CD or flash stick are necessary for a thin client to boot to LTSP [201]. This configuration significantly reduces the amount of administration required within the network as thin clients require very little software maintenance at all and only typically require hardware maintenance as components fail [201].

The LTSP server is configured such that a chroot (a Unix operating system command that changes the apparent root directory for the current running process and its children) environment is created with a minimal Linux operating system and X-server installation [201, 200]. It is this minimal environment that will be provided to the clients to run. This scaled down GNU/Linux installation is customized so that it is efficient to boot over the network. It is possible to have several different chroot environments on one server, one for each CPU architecture required (assuming the thin clients are heterogeneous and have multiple architectures) [201].

The clients on the network will boot via the PXE protocol; PXE will send out a request on the network for an IP address from a local DHCP server. The DHCP service, running on the LTSP server will respond and pass additional parameters to the thin client, specifically the PXE bootloader (PXElinux). PXE [107] is part of the "Wired for Management" specification [203], which included a specification for boot ROM technology. The client loads the PXE bootloader which then fetches the kernel and initram image from a Trivial File Transfer Protocol (TFTP) service also running on the LTSP server and loads them into the client's RAM in order to continue the booting process. Using utilities in the initram image, the client then requests a new IP address (via DHCP) and the IP address of the server which it needs to mount its root file system (the chroot previously mentioned). The client then mounts the root file system via one of two methods, either the NFS or NEB or NBD mounted root system and starts the X window system. Once done the client creates a SSH tunnel to the LTSP server's X environment and via this tunnel starts the LTSP Display Manager (LDM) login manager locally. Users can then log into the server (authentication done by the LTSP server) via the LDM running on the client and launch applications and modify data files, all of which will be run on the LTSP server but displayed and operated via the client [201, 200].

#### 5.3.3 Network File Systems

The Network File System (NFS) is a Remote Procedure Call (RPC) based protocol with a client-server relationship between the machine sharing the file system and the machine wishing to access the shared file system [204]. The NFS kernel server threads run on the server machine and respond to RPC calls from the client machines. While on the client systems, NFS-mounted files make RPC calls to the NFS server from which the files were mounted (see Figure 5.11) [204].

In Unix there is a mount operation that makes all the file systems appear as a single tree while in Windows-based systems file systems are assigned to disk letters or top-level names [12]. In NFS, fsid is a file attribute that uniquely identifies a file system on a server, while the file-id attribute identifies a file or directory within the file system and NFS clients will assume that a fsid/file-id pairing are unique for each file on a server [12]. NFS provides access to several different kinds of file system objects, namely, files, directories, symbolic links (which contain a text pathname that is an indirect reference to another file or directory) and special files (NFS provides access to a number of special files or device types; block special, character special, socket and named pipes. However these file types are just placeholders) [12].

NFS is a stateless protocol, meaning that all NFS RPC requests must completely describe the operation to be performed and most NFS requests are idempotent, meaning that clients can send the same NFS request multiple times (except in the obvious case of deleting a file or directory, as once they have been deleted attempts to "re-delete" would fail) [204]. The main reason for NFS to be stateless protocol is to minimise



Figure 5.11: Typical NFS client and server arrangement and communication, adapted from [11]

crash recovery. This means that when a client fails the server is not affected, and should the server fail, all the client needs to do is wait for the server to be rebooted; each NFS request contains enough information to be completed without any need for reference to state on either the client or the server [204].

NFS is a good choice for a Network Attached Storage (NAS) deployment in a Linux/Unix environment. It is a native Linux/Unix protocol as opposed to Samba, which uses the Server Message Block (SMB) protocol developed by Microsoft. When configured correctly NFS and the location of the files (either local or remote) will be completely transparent and irrelevant to the user; they can access the files and directories in a manner that they expect. Even when the NFS server and the NFS client are running on completely different operating systems, for example a Linux client and a Windows NT server, the interactions between them will be seamless and transparent to the user.

#### 5.3.4 LDAP

The Lightweight Directory Access Protocol (LDAP) is a simplified version of the X.500 protocol. It is a standard, extensible, asynchronous message-based, client-server protocol which provides an application protocol for reading and editing directories over an IP network; a common language that LDAP clients and servers use to communicate with one another. The latest version of LDAP is Version 3, which is specified in a series of Internet Engineering Task Force (IETF) Standard Track Requests for comments (RFCs) as detailed in RFC 4510 [205]. Typically LDAP is used for authentication but can be used in other ways: shared directory (for mail clients), address book, etc.

A directory can be thought of as an organized set of records: for example, a telephone directory is an alphabetical list of persons and organizations with an address and phone number in each "record". LDAP uses a simplified set of encoding methods, and is reduced to a core of nine operations [206, 207]. The nine protocol operations can be divided into three categories: interrogation operations, such as search and compare; update operations, such as add, delete, modify and modify DN; and authentication and control



Figure 5.12: File system illustration taken from [12]

operations, such as bind, unbind and abandon [207]. Every major desktop and server operating system platform currently available either ships with a TCP/IP implementation or can easily be equipped with one.

In addition to its role as a network protocol, the LDAP standard also defines four models that guide the use of the directory. These models promote interoperability between directory installations while supporting flexibility to configure the directory to specific organisational needs [207]. The four LDAP models are [207]:

- The LDAP information model defines the kind of data that can be placed in the directory;
- The LDAP naming model defines how data is referred to and organised;
- The LDAP functional model defines how data is accessed and updated; and
- The LDAP security model defines data protection from unauthorised access.

LDAP also defines the LDAP Data interchange format (LDIF), a common, text-based format for describing directory information [207] defined in RFC 2849 [206, 208]. LDIF can be used to describe a set of directory entries, a set of updates to be applied to the directory or it can be used to export data from one directory to another. The data in an LDIF file must obey the schema rules of the LDAP directory with which the administrator or user is working. An LDAP schema is the data definition of an LDAP directory [206]. When items are added or changed in a directory those changes/additions are checked against the schema for correctness [206]. Most of the command line utilities also read and write LDIF [207]. LDIF provides a simple to use format for administrators to administer their LDAP directories [206].

In LDAP, all information is stored in a tree structure, similar to DNS (discussed in subsection 5.3.5), but there are numerous differences [206]. With OpenLDAP you have freedom to determine the directory arborescence (the Directory Information Tree (DIT)) yourself. Before beginning, one should determine what the root of the LDAP directory will be. By default, the tree is determined by the Fully Qualified Domain Name (FQDN) of the organisations for example local.org. Thus the example root node would be dc=local,dc=org. An example of a basic tree which contains two nodes ("People" and "Groups" in an authentication example) below the root would be:

• "People" node where the users will be stored

• "Groups" node where the groups will be stored

Instead of using the CLI to update LDAP records there are also a number of GUI and web-based applications that can be used to update LDAP records. These applications make it easier for administrators to successfully add, delete and update user information without errors. Examples of such applications are: Jxplorer [209], a Java-based open source client for browsing LDAP servers and LDAP LDIF files; GOsa<sup>2</sup>[210], provides a framework for managing accounts and systems in an LDAP database; phpLDAPadmin [211], a web-based application for administrating LDAP servers, written in PHP; and Webmin [212], a web-based system configuration tool for (typically) Unix based systems with a module for configuring LDAP servers.

#### 5.3.5 Domain Name System

The Domain Name System (DNS) is a hierarchical naming system which translates domain names into the numerical identifiers (IP addresses) associated with ICT equipment in order to locate and address the devices from anywhere in the world [213]. The process of name resolution refers to searching through the domain name space in order to retrieve information/data about hosts in the domain [214].

DNS services are built on a distributed database which supports DNS queries from computers, services, or any resource connected to the Internet or a private network [213, 215]. In the early days of the Internet there were only a few hundred computers on the Internet and as such a single file, HOSTS.TXT, contained all the information about every host on the Internet [214]. The master copy of the file was maintained by Stanford Research Institute's (SRI) Network Information Centre and network administrators would email in changes that needed to be made to the file should any of the hosts' details change. Administrators could also FTP a copy of the file from SRI in order to configure routing information at their institutions. This solution did not however scale with the growth of the Internet and so they needed a successor to HOSTS.TXT [214].

The administrators of the Internet conceived a distributed, hierarchical system. The distributed design meant that each sub-domain could be delegated to the organisation that was using it [214]. This resulted in local administrators being responsible for each of their portions of the system and allowed them to make their own changes to their portion of the system as network information changed, while the hierarchical structure facilitated unique names in the system; thus DNS was born. Organisations administering a domain could further subdivide the domain and delegate use and responsibility for those to other organisations [214]. The design of DNS allows parent domains to contain pointers to the sub-domains' data in order to be able to refer queries to the right places.

The structure of a DNS database can be pictured as an inverted tree, with the root node at the top. Each node in the tree has a label which identifies it relative to its parent. The root node has a null label ("") which is represented by a single dot; the null label is reserved for the root node and cannot be used by another node in the tree. Each node in the DNS tree can be further divided into sub-nodes called sub-domains which are drawn as children of the parent (or root) domain [214]. The depth of a DNS tree is limited to 127 levels. Every domain has a unique name which identifies its position within the database. DNS requires that all sibling nodes have unique labels in order to preserve unique domain names (identifiers) for all computers within the network. The domain name is made up of the sequence of labels from the node at the root of the domain (the last node within the tree) to the root of the whole tree (at the top of the tree), with dots (or full stops) separating the labels [214]. An example would be loki.ict.ru.ac.za, where loki is the bottom of the tree (last node within the tree) and za is the top of the tree. Domain names are always read from the node (not necessarily the last node in the tree, i.e. "ru.ac.za", "ict.ru.ac.za" and "loki.ict.ru.ac.za" are all domain names) traveling up the tree towards the root at the top. Therefore a domain is a subtree of the domain name space. There are two kinds of domain names: FQDN and relative domain names. A FQDN ends in a period (full stop or dot); relating to the root domain. While, a relative domain relates to the current domain [215].

In other words, a FQDN looks like loki.ict.ru.ac.za. while a relative domain name is loki.ict (the ru.ac.za. would be assumed by the local DNS servers in the ru.ac.za. domain) [215].

The first implementation of DNS was called Jeeves, which was later followed by BIND (Berkeley Internet Name Domain). BIND is the most popular implementation of DNS today and is available in most versions of Unix and was even ported to Microsoft's Windows NT [214, 215]. The systems that store information about the domain name space are called name servers.

As mentioned earlier, resolvers are the programs that run on the client hosts that access and query the name servers. Other programs running on a host that require information from the domain name space will use the resolver to do so [214]. The resolver handles: querying a name server; interpreting responses; and returning the information to the programs that made the original requests [214]. The BIND resolver consists of a set of library routines that are linked into programs like telnet and FTP [214].

DNS provides the information needed to connect remote systems in the form of Resource Records (RRs). These resource records include: A (Address) records; PTR (Pointer) records; MX (Mail Exchange) records; SOA (Start of Authority) records; NS (Name Server) records; HINFO (Hardware Information) records; and CNAME (Canonical Name) records [215]. A records translate the host names to IP addresses. PTR records provide translations from IP addresses to host names. MX records specify the IP addresses of the email servers for the domain. SOA records define the zones within a domain. HINFO records describe the hardware and software that run on a particular system. Finally, CNAME records provide the alternative names (if specified) for a particular system [215].

#### 5.3.6 Sikuli

Sikuli (which means "God's eye" in the language of Mexico's Huichol Indians [216]) is a visual technology to automate and test GUIs using images (screen-shots) and/or key strokes. Sikuli includes Sikuli Script, a visual scripting API for Jython (an implementation of the Python programming language written in Java), and the Sikuli IDE, an integrated development environment for writing visual scripts with screen-shots, able to reproduce key strokes and click events. Jython programs allow coders to import and use any Java classes. (These are often used instead of Python modules by programmers.) Jython includes almost all the modules in the standard Python programming language distribution except some modules that were originally implemented in C [217, 218]. Sikuli is an open source research project developed at the User Interface Design Group at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) [217, 218].

The Sikuli Script automates anything you see on the screen; programmatically allowing you to control anything from a web page, a Windows/Linux/Mac OS X desktop application, or even an iPhone or Android application running in a simulator. Usually, when attempting to have a computer run a repetitive task one needs to have an understanding of a programming language (e.g. Java, C, etc). It would involve having to understand the API of the application that you wish to automate and fair amount of coding. However, Sikuli allows users to circumvent this by using what MIT call "Picture-driven computing". Instead of having a sound knowledge of the API of the application you wish to interface with you are able to use screen-shots and shortcut keys to allow the scripts to interact with applications automatically [217, 218], all that is required is some familiarity with the Python programming language [216]. One use for an application like Sikuli would be to simulate user interactions with systems, in order to test their functionality or benchmark their use.

## 5.4 Connectivity

In this section I discuss various connectivity options, specifically focusing on wireless technologies that could be employed in providing sustainable ICTs in schools in rural South Africa (wireless solutions are also applicable in urban areas naturally). I begin by discussing the South African Broadband Policy, SA Connect, and its impact on providing connectivity to South African school. This is followed by an explanation of the concept of "broadband islands" for use in rural communities. Lastly I cover, in greater detail, the wireless technologies of WiFi, WiMAX and VSAT as enablers of broadband islands in rural areas of South Africa.

#### 5.4.1 SA Connect: The SA Broadband Policy

On the 4th of December 2013, the South African Cabinet adopted "SA Connect"; the country's broadband policy and strategy.

"South Africa Connect, the national broadband policy and the associated strategy and plan, gives expression to South Africa's vision in the National Development (NDP) of 'a seamless information infrastructure by 2030 that will underpin a dynamic and connected vibrant information society and a knowledge economy that is more inclusive, equitable and prosperous'. As envisaged in the NDP, at the core of this will be 'a widespread communication system that will be universally accessible across the country at a cost and quality that meets the communication of citizens, business and the public sector and provides access to the creation and consumption of a wide range of converged applications and services required for effective economic and social participation. This ecosystem of digital networks, services, applications, content and devices, will be firmly integrated into the economic and social fabric of the country. Together, these broadband elements provide an enabling platform for economic enterprise, active citizenship and social engagement and innovation. It will connect public administration to the active citizen; promote economic growth, development and competitiveness; drive the creation of decent work; underpin nation-building and strengthen social cohesion; and support local, national and regional integration.' (NDP 2012)" [15, p2].

The purpose of SA Connect is thus to present a vision, strategy and long-term plan to catalyze broadband connectivity in South Africa [219]. The Department of Telecommunications and Postal Services (DTPS) [220] claimed that by mid 2014 the Department of Communications (DoC) had already connected 788 schools to "fast" Internet access. It is difficult to find data pertaining to the number of schools with access to the Internet as of 2016, but one would assume that the number will not be greater than the number of schools with computers, which means that if one recalls from section 5.1, that would be approximately 10% of public schools or less probably have access to the Internet in South Africa. That said, the data published in the SA Connect plan claims that the baseline of schools connected in 2013 was 25% [15]. Whichever of these figures are correct, it remains that there is a significant amount of work ahead in order to connect the numerous public schools across South Africa.

Realising the potential of providing access to networks and the network society, the South African broadband policy proposes to prioritise schools and sets the target that 50% of schools will be connected by 2016 at 10 Mbps (see Table 4.1 for all school targets) [15] (incidentally, this target was not met). However, the National Broadband Policy [15] and the Eastern Cape Broadband Master Plan [36] do not specify the IT infrastructure that should be available in schools in order to access and unlock the potential of the proposed, newly available network access.

#### 5.4.2 Broadband Islands

As described in section 3.2.3, the SLL has deployed a community-wide broadband network that connects a number of DANs, housed on school properties to one another. This allows the sharing of resources such as Internet access, as well as permits local communication and content/information sharing. We refer to this community-wide broadband network as a "broadband island" [221].

Similarly and in parallel to the SLL, Nungu [222] describes a broadband island as a "local area network that provides services in a local community" [222, p15]. A broadband island is thus a community wide area network, which spans over the entire community - typically a village (or villages) or municipal district - providing access to network services such as VoIP, email and the Internet for local community members. It can be thought of in a similar manner as a metropolitan area network (MAN) but obviously the broadband island is not within a metropolitan but rather rural communities. The network hosts services and applications within the broadband island facilitating a grassroot approach [221]. The advantages of this approach to network design are: content can be produced in the local languages; outside content can be saved locally to limit bandwidth consumption; it provides local communication channels for community members; allows the sharing of Internet access across the community members; and facilitates collaboration and cooperation [221, 31, 223, 222, 224].

A number of technologies can be used to provide broadband islands, namely, Optical Fibre, Terrestrial wireless (such as WiFi, WiMAX and cellular networks), VSAT and Delay Tolerant Networks (DTN) [221, 100]. In the context of rural South Africa access to Optical Fibre networks is prohibitively expensive and currently (2016) not a sustainable solution [225, 226]. Alternative wired solutions include dial-up, Integrated Services Digital Network (ISDN), DSL, Telkom's Diginet and Leased Lines. While dial-up access for Internet connectivity is being phased out in first world countries to be replaced by broadband technologies, such as DSL, cable modems and suchlike, it was still an important form of Internet connectivity in South Africa in 2004 [227].

There are disadvantages in using wired solutions to more rural and remote areas. The most obvious of these is that the more rural areas do not have any wired telecommunications infrastructure i.e. low teledensities [97] and copper/fiber would have to be put into the ground and exchanges built. The costs incurred in laying new copper/fiber and building such exchanges is high compared to potential revenue and so unattractive to telecommunication companies. A further disadvantage of wired Internet connections in disadvantaged areas of the country is the high probability of bad quality copper, which results in bad data lines, and copper cable theft [228, 229, 230].

The main disadvantage of dial-up is that it is narrow-band communication with a maximum download speed of 56 Kbps and maximum upload of 33.6 Kbps. Furthermore, if you make substantial use of your Internet connection via dial-up then you will be paying more than you would with an "always on" broadband Internet connection [231]. ISDN improves upon dial-up with access speeds of up to 128 Kbps full duplex (2 x 64 Kbps channels), paying the same price per call as one would on a normal phone line [232], but the rental will be higher than dial-up over POTS (Plain Old Telephone System). In addition, ISDN has a local loop limit of 6-8 km [233]. Solutions like DSL have their own disadvantages, in that they are obviously more expensive and in the case of DSL it has a maximum distance of approximately 5 km, which limits it to use in urban areas or areas very close to an exchange [233]. The limitation of 5 km for DSL is a major disadvantage in South Africa and especially for rural South African schools that may wish to take advantage of DSL [234]. As with dial-up and ISDN, copper cable theft in South Africa tends to be problematic, as well as the presence of bad quality copper cable, both of which will result in poor or interrupted data transmissions for DSL technologies [234].

Cellular networks are near ubiquitous in South Africa, however, often data services are not and when they are bandwidth can be limited [235]. The use of LTE as the technology for building a communitywide broadband island will be discussed in section 5.4.5 as it is likely to dominate over WiMAX going into the future. Nungu [222] notes that while DTN protocols can provide high bandwidth they also have very high latency and thus eliminate real-time services such as VoIP, Skype or streaming services. Thus for the purposes of the SLL we have focused on the three remaining technologies that can be sustainability employed in the context of rural South Africa, namely, WiFi (discussed further in section 5.4.3), WiMAX (discussed further in section 5.4.4) and VSAT (discussed further in section 5.4.6), in the provision of a broadband island and access to the Internet.

#### 5.4.3 IEEE 802.11 (WiFi)

The IEEE 802.11 standard is part of a family of standards for local and metropolitan area networks. This family of standards deals with the Physical (PHY) and Data Link layers (also known as the Medium Access Control (MAC)) as defined by the International Organisation for Standardisation (ISO) Open Systems Interconnection (OSI) Basic Reference Model. This standard defines the protocol and compatible interconnection of data communication equipment via the air, radio or infrared in a LAN using the carrier sense multiple access protocol with collision avoidance (CSMA/CA) medium sharing mechanism [236]. It defines the PHY and MAC layers so that they are compatible with the existing standards for higher layers (Logical Link Control and higher) [237] allowing stations to move and roam freely through a wireless LAN (WLAN) and still appear stationary to the Logical Link Control (LLC) sub-layer and above. This allows existing network protocols such as TCP/IP to operate transparently over IEEE 802.11 WLANs without any special considerations [238].

The IEEE 802.11 architecture consists of several components that interact to provide a wireless LAN. The coverage area (also known as the basic service set (BSS)) is the basic building block of an IEEE 802.11 LAN. Within the coverage area of BSS, stations (a station is any device that contains an IEEE 802.11 conforming MAC and PHY interface to the wireless medium) may communicate directly with other members of the BSS. Should a station leave the BSS, direct communication will cease. To become a member of a coverage area, a station is (dynamically) "associated" [236] with a single server, called an Access Point (AP) [237].

The basic form of BSS is as stand-alone, however it is possible to connect many BSS to each other through a distribution system. A distribution system is defined as a system used to interconnect BSSs and integrated LANs to create an extended service set (ESS). The access points are the stations that provide access to the distribution system, allowing data to move from the coverage area of the AP to the distribution system, via the AP [236]. IEEE 802.11 WLANs can be integrated with a traditional wired LAN through an access point, which is a logical point at which packets/data from wired LANs can enter an IEEE 802.11 WLAN's distribution system providing logical integration between the IEEE 802.11 architecture and the existing wired LAN.

IEEE 802.11b of 1999 extends the IEEE 802.11 standard (1997) by building on the data rate capabilities to provide 5.5 Mbps and 11 Mbps payload data rates in addition to the 1 Mbps and 2 Mbps rates [239]. The IEEE 802.11g of 2003 further extends the 802.11 standard, building on the payload data rates of 1 and 2 Mbps that use direct sequence spread spectrum (DSSS) modulation to include payload data rates that use either DSSS, complementary code keying (CCK) or optional packet binary convolutional coding (PBCC) modulations. The extended rates of 802.11g provide additional payload data rates of 6, 9, 12, 18, 24, 36, 48 and 54 Mbps [240]. The IEEE 802.11n amendment, published in 2009, improves upon the previous 802.11 standards by adding MIMO antennas. The extended rates of 802.11n provide additional payload data rates from 54 Mbps to 600 Mbps. For more in-depth detail about the IEEE 802.11 standard, and its "b", "g" and "n" extensions, see [236, 239, 240, 241].

There are several advantages to using the IEEE 802.11 standards, namely cost, standardisation and interoperability and certain wireless advantages over wired infrastructures. Standardisation and interoperability make these networks cost-effective and easy to deploy and manage. The advantages of wireless communications over wired communications are similar to those of standardisation and interoperability, namely cost and ease of deployment. While the infrastructural costs of wired solutions tend to be high and often not attractive when potential revenue may not be high [234], wireless technologies result in less financial pressures due to minimal infrastructural deployment when compared with wired counterparts [242]. The low cost, ease of use and ubiquitous nature of wireless technologies, and indeed of the IEEE 802.11 standard, make them attractive to use by schools that do not necessarily have sufficient money or expertise.

The greatest disadvantage of 802.11 as a WAN, MAN or community-wide wireless network, is that it is specifically designed to be a LAN network standard and thus not particularly suited to covering distances greater than 100 m [236]. However, the 802.11 technology can and has been stretched to cover much larger distances in the range of tens of kilometres. In such instances it involves signal boosting to stretch beyond regulation limitations. For this reason rural or peri-urban schools, which are not within the access loop of an exchange, may find 802.11 connectivity offerings attractive. WiFi is however not a good provider of long-distance back-haul and thus for deep rural schools it may not be a good solution. Furthermore, schools in urban areas may choose wired counterpart products, such as DSL or fibre optics, as they offer better reliability within metropolitan distances when compared to WiFi.

Another important disadvantage to 802.11 is interference. Interference problems within some of the 802.11 radio frequency range tend to be substantial as many other household appliances operate in the same range – 2.4 GHz is the Industrial, Scientific and Medical (ISM) band. In the 5 GHz range this is less so as there is less congestion. This can result in an unreliable signal for a school's Internet connectivity, especially in urban areas where use of the ISM band is prolific and the use of 5 GHz is increasing. Common examples of equipment which use the ISM band are microwaves, electric door openers and car alarms.

In addition, up until a few years ago wireless communication was not permitted except for the holders of a carrier of carriers license [243]. In the past this meant a rather limited number of service providers, namely just Telkom and Neotel. However, in today's more open market there are a lot more companies which hold valid licenses in order to provide access to network services via WiFi technologies.

With the need to improve on signal strength and decrease interference as well as to cover distances greater than 100 m, and maintain regulations limitations, one could either consider utilizing some of the newer WiFi technologies that specifically target outdoor use, or consider alternative wireless technologies such as WiMAX.

#### 5.4.4 IEEE 802.16 (WiMAX)

Broadband wireless access (BWA) generally refers to fixed radio systems used primarily to convey broadband services between users' premises and core networks. IEEE 802.16, otherwise known as WiMAX, is such a method of wireless access. A typical Fixed BWA (FBWA) network supports connections to many user premises within a radio coverage area [244] of 31 - 50 miles [245] (49.9 - 80.5 kilometres). It provides a pool of bandwidth, shared automatically among the users. Such networks deliver significant bandwidthon-demand to many users with a high level of spectrum efficiency [244]. Often significant frequency reuse is employed. The range of applications that these networks can support is wide and evolving all the time. They include voice, data, and entertainment services of many kinds. Traffic flow may be unidirectional, asymmetrical, or symmetrical, or change with time [244].

FBWA systems often employ Multipoint (MP) architectures. MP includes point-to-multipoint (PTMP) and mesh. PTMP is a topology where a base station (BS) services multiple geographically separated subscriber stations (SSs), and each SS is permanently associated with only one BS, while mesh is a wireless network topology in which a number of SSs within a geographic area are interconnected and can act as repeater stations (RSs) [244]. The IEEE 802.16 Working Group on Broadband Wireless Access has developed standards containing a fully specified air interface for PTMP (2 - 66 GHz) and mesh (2 - 11 GHz) systems. Similar standards have been developed within the HiperACCESS and HiperMAN working groups of the European Telecommunications Standards Institute (ETSI) Broadband Radio Access Networks Project. In addition, a number of proprietary FBWA systems exist for which the air interface is not standardised [244].

FBWA systems typically include at least one BS, a number of SSs, terminal equipment, core network equipment, inter-cell links, RSs, and possibly other equipment [244]. The BS connects to the network backbone and uses an outdoor antenna to send and receive high-speed data and voice to subscriber equipment, thereby eliminating the need for extensive and expensive wired infrastructure and providing a flexible and potentially cost-effective last-mile solution [246]. In a PTMP system, RSs are generally used to improve coverage to locations where the BS(s) have no LOS or to extend coverage of a particular BS beyond its normal transmission range [244].

The core 802.16 specification is an air interface standard for broadband wireless access systems using PTMP infrastructure designs, and operating at radio frequencies between 10 GHz and 66 GHz, addressing the line of sight environments [247]. It targets an average bandwidth performance of 70 Mbps and peak rates of up to 268 Mbps [248]. Wavelengths in this region become shorter and shorter resulting in an increase in attenuation of the signal when propagated through the air and are often affected by atmospheric conditions - rain in particular [249]. Trees and buildings are also problematic and result in an increase in signal degradation [249]. In cities where there are many tall buildings, LOS requirements may be a problem for service providers and thus limit their clientèle. For these reasons the IEEE ratified its core standard to incorporate the 802.16a which operates in the 2 - 11 GHz spectrum and thus suffers less from the above mentioned problems of attenuation and necessary LOS [249].

The 802.16a (later rolled into IEEE 802.16-2004) collection of amendments took into account the emergence of licensed and license-exempt broadband wireless networks operating between 2 GHz and 11 GHz, with support for non-line-of-sight (NLOS) architectures that could not be supported in higher frequency ranges [248]. The 802.16-2004 standard or the HiperMAN counterpart for 2 - 11 GHz is a wireless MAN technology that provides broadband wireless connectivity to Fixed, Portable and Nomadic users [247]. This OFDM and NLOS technology can be used to back-haul 802.11 hot-spots and WLANs to the Internet, and enable a wireless alternative to DSL for last mile broadband access [247]. It provides up to 50 kilometres of service area range, allows users to get broadband connectivity without needing direct line of sight with the base station, and provides total data rates of hundreds of Mbps per base station - a sufficient amount of bandwidth to simultaneously support hundreds of businesses with T1/E1- type connectivity and thousands of homes with DSL-type (512 Kbps) connectivity with a single base station [246, 247].

Support for NLOS performance was one of primary PHY differences in 802.16-2004 as well as being able to operate in lower frequencies [247]. The OFDM signaling format was selected in preference to competing formats such as CDMA due to its ability to support NLOS performance while maintaining a high level of spectral efficiency – maximising the use of available spectrum [247]. OFDM is a frequency division multiplexing modulation scheme which divides the data transmission into several streams; each one assigned to one orthogonal subcarrier [250]. In OFDM individual transmissions are distributed across the entire available spectrum in complex inter-leavings that leave relatively little spectrum unoccupied for any length of time during periods of heavy traffic [249]. OFDM supports several hundred narrow-band subcarriers, using Inverse Fast Fourier transforms (IFFT) and FFT to modulate the signal [250].

IEEE 802.16e (IEEE 802.16e-2005), better known as Mobile WiMAX, extends the IEEE 802.16 standard to include broadband connectivity to mobile users without the need for LOS [251]. The IEEE 802.16e standard is optimized for dynamic mobile radio channels and supports hand-offs and roaming at vehicular speeds of up to 75 mph (120.7 kph) [252]. The key technologies in 802.16e at the PHY layer are OFDMA and Scalable OFDMA (SOFDMA) - an enhanced version of OFDMA. OFDMA uses a multicarrier modulation scheme where carriers are divided amongst the users to form subchannels [252]; its OFDM modulation for multiple users [250]. The coding and modulation are then adapted separately for each subchannel allowing channel optimization on a smaller scale, thus optimizing the use of spectrum resources. Users are then allocated to any of the subcarriers in the used frequency band [250]. SOFDMA adds scalability support to OFDMA; scaling the FFT size with the bandwidth in order to keep the subcarrier spacing constant regardless of the adapting bandwidth size [250]. Both, OFDMA and SOFDMA allow for a more efficient allocation of spectrum and reduced interference [252]. Another multiplexing scheme employed in WiMAX networks is that of MIMO [250]. The basic principal behind MIMO is spacial multiplexing; the use of multiple antennas at the sending and receiving end of the communications channel [250]. When using MIMO, the data stream is divided into smaller data streams that are then sent and received over multiple antennas. MIMO takes advantage of multi-path propagation of radio signals increasing peak data rates [250]. WiMAX also employs a combination of modulation schemes (together with the multiplexing schemes of OFDM and MIMO), such as QPSK (), BPSK (Binary PSK), and QAM (Quadrature Amplitude Modulation). The modulation schemes allow the signal to adapt to different demands, such as distance or throughput [250]. In QPSK the carrier frequency phase is shifted in order to send and receive data; this modulation scheme only recognises phase-shifting and so is not susceptible to amplitude changes caused by interference [250]. QAM, a higher order modulation scheme, combines two techniques, namely phase-shift keying and amplitude-shift keying; QAM changes the amplitude of two carrier signals which are 90° out

of phase with each other [250]. The use of higher order modulation schemes supports the transmission of more data – higher transfer rates – but is prone to higher error rates and requires a higher signal to noise ratio (SNR) [250].

802.16e also supports hard and soft hand-offs as users move from one cell to another [252]. In order to make WiMAX meet the requirements of 4G networks, the IEEE released 802.16m (also known as WiMAX 2.0) [253]. 802.16m was released in 2011 and amends the WiMAX standard in order to cater for 100 Mbps while mobile and 1 Gbps data rates for fixed installations [254]. IEEE 802.16 standards integrate seamlessly into most wired networks, much like 802.11 standards. For further detailed information on the IEEE 802.16 standards and specifications see [244], [255], [256], [252], [257], [245] and [249] amongst others.

Similar advantages that were noted for the IEEE 802.11 standards can be noted here for the IEEE 802.16 standards. Additional advantages are that WiMAX is designed to be a MAN/WAN technology and as such will be better at creating WAN networks when compared with WiFi. While according to [258], the main advantage of FBWA technologies over wired systems results mainly from the high costs of the labour intensive deployment of cables. "A 200-square-kilometre service area costs a DSL provider over \$11 million. The same area can be served wirelessly for about \$450,000 [258]."

While the IEEE 802.16 standard for wireless communication better addresses the issues found in the IEEE 802.11 standard, namely, interference and distance, these improvements are still going to be limited to some degree. As previously mentioned, transmissions become increasingly subject to atmospheric conditions and in particular rain, while other structures/obstructions found in urban areas, such as trees and buildings, can attenuate the signal. Furthermore, at the higher frequencies data can only travel over very short distances, and line of sight is necessary in order for communication to take place. In the highly populated Radio Frequency (RF) spectrum, such as those of the unlicensed spectrum (i.e. 2.4 GHz ISM band and others) interference is still going to be an issue for network providers. In this case operators will need to seriously consider purchasing licensed spectrum, which is often expensive [249].

Another disadvantage to 802.16 is that the data rates are not nearly as high as some of its wiredline counterparts, such as optical fibre [249]. In addition, few WiMAX products ever entered the market and as such are still rather expensive when compared with WiFi technologies. Furthermore, there are currently few telecommunication companies operating WiMAX technology on a commercial basis in South Africa, and given that it would seem that WiMAX has "lost" its battle to LTE (many networks in the USA which employed WiMAX have shut them down [105, 106]), there seems very little chance that WiMAX technologies will be deployed on a large scale anywhere in the world. Rather more likely, will be large scale deployments of LTE and, more relevant in this context, fixed LTE networks.

#### 5.4.5 LTE

LTE (Long Term Evolution) is the brand name of 3GPP's (3rd Generation Project Partnership) near 4th generation technology development efforts in both Europe and North America [259]. I say near 4G because LTE in its original release did not meet the ITUs specifications of providing 1 Gbps transfer rates and so despite being referred to as a 4G technology by the media, its is considered to be a 3.9G technology [253, 260]. The evolution of wireless/mobile telephone technologies has been grouped into (currently, 2017) four generations, which are based on the "maturity" of the mobile technology. There are also subdivisions within each generation (e.g. 2.5 and 3.5). The classification into the various generations represents a commonly agreed upon perspective from both industry and academia [259].

The first generation (1G) of mobile telephone technologies was an analogue wireless access system primarily used for voice traffic; the "advanced mobile phone system" (AMPS) in North America and the "total access communication system" (TACS) in Europe are the most well known examples of 1G mobile technologies. The analogue nature of the 1G system was susceptible to static noise (interference) and provided no security against eavesdropping on the shared media (air). That said, AMPS pioneered the use of small hexagon-shaped service areas, referred to as "cells", in order to support frequency use without interference; a method of deployment still used in modern mobile phone systems today [259].

As a result of the noise susceptibility and the lack of security, the 1G technology was replaced by the second generation (2G) technologies; this represented a move from analogue to digital technologies. The digital technologies were superior to their analogue predecessors as the digital radio waves could be subjected to superior processing techniques that made them less susceptible to noise. Furthermore, the digital technologies are based on discreet bi-level signals (as opposed to the continuous analogue signals) making them easier to calibrate and maintain and as such cheaper than their analogue predecessors [259]. The two best known 2G technologies are TDMA (Time Division Multiple Access) and CDMA (Code Division Multiple Access); TDMA was adopted in Europe and called the Global system for mobile communications (or GSM), while CDMA in North America and referred to as CDMAOne [259]. While GSM is the more popular of the two technologies, CDMA has the advantage that it is able to support more users than GSM because it utilizes the available spectrum better. 2G technologies were later updated by the interim 2.5G, which were now included packet switched in addition to the circuit switched domain of 1G and 2G networks. GPRS (General Packet Radio Service) is an example of the 2.5G network technologies adopted in the GSM networks [259, 253]. Following from 2.5G was 2.75G, which included EDGE (Enhanced Data Rates for GSM Evolution) and CDMA2000 1xRTT [259, 253]. Arguably these could be considered 3G technologies as they both resulted in data rates of 144 kbps or more (EDGE data rates are 236.8 kbps and CDMA2000 1xRTT are 144 kbps) [259].

The development of 2.5G and 2.75G technologies led to the next level in the mobile network technologies, referred to as third generation (3G) mobile technologies. The ITU (International Telecommunication Union) specified that the minimum data rate for 3G was 144 Kbps in order for a technology to qualify as 3G. In actual fact, 3G technologies typically far exceed this minimum level, providing data rates of between 5 and 10 Mbps [259]. 3G technologies employ better methods to obtain spectral efficiency; in other words, they are able to send more bits per hertz, allowing for higher data rates and improved services [259].

3G technologies in the GSM network are UMTS (Universal Mobile Telecommunication System) based, while the CDMA networks advanced to 3G status through the CDMA2000 family of protocols, specifically, EV-DO (Evolution-Data Optimised). UMTS employed the use of W-CDMA (Wideband CDMA) as the air interface, while EV-DO uses CDMA and TDMA multiplexing techniques to increase the per user and system throughput [259].

Following 3G were 3.5G technologies which included HSDPA (High Speed Downlink Packet Access) and HSUPA (High Speed Uplink Packet Access) in the GSM based networks [259, 253]. HSDPA allowed for data throughput rates of up to 28 Mbps and HSUPA provided up-link speeds of up to 5.76 Mbps, and were the stepping stones for GSM network's transition towards 4G technologies [259, 261, 253]. CDMA based networks attempted to create their own counterpart to LTE, called UMB (Ultra-Mobile Broadband) [259], however, in 2008 the lead developer of UMB technologies, Qualcomm, announced that it would be discontinuing development in favour of LTE. As such, both streams of mobile technologies will converge at the 4G level and employ LTE [262].

The technical specifications of LTE, as approved by the 3rd Generation Partnership Projects (3GPP) – a collaborative body including a number of telecommunications standard bodies that co-operates to produce a globally accepted set of standards to govern 3G and higher mobile phone technologies – include the use of OFDMA in the downlink and Single Carrier - Frequency Division Multiple Access (SC-FDMA) for the up-link (also known as DFT-Spread-OFDM (DFT-S-OFDM)) as well as MIMO technologies and the resulting hardware should provide data rates of up to 300 Mbps downlink and 75 Mbps up-link in 20 MHz bandwidth portions [259, 13, 261]. SC-FDMA was chosen for the up-link because it is more power efficient than OFDMA which is important given that handsets are powered by batteries [263, 250]. Furthermore, the specifications indicate that the round-trip time (RTT) between the base station and a handset may not be more than 10 milliseconds. The specifications for LTE also mandate that an LTE network is an All IP network [259]. The key features of an LTE network are: firstly, its an all IP network, meaning that the core of the network has changed to a packet only all IP based architecture; secondly, the use of radio technologies that facilitate higher throughput and lower latencies; thirdly, radio access technologies are interoperable across conforming manufacturers; and lastly, the support of vertical handover between multiple radio access technologies [259].



Figure 5.13: EPS architecture, taken from [13]

The system architecture of LTE consists of new radio network and air interface architecture of LTE together with the Service Architecture Evolution (SAE) which focused on the design of the core network

architecture. Together these two components are known as the Evolved Packet System (EPS) program [13]. The EPS architecture can be seen in Figure 5.13, taken from [13]. The EPS consists of the access network (referred to as the E-UTRAN) and the core network (EPC). In turn, the E-UTRAN network consists of the enhanced NodeB (eNodeB, also known as the base station) and the user equipment (UE), such as mobile handsets [13]. There can be many eNodeBs (connected to one another) and many UEs in the E-UTRAN. The E-UTRAN is connected to the EPC via a Serving Gateway (S-GW) and the Mobility Management Entity (MME). The MME is the control plane entity and controls the high level operations of the mobile unit using signaling messages and interaction with the home subscriber server (HSS – a central database which contains the information about subscribers). The S-GW is the user data plane entity and is responsible for forwarding IP packets from the UE to the Internet via the Packet Data Network Gateway (P-GW) [13]. The distribution of functionality within an LTE network is illustrated in Figure 5.14, taken from [13].



Figure 5.14: Distribution of the functionality within an LTE network, taken from [13]

The successor to LTE is LTE-Advanced, which completely meets the standards to qualify for 4G as it provides data rates which can peak at 1 Gbps [253]. Furthermore, LTE-Advanced supports faster switching between power states and higher bandwidth, as well as scalable system bandwidth up to 100 MHz and *clustered* Discrete Fourier Transform-Spread-OFDM (DFT-S-OFDM) for the up-link [253, 264].

Currently in South Africa there are not many fixed LTE/LTE-Advanced deployments or service providers. Furthermore, it is only possible to have fixed LTE/LTE-Advanced in those areas where LTE/LTE-Advanced is available from the cellular providers, which is currently (2017) typically limited to urban areas. For wire-less Internet access in more remote or rural areas it is more likely that VSAT services are available to the end user (or school) in South Africa.

#### 5.4.6 Satellite Communications: VSAT

Satellite communications offer attractive solutions due to their ability to connect widely dispersed geographical sites. Satellites offer the ability to service many users and solve the expensive 'last-mile' issue without having to dedicate to each user cable, optical fibre, switching equipment ports, and other expensive equipment. These factors makes satellite use attractive for broadband communication [265].

Satellite communication systems can be classified into three categories according to their orbital altitudes [266]:

- · Geostationary Earth Orbiting (GEO) satellites, with approximate altitudes of 35000 km
- Medium Earth Orbiting (MEO) satellites, with approximate altitudes of 10000 km
- Low Earth Orbiting (LEO) satellites, with approximate altitudes of <1000 km

LEOs can be further sub-divided into Big LEO and Little LEO categories. The Big LEO satellites are to offer voice, fax, telex, paging and data capabilities, while the Little LEOs are to offer data capabilities only, via a "bent-pipe" method or store-and-forward service [266]. The "bent-pipe" method is a satellite technology that transmits data from one point on the Earth to a satellite and then from the satellite back down to another point on the Earth's surface. Pipe payloads simply receive and retransmit signals from the same beam's coverage area [267]. The name "bent-pipe" comes from the shape of the path that the data travels to and from the satellite. As the satellite orbits get closer to the Earth, so its footprint over the Earth decreases. For this reason, lower orbiting satellites will require larger constellations than higher orbiting satellites will, however, require less energy than the higher orbiting satellites because of the shorter average distance between transmitter and satellite [266].

Until the early 1970s satellites were mainly used for international telephone trunking and TV signal transmissions and were in geostationary orbits [265]. These GEO satellites were entirely analogue and used Frequency Division Multiplexing (FDM) [265]; they acted as simple "bent-pipe" repeaters in the sky and were very limited in power and capacity [265]. The limitation of single beam satellites was low data rates. The need for higher data rates thus led to the development of multi-beam satellites and methods of frequency re-use [265].

In order to avoid interference, when there are multiple users, there is a need to multiplex information so that when it is sent down from the satellite it can be properly separated by the users. For these reasons the choices of multiple access and multiplexing methods are important and will impact on the performance of the satellite network [265].

Two-way interactive data networks involve much more than the combination of two one-way channels. The two directions of communication flow in the channels of a two-way VSAT network with a client/server configuration and involve two fundamentally different forms of communication [268]. In the direction from the satellite to the terminals, the communication channel is one to many, or broadcast. In the direction from the terminals to the satellite, the communication channel is many to one, or multiple access [268]. Transmitting data from a single satellite station to a large number of remote terminals (the broadcast channel) is almost always configured in a simple time-division multiplexed (TDM) mode [268]. Transmitting data from large numbers of remote terminals to a single satellite (the multiple access channel) is a more challenging problem [268].

In frequency division multiple access (FDMA) each user is assigned a different frequency upon which to transmit. With FDMA many users can simultaneously share the satellite. FDMA has been used for decades in "bent-pipe" satellites for the up- and down-links of analogue signals, such as telephone conversations. In time TDMA, users are assigned positions in a quickly repeating schedule for transmitting on a common frequency to the satellite [265].

FDMA is attractive for Earth stations since an amplifier in an FDMA system operates continuously, while an Earth station transmitting with TDMA requires a higher burst power and a correspondingly more expensive amplifier [265]. A drawback of the FDMA up-links is inflexibility and often the fraction of system channels actually in use is poor, since not every Earth station may have something to transmit at all times. In a TDMA system the time slot plan can be changed dynamically to accommodate such varying demands [265].

If an FDM down-link is used then several carriers must be amplified simultaneously for transmission. Often such amplifiers, when operating at their maximum, will distort the transmitted signal and waste power [265]. Thus for a FDM down-link a large amplifier must be operated at less than half the power. If TDM is used then there is only one signal that needs to be amplified, which results in no inter-modulation products. A satellite producing a TDM down-link transmits essentially continuously. Hence, FDMA is economically preferable for up-links to reduce Earth station cost, while TDM is preferred for down-links to reduce the satellite costs [265].

Some satellite implementations have also been known to implement other multiple access techniques in various ways, see [265], [266], [269], [270] and [271].

In South Africa today there are an increasing number of satellite solutions available [272]. From Telkom [273], one may purchase one of their SpaceStream [274] products, and from Sentech [275] one may purchase one of their VStar products [276], while First Consulting are reselling StratoSAT services and Vox Telecom are selling Yahsat/YahClick [277], to name just a few (there are almost 50 Internet Service Providers (ISPs) in South Africa that offer VSAT services to customers for varying prices and services [278]). All of these services make use of GEO satellites in orbit. Currently MEO services are limited to GPS, while there are no LEO services within South Africa due to the high expense of running such a network [266].

Iridium, based in the United States of America, with a constellation of 66 LEO satellites, is able to deliver communication services to and from remote areas. Iridium services cater for industries such as maritime, aviation, government/military, emergency/humanitarian services, and many others [279].

The advantages of satellite communications are much the same as the advantages of other broadband wireless communications that have been discussed. For example, they are able to connect users to other users who are geographically dispersed without the use of wired infrastructure or by integrating into wired infrastructure [265]. They are also useful in connecting users to the Internet in areas where there are no wired alternatives. Additional advantages of satellite communications for broadband inter-connectivity include [265]:

- The broadcast nature of satellites supports the transmission of the same message to a large number of stations, making satellites a good choice for PTMP transmissions; and
- New users can be accommodated by simply installing new Earth stations at customer premises, making such networks scalable.

A major disadvantage to satellite communications in the South African context is the cost of the services. Typically, VSAT offerings in South Africa are more expensive than other wired alternatives [280] and thus are less likely to be utilised by those customers or schools in the rural areas who have financial constraints and are classified as "previously disadvantaged".

Further disadvantages of satellite communications are due to the RF spectrum shortcomings: the Kaband, which most satellite communications use, experiences significant rain attenuation. This attenuation is due to molecular water vapour absorption resonance frequency, which is located in the Ka-band, 22.3 GHz [265]. The delay in data round-trip time for GEO satellites is very long, often affecting quality of service in real time services. While lower orbiting satellites such as MEOs and LEOs would not have this problem, they are much more expensive to deploy and several companies have gone bankrupt in attempting to do so [265, 266].

# 5.5 Conclusion

In South Africa there remains a dearth of access to ICTs in schools and this presents an important obstacle for teaching and learning [25]. Power [127] discussed the need for new paradigms of educational technology that meet the needs and experiences of rural communities. These are that schools firstly need access to ICT facilities and that these facilities need to be sustainable and relevant to teaching and learning. The use of thin client computers together with grid, cluster or cloud server architectures could provide access to sustainable ICTs for use in rural South African schools. In addition, the use of wireless technologies, especially in the "broadband island" topology, which we would strongly advocate in the national broadband roll-out, could provide sustainable access to the Internet and on-line services that foster and afford learning opportunities for teachers and learners in rural South African schools. In the next chapters of this thesis I propose a number of alternative computing infrastructure configurations for rural South African schools. These alternative configurations are based on the work done in the initial phases of the SLL (Chapter 3), the literature reviews of ICTs in Education (Chapter 4) and Computing Infrastructure technologies (Chapter 5) together with questionnaire and interview data provided by a group of teachers working in six schools in the SLL (Chapter 6). The alternative configuration experimentation is then used to inform deployments within two schools in the SLL network (and a third school where I deployed the standard thin client configuration of the current SLL model for comparison), from which lessons learned are used to revise the computing infrastructure model of the SLL.

# **Chapter 6**

# Findings from the Siyakhula Living Lab survey

# 6.1 Introduction

This chapter details the findings from the SLL schools' survey. It describes results from the questionnaire conducted with participants and the follow up interviews and focus groups done at the six schools. Further, it does some analysis of those results in light of the theories discussed in Chapter 2, to better understand the findings from the survey. This analysis and discussion are then used to inform the design and eventual deployment of the proof of concept computing infrastructure within three of the six schools (which are discussed in the subsequent chapters).

# 6.2 Questionnaire results

Questionnaires were provided to each of the participants in order to gain an understanding about their teaching experience as well as some demographic data (for example, how long they had been teaching, which subject areas they taught, their level of ICT training, age, etc) as well as to highlight their perceptions and understanding about the use of ICTs for teaching and what ICTs they had available at their school. The questionnaire was divided into three sections, Section A focused on demographics and other personal information (but not including their names or any other completely unique identifiers in order to support anonymity). Section B focused on the ICT facilities available at their schools, and Section C asked about the level of ICT utilization at their school. In addition, I performed a site survey at each school and so had an independent audit of the ICT facilities at each school. This would help to highlight any misunderstandings regarding available ICTs at each school. This section begins with some of the demographic information regarding the teacher participants, followed by information provided regarding the participants' school ICT facilities and finally presents the data regarding ICT use at their schools, as perceived by the participants.

#### 6.2.1 Demographics

Each of the participants were asked questions about their demographics: age, gender, number of years teaching, qualifications, subject areas and grades, ICT qualifications, etc. The results of some of the questions from section A of the questionnaire are presented here. Not all the results of the questions are presented here, but rather just those questions and corresponding data that were of most relevance to the discussion here (for example, I don't reflect on the subject areas that the teachers taught, or the grade levels at which they taught in this dissertation).

#### 6.2.1.1 Age, Gender and number of years teaching

Teachers were offered a number of age ranges into which they might fall. Of the 12 participants the mode in terms of age was between 30 and 34 years of age. Participants were however spread out across all the age groups provided, except for 20-24 years and 55-59 years where there were no participants. Figure 6.1 shows the number of teachers in each of the age groups. From Figure 6.1 it can be seen that half of the teachers interviewed were under the age of 40 years old and the other half were over the age of 40 years old.



Figure 6.1: Number of Teachers per age group

In terms of gender, of the 12 participants 9 were female and the remaining 3 were male. When asked about the number of years they had been teaching, the largest group of participants, 4 teachers (the mode), had been teaching for less than 5 years and the next biggest group (3 teachers) for 5-10 years, meaning that the majority of participants had been teaching for less than 10 years. The results for the number of years teaching can be seen in Figure 6.2.

#### 6.2.1.2 Qualifications

In terms of qualifications, three questions were asked: "What is your highest qualification"; "What computer training have you had"; and "Have you ever received any training that focused on the integration of ICTs in teaching and learning"? In terms of their highest qualifications, 10 of the 12 participants had diplomas (9 of which were in education and one in engineering). The remaining two had degrees, one Bachelor of Education, the other a Bachelor of Science. All 12 participants had received some form of computer training; typically computer literacy short courses either formally through universities or by attending courses provided by the Siyakhula Living Lab or the Department of Basic Education. Furthermore, 10 of the 12 teachers answered that they had received some form of training that focused on the integration of ICTs into teaching and learning. Of those 10 teachers, five of them had done an Advanced Certificate in Education (ACE) specializing in ICTs. Of the remaining five, two had received computer literacy related training, one had received training on the use of DVDs and data projectors, one was trained in Master Maths and the last was trained in using the South African School Administration and Management System (SASAMS).

#### 6.2.2 ICT facilities

Section B of the questionnaire focused on the ICT facilities available at each of the schools. Within section B I asked teachers the following questions:



Figure 6.2: Number of years teaching by participants

- What kinds of computers does your school have?
- Do you know the type of computer configuration that your school has?
- How many computers does your school have?
- Where are the computers in your school located?
- What operating systems are used in your school?
- What other software does your school have access to?
- Does the school have an internal computer network?
- Does the school have an Internet connection?
- What type of Internet connection does your school have?
- Does your school have a computer technician who fixes school computers?
- Does your school have an IT teacher?
- How much money is available for computer maintenance?
- How much money us available for purchasing new ICT equipment?

In addition, I asked questions relating to who teachers approached for IT related support at school and if there was no specific IT teacher at school, did the school appoint another of the teachers to be in charge of the IT facilities? The majority of these questions provided optional answers in order to help prompt teachers with possible answers to questions. The purpose of this section was to learn about what facilities were available at each school but also so gain insight into how much teachers knew about the facilities at the schools where they taught, as I also performed a site survey in order to ascertain what was available at each school independently of what the teachers answered. This allowed me to identify what facilities were available in cases where teachers either didn't know or were unsure, or where teachers were incorrect about what they had at their school.

#### 6.2.2.1 Types and configuration of computers available

When asked about what kinds of computers were present at the schools (i.e. if they had Intel Pentium, Dual Core, Atoms, etc), 50% of the teachers were unable to answer the question. Of the remaining 50% who did answer the question, all of them first asked me what their school had and then filled in the answer that I provided to them. In other terms, none of the teachers were aware of the types of computer hardware that they had available at their schools. The same applied when answering the question pertaining to the computer configuration at their schools (i.e. thick clients, thin clients, hybrid, etc). Nine teachers (75%) were unable to answer the question, while the remaining three (25%) teachers first asked me what the computer configuration was before answering the questionnaire. This again indicates that none of the teachers knew how the computers were configured at their school.

#### 6.2.2.2 Number of available computers and location within the school

Regarding the number of computers, three teachers did not answer, three noted that they had less than five computers, two teachers said they had between 5 and 10 computers, another three between 10 and 25 computers and one teacher noted that they had between 25 and 50 computers. All teachers were able to comment on where the school computers were located: four of the six schools had computers in a computer lab, one school located their computers in individual classrooms, three of the schools had computers in the school or administrator offices and one school kept computers in the staff room.

#### 6.2.2.3 Operating systems and available software

When asked about the operating systems used at their schools, one teacher did not answer; two said that they were unsure (one of which hazarded a guess which was correct), the remaining teachers answered the question. The results of the question revealed that four schools made use of Edubuntu, three schools Windows 7 and both Ubuntu and Windows XP were used by one school (the school has dual boot computers).

Teachers were then asked to choose from a list of software those which they had available at school and also note which of the available software they were making use of. Only one teacher did not answer either question, Teacher\_A. This was unsurprising as the computer facilities at FH1 where Teacher\_A was teaching had never been in working order and so she was unable to make use of the computer lab at the time that the questionnaire was conducted. From Figure 6.3 it can be seen that typically teachers made use of the software that was available to them at their schools. There were a few instances where despite the availability of the software a number of teachers were not making use of the software, such as, Microsoft PowerPoint, Microsoft Access, Microsoft Outlook, and SASAMs. That said, the most utilized software was typically productivity suite applications and web browsers.

#### 6.2.2.4 Computer networking and Internet connections

When asked whether their schools had an internal network (LAN) two teachers didn't answer, five teachers said no and five teachers said yes. All schools had a LAN. Furthermore, when asked about an Internet connection, the same two teachers who didn't answer the LAN related question also didn't answer the



Figure 6.3: Software available to and used by teachers interviewed in the SLL

question about Internet access. The remaining 10 all noted that they did have access to the Internet (which is correct). Unfortunately the majority of teachers were not able to accurately say what type of Internet access they had at their school. Three of the teachers didn't answer the question, two answered that they didn't know. Of the remaining five, four chose WIMAX and one choose VSAT. Furthermore, of those who ventured an answer, four asked the researcher first for the answer to the question and then filled in what was provided for them.

#### 6.2.2.5 IT maintenance and support

When asked about a technician for fixing computers, three teachers said yes and three didn't answer. The remaining 6 answered no. While, in terms of an official IT teacher, only FH6, has an IT teacher, the other five schools have no official IT teacher. However, of the five schools without an IT teacher, four of them have assigned the role of managing the IT facilities to one of their teaching staff. From Figure 6.4 it can be seen that biggest subgrouping of teachers (four) noted that they relied on the SLL for support with the ICT facilities. While equal numbers (two each) either had no support, had a school IT technician or went to their principal or HOD for support. One teacher noted that another member of staff was typically asked to support ICT needs, and one teacher didn't answer the question.

#### 6.2.2.6 School finances for ICTs

When asked about the schools' budgets for maintaining computer equipment or purchasing new equipment, five teachers noted that they were unsure of how much money their school was able to allocate to new equipment while six didn't know the amount of funds available for maintenance. In both questions, one teacher did not provide any answer. In terms of funding for maintenance the remaining teachers were split



Figure 6.4: Provision of ICT support at the schools

across the categories of "No money for maintenance" with two respondents, "between R5000 and R10000" for maintenance with two respondents and "less than R5000" with one respondent. In terms of purchasing new computers, the remaining teachers were split across "between R10000 and R25000" for new equipment with two respondents, "between R5000 and R10000" for new equipment with three respondents and lastly "less than R5000" for new equipment with one respondent.

#### 6.2.3 ICT use

The third section of the questionnaire focused on how teachers were making use of the ICT facilities at their schools (assuming that they were using the facilities), the level of availability of the facilities and whether or not the available facilities are supporting their needs. In this section I asked questions about levels of ICT use, levels of reliability of the computers, levels of access to the facilities and ease of use.

#### 6.2.3.1 Levels of ICT use

When asked if the teachers made use of the ICT facilities at their school, one teacher said no and another did not answer. The remaining 10 said that they did make use of the facilities. Furthermore, I asked those who were using the computers to elaborate on the nature of their use of the computers; Figure 6.5 reflects their answers. From Figure 6.5 it can be seen that the majority of the teachers who are using computers (80%) are using them for "learning purposes". This included activities such as asking learners to type out essays or search for information on the Internet. The next two highest categories, each with 60%, are for the use of computers for classroom and school administration, these include capturing marks, making worksheets and school management activities. Only five teachers (50%) who were using computers were using them for teaching activities, such as preparing worksheets, tests and presentations.

In addition, I asked about the numbers of teachers in their schools that were making use of the school computing facilities for teaching purposes. The answers are reflected in Table 6.1. In Table 6.1 it can be seen that teachers from two of the schools did not answer, FH2 and FH5, while FH6 and FH1 (both the secondary schools) had the highest number of teaching staff making use of the ICT facilities. FH3 had the



Figure 6.5: How teachers are making use of their ICT facilities at school

lowest number reported, and FH4 had a mid-range figure. Unfortunately I did not ask how many teachers each school employed in order to ascertain the percentage of teachers using the facilities.

School	Number of teachers using ICTs for teaching purposes		
FH1	6		
FH2	unanswered		
FH3	3		
FH4	4		
FH5	unanswered		
FH6	7		

Table 6.1: Number of teachers using computers for teaching purposes per school

#### 6.2.3.2 ICT reliability and usefulness

Teachers were asked two questions, firstly about the reliability of the school ICT facilities and secondly about the usefulness of the facilities (i.e. were the facilities meeting their teaching and learning needs). In terms of ICT facility reliability, only two teachers, both at the same school, FH6, said that their school ICT facilities were always in working order. The majority of teachers, eight across five schools said that their facilities worked "sometimes". One teacher, Teacher\_F, said that their ICT facilities never worked and Teacher\_A didn't answer as their school computer lab was not yet running at the time of the questionnaires.

When asked about the facilities meeting their teaching and learning needs, the results were very similar. The same two teachers from FH6 said that it was always the case that the school facilities met their teaching and learning needs. Seven teachers, across five schools said that the facilities "sometimes" met their needs for teaching and learning, while one teacher, Teacher\_E said that they rarely met their needs. Two teachers, Teacher\_A and Teacher\_F left the question unanswered.

#### 6.2.3.3 Learners and teachers levels of access to ICT facilities

With regards to the learners' levels of access to the ICT facilities, firstly, I asked if the learners had access to the computers at school and if so at which times. Of the 12 teachers, two said that the learners did not have access (both from FH3) and two didn't provide an answer (both from FH1, where the ICT facilities were not operational). The remaining eight said that the learners did have access to the ICT facilities. Their

access was broken down as described in Figure 6.6. From Figure 6.6 it can be seen that 50% of the teachers (two schools, FH5 and FH6) who said learners did have access to ICT facilities were accessing the ICTs during scheduled class times. Teachers from FH2 noted that their learners had access to the facilities after school hours in the afternoons, while the teachers from FH4 said learners had access during specific IT related training provided. Finally, one teacher, from FH6 also noted that some learners had access as a result of taking the CAT subject area.



Figure 6.6: Breakdown of when learners have access to ICT facilities at school

Teachers were also asked to rank how often learners were able to make use of the ICT facilities available at school, where 4 was "very little access", 3 was "little access", 2 was "some access" and 1 was "often". Three teachers didn't provide an answer (one from FH1 and both from FH3), three teachers said that the learners "often" had access (one from FH1 and two from FH6). The remaining six teachers (from FH2, FH4 and FH5) said that the learners had "some access". In addition, I asked the teachers to rank the ease with which learners were able to gain access to the facilities, where 4 was "very difficult", 3 was "difficult", 2 was "easy" and 1 was "very easy". Again, three teachers did not provide answers (one from FH1 and both teachers from FH3). A further three teachers (one from FH6 and two from FH4) said that it was "very easy" for learners to gain access to the facilities, while one teacher (from FH1) said it was "difficult" for learners to gain access. The remaining five teachers (from FH2, FH5 and FH6) said that it was "easy" for learners to gain access.

Finally, I asked the teachers to rank the ease with which they were able to gain access to the ICT facilities for the purposes of teaching and learning, where 4 was "very difficult", 3 was "difficult", 2 was "easy" and 1 was "very easy". Two teachers (Teacher\_A and Teacher\_F) did not answer the question, while three teachers (one from FH6 and both from FH2) said that it was "very easy" for teachers to gain access. The second teacher from FH6 described the access as "easy". The remaining six teachers (50%), from FH1, FH3, FH4 and FH5, all said that it was "difficult" for teachers to gain access to the facilities for the purposes of teaching and learning.

#### 6.2.4 ICT maintenance, management and utilization

The fourth and final section of the questionnaire focused predominantly on ICT maintenance and management, as well as the level of utilization of the facilities. This section consisted of five questions, three of which were focused on maintenance and management and two on utilization. In terms of maintenance I firstly asked how teachers would rate the maintenance of ICT facilities; Difficult, Manageable or Easy? The majority of the teachers (7 respondents) felt that it was manageable, three did not answer and one each chose Easy and Difficult (see Figure 6.7). Secondly, I asked if there was anyone at their school who was fulfilling a support role for the teachers, helping them make use of the ICT facilities. Of the teachers, eight said yes, two said no (both from FH5) and two did not provide answers.



Figure 6.7: Ease of ICT maintenance according to the teacher participants

In terms of management, I asked the teachers if their school had any policies in place that governed the use, maintenance and procurement of IT related equipment. One teacher did not answer the question. Of the remainder, seven said that there were policies in place and four said that there were no policies in place.

Finally, with regards to ICT utilization, I asked the teachers about the levels of utilization for both teachers and learners at the schools, their answers are reflected in Figure 6.8. From the graph in Figure 6.8 it can be seen that the majority of teachers felt that both the teachers and the learners were making moderate use of the facilities available at their schools. Interestingly more teachers felt that there were more teachers rarely using the available technology than learners rarely using the technology; the unspoken consensus being that typically learners were making better use of the facilities than the teachers were.

#### 6.2.5 Summary

When considering the demographics of the teachers who participated, they are evenly spread in terms of age with 50% being under the age of 40 and 50% are over the age of 40 years. Interestingly though, slightly more than 50% (58% to be specific) have only been teaching for 10 years or less. All teachers however possess a qualification, with 10 diplomas and 2 degrees, and 83% (n=10) have received some sort of ICT related training. Of those trained, however, only 50% have formal university qualification based training, the remaining 50% have been trained by either DoBE, the SLL or other education based organizations. It would thus seem that interest in ICTs and their use in teaching and learning amongst teachers is not



Figure 6.8: Reported levels of ICT utilization at schools by teachers and learners

dependent on age, as most teachers across the various age groups were trained in the use of technology to some degree.

Despite any training that they had received, none of the teachers were able to describe what kinds of computers they had, nor how they were configured. Similarly, less than half of the teachers were able to recognize that their schools had a LAN, nor were they able to identify what type of Internet access their school had. They were, however, able to identify which operating systems they had on the computers at school and a number of other software products that were available at school. It is debatable as to whether or not it is necessary for teachers to know or understand this type of information; after all, the teachers are able to successfully utilize a pencil without knowing how they were brought to the school nor the history or particular construction of the pencil. Arguably, the same applies to technology use. However, understanding some of this information would make it easier for the teacher to explain problems when needing assistance and possibly even allow them to do their own basic support or maintenance, given that schools typically do not have on-site technicians and schools have limited access to outside support.

In terms of software use, one can see that typically office products (both Microsoft and Libre) are well utilized when available at schools, followed by browser applications (such as Firefox and Internet Explorer). This would indicate that typically ICT facilities are being used in schools for office productivity related tasks and Internet searching.

School ICT facilities suffered from a general lack of support, with little access to ICT related support structures; schools were typically relying on the SLL for support and maintenance. Furthermore, schools had little access to funds in terms of maintenance or the purchasing of new equipment.

When teachers were asked about what they used the ICT facilities for, the majority claimed that it was for learning purposes, followed by both classroom and school related administration. Only 50% of teachers who answered were using the ICT facilities for teaching. This could be because when asked about the reliability of the computers, teachers typically said that the facilities "sometimes" worked. This would result in a level of distrust when wanting to use facilities in order to integrate them into the lesson. Furthermore, the majority of teachers felt that the ICT facilities only support "some" of their teaching and learning needs, again, that would make them hard to integrate when not all the learning and teaching needs are readily met. When I asked the teachers about the level of access that the learners had to the

ICT facilities, the majority indicated that the learners had "some" access, typically during schools hours, or specific training sessions, or after hours. Interestingly, when asked about the teachers ability to access the facilities, half of those who answered said it was "easy", while the other half said it was "difficult".

#### 6.3 Interviews and Focus groups results

Following on from the questionnaires, the interviews and focus groups were conducted with the willing teacher participants (as per the explanation in Chapter 2). During the analysis of the interviews and focus groups conducted with the teachers at the six schools, I used both bottom up (in vivo) and top down (application of theories) analysis techniques to analyses, interpret and understand the data coming from the interviews with the teachers. 12 broad categories were identified during the in vivo analysis, namely, infrastructure, training, ICT use, ICT integration, support, content, barriers to use, facilitate use, teachers, learners, community members and ICT benefits. Once the interviews had been coded according to these codes that emerged from the data, they were further coded according to the TPACK framework and then a slightly modified version of Bowers [91] Affordances framework.

However, the data generated through this process was very large and more than is possible to include in a single thesis dissertation. As such the thesis will focus on the reporting of the data generated around infrastructure, as the aim of this project was to produce a model of computing infrastructure for rural South African schools. Thus it will predominantly focus on the Infrastructure category from the in vivo coding process, as well as the the TK data generated through the lens of TPACK [153]. I also give some attention to the affordances of the technology as understood by the teachers. The rest of the data analysis will be written up in separate papers and only feature when they offer supporting arguments to better understand the data around the computing infrastructure.

#### 6.3.1 In Vivo coding

I made use of in vivo coding in order to interpret and understand the data that emerged from the interviews and focus groups with the teachers from the six SLL schools. From the coding process 12 broad categories (codes) emerged. Each of these broad categories was further sub-divided into sub-codes in order to be better describe and provide greater insight and meaning with regards to what the teachers were saying. Each category (or code), its sub-categories (or sub-codes) and the associated results can be seen in Figure 6.9 below. In order to obtain a sense of the scope and depth of the comments I created a mind-map of the comments and their associated categories and subcategories that were said during the interview/focus group process (see Figure 6.9). From the mind map in Figure 6.9, one can see the large quantity of data that was generated; more than could adequately be covered in a single dissertation. As such, as discussed earlier, I focus on the Infrastructure branch of the mind map for the purposes of this write-up.



Figure 6.9: Mind map of the categories and sub-categories identified through In Vivo coding of the interviews and focus groups with the SLL teachers

#### 6.3.2 Infrastructure

When asked about infrastructure at the six schools, 326 comments were made by the teachers (see Figure 6.10 highlighting the Infrastructure category). This was the most discussed topic across all the interviews and was discussed more than twice the number of times as the next highest category (see Figure 6.9). Arguably, infrastructure is a large part of the focus of this project, as it seeks to investigate appropriate computing infrastructure for schools, and this would account partly for the higher number of comments in this category. However, the fact that the comments were more than two times the next highest category, is an indication of how important it is to get the infrastructure right in order to even begin to ask how does one integrate the use into teaching and learning. The infrastructure category was further subdivided into Hardware, Software, Networking/Internet, Physical (schools building, electricity, etc), Finance, Importance of technology and maintenance. Of these seven subcategories, the first four were further subdivided in order to gain a better understanding of what the teachers were saying in their interviews and focus groups.



Figure 6.10: Mind map of the Infrastructure category with its sub-categories and sub-sub-categories
Of the seven sub-categories within the infrastructure category, the greatest number of comments and statements from the teachers pertained to Hardware (148 comments). Next was Software (121 comments), Networking/Internet (21 comments), physical (20 comments), importance of technology (8 comments), maintenance (6 comments) and finance (2 comments). These subcategories will be discussed in turn in the subsections that follow on from here.

#### 6.3.2.1 Hardware

ICT hardware was by far the most discussed topic in the infrastructure category during the course of the interviews and focus groups; it was discussed almost as often as the "ICT Use" category (the second highest category of discussion (see Figure 6.9)). Again, this is unsurprising as the focus of the survey process (and hence the interviews and focus groups) was to investigate computing infrastructure for schools. The most obvious topic of discussion would be ICT hardware. That said, it is important to also realize the importance of getting the hardware "right" in order to begin to discuss appropriate integration methods. Within the sub-category the majority of comments clustered around the limitedness of resources (37), the teachers hardware requirements (25), the need for data projectors (21), hardware related faults (20) and hardware placement (15).

Across all the interviews and focus groups teachers discussed the limited resources at their schools. Some noted that they needed more computers, for example Teacher\_A at FH1 said, "Because we have only one lab but there are about 600 learners"; Teacher\_K at FH6 said, "The computers that we have are not equal to the students that are doing CAT. So for example, we have got three sessions for grade 10 and they are sharing the computers, like four or five to one computer"; and Teacher\_D at FH2 commented, "We are thinking of buying a data projector and so if we can have more computers than we currently have that could be better for our learners so that every educator who needs the lab at any time could reach the lab". Other teachers discussed the need for printers or data projectors; "But the problem is that we don't have printers or a data projector" (Teacher\_E at FH3), and "But we don't have a printer for those clients. So they don't print on their own" (Teacher\_J at FH7).

Hardware requirements was the next most discussed subcategory within hardware, generating a total of 25 statements from the participating teachers. Typically teachers discussed their school's need for a computer lab as well as laptops and data projectors that could be moved to other classrooms for use in lessons. Teacher\_E succinctly said, "We do need a lab where everybody can work. And then if a teacher needs to present a lesson then maybe we need two or three laptops and that could be useful. Then they can move around to the classes with a data projector".

The need for (more) data projectors at school was the third most discussed topic by the teachers and was mentioned by all participant teachers except Teacher\_H. Teachers discussed the benefits of the portable nature of data projectors, "*That would be a good thing, also the projector and the laptop are both portable so we can take them to any class*" (Teacher\_B); they also noted that a lack of data projectors prevented them from making use of computer technology within the classroom, "*I think that the only thing now that makes us not use the computer in the classrooms now is that we don't have the data projector*" (Teacher\_D); or generally deplored the lack of data projector facilities at their school, "*But the problem is that we don't have printers or a data projector*" (Teacher\_E).

Surprisingly, to this researcher, hardware faults was only the fourth most discussed topic within the hardware sub-category, with only 20 comments from participating teachers. The majority of the faults clustered around problems with printers and keyboards, and secondly with end client computers (thick, thin and laptop computers). Lastly, worth noting was the commentary regarding placement of computers within the schools. Teachers were asked whether they felt computers should be in a computer lab, in the classroom or both. Of the 11 teachers who participated in the interviews and focus groups only two teachers (Teacher\_J

and Teacher\_I both from FH5) felt that computers should be in the classrooms. All other teachers believed their school needed a computer lab and then if they wanted to make use of technology in the classroom laptops with a data projector should be available. The predominant reason behind this seemed to be related to security; that it was easier to secure computers in one room rather than throughout the school. For example Teacher\_D said, "I also think the lab and the laptops, so that if one doesn't want to go to the lab then you just take the laptop to the classroom and use the data projector. But the lab should be there", while Teacher\_E said, "We do need a lab where everybody can work. And then if a teacher needs to present a lesson then maybe we need two or three laptops and that could be useful. Then they can move around to the classes with a data projector", and Teacher\_G noted, "No, I think, for the moment, it's better for them to be in one room because it is the only room that is... I think that they are well protected".

#### 6.3.2.2 Software

Software was the second most discussed sub-category within the infrastructure category. The discussion regarding software focused largely around three areas, namely: teachers' knowledge of software (44 comments), teachers' software requirements (29 comments) and need for curriculum appropriate software (20 comments).

With regards to the teachers' knowledge of software, the majority of the comments (26) were about how their knowledge was limited. Teachers commented that they didn't know what software was available to them, particularly curriculum appropriate software. For example Teacher\_E commented, "Yes, there are a lot of things that I need for natural sciences but I don't know what programs are available", while Teacher\_D said, "The others we don't understand, like Google Chrome, so maybe with better understanding of those other software options". However, they are aware of their need for access to curriculum appropriate software (20 comments). For example Teacher\_C said, "I think it would be good to have more programs for education, maybe even like Encarta", and Teacher\_H noted, "Ja, I think we need software that is related to our curriculum". Teachers knew that they required software to support their teaching and learning (29) comments), "Yes, more especially for physical science. An example is in chemistry, we are lacking in terms of experiments, practical experiments. So if we could have software that would help us to be able to do some of those practical experiments that would really help us" (Teacher\_B) but not all were sure of what software was available, "I don't know what programs are available" (Teacher\_E). Only the teachers from FH2 and FH5 offered examples of some software packages that they would like access to, the remainder expressed a need for software but were not able to specifically say what software packages they wanted access to.

#### 6.3.2.3 Networking/Internet

Networking was the third most discussed topic within the infrastructure category, however, it was mentioned significantly less often than hardware or software by participant teachers (only 21 comments in total). From Figure 6.10 it can be seen that the two most (equally) discussed sub-categories within the Networking/Internet category were the need for access to the network/Internet from anywhere in the school and network stability (both eliciting six comments). Interestingly, the six comments regarding to access to the network everywhere were all made at one school, FH5, by Teacher\_J and Teacher\_I. Incidentally, FH5 is one of the schools that has been in the SLL network for the longest and anecdotally appear to be the school with the most active users of their facilities.

The comments about network stability came from three schools, FH2, FH5 and FH6, all of which are probably the biggest users of the telecommunications network in the SLL. Teachers from these three schools discussed the need for a more stable network, in which they would reliably have access to the Internet at

all times of the day. Teachers from FH2 and FH6 expressed frustration at having to rely on FH5 (where the shared Internet is located) for access, as they claimed that FH5 were either turning the infrastructure off or that there seemed to be inherent problems with trying to connect to them.

As such, closely related to these two sub-categories is the "Faults/Problems" sub-category which had four comments. Again, all four comments came from the same three schools, FH2, FH5 and FH6. All four comments pertained to faults with Internet access. FH2 and FH6 focused on unreliable access to or a lack of reliability in terms of the Internet being available from FH5. While FH5 noted that there had been times when despite the equipment being powered up the Internet access would be unreliable. The last two sub-categories were comments pertaining to a requirement for Internet access and the need for always available Internet access, while the comments about the need for always on access were from FH6.

#### 6.3.2.4 Physical (School buildings, electricity, etc)

The next most discussed sub-category centered around the physical spaces in which the teachers were operating. These focused on physical space issues (six comments), problems with electricity (six comments), limited resources (four comments), specifically no library facilities (three comments) and finally security issues (one comment). The six comments regarding the physical space came from two teachers at two schools, FH2 and FH3. In both cases teachers lamented the lack of computer lab space within their schools, for example, "So there is a lack of space" (Teacher\_C) and "Yes, so because of space it is not easy for us" (Teacher\_D). While typically the comments pertaining to electricity were to note issues surrounding access to electricity. For example, at FH1, the computer lab had not been properly electrified (and it turned out the entire school had not been properly electrified). Teacher\_A noted, "I don't know what is going on, but I do know that my Principal has been in contact with Eskom but still nothing has happened". In addition, FH5, were experiencing power related problems within the distribution of the power in the school lab which was resulting in damage to plugs and tripping the earth leakage. Alternatively, at FH4 teachers noted that they did have electricity throughout the school (not a given within the former Transkei Homeland of South Africa) and thus would like to have more computers throughout the school as a result.

Other topics within the Physical infrastructure sub-category included a lack of or limited resources (four comments), to which having no library (three comments) is closely related. The limited physical resources is closely tied to the lack of adequate space for ICT facilities within the schools. For example, Teacher\_D commented "Because we have such a small lab at the moment, because we are using the staff room, it is not easy to take the learners to the lab because there are still teachers busy in here or sitting in here", while Teacher\_E explained that they had no formal space to use as a computer lab and so would have to make use of one of the existing classrooms at the school which would mean that teachers would have to continually swap classrooms in order to gain access to any ICT facilities deployed there. Related to limited resources was the fact that not one of the participant schools had a library; "Yes, because we don't have a library so if we could have Internet access that would be great to use for research and other things" (Teacher\_A) and "Even us, we don't have a library. So it is only there [the Internet] that the learners can have information" (Teacher\_D).

Interestingly only one teacher discussed the lack of security with regards to the physical premises, Teacher\_E, however, teachers from FH2 and FH5 noted that they didn't believe the use of laptops would be safe in schools as they'd be easy to remove, while Teacher\_G commented that computers should be limited to one location within the school in order to limit the security threats.

#### 6.3.2.5 Importance of Technology

During each focus group or interview teachers were asked if they thought access to technology at school was important. All responses were that it was, for a number of reasons. Teacher\_B, Teacher\_G and Teacher\_I all highlighted that the world incorporates and uses computers and thus its important for the learners to know how to use them. Teacher B said, "Everybody is interested because it's our daily thing, we need technology", while Teacher\_G said, "Well because now everything is... we use technology. Then we want our children to be good at it. And I think it's the easiest way of doing things" and Teacher\_I said, "I think it's important because teachers will be always hands on. And the learners too. And even the members of the community. And since the times are changing fast, each and every one must be computer literate because the world at large needs people with ICTs. So I think its good for us to have the computer lab so that we are to communicate abroad". Teacher\_I's comment also highlighted the affordances offered by ICTs in terms of communication with others. There were other teachers who also highlighted affordances of technology, most especially how it helps to reduce and/or support their work. Teacher\_E said, "It is very important because it is difficult for us to do everything on our own like paper work. We are writing down everything, we write from textbooks and we write on the board. And that makes the learners to become bored because they must write everything. But when there is [sic] computers you can present the lesson easily and then it makes the learning quite interesting for them. Like when you are explaining leaver and gears in technology they can actually see what is happening there. And the other thing is you can type the exam papers, make use of lesson plans from the Internet, and all sorts of other things", while Teacher\_L said, "OK, for me I think that computers at school helps to reduce a lot of paperwork. Because most of the things that we do we are hand writing. But now we can do them with the computer which makes our task as teachers easier. Also it creates a conducive learning environment. It also attracts the concentration of the learners when you are using the computer together with the data projector, so it attracts them". Teacher\_L's comment also highlights that some teachers believe that the novelty factor of ICTs interests the learners and holds their attention. Finally teachers also noted that they needed ICTs for teaching and learning and most especially to facilitate access to information for the learners (we know from 6.3.2.4 above that no schools had libraries available for learners). Teacher\_D noted, "I think we need them for teaching and learning", and Teacher\_K commented, "And also, it's the only way that they [the learner] can get the information is from the Internet. If they need to search for information for an assignment, it's the only way".

#### 6.3.2.6 Maintenance and Finance

I have grouped maintenance and finance together here as these were the two least discussed topics within infrastructure. There were only two comments about finance, both from the interview with the teachers at FH1. This was because of all the schools, this was the only school to not answer the question in the pre-interview questionnaire about finances for ICTs. When asked if either of the teachers were aware of how much money the school had available for purchasing new and repairing ICT facilities, both teachers indicated that they did not know, saying, "I'm not going to lie, I don't know" (Teacher\_A) and "Yeah, me neither. I think the people who might know are the people in the school finance, but unfortunately we are not" (Teacher\_B).

In terms of maintenance, all schools noted that they did not have a technician on campus ("And also, *if we don't have someone that can fix the computer*..." (Teacher\_K)), nor did they have access to one in the area - the nearest being the DoBE's technician in Idutywa. As such they were reliant on outside help for maintenance. Some teachers were vocal in noting that it was impossible for them to fix the computers themselves, saying "*No we don't. We are unable to fix thing*" (Teacher\_H) and "*Yes! It's quite complicated* 

process because we only have one guy in Idutywa and he is very, very busy. Like now we have the problem with our school laptop and he can't fix it. So we do have a problem" (Teacher\_G).

#### 6.3.2.7 Summary

The teacher participants in the interviews and focus groups clearly expressed a desire and need for computing infrastructure at their schools. They noted that computing infrastructure, together with associated networking technology (specifically the Internet) were typically the schools' only access to information sources and data. Furthermore, teachers commented that in the world today learners and teachers need to be ICT capable; justifying their need for access to computing infrastructure and the Internet. Teachers discussed how they would like access to more computing infrastructure, in order to make facilities more readily available and better able to service the number of teachers and learners. They also discussed the need for specific hardware components (such as laptops and data projectors) in order to facilitate classroom teaching and ICT integration. Furthermore, teachers highlighted the need for reliable hardware solutions, while noting that they had very little access to maintenance and support and were typically reliant on outside sources of help, such as the SLL. Teachers also noted their lack of knowledge of available software products, but also expressed a clear need for software that supports the South African school curriculum.

From the emerging data around infrastructure from the interviews and focus groups, it can be seen that there is a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content).

#### 6.3.3 TPACK coding

As explained in Section 2.9.1 of Chapter 2, after the initial categorization of the interview data, I analysed the data through the lens of TPACK, coding each code(s) per statement with the most appropriate TPACK colour, based on which of the knowledge groups it fell into (Technological Knowledge, Pedagogical Knowledge, Content Knowledge, Technological Pedagogical Knowledge, Technological Content Knowledge, Pedagogical Content Knowledge, Technological Pedagogical Content Knowledge or Context) as can be seen in Figure 4.1.

Category	Pedagogical Knowledge (PK)	Technological Knowledge (TK)	Content Knowledge (CK)	Pedagogical Content Knowledge (PCK)	Technological Pedagogical Knowledge (TPK)	Technological Content Knowledge (TCK)	Technological Pedagogical Content Knowledge (TPACK)	Context	Totals per category
Infrastructure	0	55	0	0	25	21	1	223	325
Support	0	10	0	0	0	0	0	63	73
Training	0	49	0	2	4	5	4	6	70
Content	0	0	38	0	0	16	1	0	55
ICT Use	0	19	0	2	95	21	12	11	160
ICT Integration	0	3	0	0	1	0	28	24	56
Barriers to Use	0	15	0	1	0	0	0	47	63
Facilitate Use	0	7	0	0	0	0	0	16	23
Teachers	0	13	0	1	1	0	1	68	84
Learners	0	2	0	5	7	6	5	13	38
Community Members	0	0	0	0	0	0	0	3	3
ICT benefits	0	4	6	1	29	5	1	9	55
ICT drawbacks	0	0	0	0	0	0	0	0	0
Totals per TPACK	0	177	44	12	162	74	53	483	

Figure 6.11: Spreadsheet of interview data categories vs the TPACK framework knowledge types and context



Figure 6.12: Total number of comments per TPACK knowledge types and context

If we consider the spreadsheet in Figure 6.11 and the graph in Figure 6.12, it can clearly be seen that single biggest grouping is that of context with 48% of the comments made by teachers during the course of the interviews. The next highest category is Technological Knowledge (18%) followed by: Technological Pedagogical Knowledge (16%); Technological Content Knowledge (7%); Technological Pedagogical Content Knowledge (16%). Interestingly, Pedagogical Knowledge was not classified once, meaning that no teacher talked about pedagogical knowledge (teaching strategies) during the course of the interviews and focus groups.

As explained earlier in this chapter, and in Chapter 2, the focus of this dissertation is on the Infrastructure or TK related commentary by the teachers. As such, I will only discuss the TK category of the TPACK analysis here. The rest of the TPACK related analysis will be addressed in follow up papers to this dissertation.

#### 6.3.3.1 Technological Knowledge

169 of the comments recorded during the interviews and focus groups were classified as TK. The majority of comments classified as TK were in the in vivo categories of Training (60 comments) and infrastructure (50 comments) (see Table 6.2 for the complete break down of TK comments per in vivo category and Figure 6.13 for the graphical representation of the same results (reflecting the percentage of non-zero categories as a part of a whole)).

In vivo category	Number of comments classified as TK			
Infrastructure	50			
Support	0			
Training	60			
Content	0			
ICT use	18			
ICT integration	3			
Barriers to use	15			
Facilitate use	7			
Teachers	13			
Learners	1			
Community members	0			
ICT benefits	2			

Table 6.2: Technological Knowledge classified comments per in vivo category



Figure 6.13: Pie chart of the Technological Knowledge (TK) classification of the in vivo categories; representing the non-zero categories percentage as part of the whole

It is not surprising that the results of the TPACK analysis of the Technological Knowledge clusters around "Training" and "Infrastructure". In order to possess Technological Knowledge a person would need to have had some sort of training or education pertaining to the technology. In fact, the majority of comments that were classified as TK within the Training category were comments from teachers about the need for or importance of training, accounting for 44 comments (73% of the Training TK comments). Other comments included teachers' limited knowledge and that ICT integration into the curriculum is aided by having teachers adequately trained with sufficient, appropriate technological knowledge.

With regards to the Infrastructure category, comments centered around teachers' knowledge of infrastructure or lack there of, and more especially, software (42 comments; 84% of the Infrastructure TK comments). Teachers talked about their limited knowledge of (appropriate) software (23 comments) for teaching, while indicating that they needed software for teaching (specifically within their context and curriculum) but were not able to articulate which or what software they wanted/needed. They also discussed how they did possess some knowledge about software (19 comments), which were typically the software products that they were making use in the course of their work (teaching and administration).

The next two in vivo categories (codes) that received a number of TK classifications were "ICT Use" (18 comments) and "Barriers to use" (15 comments). Of the comments in ICT Use, five comments were about schools using their facilities for providing literacy related training; a further five comments were about how ICT training will increase the use of the facilities at the schools; and four comments were about how current computer use amongst teachers was for personal related tasks. The Barriers to use category centered around teachers limited ICT knowledge being a barrier to ICT use in their schools (14 out of 15 comments). The remaining comment in the category was one teacher noting that poor quality facilities were also a barrier to ICT use.

The final category worth noting, the Teachers in vivo category (code), had 13 comments. The comments in this category centered around teachers lack of ICT knowledge, predominantly, and their lack of technical knowledge (i.e. their inability to fix technical problems).

#### 6.3.3.2 Summary

From the results one can see that typically teachers commented that they didn't possess enough knowledge, or that their knowledge was limited with regard to software for use in their teaching. Given, that they commented more often about not possessing adequate knowledge as compared to comments about the knowledge that they did possess, it then follows that there were so many comments stressing the need for and importance of adequate technology-related training. Understandably, teachers also discussed how their lack of ICT knowledge (Technological Knowledge) was a barrier to ICT use and by association, ICT integration.

#### 6.3.4 Affordances

Lastly the data was analyzed in terms of the concept of Affordances (as discussed in section 4.3). Specifically, each comment/statement from the teachers was assessed to see if it discussed an Affordance of technology. I used the Affordance classifications described by Bower [91], as well as three additional affordances, that presented within the data but had no current classification by Bower, namely Efficiency affordances, Communication affordances and Social affordances. Efficiency affordances refer to how technology improves ones levels of efficiency or help to reduce workload. Communication affordances refer to when technology affords opportunities for communication with other people and places. Social affordances refers to when technology affords users opportunities to engage with others socially; forging or supporting relationships. The other affordance classifications were explained or described in Bower's paper [91] and were discussed in section 4.3.

The results of the analysis when considering the Affordance framework can be seen in Figure 6.14. From the pie chart in Figure 6.14 one can see that most referenced affordance by teachers was that of Media affordances (37%). The next most referenced affordance was Navigation affordances (21%). Together these two affordance make up the majority of the affordance classified comments from the teachers during the course of the interviews and focus groups.

As discussed in section 4.3, Bower [91] describes Media affordances as those which offer the functionality of reading, writing, viewing, drawing, listening, and speaking. While Navigation affordances are those which offer the functionality of browsing, linking, searching, and data manipulation [91]. In terms of the comments from the teachers, their media affordance comments centered around the ability to write (or type), read, and view. While, their navigation affordance comments centered around browsing and searching, specifically with regards to the Internet.

The next three most referenced affordances by the teachers were Temporal affordances, Reliability affordances and Technical affordances. Bower [91] defined Temporal affordances as that which offers the ability to access, play back, record and synchronize. Reliability affordances were described as that which refer to the reliability or robustness of the platform, and technical affordances refer to the cross platform nature of many software packages, as well as the ability to adapt to various technological setups such as varying bandwidth and abilities of computer hardware [91].

In terms of the comments from the teachers in this study, the majority of the temporal affordance classified comments centered around the ability to record. Specifically, that teachers could record (save) work covered in order to either recap that with the learners at a future date, or provide them with printed hard copies or soft copies, as well as have access to that same content for future classes in subsequent years. With





regard to the reliability affordance, comments centered around hardware, software and networking faults resulting in the functionality of the ICT infrastructure being unreliable and how that had a negative impact on the (successful and continued) use of the facilities. Lastly, the technical affordances reflected comments from teachers pertaining to a need for software that supports practical work for subjects like the sciences; the inability of the computing infrastructure platforms or configurations to adequately support the teachers in their use of the facilities while teaching learners; and fraudulent software on their systems.

#### 6.4 Discussion

From the data generated during the survey, it can be seen that teachers were unable to describe what kinds of computers they had, nor how they were configured at each of their schools. Similarly, less than half of the teachers were able to recognize that their schools had a LAN, nor were they able to identify what type of Internet access their school had. However, they did clearly express a desire and need for more computing infrastructure at their schools. They noted that computing infrastructure, together with associated networking technology (specifically the Internet) were typically the schools' only access to information sources and data. Furthermore, teachers commented that in the world today learners and teachers need to be ICT capable; further justifying their need for access to computing infrastructure and the Internet. Teachers discussed how they would like access to more computing infrastructure, in order to make facilities more readily available and better able to service the number of teachers and learners. They also discussed the need for specific hardware components (such as laptops and data projectors) in order to facilitate classroom teaching and ICT integration. In addition, teachers highlighted the need for reliable hardware solutions, while noting that they had very little access to maintenance and support and were typically reliant on outside sources of help, such

as the SLL. Furthermore, schools had little access to funds in terms of maintenance or the purchasing of new equipment. Indeed, during the Affordances analysis of the interview/focus group data generated, 8% of the data was categorized as relating to "Reliability affordances" and specifically that facilities at schools were not reliable, making their integration and successful utilization difficult.

Teachers were able to identify which operating systems they had on the computers at school and a number of other software products that were available at school. Looking at the questionnaire results one can see that typically the office productivity suites and Internet browsers were the most known and utilized software. This would indicate that typically ICT facilities are being used in schools for office productivity related tasks and Internet searching. Further supporting that notion, is the Affordance analysis; the two biggest affordance categories (together making up more than 50% of the affordance classifications) were those of Media Affordances and Navigation Affordances. As such, it would appear that teachers are typically using school computing infrastructure as productivity tools and for web searches. Teachers also noted their lack of knowledge of available software products, but also expressed a clear need for software that supports the South African school curriculum. As discussed in Section 4.5, despite the policy documents stipulating the use and integration of ICTs in teaching and learning there is very little in terms of direct guidance to teachers regarding how to integrate technology in the classroom or which technologies best support which classroom activities or pedagogical practices. The closest set of guidelines to the use and integration of ICTs in teaching and learning is "Managing ICTs in South African Schools: A Guide for school Principals", the purpose of which is to "give principals and senior management information on using and managing ICT resources so that they can provide leadership in their schools" [151].

While useful in attempting to assist school management to support the implementation of technology in teaching and learning, the Principal's Guide [151] lacks the level of detail to wholly achieve its objectives [100]. Furthermore, there is no guidance regarding use of technology in the classroom or for teaching and learning specifically. Thus there remains a need to support teachers in the classroom; a need for a guideline or framework that supports teacher's integration of technology in their pedagogical practices (or classroom activities). Further support for a need for a guideline of this nature is the fact that at no point during the course of the interviews and focus groups did teachers discuss pedagogical strategies for the effective inclusion of computing infrastructure; teachers appear to not be adopting or adapting teaching strategies in order to meaningfully include ICTs into their teaching processes.

All the teachers surveyed possess a qualification; 10 have diplomas and two have degrees, and 83% (n=10) have received some sort of ICT related training. Of those trained, however, only 50% have a formal university qualification in some form of ICT-related training, the remaining 50% have been trained by either DoBE, the SLL or other education based organizations. Despite their training, they still expressed a need for more training, and training for their colleagues. From the analysis of the data through the TPACK lens, and specifically looking at the Technological Knowledge classification, we saw that 35% of the comments classified as TK from the interviews and focus groups were about training. And that 73% of the training related comments from teachers were about the need for or importance of ICT related training for teachers. Thus the need for appropriate ICT related training (both technical and pedagogical) for teachers is imperative to the successful adoption and use of computing infrastructure in the teaching and learning processes. Furthermore, training would support teachers in better knowing and understanding the software that they have access to or could get access to for the purposes of teaching, as well as, a better understanding of hardware and how they work, in order to be able to do rudimentary trouble shooting in order to be able to better make use of their facilities.

When teachers were asked about what they used the ICT facilities for, the majority claimed that it was for learning purposes, followed by both classroom and school related administration. Only 50% of teachers who answered were using the ICT facilities for teaching. This could be because when asked

about the reliability of the computers, teachers typically said that the facilities "sometimes" worked. This would result in a level of distrust when wanting to use facilities in order to integrate them into the lesson. Furthermore, the majority of teachers felt that the ICT facilities only support "some" of their teaching and learning needs, again, that would make them hard to integrate when not all the learning and teaching needs are readily met. Furthermore, teachers had also requested more training and so the lack of use of the facilities for teaching could also be a reflection of their confidence levels with respect to their skill set. When I asked the teachers about the level of access that the learners had to the ICT facilities, the majority indicated that the learners had "some" access, typically during schools hours, or specific training sessions, or after hours. Interestingly, when asked about the teachers ability to access the facilities, half of those who answered said it was "easy", while the other half said it was "difficult".

From the data generated by the survey, it can be seen that there is a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content) that is also easy to maintain and replace/repair. Arguably, this is the first step in realizing the dream of ICT integration in schools in rural (and indeed urban) South African schools. We acknowledge however, that even with fully working, reliable, plentiful, curriculum appropriate computing infrastructure, it can still occur that facilities will not be readily or appropriately utilized for teaching and learning. In order to fully utilize computing infrastructure to its fullest potential in appropriate manners to support teaching and learning, teachers need to be adequately trained (in both technological and pedagogical knowledge). However, for the purpose of this research work I focus on the first step of attempting to provide appropriate computing infrastructure for rural South African schools. I hope that the future work from this research project will begin to contribute to a guide where computing infrastructure affordances can be matched with curriculum-based classroom activities (as per the work described by Harris, Mishra and Koehler and discussed in sections 4.5 and 4.6) in order to attempt to support teachers in their endeavours to successfully and appropriately integrate the use of ICTs into teaching practices.

#### 6.5 Conclusion

This chapter described the survey (questionnaires and interviews and focus groups) undertaken with six schools in the SLL. Furthermore, it detailed the analysis of some of the data generated by the survey, specifically focusing on the in vivo category of "Infrastructure", the TPACK analysis of the Technological Knowledge and the Affordances analysis of the interviews and focus groups. From the analysis I found that there is a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content) that is also easy to maintain and replace/repair. As a result of the relative novelty ("newness") of computing infrastructure within the environment of the SLL, the requirements elicited from the teachers are still relatively generic, such that the facilities that they require could be used to support other initiatives in the communities outside of the explicit purposes of education. However, this currently plays to the strengths and desires of the SLL, which would like to see communities utilizing computing infrastructure within schools, outside of normal school hours.

In the next chapter I propose models of computing infrastructure for the SLL schools. The chapter describes the various computing infrastructure options investigated and design choices made (in order to meet the objectives of a reliable, cost effective, curriculum appropriate and easy to maintain computing infrastructure) together with the computer laboratory experimentation conducted with each, in order to propose three possible models for testing in three schools within the SLL.

### **Chapter 7**

## Computing infrastructure model: Laboratory tests

#### 7.1 Introduction

This chapter details the experimentation with various clustering techniques, including performance tests utilized to assess the viability of cluster LTSP vs standard thin client LTSP and LTSP fat/hybrid client configurations. The results of the various laboratory experiments are used to inform whether or not clustering LTSP would provide a potentially viable solution to computing infrastructure requirements within a school environment.

The chapter begins with a review of the current computing infrastructure in the SLL and highlights the shortcomings of the current configuration. This is followed by a discussion of various clustering techniques that could be employed within the SLL schools in order to address the shortcomings of the current model. Next, the chapter shows how Sikuli, was utilized to compare the various infrastructure configurations and compare hardware and software performance and user experiences. The chapter concludes with the results of the tests and their implications.

#### 7.2 Current computing infrastructure model in the SLL: A recap

In Section 3.2.4 I explained that from the very beginning of the project the SLL has pushed the thin client computing topology as the primary means of computing infrastructure in schools (for more about thin clients see Section 5.2.1.2, and think clients in Section 5.2.1.1). Furthermore, the project utilizes free and open source software solutions (operating systems and user applications), specifically Edubuntu, as they are cost effective, robust and promote access to knowledge (i.e. users and researchers are able to see the source code, understand how software works and possibly modify it in order to better meet their needs). In addition, as discussed in Section 3.2.4.3, Edubuntu is is an operating system aimed at education and the SLL computer labs are in the local schools.

The choice of thin clients over thick clients is based on cost effectiveness and sustainability, the reason for retaining the historical form of "standard desktop PC" deployment over the more trendy tablet, is, as previously mentioned in Section 3.2.4, that the desktop PC configuration lends itself more readily to being a shared resources and promotes the users to being both producers and consumers of information and knowledge. High end cell phones and tablets are devices that essentially promote the consumption of information but do not readily afford production of information – for example, it would incredibly tedious to attempt writing this thesis on a tablet rather than at a computer with a keyboard, monitor and mouse. As producers of information members of the school and local communities can actively contribute and

take part in the digital "networked society" [27]. It is not to say that this will not change in the future; as technology advances manufacturers might provision new methods in order to support mobile devices being more readily useful in the production of knowledge or artifacts, but this is not the case right now.

It is worth noting that while the underlying focus of computing infrastructure in the SLL schools has been on a 'computer lab', as in a dedicated room within the school where a number of computers are colocated for specific classes, the architecture we have proposed, implemented and deployed in the schools in the SLL supports computers in the classroom for typical class work such as presentations, demonstrations, document/information generation or Internet surfing. Furthermore, the terminals can be located in the secretary's and principal's office or the school staff rooms for more administrative types of work.

However, from Section 3.2.4.3 it was shown that there are some disadvantages to the use of thin client computers with a central server, primarily that the single central server represents a single point of failure. This does not present a robust solution, especially when considering that the environment in which the SLL operates is harsh on the life of a computer – "dirty" electricity, lightening strikes, heat, dust, etc – resulting in a high failure rate of the server hardware, even when on UPS backups to attempt to protect from electrical compromise.

Secondly, when all the clients are being utilized the load on the server is high, and so the user experience is compromised because the response time slows down. It was seen from the MRTG graphs in Section 3.2.4.3, specifically in Figure 3.6, that the load on the CPU and the memory resources of the server, when servicing a large number of simultaneous clients was high. Naturally the implication is that the server would benefit from increased computation resources (specifically CPU and memory). However, servers that support greater amounts of memory and powerful CPUs are typically more expensive, which might not suitable to a development project like this.

Finally, the survey conducted with teachers and reported on in Chapter 6 revealed that teachers expressed a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content) that is also easy to maintain and replace/repair.

Thus we need computing infrastructure that is more easily healed (improving reliability and maintenance efforts), and has greater amounts of memory and CPU resources to readily serve an increasing number of thin clients (increasing the amount of computing infrastructure available) in a cost effective manner, while still utilizing Edubuntu in order to provide access to education applications. All this, however, within the important constraints expressed by an important group of users, the teachers.

# 7.3 Proposed improvements to current computing infrastructure model in the SLL

In order to improve the current computing infrastructure model in the SLL it was decided to replace the single central server with a cluster or grid. The use of clusters or grids would help to improve the ease with which the computing infrastructure can be healed and build redundancy and high availability into our solution. I wanted to retain the benefits of thin client computing, while making the infrastructure more easily healed and increase the available computational resources of the computing infrastructure without increasing the overall costs very much.

While thin clients can be acquired relatively cheaply (particularly when using recycled computers) a thin client server can be expensive. It should be noted that a robust server with a minimum of 4 GB RAM - preferably more - would be needed [281]. The use of recycled desktops or server computers configured as cluster servers offer a cheaper alternative to purchasing new and expensive servers.

I investigated four cluster computer options and one grid-like configuration. The cluster options were: DNS Round Robin, Ubuntu LTSP Cluster, SSI and Virtual Desktop. The grid-based solution is that of the Edubuntu fat/hybrid clients. Each of these solutions is discussed in the subsequent sections below.

#### 7.3.1 DNS Round Robin

As described in section 5.2.4, DNS Round Robin is a method for load balancing or fault-tolerance, providing multiple, redundant Internet Protocol service hosts through controlling the DNS responses to address requests from client computers [189]. A DNS-based distributed server architecture matches a domain name, e.g. google.com, to multiple IP addresses of multiple servers within a cluster of servers. When a client resolves a DNS name to an IP address, the DNS server responds with a rotated set of associated IP addresses. Each client generally selects the first address from the list. In this way, a group of clients will be distributed across the servers associated with the shared domain name (i.e. the cluster) [189].

In order to configure a DNS Round Robin cluster server a group of Edubuntu LTSP servers must share the same DNS name and file storage location. In the LTSP DNS Round Robin cluster which was developed, NFS (see section 5.3.3 for more details about NFS) was used to provide user home directories to the multiple cluster servers and LDAP (see section 5.3.4 for more details about LDAP) to authenticate users. Bind was utilized in order to provide the DNS service (see section 5.3.5 for more details about DNS) on the network.

The basic design of the LTSP DNS Round Robin cluster consisted of five servers, all running Edubuntu Linux [282]. One of the servers was configured as an NFS and LDAP server (named the Parentnode), while the other four were LTSP servers for the thin clients. The NFS server (Parentnode) exported the home directories of the users, which the four LTSP servers (which had NFS clients) mounted. This configuration provided the users with access to their home directories and consequently all their files regardless of which LTSP server their thin client connected to. The Parentnode being also an LDAP server, allows centralized authentication with the same user-name and password for each user regardless of which LTSP server they were connected to. The four LTSP services for the thin clients' LTSP requests. One of the LTSP services also provided DNS and DHCP services for the thin clients (called DNSnode, while the remaining three were called Generalnodes).

The DHCP server "next server" option provides the PXE-booting thin clients with the IP address of the server from which it can transfer its kernel via TFTP (and initramfs image, if present).

In order to configure the thin clients to use any one of the LTSP servers to access their files and applications one needs to set the LDM\_SERVER variable in the lts.conf file to instruct each client that LDM must communicate with the cluster. This is again achieved using the DNS Round Robin configuration by setting the variable to refer to a domain name, rather than an IP address, so that when the clients query the DNS server they choose an IP address from the rotated list of IPs they receive from the DNS server. This results in the client load being balanced (largely evenly distributed) across the LTSP server nodes within the cluster. A diagrammatic representation of the LTSP DNS Round Robin cluster can be seen in Figure 7.1.

This solution provides increased memory and processing capacity via less powerful central hardware to the thin clients to provide a better overall experience for the end user, while improving the self healing ability of the LTSP service. That said, the single point of failure of the NFS server still needs to be addressed (possible solutions to this are discussed in the future work (Section 9.5) of Chapter 9). Details of the experimentation with the configuration and implementation of a DNS Round Robin LTSP cluster can be found in Appendix H.



Figure 7.1: Diagrammatic representation of the LTSP DNS Round Robin cluster

#### 7.3.2 Ubuntu LTSP Cluster

The Ubuntu LTSP cluster is relatively similar to the DNS Round Robin implementation except that instead of using DNS Round Robin to load balance the client connections, a controller evaluates the hardware resources of each node within the cluster (assuming not uniform) as well as the number of clients currently connected to each node and then determines the least loaded node and connects new clients to that node in the cluster. The implementation of the LTSP cluster in Ubuntu requires one root server (that provides services to the clients such as DHCP, NFS, TFTP, PXE booting, DNS, routing to the Internet, LDAP authentication, etc.) and one or more application servers which facilitate the clients access to applications.

The Ubuntu LTSP cluster provides the following features [14]:

- · Central Configuration web interface to load balance thin clients across multiple servers
- Complete auto-login support with account creation
- Store hardware information for all clients in the control center
- Managed PXE configuration, creating links to a specific configuration for a specific node

In the Ubuntu LTSP cluster implementation the "LTSP-Cluster" is integrated into the LTSP boot image (also called chroot). The only difference between a standard LTSP boot image and the LTSP-Cluster boot image is a single configuration file called getltscfg-cluster.conf found in /etc/ltsp of the chroot. This file specifies the information pertaining to the load balancing server. The clients can send a hardware inventory along with the requests to the load balancer. The hardware information is then used by the LTSP-Cluster Control Center to send configuration information to clients depending on their hardware components. The LTSP-cluster consists of various components/plugins, namely, the control center; the load-balancers; NX

load-balancer; PXE configuration editor; and Account manager [14]. The most user-visible component is the LTSP-Cluster Control Center which provides the web interface for the system [14]. Each of these components is explained below, and a diagram of the overall system can be seen in Figure 7.2 [14].

During the boot process, the LTSP-Cluster boot image talks exclusively to the LTSP-Cluster Control Center using HTTP requests (GET/POST). Communication typically occurs on three occasions: during the beginning of the boot process when the LTSP-Cluster boot image gets lts.conf in order to initialize services; during the second part of the boot process as variable information is lost after the first step; and lastly just before the session starts when the client requests its lts.conf and a refresh of the configuration for login [14]. After the boot process has completed, the client will make further requests to the control centre and the account manager (also via HTTP) when using LDM for login. These requests are typically related to the user's session [14].

The LTSP-Cluster load-balancer server is used to contact all the LTSP-Cluster application servers and retrieve information pertaining to their current load, user list and memory usage. The load-balancer then returns the IP of the least loaded server (according to its configuration) to the thin client [14]. The LTSP-Cluster load-balancer agent is contacted by the LTSP-Cluster load-balancer server. It exists for both Linux and Windows and listens for incoming XML+RPC over HTTP requests. This agent runs on the LTSP-cluster application servers; one for each server [14].

The LTSP-Cluster NX load-balancer is a very minimal SSH load-balancer which redirects incoming SSH connections to the less loaded NX-capable application server. It contacts the LTSP-Cluster loadbalancer server using HTTP requests. An NX-server is a computer that runs the NX service that handles remote X window system connections. The application attempts to improve on the performance of the native X display protocol so that it can be used over very slow connections such as that of Dial-up. It wraps the remote connections in SSH sessions for encryption. The NX technology is included in the LTSP-cluster application to support virtual desktops for remote users using the NX technology. The NX protocol can allow remote Windows and Linux sessions to be accessed from a web browser with very low bandwidth (40 kbps) requirements and tolerance for high-latency connections. The NX client runs on various operating systems including Linux, Mac, and Windows [14].

The LTSP-Cluster PXE config runs on the TFTP server and is used to generate links between the thin client MAC address and its TFTP configuration. It contacts the LTSP-Cluster Control Center using direct PostgreSQL database access.

LTSP-Cluster Account Manager is the service in charge of managing the sessions on the Linux LTSP-Cluster Application server. It's used to [14]:

- Return the SSH key signature (using a plain-text protocol)
- Answer auto-login session creation (using a plain-text protocol)
- · Handle user session clean-up on logout
- Handle mount point clean-up on logout

The current code also supports an encrypted protocol (using blowfish) to kill a user session and get a process list (to be used by the new LTSP-Cluster Control Center) [14].



Figure 7.2: LTSP Cluster implementation [14]

In principle this is a better idea than DNS Round Robin in circumstances where each node in the cluster is not uniform as it would allow for a better utilization of available resources and provide users with a good overall experience. However, there is a pitfall in the solution. It takes some time for the controller to be notified of increases in load on any given node, so that should a large number of clients attempt to connect in a short space of time - which is standard practice in the classroom environment - then most of those clients will be connected to one of the nodes, the most powerful, leaving the other nodes relatively unused and thereby not distributing the load across the available resources. In instances like this, the DNS Round Robin would be a better solution as it would relatively evenly distribute the large number of incoming connections across all the servers instead of overloading the most powerful of the nodes and leaving the rest under loaded. Furthermore, this solution provides access to increased memory and processing capacity for the thin clients, as well as improved self healing for the LTSP service (like that of the DNS Round Robin cluster). However, the single root server still presents as a single point of failure to this solution. Details of the exact method of configuration and implementation of an Ubuntu LTSP cluster can be found in Appendix I.

#### 7.3.3 Single System Image

A SSI machine was implemented using Kerrighed (for a detailed explanation of SSI see section 5.2.4). Kerrighed is an open source SSI project that provides the view of a single SMP machine on top of a cluster. The Kerrighed solution consists of a set of kernel distributed services that are in charge of the global management of the cluster resources [283, 193]. According to the project web page, the goals of Kerrighed are "ease of use, high performance of applications, high availability of the cluster, efficient resources management, and high customizability of the operating system" [283], while the implementation is an extension of the Linux operating system [193].

One of the Linux distributions that Kerrighed has been ported to is Debian, which is very similar to the Ubuntu and Ubuntu-related flavours like Edubuntu. I used the Edubuntu Linux distribution together with Kerrighed to build a SSI thin client cluster server. The SSI cluster had one master server from which the common file system was shared via NFS. The hard disk of the master server was divided into two partitions. The Ubuntu operating system was installed in one of the partitions while the Edubuntu operating system was installed in the second partition. The Edubuntu partition, patched with the Kerrighed extensions, was then mounted by the Ubuntu system and exported to the network. This meant that the thin client computers were connecting to Edubuntu Kerrighed Linux nodes (Figure 7.3). All thin clients will open connections to the IP address of one of the cluster nodes. Kerrighed can then potentially migrate processes and connections to other cluster nodes after they have been established in order to keep the load balanced across the cluster. While each node has its own IP address, thin clients need only connect to the one IP address, associated with the NFSRoot, once connected the Kerrighed system would load balance the incoming connections and processes across all the nodes. This solution could improve upon that of the DNS Round Robin because not only are thin client connections load balanced but also all subsequent processes that thin clients initiate during their use of the system. In addition, if one node fails processes can be migrated to other nodes because of the high availability nature of Kerrighed (see Figure 7.3).



Figure 7.3: SSI Cluster with Ubuntu NFS server (with file system and storage) and Edubuntu Kerrighed Linux cluster nodes, serving thin client computers

The SSI was the most difficult of all the solutions to implement as it required manual patching of the Linux kernel. Once done however you have a cluster of servers that allow you to migrate processors from one processor on one server to another. There is however a large disadvantage that I discovered when attempting to implement this solution with LTSP clients and desktop computing. The problem is that applications used in desktop computing, such as OpenOffice.org and LibreOffice Writer or Firefox cannot be migrated from one processor to another because these applications fire off a number of subprocesses which need to know where the parent process is at all times making this much harder to migrate than simpler processes. As such I was unable to migrate complex applications like Firefox and OpenOffice.org Writer from one processor to another. For this reason the SSI solution is better suited to simpler processes performing mass computations or data analysis as these applications are typically written to be migrated and implemented across multiple processors which typical desktop applications are not. Thus SSI and Kerrighed are great solutions for scientific problems written to be solved on mass processors systems. The Kerrighed configuration, like both the DNS Round Robin and the Ubuntu LTSP cluster, improves the self

healing of the LTSP service but doesn't eliminate the single point of failure represented by master server. Details of the exact method of configuration and implementation of a Kerrighed SSI LTSP cluster can be found in Appendix J.

#### 7.3.4 Virtual Desktop

As discussed in section 5.2.7, Desktop visualization (or Virtual Desktop infrastructure, VDI) refers to the virtualization of the end user's desktop by separating the PC desktop environment and its applications from that of the typical PC hardware and instead making the desktop available on a virtual server in a client-server model [10, 196]. In section 5.2.7 I listed the different types of VDI, namely, user state virtualization, application virtualization, session virtualization, virtual desktop infrastructure, blade-based virtual desktops and single-desktop virtual machine. Session virtualization VDI was essentially experimented with during the various LTSP cluster configurations (DNS Round Robin, Fat clients, Ubuntu LTSP Cluster and the SSI). In this section, I discuss the idea of creating a Virtual Desktop Infrastructure.

Session virtualization and Virtual Desktop infrastructure have similar benefits, namely, centralized control and flexibility in terms of offering different options for different user categories. VDI also offers availability benefits because a virtual machine can be more readily and easily copied, moved and backed up than a physical machine. The only downside that VDI has when compared to session virtualization, is that it requires more server and storage resources which could potentially make it slightly more expensive in terms of cost when compared with session virtualization [10].

The system design that I propose is a central server or servers that serve an image for the thin clients to boot – much like LTSP; a server that responds to DHCP requests and serves TFTP files to booting thin clients. Once the clients are booted instead of connecting to an X-server via ssh, which is the standard LTSP installation/configuration, they connect using VNC to a virtual desktop running on the central server(s) using Virtualbox. The required services can be balanced across multiple machines. One server could for example provide the DHCP, TFTP and file services (NFS) to the clients and virtual server(s). Then one or more machines could host the required virtual desktops. Each virtual desktop will require a minimum of 512 MB of RAM, 1 GB however, would be preferable. Thus depending on the number of virtual desktops that could connect at any given time this solution has the greatest memory requirements when compared to the other cluster solutions discussed thus far. Assuming 20 clients, you need a minimum of 10 GB of RAM, but 20 GB would be better; making the cluster hardware requirements a more high-end than those of the LTSP solutions.

The VDI implementation was also only done in a virtual environment and not tested in the classroom or with many incoming clients. It is also a complicated solution that requires a lot more memory in order to serve the clients as compared with any of the other solutions as each client is provided a full virtual desktop environment. Each environment would require at least 1 GB of memory to run effectively which would mean that in order to support 8 clients a server would need more than 8 GB of RAM (1 GB for each client and some memory for itself in order to run its own operating system and perform all the overhead in managing the various VMs). Implementing this solution on a single server is the simplest method as all the virtual desktops would be run off one single server but it would require a staggering amount of RAM in order to provide enough virtual desktops to service a school environment. Running the solution on a cluster of computers is a bit more complicated as it would require a central controller starting up new instances of virtual desktops on nodes in the cluster. As a solution for school computing it is an option but perhaps a rather more expensive option than the other two solutions of DNS Round Robin and the Ubuntu LTSP cluster. For schools with large financial resources at their disposal however this could be a possible solution for them to employ especially as there are hardware solutions already available for sale, such as iWyse and N-Computing, which can also be implemented with Windows Operating systems. There has been recent

research done in India in using VDI solutions to support the ICT needs of rural schools [198], and indeed in education in general [197], as well as the growing popularity and maturity of open source VDI solutions such as Ulteo [284] and FOSS-Cloud [285]. Details of the experimentation with the configuration and implementation of a Virtual Desktop LTSP cluster can be found in Appendix K.

#### 7.3.5 Edubuntu Fat clients

The discussed benefits of thin client computing are lowered costs in initial deployment as well as running costs in classrooms, libraries and telecentres, together with eased maintenance, but they do have some computational limitations, especially when it comes to graphical and CPU intensive software (see section 5.2.1.2). For example, playing a video on one or more thin clients can be enough to saturate the network. In recent LTSP versions, it became possible to run some applications locally on the clients, making it possible to use the client's CPU/GPU/memory and thereby improve performance and user experience. However, since LTSP 5.2.1, it's been possible to run everything locally. This essentially makes a client terminal a complete fat/hybrid client (similar to the thick clients discussed in Section 5.2.1.1 but without a hard-disk) that simply uses the network as a storage device for its files and its operating system. It combines many of the benefits of thin clients and thick clients, while also requiring a less powerful central server which becomes just a file server. It is also useful to note that the fat/hybrid client image can be used for both fat/hybrid and thin clients on the network, which is useful in situations where you have a mixture of both new and old client hardware [17, 286].

Similarly to thin clients, fat/hybrid clients use the server as a network disk for /home and for the root file system, and authenticate with LDM/SSH. As mentioned, fat/hybrid clients use their own RAM (memory) and CPU power to run the applications. Of course this also means that their hardware requirements are the same as for running Ubuntu on normal workstations. LTSP fat/hybrid clients are similar to the functionality available in Diskless Remote Boot in Linux (DRBL) and Netlive (an alternative to LTSP developed in Italy), with the exception that LTSP fat/hybrid clients use NBD for offering the client's root file system to the client, which is faster than NFS [17]. For a comparison of thin and fat/hybrid clients see Table 7.1.

Edubuntu fat/hybrid clients together with a single central server might not be considered as a grid. However, I would argue that the load of servicing the users of the system is distributed across the server and the fat/hybrid clients, instead of all computational requests having to be serviced by a single server alone. This solution does provide users access to more memory and processing capacity and thus provide overall speed and a better user performance; and makes better use of all available computing resources. Furthermore, it removes the true Terminal Services part of LTSP, making the clients not as susceptible to failures of the LTSP service, improving the self healing aspect of the overall configuration. Unfortunately, however, it does not eliminate the single point of failure of the server, which is still responsible for serving the clients with their images and providing home directories and authentication. Thus, if the server fails the computing infrastructure facilities are still rendered useless. For an explanation on how to configure an Edubuntu LTSP server with fat/hybrid clients see Appendix G.

#### 7.4 Sikuli Scripts

In order to test the performance and user experience of the various computing infrastructure configurations experimented with in the laboratory I used Sikuli to simulate user interactions (further details of the experiments are described in section 7.5 and section 7.6). I specifically focused on user interactions with two applications, Firefox and OpenOffice.org (or LibreOffice) Writer; two applications that one would expect would typically be used on a regular basis by most desktop users. In the Firefox example, I launched the application, then typed in the address of a local website (local to the cluster server, in order to remove time

Fat Clients	Thin Clients
The home directory is mounted on login	The client runs directly off the server
(with sshfs, optionally with NFS)	
The fat client has a larger NBD root file	The NBD root on the thin client version
system containing a complete OS with all the	simply has to get the client to SSH/LDM
desired additional programs	logon screen and provide an X-Window
	server
Processes are not running on the server but	All processes are running on the central
locally on the clients preventing an admin	server and can be controlled by the admin
from interacting with them from the server	from the server
(i.e. to kill processes)	
Fat clients have to maintain their own	Conversely, as processes are not running on
processes and are responsible for killing	the server there is less maintenance required
'stale' processes as the administrator on the	to kill 'stale' processes that accumulate from
server is not able to	the thin clients
Internet connectivity needs to be provided	Client apps run on the central server and so
directly to the fat client, i.e. the client needs	all network traffic originates from and routes
to be directed to an Internet gateway via	through the server and thus clients need not
DHCP services	know where the Internet gateway is
Runaway processes will not bring down the	Runaway processes can negatively impact on
central server, but rather just the misbehaving	the central server rendering it slow or
fat client	completely unusable

Table 7.1: Similarities and differences between Fat and Thin clients [17]

factors of accessing Internet-based web sites). Once loaded, the simulated user clicks on a link and then scrolls to the bottom of the new page and then closes Firefox. Similarly, for OpenOffice.org Writer, the application is launched and opens a local file on the user's home directory. The script then scrolls to the bottom of the file, saves the file and then close the application. These are relatively normal interactions; Firefox being used to view a web site and OpenOffice.org Writer being used to load a document.

Sikuli was installed on each of the cluster computers. The Sikuli installation provides an IDE (sikuliide) that can be run in order to create and edit scripts. Once the script is completed it is then exported as an "executable"; a .skl file. The IDE program can be called from the command line and passed the executable which it will then run.

Testing in single user mode worked flawlessly. However, when moving to a multi-user environment I encountered a problem. The application dynamically writes the libraries found in the Sikuli root directory (the libs folder) to a /tmp/tmplibs folder as part of running the Sikuli executable files. However, in a multi-user environment the first user will create the directory and own it which results in errors for all subsequent users. Thankfully though the Sikuli developers also allow for the temporary library files to be written to the current directory from which the application is being invoked (in this case the home directory of each user). Thus creating a libs directory in the home directory of each user, allowed each user to successfully run their own instance of the Sikuli application and the corresponding Sikuli executable script that I wrote. Once I had the scripts working in the multi-user environment I could begin the testing. The bash script used to invoke the tests for Firefox can be seen in Listing 7.1.

The script took parameter inputs, namely the number of times the user interaction should be iterated (ITERATIONS) and the offset (in seconds) at which the application should begin (OFFSET). Through experience I implemented the offset because the server was not able to process all the clients beginning one at a time, they needed to be offset from each other by 20 seconds, this gave the server time to load

```
Listing 7.1: Firefox bash script to invoke the Sikuli tests
```

```
#!/bin/bash
ITERATIONS=$1
OFFSET=$2
STARTTIME="15:42"
while ! date | grep -q $STARTTIME ; do
        date
        sleep 0.1
done
echo
sleep $OFFSET
date
for ((i=0; i < SITERATIONS; i++)); do
        echo "iteration $i"
        /usr/bin/time - f "\%U, \%S, \%E" - a \
                -o /tmp/$USER-sikuli-firefox-output-$ITERATIONS.csv \
                /etc/Sikuli-X-1.0rc2-linux/Sikuli-IDE/sikuli-ide.sh \
                -r ~ings/firefox-loadPage-gnome-type.skl
        sleep 60
done
grep -iv command /tmp/$USER-sikuli-firefox-output-$ITERATIONS.csv \
        > $USER-sikuli-firefox-output-$ITERATIONS.csv
```

the Java environment necessary to run the applications. Without doing this, I could not reliably run the tests – its not to say that the machines were not able to support multiple users running applications such as OpenOffice.org or Firefox, but rather loading many instances of Java takes time, hence the 20 second delay, which did not remain in sync but will vary as the tests run anyway. I kept this delay to start the initial tests throughout all the different experiments conducted in order to allow me to compare results. The 60 second sleep in the script also helped all the processes to finish off from one test to another before the next began. Note, I are not trying to produce software that runs quickly, but rather compare the load and speed at which one platform processes requests compared with another (single server, versus multiple servers working together in a cluster), so I was not interested in tweaking it to be as fast as possible, but rather finding a test that would run repeatedly on multiple platforms.

At the start of each test, I adjusted the start time to some future time, so that all scripts began simultaneously, each being passed the number 50 (for 50 iterations) and then a number between 0 and 1500 to indicate the number of seconds they would be offset by. The OpenOffice.org or LibreOffice script was nearly identical to that of Firefox, and performed similar tasks and can be seen in Listing 7.2.

The scripts capture the time it takes for each iteration of the user interactions. These are then collected in a csv file in /tmp. Once the test is over, they are written to a csv file in the user's home directory, removing any superfluous information, so that the time taken is recorded. The csv file contained 3 entries per record, one for the User land time, one for the system time and one for the total elapsed time. It is the last time per record that I was interested in as it reflected the total time that it took for the simulated user to complete the interaction with the application (be it Firefox or OpenOffice.org).

During the initial testing phases (prior to conducting the experiments in a multi-user environment) I experimented by running the tests in incremental iterations, namely, 10, 20, 30, 50 and 100 times. I then calculated the standard deviation for each batch of iterations and found that for 50 or 100 iterations the standard deviation tended to be 1% or less. Thus, I ran all of the experiments (which are discussed in sections 7.5 and 7.6), for both of the applications, a total of 50 times per user.

Listing 7.2: OpenOffice/LibreOffice bash script to invoke the Sikuli tests

```
#!/bin/bash
ITERATIONS=$1
OFFSET=$2
STARTTIME="09:57"
while ! date | grep -q $STARTTIME ; do
        date
        sleep 0.1
done
echo
sleep $OFFSET
date
for ((i=0; i < \text{SITERATIONS}; i++)); do
        echo "iteration $i"
        cp ~ings/openoffice-bckup-testfile.odt ~ings/openoffice-testfile.odt
        /usr/bin/time - f "\%U, \%S, \%E" - a \
                -o /tmp/$USER-sikuli-openoffice-output-$ITERATIONS.csv \
                /etc/Sikuli-X-1.0rc2-linux/Sikuli-IDE/sikuli-ide.sh \
                 -r ~ings/openoffice-app-write-save.skl
        sleep 60
done
grep -iv command /tmp/$USER-sikuli-openoffice-output-$ITERATIONS.csv \
        > $USER-sikuli-openoffice-output-$ITERATIONS.csv
```

#### 7.5 Cluster server vs standard thin client server laboratory testing

In order to get a feel for the level of performance of a cluster in general, I experimented with a standard thick client, a single cluster node supporting one client, a cluster node supporting the maximum number of clients possible per node (heavily loaded) and a cluster of four nodes supporting 20 clients.

The initial tests I did were to investigate the use of older, Pentium 4 desktop computers with 2 GB of RAM each as a cluster server solution. In order to learn more about the solution I compared simulated user interactions of 20 thin clients connected to the cluster against a single thin client connected to one of the cluster nodes, a single node as a thick client computer and a single node heavily loaded. I found that in LTSP desktop computing memory is the bottle neck rather than the processor (unsurprisingly given that this is desktop computing). The results of the tests indicate that older desktops could be used to build a LTSP cluster server to support a greater numbers of LTSP clients as compared to a newer, single server with a more powerful processor but less available memory.

#### 7.5.1 Single cluster node as thick client

The first set of benchmarks was to run the Sikuli tests on a single node of the cluster as though it were a thick client. A single user was logged into one of the cluster nodes (as though it were a standard thick client) and the Sikuli tests were run (50 iterations each) for both Firefox and OpenOffice.org Writer and the times were saved to a csv file for later analysis. When running the tests with the Firefox application the average time taken to complete the user interactions was 39.99 seconds with a standard deviation of 1.13 seconds (2.83%). When running the tests with the OpenOffice.org application the average time taken to complete the user interactions with a standard deviation of  $0.2 \times 10^{-5}$  seconds ( $3.46 \times 10^{-6}\%$ ).

	-			-	
Mar 21	14:57:12	DNSNode	kernel:	[941920.458928]	543 total pagecache
pages					
Mar 21	14:57:12	DNSNode	kernel :	[941920.458933]	0 pages in swap cache
Mar 21	14:57:12	DNSNode	kernel:	[941920.458938]	Swap cache stats: add 0,
delete	0, find	0/0			
Mar 21	14:57:12	DNSNode	kernel:	[941920.458942]	Free swap = $0kB$
Mar 21	14:57:12	DNSNode	kernel:	[941920.458946]	Total swap = 0kB
Mar 21	14:57:12	DNSNode	kernel:	[941920.473104]	522096 pages RAM
Mar 21	14:57:12	DNSNode	kernel :	[941920.473108]	294770 pages HighMem
Mar 21	14:57:12	DNSNode	kernel:	[941920.473111]	8835 pages reserved
Mar 21	14:57:12	DNSNode	kernel:	[941920.473114]	13542 pages shared
Mar 21	14:57:12	DNSNode	kernel:	[941920.473117]	498252 pages non-shared
Mar 21	14:59:39	DNSNode	kernel:	[942069.760159]	ratelimit: 12
c all b a	cks suppr	essed			
Mar 21	14:59:39	DNSNode	kernel:	[942069.760167]	java invoked oom-killer:
gfp_mask=0x280da, order=0, oom_adj=0					
Mar 21	14:59:39	DNSNode	kernel :	[942069.760175]	java cpuset=/
mems_a	allowed=0				-

Listing 7.3: Syslog file of a single node in the cluster when heavily loaded

#### 7.5.2 Single node with one client connected

The second set of benchmark tests involved connecting one thin client to one of the nodes in the cluster and running the same Sikuli tests with Firefox and OpenOffice.org. The times measured in this scenario were extremely close to the times measured when treating the single node as a thick client - as one would expect. The average time taken to run the Sikuli Firefox test was 39.27 seconds with a standard deviation of 1.31 seconds (3.34%). While the average time taken to run the Sikuli OpenOffice.org test was 57.20 seconds with a standard deviation of 0.29 seconds (0.507%). Thus the ssh overhead to export the image to the client together with NDB do not add significant overhead when serving just one client.

#### 7.5.3 Single node, heavily loaded

In this scenario, I was experimenting to see how many clients one cluster node could reliably support (running the same Sikuli tests with both Firefox and OpenOffice.org). I found that it the answer was 6 (for the Pentium 4 hardware with 2 GB of RAM), and so the test was effectively run 120 times on the individual cluster node. These were then repeated in OpenOffice.org (having started with Firefox). From the results, it seemed that memory is a bottle neck to thin client desktop computing – I found I ran out of memory while testing as a general error. For example, see Listing 7.3 depicting a segment of the log files when trying to overload a single server within the cluster.

As I added more user sessions, thus increasing the number of instances of the user simulations, the operating system eventually started killing the additional processes as the cluster node had run out of memory (RAM) that it could address. This generally happened at 8 or more simultaneous users per node. However, with the additional load added by the Java overheads 7 users were also not able to run reliably; the tests would run, but Sikuli would be attempting to complete the next event in the sequence before the application had fully completed the last event, which would crash Sikuli . However, with real users (and therefore no need for Sikuli) I suspect that each machine (Pentium 4 node with 2 GB of RAM) could probably support 7 users, and possibly even 8 (as the extra Java run time environment would not be loaded). In total, I expect that the four server cluster could support a maximum 24 - 32 clients (when using the Pentium 4 processors with 2 GB of RAM each).

#### 7.5.4 Pentium 4 cluster with 20 clients spread across them (DNS Round Robin)

In this scenario I was investigating the usability of the system when a cluster of four nodes with Pentium 4 CPUs and 2 GB of RAM each was supporting 20 evenly distributed clients. Each node had 5 clients connected (less than the maximum limit of 6 clients to reliably run the Sikuli tests). The average time measured to run the Sikuli Firefox test across the four nodes was 62.81 seconds with a standard deviation of 3.63 seconds (5.87%). While the average time measured to run the Sikuli OpenOffice.org tests across the four nodes was 82.14 seconds with a standard deviation of 5.67 seconds (6.74%). The difference between one client per node and 5 clients per node is substantially greater when you compare the results in Table 7.2. This difference is, however, to be expected as a single node is doing considerably more work supporting five thin clients rather than just one. The difference between the two in the case of Firefox is 23.90 seconds which represents a 61.42% increase in time to complete a single iteration of the Firefox test, while the difference between the two for the OpenOffice.org test is 24.82 seconds which represents a 43.30% increase in the time taken to complete a single iteration of the OpenOffice.org test. It is important to note that these tests compress "normal" user interactions and that real life use of the software by users in real world environments will probably take longer than the approximate 1 minute interactions timed here – as a result of web page loading times and users interactions being less mechanical/methodical than simulated interactions. As a result, I believe that even the increased times recorded on the heavily loaded nodes of the cluster still represent performances which would be acceptable for everyday use.

Application tested	Thick client scenario (average time in seconds)	Single thin client per cluster unit (average time in seconds)	20 thin clients uniformly distributed across a 4 node cluster (average time in seconds)
Sikuli Firefox test	39.99 (std dev: 1.13)	39.27 (std dev: 1.31)	62.81 (std dev: 3.63)
Sikuli OpenOffice.org Writer test	$57.76 \text{ (std dev:} 0.2 \times 10^{-5}\text{)}$	57.20 (std dev: 0.29)	82.14 (std dev: 5.67)

Table 7.2: Timing results of Firefox and OpenOffice.org across the various hardware configurations

#### 7.5.5 Quad core single LTSP server with 12-14 clients

In order to compare the results of the clustered LTSP server, I performed the same user simulation tests (using Sikuli) on a Quad-core single server with 4 GB of RAM (typical bottom end hardware of a thin client server that could be purchased new at the time). The quad-core server, with its faster CPUs was able to complete the test iterations significantly faster than the cluster server did (compare Table 7.3 with Table 7.2). The Firefox test ran 46.12% faster on the quad core while the OpenOffice.org ran 35.26% faster on the quad core when compared with multiple clients supported on the cluster server. However, the quad core server ran out of memory rather quickly and was only able to support a maximum of 14 clients running the Firefox test and 12 clients running the OpenOffice.org test (approximately 50% of the number of clients compared with the cluster solution – to be expected when in total the cluster can address double the RAM, 8 GB vs 4 GB).

Application tested	Time taken on Quad core server - 4 GB test (average time in seconds)		
Sikuli Firefox test	33.84 (std dev: 0.53)		
Sikuli OpenOffice.org test	53.17 (std dev: 0.44)		

Table 7.3: Timing values of Quad core single LTSP server

Thus the DNS Round Robin cluster server effectively facilities addressing more than 4 GB of RAM (the 32-bit limit) without having to install 64-bit operating systems, allowing older computers to easily be used while still increasing the amount of RAM available to thin clients. Thus it is more important to have more memory to support a greater number of LTSP clients than it is to have very fast processors. Supporting more clients (and simultaneous users) is of greater value than servicing their requests at a faster rate when providing general desktop computing services as long as the user experience is not negatively impacted.

#### 7.6 Cluster server laboratory testing

One practical disadvantage of using older desktop computers as a cluster server is that they are not rack mountable and are limited in both their memory and CPU capabilities. So I performed some follow up tests using Supermicro servers that had dual core atom processors and 4 GB of memory each (using 64-bit operating systems). In addition, I investigated a standard single central server configuration (Quad-core with 8 GB of RAM) and a fat/hybrid client server configuration (also Quad core with 8 GB of RAM) in order to compare the results with the cluster server when supporting multiple clients. Twenty-one clients were used during this second phase of testing. The results of the tests are described below.

#### 7.6.1 Standard Thin Client Configuration

In the standard thin client configuration, I had 21 clients connected to a single Quad-core server with 8 GB of RAM. I ran the two Sikuli tests, Firefox and LibreOffice (the OpenOffice.org free software fork), 50 times per client and recorded the results. I found that on average the Firefox test ran for 28.72 seconds with a standard deviation of 1.42 seconds (4.94%) when we removed the "dirty" data that was recorded as a result of some of the clients crashing (running out of memory on the central server). If we consider the whole data set then the Firefox test ran for 1:03.74 minutes with a standard deviation of 01:51.13 minutes or 174.34%. These results once again highlight the need for memory when serving thin client computers and highlight the high demand placed on memory resources in a single central server in this configuration. Figure 7.4 depicts the Sikuli Firefox graphs for (a) the "cleaned" data set reflecting times without the crashed clients, while (b) shows the full data set including the high times recorded from the crashed clients. The LibreOffice tests ran on average on the thin clients for 15.09 seconds with a standard deviation of 0.46 seconds (3.04%).







Figure 7.4: Sikuli Firefox test results on a standard thin client configuration with single central server (a) "Cleaned" data set (b) Full data set including times of crashed clients

If we consider the load on the CPU and the amount of Free memory available on the system for the standard thin client image (both of which can be seen in Figure 7.5, (a) CPU utilization and (b) Free Memory), we can see that during the tests the CPU was heavily utilized, recording a 100% load. The amount of memory available during the tests was around 2 GB, meaning that the other 6 GB were being

used by the thin clients to run the Sikuli tests. As soon as those tests were done the CPU utilization dropped to zero while the amount of free memory went back up to nearly 7 GB.



Figure 7.5: Hardware resources utilization on the standard thin client server (a) CPU utilization (b) Free Memory available

#### 7.6.2 Fat Client Configuration

In the fat client tests, I connected 21 Fat clients to a single central Quad-core server with 8 GB of RAM. The individual thin clients were also quad-core processors with 8 GB of RAM. The same Firefox and LibreOffice Sikuli tests were run on the fat clients, each test run 50 times per client. The average time of the Firefox Sikuli tests on the fat clients was 25.95 seconds with a standard deviation of 1.20 seconds (4.62%). The average time recorded for the LibreOffice Sikuli tests on the fat clients was 14.15 seconds with a standard deviation of 0.58 seconds (4.09%), when considering the data that did not include the crashed clients. If we consider the full data set of the LibreOffice tests then the average time was 31.17 seconds with a standard deviation of 08:39.48 minutes (1666.46%). Figure 7.6 depicts the results of the LibreOffice tests on the fat clients, (a) the "cleaned" data set and (b) the full data set.



(a)



Figure 7.6: Sikuli LibreOffice test results on a Fat client configuration with a single central server (a) "Cleaned" data set (b) Full data set including the times of the crashed clients

If we consider the load on the CPU and the amount of free memory available on the central server for the Fat client configuration, we can see that there is almost no load on the CPU of the central server (see Figure 7.7 (a) ) and almost all the memory on the system is available (nearly 7 GB available) (see Figure 7.7 (b) ). This is not surprising as the central server acts only as a file server, TFTP server and DHCP server

for the clients. All the processing in terms of running the applications and hence the Sikuli scripts is done on the individual clients.



Figure 7.7: Hardware resource utilization on the central server serving in the fat client configuration (a) CPU utilization (b) Free Memory available

If we then consider the hardware utilization of the CPU and free memory on one of the fat/hybrid clients, we see a much greater load placed on both resources as the Sikuli tests are being run on the hardware of the fat/hybrid client directly. From Figure 7.8 we can see that in (a) the CPU utilization on the client was less than 25%, while in (b) the amount of free memory available during testing was around 6 GB. Thus even when running the Sikuli tests the hardware of the fat/hybrid clients was not overloaded. This is also to be expected as each client computer is only running one instance of each of the tests while the single standard thin client server was running 21 instances of the tests - one for every client connected. As soon as the tests were completed the load on both resources dropped back down to almost nothing.



Figure 7.8: Hardware utilization on single fat/hybrid client image (a) CPU utilization (b) Free memory available

#### 7.6.3 DNS Round Robin (Dual core atom cluster)

In this round of testing I had the clients connected to and distributed across the LTSP server cluster, using DNS Round Robin. DNSNode had 5 clients connected, GenNode1 also had 5 clients connected, GenNode2 had 4 clients connected and lastly, GenNode3 had 7 clients connected. Each of the clients ran the same Sikuli Firefox and Sikuli LibreOffice tests, 50 times each and recorded the times of those interactions. Overall the average time taken for Sikuli Firefox test was 01:00.82 minutes with a standard deviation of 18.54 seconds (30.48%). While the Sikuli LibreOffice overall average time taken was 54.04 seconds with a standard deviation of 16.20 seconds (29.98%). The average times for each of the tests on each cluster node as listed in Table 7.4.

Test	DNSNode	GenNode1	GenNode2	GenNode3
Sikuli Firefox	49.76 seconds (std	49.82 seconds (std	44.93 seconds (std	01:25.67 minutes
	dev: 2.60 seconds;	dev: 2.94 seconds;	dev: 3.33 seconds;	(std dev: 8.83
	5.22%)	5.90%)	7.40%)	seconds; 10.31%)
Sikuli LibreOffice	44.12 seconds (std	43.84 seconds (std	40.85 seconds (std	01:15.97 minutes
	dev: 2.61 seconds;	dev: 2.82 seconds;	dev: 3.76 seconds;	(std dev: 6.60
	5.91%)	6.43%)	9.20%)	seconds; 8.69%)

Table 7.4: Sikuli Firefox and Sikuli LibreOffice tests average times and standard deviation per cluster node

From the results listed in Table 7.4 we can see that the times for DNSNode and GenNode1 are very close; as to be expected as they both had the same number of clients connected. While the times recorded for GenNode2 are the fastest, again this is to be expected as it had the fewest nodes connected to it. Finally, GenNode3 with the most connected clients had the highest recorded times for the Sikuli user interaction simulation tests.

If we look at the hardware utilization of the servers in the cluster, we can see that the CPU and memory utilization of the file server, LDAP server and DHCP server are very low (Figure 7.9). This is to be expected as the bulk of the work, namely running the applications (and hence the Sikuli tests) are done on the LTSP servers in the cluster. Figure 7.9 depicts the low load placed on the file server, while Figures 7.10, 7.11, 7.12, 7.13 show the load placed on the four LTSP servers. From these graphs we see near identical patterns for the first two LTSP nodes (DNSNode and GenNode1) in Figures 7.10 and 7.11. This is to be expected as both nodes were serving the same number of clients, namely five. While the third cluster node, GenNode2 (Figure 7.12) had the least demand on resources as it was serving the fewest number of clients and the last node, GenNode3 (Figure 7.13) had the highest demand on resources as it was serving seven clients did not over tax the dual core atom processors of the server, but that rather the resource most heavily utilized was the memory and at times during testing there was less than 800 MB of memory available on the system. This again highlights the memory bottle neck in LTSP computing.



Figure 7.9: Hardware utilization on the cluster's File, LDAP and DHCP server (a) CPU utilization (b) Free memory available



Figure 7.10: Hardware utilization on the DNSNode of the LTSP cluster server (a) CPU utilization (b) Free memory available



Figure 7.11: Hardware utilization on GenNode1 of the LTSP cluster server (a) CPU utilization (b) Free memory available



Figure 7.12: Hardware utilization on the GenNode2 of the LTSP cluster server (a) CPU utilization (b) Free memory available



Figure 7.13: Hardware utilization on the GenNode3 of the LTSP cluster server (a) CPU utilization (b) Free memory available

#### 7.7 Discussion

There are numerous methods for providing computer facilities in schools. Each of the above solutions have numerous pros and cons.

Standard thick clients require the greatest effort in terms of maintenance and capital investment. Furthermore, without a central server, should a client fail all data stored on the client will be lost. In addition, without a central server all data is stored locally on individual computers meaning that the user's data and profile does not follow them from machine to machine but is specific to the last computer that they were working on.

Thin clients help to eliminate some of the short comings of thick clients. They are cheaper to acquire as they cost less than thick clients and older, second-hand computers can be used as thin clients. Furthermore, all the data is stored on the central server so should a client fail then no data is lost. In addition, the user's data and profile follow the user irrespective of which of the thin clients the user makes use of. At the same time, the single central server represents a single point of failure: should the server fail then all the thin client computers are unusable and all data is lost. In addition, the single central server needs to be relatively powerful and requires a minimum of 8 GB of RAM in order to support 20 simultaneous thin clients and the software running on it. Indeed, my tests have shown that memory is typically a bottle neck in LTSP thin client desktop computing. That said, increased memory and CPU requirements will result in a more expensive server.

The fat/hybrid clients, however, require a less expensive central server as the clients are able to make use of their own hardware resources to run the user's software. This results in a much better utilization of all the hardware resources available within the computing infrastructure configuration. All the while, retaining the major advantages of thin client computing, namely, that the clients fetch all their software and files from a server, reducing the overall maintenance of the configuration.

In this chapter I presented four cluster server options and one grid-based solution. The first two cluster options, the DNS Round Robin and the Ubuntu LTSP Cluster, as well as the grid-based solution of the fat/hybrid clients are the simplest options to implement. However, the Ubuntu LTSP Cluster has flaws in terms of how it distributes the client load across the LTSP servers, making it less desirable than that of the DNS Round Robin cluster. The third cluster option, the SSI, was the most complicated to implement but

if done correctly would allow for one of the most efficient use of hardware resources of all the solutions. However, SSI requires that processes can be easily migrated from one processor to another, which is not the case when using typical desktop software such as Firefox. The last of the clustering methods investigated, Virtual desktops, is probably the most expensive as it requires the most hardware resources to implement, specifically in terms of memory access, making it prohibitive in a developmental context.

#### 7.8 Conclusion

This chapter presented the experimentation work undertaken with various clustering and grid techniques to investigate the possibility and feasibility of using cluster LTSP servers or grids in schools in order to provide significantly more memory and processing capacity for utilization by the thin clients, while retaining the advantages of thin client computing (ease of maintenance, central storage of software and files and low costs). This chapter began by describing the current computing infrastructure configuration and model utilized in the SLL, highlighting the problems with it with a view to addressing those issues, namely improving the ease with which the facilities can be healed and the need for more computing resources. The chapter then described four clustering and one grid-like method that could be employed in the SLL to attempt to address the current shortcomings. Next, the Sikuli environment and tests used to simulate user interactions with a Linux system were introduced. These tests were used to experiment with some of the computing infrastructure configurations, including one of the clustering methods (DNS Round Robin) and the grid-based method (Edubuntu (LTSP) Fat/Hybrid clients), in order to provide a clearer understanding of the nature of LTSP thin client computing and for comparison with a standard thin client configuration. The results showed that memory is a bottle neck in LTSP thin client computing and that clustering of the LTSP servers can provide the necessary computing resource requirements in order to server a growing number of thin clients. Furthermore, utilization of LTSP fat/hybrid clients, instead of standard thin clients, further reduces the load on the servers, potentially increasing the life span of the servers and decreasing rate of failure. The next chapter presents the deployment of some of the solutions tested to a selection of SLL schools in order to ascertain their performance in a real world environment. From these deployments, together with the findings from the laboratory tests and the SLL school survey, a revised computing infrastructure model for the SLL is presented.

## **Chapter 8**

# **Computing infrastructure deployment:** evolved model

#### 8.1 Introduction

Robson defines case study methodology as "a strategy for doing research which involves an empirical investigation of a particular contemporary phenomenon in its real life context using multiple sources of evidence" [62, p52]. In this chapter I discuss three case studies conducted at three schools, FH5, FH1 and FH2, in the SLL. At each of the schools a different computing infrastructure configuration (all of which were described, tested and discussed in Chapter 7) was deployed. At FH5 I deployed the DNS Round Robin cluster, while at FH1 I deployed the Edubuntu Fat/Hybrid client configuration. Lastly, at FH2 I deployed a standard thin client configuration for comparison with the two alternative configurations. The two alternative configurations were chosen as they showed the most promise in the laboratory tests discussed in Chapter 7. After the case studies have been described the chapter then presents a detailed discussion in which the lessons learned are highlighted and as well as key results from Chapters 6 and 7. This is followed by the proposal of a revised computing infrastructure model for the SLL.

#### 8.2 DNS Round Robin deployment to FH5

In this section I present the case study undertaken at FH5, where the DNS round robin LTSP cluster was deployed and tested. The section is broken down into two subsections: firstly I describe the DNS round robin LTSP cluster and computing facilities configuration at FH5, focusing on the computing and networking infrastructure. This is followed by a report on the hardware resources performance results of the cluster as logged by the software MRTG.

#### 8.2.1 Configuration

The configuration of the facilities at FH5 can be broken down into two main sections, namely, the computing infrastructure (located in the computer lab, but could have been deployed to any venue within the school) and the networking infrastructure.

#### 8.2.1.1 The computing infrastructure

The computing infrastructure at FH5 consists of a DNS round robin LTSP cluster server together with the thin client computer desktops. There are five Edubuntu LTSP servers (nodes) in the lab at FH5 appearing as

though they are one large Edubuntu server, but with no co-ordination between the individual LTSP servers. Each of the five servers is numbered and each has a specific set of tasks or services that they offer.

- The first is the NFS File server and LDAP server. This machine is responsible for storing all the users files and folders as well as authenticating user accounts at login.
- The second is the DHCP server, also providing TFTP and LTSP services to the clients supporting them booting and then allowing them to log in and run applications.
- The third, fourth and fifth all offer TFTP and LTSP services, again supporting booting and running applications after login.

Servers 2-5 all mount the home directory space of the users stored on Node1 (the NFS and LDAP server) in order for the user's files to always be accessible. What this configuration supports over a standard, single Edubuntu LTSP server is that should one of the nodes 2-5 fail for whatever reason, the remaining nodes (servers) can provide the TFTP and LTSP services to the clients. Should one of the nodes 2-5 crash during use, the client machine only needs to be rebooted, but if one of nodes fails prior to a client connection, the clients will simply not connect to the faulty node. This configuration improves the overall reliability of the system, but does not completely eliminate the problem of a single point of failure. There are two problems, namely, that a File/LDAP server failure (Node1) or a DHCP server failure (Node2) remain single points of failure. Configuring another of the nodes as a secondary DHCP server will fix the single point of failure for that service, but in order to prevent Nodel being a single point of failure we would need a back-up file server, or some other solution (more of which will be discussed later in section 8.5). That said it is worth noting that the file server (Node1), as a result of the reduced load (its just serving files) has, in theory, a reduced chance of failure. In addition I implemented RAID on the hard disks, so should one of the disks fail data stored on the file server was less likely to be lost. In addition, the cluster made more resources available to the clients – there was more RAM and processing capacity available via the cluster at a cheaper price than via a single server of a similar capacity. The file server at FH5 (Node1) is a Quad core desktop computer with 2 x 500 GB RAID'ed disks and 4 GB of RAM. Each of the other servers, (nodes 2-5) are Proline supermicro 1U Dual-core Atom processors with 2 x 320 GB disks and 4 GB of RAM each. All five nodes run Edubuntu 12.04 64-bit edition. Node1 binds the IP 192.168.2.2/24, while the rest of the cluster (nodes 2-5) bind the IPs 192.168.2.4/24, 192.168.2.5/24, 192.168.2.6/24 and 192.168.2.7/24. The DNS services on the network will translate each of the IP addresses of nodes 2-5 as the hostname, Itspserver.local.org; so a client connecting to Itspserver.local.org could receive anyone of the IP addresses of any one of the cluster nodes 2-5.

FH5 has a computer lab of 20 thin client desktop computers. The thin clients boot off the DNS round robin LTSP cluster servers. This means that after receiving their IP information from the DHCP service, they TFTP a kernel and an init RAM image from one of the LTSP servers which they then load into their memory and boot. Once booted, users are presented with a login screen. Users can log in with any of the generic user account or a personal account if one has been created (as as has been for the teaching staff). At login the users are authenticated against the LDAP service running on Node1, and are logged into the LTSP server that they happen to be connected to (the round robin nature of the cluster configuration means that the clients will be relatively evenly distributed across the available LTSP servers). Once logged in, they can access stored files, which are stored on Node1 and mounted by the LTSP server that they are connected to, and can run the numerous applications available in Edubuntu that run directly on the LTSP server. The LTSP cluster is connected to the core router at FH5 in order to be able to communicate with other computers/services at other schools in the broadband island or to reach the Internet via the VSAT connection at the school. Furthermore, a proxy service was provided via the core network router (a


Figure 8.1: Computing and networking infrastructure configuration at FH5

FreeBSD 9.3 server) at FH5, in order to facilitate the sharing of the Internet connection across the multiple clients while helping to lessen the load on the Internet link by caching pages so that they don't have to be retrieved as often. All the internal clients at FH5 as well as the client machines at the remote schools which connect to FH5 utilize the proxy service (implemented using Squid [287]) at FH5 for web related traffic. The configuration of the computing and networking infrastructure available at FH5 is represented graphically in Figure 8.1.

### 8.2.1.2 The networking infrastructure

FH5 is one of the schools that houses a VSAT connection for use by the SLL broadband island; satellite schools connect to FH5 in order to reach the VSAT link and the Internet. In order to share out the Internet access, FH5 also hosts a WiMAX base station which is used by the incoming remote schools to connect to FH5. Furthermore, there is also a link from FH5 to FH1 (where another VSAT unit is hosted) in order to support redundancy of the Internet access; should one of the VSAT units fail then all traffic that usually connects to FH5 for Internet access can be rerouted to FH1 for Internet access and vice versa.

The VSAT unit is provided by the service provider Vox/Yahclick and provides a 10M down-link and 2M up-link with a 30G maximum download limit in a month. The VSAT link is terminated by the VSAT router and from there connects directly to the core router at FH5, a supermicro 1U server running FreeBSD 9.3 which is responsible for routing all traffic both internally and externally within the network at FH5 and the larger broadband island. Also attached to the core router is the WiMAX base station which terminates all incoming WiMAX connections from the remote schools that connect to FH5. Traffic generated at the remote schools is either routed to the Internet (via the VSAT) or to another school via a WiMAX link. The incoming VSAT connection from Vox/Yahclick has a real world IP address associated with it, through port

forwarding implemented by the service provider. As a result, I am able to ssh to the core router at FH5 and from there to the other servers within the broadband island in order to effect maintenance when needed.

There is also an internal LAN present at FH5; there are physical cat5 Ethernet connections within the computer lab room connecting all the thin clients to the servers, Internet access and WiMAX network. Furthermore, there is a WiFi network deployed at FH5 in order to support wireless connections to the network and its services from mobile devices such as mobile phones, tablet PCs, and laptop computers. The configuration of the computing and networking infrastructure available at FH5 is represented graphically in Figure 8.1.

### 8.2.2 MRTG results

In order to assess the performance of the cluster hardware I used MRTG [288] to monitor the load on various hardware resources within the cluster, namely, the CPUs, network interfaces, memory, disk utilization and the number of processes running at any given time on each node. MRTG is based on a Perl script which employs the use of SNMP (Simple Network Management Protocol) to monitor any SNMP variable of your choice [288]. Counters are used to create log files of data which are then used to create graphs which represent the values of the variable being monitored (CPU, network, memory, etc.). Over time the graphs are embedded into a web page that can then be viewed from any web browser. MRTG reflects the data in a number of views: daily, weekly, monthly and yearly [288]. This is possible because MRTG keeps logs of all the data it obtains about the variables it is monitoring via SNMP for up to two years. The log is automatically consolidated so that it doesn't grow excessively over time, but still contains all data for all the variables monitored over the last two years [288]. However, the consolidation means that you do loose the fine grain data and rather are left with the average of the average reading per day, and the average of the maximum reading per day, for both inbound and outbound data/traffic per SNMP variable.

As a result of only keeping two years worth of data, and that it is consolidated over time, I only have data from November 2014 to January 2017 (when this section of the dissertation was written). Furthermore, because of the consolidation of the latter data, using the graphs generated by MRTG didn't reflect some of detail. Thus, I imported the raw MRTG log files into Excel and created graphs of the data for the year 2015 personally, using spreadsheet software, in order to be able to get a better representation of one year of data in order to ascertain the performance of the hardware in the cluster at FH5. Figures 8.2, 8.3, 8.4, 8.5, and 8.6 reflect the MRTG logs of the CPU, Network interface, Free Memory and number of processes per node for the year 2015.

From the graphs in the Figures 8.2, 8.3, 8.4, 8.5, and 8.6 it can be seen that separating the LTSP service from that of the file sharing services significantly reduces the load on the file serving node (Node1). The load on the CPU of Node1 is noticeably less than that experienced on the four LTSP nodes (however, it must also be noted that the CPU of Node1 exceeded the capabilities of the CPUs of Nodes 2-5). Memory utilization on Node1 was comparable to those of the other nodes; naturally the work of the file server (Node1) is more memory intensive than processor intensive. There is also a lot of traffic on the network interfaces as data (files and folders) and software images are served to the clients over ssh tunnels (and NFS in the case of Node1). Incoming network traffic was typically less than outgoing, as the incoming data from the thin clients usually amounts to mouse and keyboard events from the clients to the various nodes.

During the course of the experimentation I experienced a lot of problems with overall reliability of some of the cluster nodes, specifically Node4 and Node5. As such you can see in Figure 8.7 that typically the CPU and Memory loads on nodes 4 and 5 were less than those on nodes 1-3. This is because typically Node4 and Node5 were not on or not booting properly. The biggest problem faced with regards to computer (both server and desktop) hardware in the SLL is that users do not understand that computers are not like a home electric appliance, such as a toaster or a kettle, where you can just switch the device off at the wall





165



Figure 8.3: MRTG data for Node2 reflecting the CPU load, Network traffic, Free Memory (load on the memory) and the number of processes on Node2



Figure 8.4: MRTG data for Node3 reflecting the CPU load, Network traffic, Free Memory (load on the memory) and the number of processes on Node3

Figure 8.5: memory) and the number of processes on Node4 MRTG data for Node4 reflecting the CPU load, Network traffic, Free Memory (load on the





Figure 8.6: MRTG data for Node5 reflecting the CPU load, Network traffic, Free Memory (load on the memory) and the number of processes on Node5

(i.e. instantly removing electricity in order to turn it off), despite explaining that to users. The operating systems of the computers need to be cleanly shut down. Furthermore, electricity supply to the Dwesa-area is very erratic and often schools in the SLL have long periods without power supply without warning or explanation. Even with the use of UPS equipment we still struggle because the batteries of the UPS are under large strain/load and don't last as long as one might hope. Finally, the users at FH5 also struggled to understand the concept of the cluster and that, if possible, they should turn all the servers on; often they would just turn Node1 (the file server) on, and maybe one other LTSP server (typically Node2). I eventually implemented a script to power up the full cluster by scripting Wake-On-LAN commands to be sent to the cluster nodes from Node1. However, due to the power interruptions to the hardware and the associated impact on the hardware and operating systems of the nodes, despite receiving Wake-On-LAN commands, not all were able to boot up properly. That said, we can see that despite the issues with some of the nodes on the cluster, the remainder did run and were able to serve clients, providing a more reliable service to the users as compared to a single server when it is unable to boot up.

### 8.3 Edubuntu Fat/Hybrid client deployment to FH1

In this section I present the case study undertaken at FH1, where the Edubuntu Fat/Hybrid clients were deployed and tested. The section is broken down into two subsections: firstly I describe the Edubuntu Fat/Hybrid clients and computing facilities configuration at FH1; focusing on the computing and then networking infrastructure. This is followed by a report on the hardware performance results of the hybrid client server as logged by the MRTG software.

### 8.3.1 Configuration

The configuration of the facilities at FH1 can also be broken down into two main sections, namely, the computing infrastructure (located in the computer lab, but could have been deployed to any venue within the school) and the networking infrastructure.

### 8.3.1.1 The computing infrastructure

The computing infrastructure at FH1 consists of a single, central file server together with LTSP Fat/Hybrid clients. The central server provides DHCP, TFTP and file sharing services. The LTSP fat/hybrid clients use NBD in order to access the client's root file system (from the server) as it is faster than using NFS, while the home directory of a user is mounted (to the fat/hybrid client) using sshfs [175]. More details about the fat/hybrid clients were discussed in section 5.2.1.3. The server hardware at FH1 is Mecer, with 4 GB of RAM, a Quad-core Intel Xeon 2.4 GHz CPU, and two 500 GB hard-disks. The server is responsible for responding to DHCP requests from the clients and providing the fat/hybrid client images to boot via TFTP (these are larger than a typical thin client image as they contain the complete operating system and all the desired additional third party applications/software (such as Firefox and LibreOffice)) [175]. Once the clients are booted, the server provides access to the user home directories (which are mounted by the fat/hybrid clients when the users log in) and user authentication. As such the load on the server after the clients have completed booting is typically limited to authentication and serving the files and folders of logged in users.

The computer lab at FH1 has 26 fat/hybrid clients which either boot off the central server (by default) or their locally installed copies of Windows 7 (from their hard drives, which are booted in cases when the server does not respond to their DHCP and TFTP requests). Once booted (off the server) the clients present a login screen to the user (which authenticates from the server) and users can log in and access their



Figure 8.7: MRTG data of (a) CPU loads and (b) Free Memory on the Five Cluster Nodes



Figure 8.8: Computing and networking infrastructure configuration at FH1

files, folders and the applications that have been built into the fat/hybrid client image. There are also 60 generic users configured on the server at FH1. Similarly to FH5, these range from tux1 up to tux60 and the username and the password are the same. The generic user accounts are also standard desktop users and are not able to perform any maintenance or updates of the software. The configuration of the computing and networking infrastructure available at FH1 is represented graphically in Figure 8.8.

### 8.3.1.2 The networking infrastructure

FH1 is another one of the schools that houses a VSAT connection for use by the SLL broadband island; satellite schools connect to FH1 in order to reach the VSAT link and the Internet. In order to share out the Internet access, FH1 also hosts a WiMAX base station which is used by the incoming remote schools to connect to FH1. Furthermore, there is also a link from FH1 to FH5 (where another VSAT unit is hosted) in order to support redundancy of the Internet access; should one of the VSAT units fail then all traffic that usually connects to FH1 for Internet access can be rerouted to FH5 for Internet access and vice versa.

The VSAT unit is provided by the service provider Vox/Yahclick and provides a 10 Mbps downlink and 2 Mbps uplink with a 30 GB cap. The VSAT unit comes with both a VSAT dish and a small formfactor VSAT router/modem, both of which are the property of Vox/YahClick. The VSAT link is terminated by the VSAT router and from there connects directly to the core router at FH5; a supermicro 1U server running FreeBSD 9.3 which is responsible for routing all traffic both internally and externally within the network at FH1 and the larger broadband island. Also attached to the core router is the WiMAX base station which terminates all incoming WiMAX connections from the remote schools that connect to FH1. Traffic generated at the remote schools is either routed to the Internet (via the VSAT) or to another school via a WiMAX link. The core router at FH1 also provides a proxy service (also implemented using Squid [287]) to all the internal clients at FH1 as well as the clients at each of the remote schools that connect to FH1, in order to facilitate the sharing of the Internet connection across the multiple clients while helping to lessen the load on the link by caching pages so that they don't have to be retrieved as often. The incoming VSAT connection from Vox/Yahclick has a real world IP address associated with it, through port forwarding implemented by the service provider. As a result, I am able to ssh to the core router at FH1 and from there to the other servers within the broadband island in order to effect maintenance when needed.

There is also an internal LAN present at FH1; there are physical cat5 Ethernet connections within the computer lab room connecting all the fat/hybrid clients to the server, Internet access and WiMAX network. Furthermore, there is a WiFi network deployed at FH1 in order to support wireless connections to the network and its services from mobile devices such as mobile phones, tablet PCs, and laptop computers. The configuration of the computing and networking infrastructure available at FH1 is represented graphically in Figure 8.8.

### 8.3.2 MRTG results

In order to assess the performance of the server hardware when servicing Fat/Hybrid LTSP clients I once again employed the use of MRTG [288] to monitor the load on various hardware resources within the server, namely, the CPU, network interface, memory, disk utilization and the number of processes running at any given time.

Similarly to the deployment at FH5, I have two years worth of data collected, going back to November of 2014. For the sake of comparison, I will also discuss the results of the year 2015 for FH1. Again, because MRTG consolidates the data over time and only draws graphs for the last year, I imported the raw readings of the SNMP variables into a spreadsheet and only drew graphs for the year of 2015. Furthermore, I again am only reporting on the results for the CPU, memory, network interface and number of processes running (leaving out the hard-disk utilization data). Like FH5, FH1 had access to more than sufficient disk space (500 GB) and only utilized 22 GB of that; again not adding much to the overall discussion, other than to note that access to a couple of hundred gigabytes of hard disk space for permanent storage is more than sufficient, currently, for the schools in the SLL. Naturally, in order to prevent data loss, I used RAID to mirror the hard disks for redundancy and data recovery should one of the hard disks in the server fail. The graphs representing the various loads on the server hardware can be seen in Figure 8.9.

Looking at the graphs in Figure 8.9, it can be seen that, the load on the CPU of servicing 26 Fat/Hybrid clients is not excessively heavy, especially when compared with the original single central server model example presented and discussed in section 3.2.4.3 (reflected in Figure 3.6) of Chapter 3. The CPU graph reflected in Figure 3.6, shows that the CPU often experienced prolonged loads of 100%. However, on the server at FH1, the maximum load in the year 2015, was 89% over a shorter period and only ever happened once. In terms of memory utilization however, it appears that the server at FH1 was under tremendous load in terms of memory use. The server has 4 GB of memory, but over the course of the year MRTG reports that it had very little free memory. I suspect, however, that something was wrong with regards to either how MRTG was accessing the SNMP variables, or perhaps there was some sort of technical software issue that was resulting in a massive amount of memory utilization. If I compare the results of 2015's memory utilization with a graph of the memory utilization of the server over three days at the start of the school year in 2017 (see the graph of the results in Figure 8.10), one can see that the memory utilization of the server is more along the lines of what one would expect; specifically that there is between 3.1 GB and 1.2 GB of free memory on the server (which has a maximum of 4 GB). These results are similar to those seen during the laboratory testing of a server with Fat/Hybrid clients and reported on in section 7.6.2 and Figure 7.7 of Chapter 7.

There was also a lot of activity on the network interface of the server, but that is unsurprising, as the server is still responsible for serving the boot images to the fat/hybrid clients, providing user authentication,







Figure 8.10: Memory utilization on the Fat/Hybrid client server at FH1 over three days at the start of the 2017 school year

and serving all the files and folders of the users (the fat/hybrid clients mount the user's home directory) as well as mounting the client's root directory from the server (as discussed in section 7.3.5 in Chapter 7). The graphs from FH1 reflect that a single server, with just 4 GB of RAM is able to adequately service 26 clients, when those clients are fat/hybrid clients as the load of providing the Desktop as a Service is delegated to the client hardware, freeing the server to act as a file server, DHCP server, and authentication server. It is also interesting to note that during the time that I ran this case study at FH1, they experienced significantly fewer server related errors than had been experienced when a standard thin client configuration single server served thin clients, and in fact, less down time than FH5 experienced during the testing of the DNS round robin case study.

### 8.4 Standard thin client deployment to FH2

In this section I present the case study undertaken at FH2, where standard Edubuntu thin clients and a single central server configuration was deployed and tested. The section is broken down into two subsections (as per the first two case studies): firstly I describe the standard Edubuntu thin client configuration at FH2; focusing on the computing and then networking infrastructure. This is followed by a report of the hardware performance results of the thin client server as logged by the MRTG software.

### 8.4.1 Configuration

The configuration of the facilities at FH2 can also be broken down into two main sections, namely, the computing infrastructure (located in the computer lab, but could have been deployed to any venue within the school); and the networking infrastructure.



Figure 8.11: Computing and networking infrastructure configuration at FH2

### 8.4.1.1 The computing infrastructure

The computing infrastructure at FH2 consists of a single central standard thin client Edubuntu LTSP server, together with 20 thin client computers. The server has a Quad-core Intel Xeon 2.4GHz CPU, together with 16 GB of RAM and 1.4 TB of hard-disk space configured in RAID 5. The thin client server is responsible for DHCP, TFTP (providing the client kernel and initram image), and providing the root file system of each client, user authentication and finally running all user applications and providing access to all user files and folders (once logged in – Desktop as a Service). A full description of how LTSP thin clients boot and work (specifically the Edubuntu implementation thereof) is described in detail in Appendix F.2.

Like the fat/hybrid clients the thin clients also boot from the server, over the network, using PXE [107]. Once the clients have booted off the server they present a login screen to the user (which authenticates from the server) and users can log in and access their files, folders and the applications that are available on the server; the server essentially provides the Desktop as a Service. There are also 60 generic user accounts configured on the server at FH2. Similarly to both FH5 and FH1, these range from tux1 up to tux60 and the username and the password are the same. The generic user accounts are also standard desktop users and are not able to perform any maintenance or updates of the software. The configuration of the computing and networking infrastructure available at FH2 is represented graphically in Figure 8.11.

### 8.4.1.2 The networking infrastructure

The networking infrastructure configuration is similar at FH2 to those of FH5 and FH1, except that FH2 does not have a WiMAX base station; there are currently no remote schools connecting to FH2. It does however also have a VSAT link. The VSAT link at FH2 is provided by SAAB Grintek, one of the industry sponsors in the Telkom Centre of Excellence programme at the University of Fort Hare. The VSAT link at FH2 provides a 2 Mbps downlink from the satellite and a 512 Kbps uplink to the satellite. The link is uncapped; meaning that there is no limit on the amount of content that can be downloaded over the period

of a month. Currently there is no WiMAX base station at FH2, so they have the sole use of the VSAT link at their school. It is planned, however, that in the future another WiMAX base station will be installed at FH2 and new schools, that have line of sight to the school, will be able to join the network and make use of the VSAT link at FH2; in order to further expand the reach of the SLL.

In order to allow FH2 to reach the rest of the SLL network, they are a subscriber node to the WiMAX base station at FH5; FH2 are linked to FH5 via a WiMAX CPE and the WiMAX base station at FH5. Thus traffic bound for the another school in the SLL network exits the FH2 network via its WiMAX CPE, while all other traffic bound for the Internet exits the FH2 network via the VSAT link. The Edubuntu server at FH2 acts as the router between the thin clients and the VSAT network and the thin clients and the WiMAX network. The Edubuntu thin client server at FH2 also provides a proxy service (also implemented using Squid [287]) to all the internal clients at FH2 in order to facilitate the sharing of the Internet connection across the multiple clients while helping to lessen the load on the link by caching pages so that they don't have to be retrieved as often. The incoming VSAT connection from SAAB Grintek also has a real world IP address associated with it; statically configured on the server. As a result, I am able to ssh to the Edubuntu thin client server at FH2 in order to effect maintenance when needed.

There is also a LAN within the school premise at FH2, in order to connect the thin clients to the server, the Internet and the WiMAX network; there are physical cat5 Ethernet connections within the computer lab room connecting all the thin clients to the server, Internet access and WiMAX network. In addition, there is a WiFi AP on the LAN so that wireless devices (laptops, mobile phones and tablet PCs) can access the computing infrastructure resources, such as the Internet, or data stored on the Edubuntu thin client server. The configuration of the networking infrastructure at FH2 is also depicted in Figure 8.11.

### 8.4.2 MRTG Results

In order to assess the performance of the server hardware when servicing Edubuntu LTSP thin clients I once again employed the use of MRTG [288] to monitor the load on various hardware resources within the server, namely, the CPU, network interface, memory, disk utilization and the number of processes running at any given time. FH2 was the last of the case study deployments and as such I do not have as much data about the performance of the server. Thus the results reported here are from this year (2017), and reflect the first few weeks of utilization in the new academic year at the school. I did, however, import the raw readings of the SNMP variables from MRTG into a spreadsheet, as I had done for the other two case studies, and then drew the graphs of the resource utilization using the spreadsheet. These results can be seen in Figure 8.12 below.

From Figure 8.12 we can see that the CPU load peeks at about 130 % (because the server has a quad core CPU, the CPU reading could go up to 400 %), telling us that one of the CPU cores was saturated at 100% and another core was servicing the extra demands. From this one can see just how resource intensive servicing 20 thin clients can be on a single server. The server at FH2 has 16 GB of memory, and from the memory graph in Figure 8.12 we can see that during the course of the first two weeks of the year, the lowest that the free memory dropped was down to just under 12 GB, thus only utilizing 4 GB at the time. With more data over time and clients doing more memory intensive work, such as all 20 clients using Firefox, we might see bigger drops in the amount of free memory. The network traffic graph in Figure 8.12 is as would be expected, there is more outbound traffic from the server than inbound, as the inbound traffic is typically just I/O events, while the outbound traffic contains the graphics for each client. From the graphs in Figure 8.12 it would seem that the high-end server at FH2 with its large pool of hardware resources is more than able to service the 20 thin clients present at the school. However, the single server still represents a possible single point of failure, and the high end hardware is expensive, meaning that having a spare of



Figure 8.12: MRTG data for FH2's server; reflecting the CPU load, Network traffic, Free Memory (load on the memory) and the number of processes running

Resource	Node1	Node2	Node3	Node4	Node5	Single server
Free Memory	2.37456 GB	1.21665 GB	3.07012 GB	3.106259 GB	3.172747 GB	5.728518 GB

Table 8.1: Average maximum Free Memory available on the cluster Nodes (in the year 2015) as well as the average maximum Free Memory available on the single central server (in the first three months of 2013) (all deployed at FH5)

similar capacity would be very expensive for a rural school, especially considering that the server that they have was donated to them.

### 8.5 Discussion: lessons learned from the case studies

This section is separated into three subsections, each of which discuss the findings of the hardware deployments at FH5, FH1 and FH2 as well as the lessons learned and what one can take away from each of the deployments. This section highlights all the necessary considerations that together with the findings from Chapters 6 and 7, will feed into the discussion of revising the computing infrastructure model for the SLL (which is discussed in greater detail in section 8.6).

### 8.5.1 DNS Round Robin solution

### Computing resources and reliability

The purpose of providing a cluster server solution was to attempt to make available more computing resources, especially memory (because for the types of processing being done memory typically was the bottleneck; modern CPUs being typically more than powerful enough for desktop processing needs), to the thin client computers at schools in order to improve the overall experience for the end user, without significantly increasing the overall cost of the server.

In providing the DNS round robin cluster as the "server" to the thin clients, the total memory available to the thin clients increased significantly from approximately 7.5 GB (in a single server with 8GB of RAM) to a shared total of 16 GB (across the four nodes providing LTSP services) as well as another 4 GB in the node serving their files (Node1); effectively more than doubling the overall RAM available to provide the various services required by the thin clients. Furthermore, if one compares the average Free Memory of the single central server (that was discussed in section 3.2.4.3 of Chapter 3 and is reflected in Table 8.1) to the average Free Memory of the cluster nodes (also available in Table 8.1) the total average available (Free) memory to the thin clients via the cluster is nearly double (10.6 GB vs 5.7 GB) that of the available (free) memory on the single server. As such, the cluster is better able to respond to greater demands for memory from the thin client computers when the end users are making use of them.

Furthermore, the LTSP cluster improved the availability and overall reliability of the LTSP service to the thin clients; when Node4 and Node5 were unable to boot up properly due to power issues negatively impacting their operating systems (see the latter days of the CPU and Memory graphs for nodes 4 and 5 in Figure 8.7) the remaining cluster nodes and specifically the LTSP servers (Node2 and Node3) were still able to service the incoming connections and requests from the thin clients. As a result the computing infrastructure at FH5 remained available to users, albeit in a more degraded state, as the service would be slower when serviced by fewer cluster nodes.

### Added complexity

The use of the DNS round robin LTSP cluster at FH5 added a level of complexity to the computing infrastructure that the end users (teachers, learners and even community members) were just not able to adequately navigate. The idea that a number of servers, working together, were providing the functionality of one server, as though it were just one server, confused them. As such, they were never really sure what servers needed to be turned on in order to access the computing resources, despite numerous diagrams and posters in the computer laboratory at their school together with verbal conversations and telephonic support attempting to explain the configuration. This resulted in reports from the school asking for assistance because either "the Internet was not working" or "the clients couldn't boot" or "the clients could not log in". Typically, one of the servers within the cluster, or their core network router had not been turned back on after being switched off at the end of the previous day or over the weekend. Furthermore, the added complexity of the cluster solution made it harder for teachers to self diagnose computing problems and even more so to effect repairs. Arguably however, that line of work should not be in a teacher's portfolio, as their's is the business of teaching, not providing technical support, and a school technician should be present at each school (or one person who works for a group of schools).

In order to reliably provide an LTSP cluster server to a school either more obfuscation needs to happen (placing the cluster in a "black box" so that user interactions are significantly simplified and all layers of complexity are abstracted away from them) or there needs to a well trained school technician in order to provide on-site support to the users. The "black box" solution would remove the end user from the server completely, but would be, naturally, only a temporary solution. It would be a much better solution to have locally trained technicians who remain on-site at the school(s) in order to provide technical support to remove the necessity of teachers needing to fully understand the complexity of the computing infrastructure available, as well as perform maintenance.

### 8.5.2 LTSP Fat/Hybrid client solution

### Hardware utilization

One of the reasons for the need to improve upon the current computing infrastructure model for the SLL was that the single central server didn't have enough hardware capacity to support the growing number of thin clients and their simultaneous interactions, specifically with regards to memory (as a bottle neck) and CPU. The thin client servers would have to ignore requests for Desktop services or application use from clients if they didn't have enough memory with which to process all the incoming requests from the various users and clients (as was seen in Section 7.5.3 of Chapter 7). However, central servers with large memory and CPU capacity are typically more expensive and schools in rural areas of South Africa have significant financial constraints given the history of South Africa and more especially those schools that are located in the former Homelands of South Africa.

An alternative solution to that of the provision of a cluster server is to utilize the hardware of the clients on the network more efficiently, in a grid-like configuration. In the standard thin client model the thin clients are essentially dumb terminals which just send the keyboard and mouse events to the server and display the corresponding changes on the monitor from the server. Thus the only hardware being utilized on the client are the I/O devices, graphics card and the network interface. The CPU and the memory of the client are essentially unused. The fat/hybrid client solution allows for the utilization of these hardware resources on the clients. Clients still fetch all their content (operating systems, root device, user files and folders) from the server, but the operating system of the client, together with all the applications for the user run directly on the client hardware. Thus the server acts as a file server (of various types), DHCP server and an authentication server. This significantly lessens the load on the server and makes better use of all the available hardware resources. Unsurprisingly, the clients in the lab at FH1 were typically faster in their response times and interactions than experienced at both FH5 and FH2 (which both had standard thin clients).

### 8.5.3 Standard Edubuntu thin client solution

### Hardware utilization

The hardware configuration at FH2 is that of the current computing infrastructure model within the SLL, namely, the standard Edubuntu LTSP thin client configuration. As such, the processing for all the applications used by the users, as well as all their data storage is performed by the central server. The thin clients themselves run a very stripped down version of the Ubuntu OS which allows them to send input from the user to the server and display the output from the server on the client's monitor. Thus, the computing resources/hardware available on the client are not fully utilized. The clients at FH2 are HP thin clients; specifically the HP t510 thin client. This thin client has 2 GB of RAM and a VIA Eden X2 U4200 (1 GHz, 2 cores) CPU, neither of which are fully utilized in this standard thin client configuration. This is essentially a waste of the resources, and places additional resource strain on the server (although in their case they have a high end server with lots of CPU and memory resources) in order to service the 20 thin clients that the school has.

# 8.6 Computing infrastructure for rural South African schools: a revised model

The current computing infrastructure in the SLL has been the standard Edubuntu LTSP thin client model, as described in section 3.2.4.3 and section 7.2. In both sections reliability, memory and CPU limitations of the server were highlighted. In order to address these concerns I experimented with different server clustering and grid techniques and through the laboratory testing decided on two to field test in order to compare with the standard solution, re-implemented on high-end hardware.

### The DNS Round Robin LTSP Cluster

The first of the case studies, at FH5, tested the most plausible of the "conventional" server cluster configurations, namely, that of the DNS Round Robin cluster. The DNS round robin cluster retained the use of Edubuntu and open source software, with the financial sustainability of the configuration more readily achievable because lower end servers or refurbished servers or desktop computers can be employed in order to build the server cluster. Furthermore, schools could more easily purchase spare servers or desktops and keep them ready to swap in to replace faulty components within the cluster, or just add them to the cluster in order to add more resources for the thin clients to utilize. As such, the solution also readily lends itself to addressing the hardware limitations experienced with the current thin client computing infrastructure model, as more nodes could be added to the cluster to increase the overall resources available to clients. The cluster configuration is easier to expand, as compared to the thin client solution at FH2, where there is a limit to the amount of memory modules and the processing capacity that the server motherboard can support. Thus, eventually the server would need to be replaced entirely by another server with greater capacity (and probably a greater price tag) in order to continue adding to the growing number of thin clients.

The high availability of the overall system – where even if one or two nodes fail or are not running there are still enough remaining nodes to service clients – should ease the urgency of maintenance. This provides

the technicians with time to assess the faulty node and either make repairs or replacements and then return it into the cluster to continue servicing clients.

There proved to be disadvantages to the DNS round robin cluster solution, however. The added complexity of the configuration of the DNS round robin cluster was never understood by the end users (teachers, learners and community members). This resulted in the users not turning the correct server hardware on in order to boot clients or access the Internet. It also meant that they were not able to clearly report technical faults to the remote technicians (myself included) at either Rhodes or Fort Hare, making remote trouble shooting and maintenance more complicated than was expected. Methods to address these issues would be to either abstract the solution so much that the entire system becomes a "black box" that the users interact very minimally with, or have on-site support through a technician, or increase user knowledge and understanding of technology (a long term goal, which will come over the course of time).

### The LTSP Fat/Hybrid clients configuration

The second case study, at FH1, tested the grid-like configuration of Fat/Hybrid clients; the fat/hybrid clients make use of all the available computing resources in order to provide functioning computing infrastructure to the end user.

The LTSP fat/hybrid client configuration implemented at FH1 retained the use of the Edubuntu operating system, as well as the corresponding open source software. Furthermore, the overall ease of maintenance was retained, in that a technician still has only got one computer to maintain, namely the server. The fat/hybrid clients do not store any data or operating system, so the only maintenance that they require would be hardware based, which either means replacing components or the entire client. Either of these activities does not render the computing infrastructure of the school useless, and normal functioning of all other clients can continue unhindered. Furthermore, the central server needs not be a high end server as the bulk of the computing (or processing) has been handed over to the fat/hybrid clients. All that the server is responsible for it booting the clients (including all associated services there in – see Appendix 7.3.5 for details), performing user authentication and serving root directories, files and folders. This can reduce the overall costs of the computing infrastructure and contributing to the financial sustainability of the facilities.

The results of the different utilization patterns of this configuration when compared with the standard thin client configuration implemented at FH2, resulted in more clients supported (26 vs 20) with less memory and CPU power on the server. The fat/hybrid clients do seem to bring together the best of thin and thick client computing.

### Summary of the findings from the SLL Survey

From the analysis conducted in Chapter 6, I found that teachers were unable to describe what kinds of computers, Internet access, networking technology or computing infrastructure configurations were present at their schools. They were very vocal, however, in discussing the need for technology and access to the Internet, stressing that the computing and networking infrastructure were typically the schools' only access to information sources and noting that learners needed to be ICT proficient to function in the current society. Along with a general call for "more computers" teachers were also specific about the need for laptop computers and data projectors to support the integration of ICTs into teaching and learning practices. Furthermore, teachers stressed the need for reliable hardware solutions, while noting that they had very little access to maintenance, support and funding and were typically reliant on outside sources of help, such as the SLL researchers.

The survey, through the questionnaire, interviews and focus groups, revealed that typically computing infrastructure in the schools were being used for office productivity related tasks and Internet searching;

so typically the office productivity applications and Internet browsers. During the course of the interviews, teachers' acknowledged their lack of knowledge of available software products, but also expressed a clear need for software that supports the South African school curriculum. As discussed in section 4.5, despite the policy documents stipulating the use and integration of ICTs in teaching and learning there is very little in terms of direct guidance to teachers regarding how to integrate technology in the classroom or which technologies best support which classroom activities or pedagogical practices. While the creation of support material of this nature is out of the scope of this dissertation, I do acknowledge its need and importance, and I hope that this work will contribute and feed into future work that will result in its creation.

Despite the acknowledged ICT related training that all the surveyed teachers had previously received they still called for further training for themselves and their colleagues. Teachers' also noted that training would support their understanding of the software that they have access to for the purposes of teaching, as well as better equip them to do rudimentary troubleshooting in order to support off-site maintenance.

In summary, the data generated by the survey revealed that there is a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content) and is also easy to maintain and replace. Arguably, this is the first step in realizing the dream of ICT integration in rural South African schools, even if, in order to fully and appropriately utilize computing infrastructure to support teaching and learning, teachers need to be adequately trained.

### A revised computing infrastructure model for the SLL

Drawing from the findings of the SLL survey, the laboratory experiments and the real world deployment case studies within the SLL, it is my recommendation that the revised computing infrastructure model for the SLL consist of a single Edubuntu LTSP fat/hybrid client server, together with Edubuntu LTSP fat/hybrid clients and a back-up server in order to address the single point of failure. Furthermore, the server (and the back-up) should be configured to offer file sharing services to other non fat/hybrid clients (such as through the use of SAMBA) in order to facilitate users access to their files and folders regardless of the device from which they are operating within the school property. In addition, the server (and the back-up) can be configured to provide access to other services like a proxy service in order to conserve Internet bandwidth, distributing it equitably and exercising some control on content. In terms of routing functionality for the school, I would suggest the use of a Mikrotik router-board, which will more than adequately support the connection of the server and fixed Ethernet clients to the Internet, as well as facilitate the connection of WiFi Access Points to the Mikrotik to support wireless devices in and around the school property accessing the local services on the network (such as the server for serving files) and the Internet access. In the rest of this section, I will elaborate upon my reasoning for my proposed revisions.

We know from the results of the survey conducted with the six SLL schools that teachers emphasized the need of a lot of resources (in other words, lots of computers) in order to accommodate all the learners in a class. Thin client computing would appear at first to be the best choice, because it allows schools to purchase a number of low-end computers or refurbished computers to be thin clients. However, the best way to lessen the load on the server is to rather configure the clients as fat/hybrid clients. As such the need for a powerful server is mitigated, as the memory load and the CPU load is spread across the clients. Naturally, it is critical that the clients are at the correct price/performance point.

A fat/hybrid client configuration also facilitates ease of maintenance, as in the pure thin client model, because there is only one computer to maintain, namely the server. All the clients are still reliant on the server for retrieval of their OS, software, files, folders and authentication. Thus, should they fail, the only repair work to be done is to either repair or replace the hardware.

Finally, in order to provide the reliable infrastructure that was desired by the local teachers, I propose the use of a back-up server. The server will not be active until such time as the primary server fails. The server should be configured to boot once a day, preferably at night when the school infrastructure is not in use, to sync all the hard-disk changes on the primary server (so that as little data as possible might be lost in the event of the primary server failing) as well as update its own operating system so that it remains as up to date as possible. The SLL survey revealed that teachers are not ICT "power users" and so a simple solution as the one just proposed is going to work best. Should the primary server fail, a remote technician could talk a teacher through swapping the failed server for the back-up, and the technician could possibly perform basic remote configuration in order to replace the primary server in functionality. This would then keep the computing infrastructure up and running with minimal downtime and provide time for the technician to effect repair to the primary server, which will then be re-introduced as the new back-up server. Naturally, a fully automated solution is also possible. However, based on the experience accumulated in this thesis I know that it is important that the automatic solution is not "brittle" and does not introduce additional complexity.

It is important to note, that the ideal would be to have teachers who are more proactive and take on greater responsibility for the facilities available and so are able to undertake more maintenance related tasks over time, thus requiring less and less assistance from remote support. One of the tenents of the SLL has been to promote this type of growth amongst the teachers within the SLL, however, with little success. We have had a few very proactive teachers over the ten year period that the SLL has been in operation; two at FH2, one at FH5 (one of the two at FH2 however, after becoming better qualified used this as a opening to leave the rural area of the SLL). This is a sociological problem, rather than technical, and we hope that through continued efforts we will see a mental shift in more teachers in the SLL to becoming more proactive, and more strongly embracing ownership of ICT facilities within their schools.

Teachers also specified the need for educational based software, that supports their curriculum. While Edubuntu is not specifically designed for the South African curriculum, it does contain a significant proportion of the Free and Open Source Software for education within the image. This means that there is a lot to choose from for the teachers; specifically, there are over 45 different education packages available within the standard Edubuntu installation [289]. Unfortunately, as yet, there is no specific list of pre-approved software that supports the South African school curriculum from the Department of Basic Education. So teachers will have to make their own choices about what constitutes appropriate software for their classroom activities. Thus the best that the SLL can do for now is to make available as much of the free and open source education based software as possible to the local SLL teachers; Edubuntu affords that opportunity.

This configuration can support fat/hybrid clients in a computer laboratory facility as well support clients deployed within the classrooms or administration offices, libraries, staffroom, etc – the requisite networking infrastructure would need to be in place (i.e. Ethernet cables just need to be pulled through to the venues where the clients are to be placed). Furthermore, through the use of WiFi networking, mobile infrastructure such as laptop computers, tablet PCs and mobile phones, can be supported in accessing the network and the computing infrastructure resources available at a school as well as the Internet access. Software, like SAMBA, can be installed on the servers to facilitate file sharing to non-fat/hybrid client devices, so that users can always access their files and folders regardless of which types of devices they use. Laptop computers can also be configured to boot as fat/hybrid clients on the network. Thus, this configuration would support laptops with data projectors for use in the classrooms in order to support the integration of ICTs into the traditional classroom – a request highlighted often during the SLL survey.

In addition, the server can be used to store off-line copies of on-line content and resources, such as Wikipedia, Project Gutenberg and the Khan Academy; all of which are on-line resources that are really useful in education. Off-line copies would facilitate the use of the contents even during periods when either the Internet is not accessible and reduce bandwidth consumption.

Computing infrastructure of this nature would allow schools to unlock and access the Internet resources that the South African government proposes to deploy through the implementation of the SA Broadband policy (discussed in section 5.4.1 of Chapter 5); through both fixed clients like the fat/hybrid clients and mobile devices where learners, teachers and community members can be both active contributors (producers) and consumers of knowledge and data on-line.

### 8.7 Conclusion

In this chapter I discussed the three case studies undertaken at three of the six surveyed schools in the SLL, namely, FH1, FH2 and FH5. The case studies investigated two alternative computing infrastructure configurations for comparison with the standard thin client configuration (implemented in one of the three case studies) that had been used as the computing infrastructure deployment model in schools in the SLL up until this point. Each of the case studies describes the computing and networking infrastructure configurations at each of the three schools, and then discusses the performance of the computing infrastructure resources as monitored by MRTG. Based on performance data collected, I discuss the lessons learned about each of the configurations, highlighting problems encountered and how they might be solved. I also summarised the findings of the SLL survey, described in Chapter 6, in order to draw from those findings as well as the findings from the case studies and the laboratory tests to propose the revised computing infrastructure model for the SLL. The next chapter, the final one of this dissertation, will address the research questions proposed at the beginning and discuss future work that could be undertaken to further improve ICT integration in South African schools.

### **Chapter 9**

### **Conclusion and recommendations**

### 9.1 Summary

This dissertation described the research work being done in attempting to provide robust, pedagogically appropriate, easy to maintain and financially sustainable computing infrastructure in the long term experiment of the Siyakhula Living Lab; focusing on answering the main research question of "What efficient and effective off-the-shelf computing infrastructure, that is easy to maintain, robust, pedagogically appropriate and financially sustainable, can support rural schools in South Africa in implementing their vision for integrating ICT use into teaching, learning and administration?" The SLL is a multi-disciplinary project, bringing together researchers from multiple disciplines. My contribution to the SLL, since its inception in 2006, has been to implement, test and deploy the evolving computing and networking infrastructure (as the primary researcher in this element of the project) within the various communities and schools that make up the SLL. This dissertation consisted of nine chapters that described the computing infrastructure based research that I undertook within the SLL and the SLL school environments, attempting to bridge the two disciplines of Applied Computer Science and Education.

Chapter 1 introduced the context in which this project was undertaken; presenting the background to the study as well as a brief overview of the research design, focusing on the research aims, questions, methodology and delimitation of scope. The chapter explains that this project aims to investigate alternative computing infrastructure configurations in order to eliminate the single point of failure in the current SLL computing infrastructure model and improve reliability and system performance as well ease maintenance of the computing infrastructure such that the schools and communities are able to meaningfully make use of the available facilities; hopefully promoting education and general socio-economic development. The chapter concludes with the structure of the dissertation.

Chapter 2 described the research design and methodology underpinning this study, specifically, the research orientation, methodology, research site and participants, tools, activities and data analysis methods. The post-positivist critical realist research orientation that I adopt in this study enables me to attempt to bridge the Science and Social Science disciplines in which the study was conducted. Furthermore, it paves the way for the use of mixed methods in order to make a serious attempt to fully understand how best to provide computing infrastructure for the rural school environment in South African. The project utilizes both qualitative and quantitative methodologies, data and data analysis techniques in order to attempt to provide a more holistic explanation of how computing infrastructure is being used in rural schools, and what variables either hinder or help its adoption. It also seeks to begin to understand what teachers think about, or understand about the integration of technology into teaching practices, and how they envisage the use of technology for teaching and learning. As such the research orientation, methodologies, tools and activities employed are all with a view to gaining as much understanding about the use or lack thereof of

computing infrastructure in education practices in rural schools in South Africa in order to appropriately inform design choices pertaining to the deployment and use of different computing infrastructure within rural South African schools.

Chapter 3 gave a detailed description of the deployments done in the SLL from the years 2006 to 2013, reflecting support for the overall ideas of using thin client computing infrastructure for education and development in rural schools. This chapter provided the necessary details and context of the SLL and the computing and networking infrastructure deployments I made within the SLL in order to begin to address the shortcomings of the current solutions.

Chapter 4 looked at the use of technology in education; despite the profound effect that technology has had on the world and our society teachers still struggle with its influence both inside and outside the classroom. For some that struggle is through providing access to appropriate technologies, while others focus on how to integrate the use of technology in the classroom environment. In the case of rural schools in South Africa the struggle incorporates both the provision of technology as well as the integration of its use into teaching practices. This chapter has described how technologies can afford teachers many opportunities for supporting the teaching and learning activities of the classroom, but it also highlighted that when teachers do not have the necessary theoretical knowledge (TPACK) that they are not best placed in order to unlock the affordances of technologies and utilize them in an effective manner, regardless of the government policies in place in order to support or force teachers to use available computing infrastructure.

Chapter 5 discussed tools for implementing computing infrastructure in education, specifically schools. In South Africa there remains a dearth of access to ICTs in schools and this presents an important stumbling block in access to technology for the purposes of teaching and learning. Research has found that there is a need for new paradigms of educational technology that meet the needs and experiences of rural settings and communities. These are that schools firstly need access to ICT facilities and that these facilities need to be financially sustainable and relevant to teaching and learning. This chapter describes a number of computing and networking infrastructure/technologies that could potentially be utilized in providing robust, reliable and financially sustainable ICTs that are relevant to their context.

Chapter 6 described the survey that was undertaken with twelve teachers from six schools within the SLL together with the findings and analysis from the survey. The analysis revealed that there is a strong need for computing infrastructure that is reliable, plentiful (and thus cost effective) and meets the curriculum needs of South African schools (in terms of access to software and content) that is also easy to maintain and replace/repair.

Chapter 7 presented the experimentation undertaken with various clustering techniques, as well as performance tests utilized to assess the viability of cluster LTSP vs standard thin client LTSP and LTSP fat/hybrid client configurations, in order to eliminate the single point of failure of the standard thin client server configuration, as well as, provide significantly more hardware resources for utilization by client workstations. The results showed that memory can be a bottleneck in LTSP thin client computing and that the clustering of the LTSP servers can provide the necessary memory requirements (as well as necessary CPU requirements) in order to serve a growing number of thin clients. Furthermore, utilization of LTSP fat clients, instead of standard thin clients, further reduces the load on the servers, potentially increasing the life span of the servers and decreasing rate of failure.

Chapter 8 described the real world deployments of two alternative computing infrastructure configurations to two schools within the SLL; the DNS Round Robin Edubuntu LTSP cluster and Edubuntu Fat/Hybrid Clients as well as a third case study of a standard thin client configuration for comparative purposes. The deployments drew from the findings in Chapters 6 and 7 in order to qualify the choice for the two selected alternative configurations and investigate solutions that showed the most promise in the given context of the SLL and teaching and learning in rural South African schools. The findings from the two case studies revealed that cluster server solutions were confusing to the users in the SLL. Fat/Hybrid clients however, with a single server were more intuitive to the end user, and the single server, with significantly reduced load (as the load of the clients was essentially distributed across the clients) was more reliable than the cluster had been and certainly more reliable than a single standard thin client server. In order to fully address the single point of failure, the revised computing infrastructure model for the SLL suggests the inclusion of a back-up server, that can readily and easily replace the primary server. Furthermore, the revised model includes the use of software to support non fat/hybrid client devices on the network, in order to provide a computing infrastructure model that attempts to meet the teaching and learning needs (as specified by teachers in the SLL survey) of rural schools as best as possible.

Chapter 9 concludes this dissertation by presenting a summary of the chapters, re-examining the research questions posed in Chapters 1 and 2 and discussing the answers to the questions based on the findings from the research work undertaken. Furthermore, the chapter describes the contribution that this research work has made, recommendations that were derived and suggests future research work that could be undertaken in this area.

### 9.2 Research questions revisited

In this section I revisit the research questions posed in Chapters 1 and 2, beginning with the subsidiary questions and working towards answering the main research question which is discussed in section 9.3.

## **9.2.1** What is the currently deployed computing infrastructure and computer networks in the schools in the SLL?

I have broken down this question into two parts and answered each under its own subheading; first looking at the computing infrastructure of the SLL from its inception up to 2013, and then similarly, the SLL's networking infrastructure over the same period.

### **Computing infrastructure**

The original computing infrastructure configuration deployed within the SLL as part of my work can be seen in Figure 9.1. The choice of hardware within the computer labs at each school was to deploy thin client computer laboratories. A thin client computer is one which depends primarily on a central server for all its processing activities; they are used to collect input from the user (to the applications running on the server) and convey the output (from the application on the server) to the users. Using PXE enabled network cards the thin clients are able to request the necessary software from the central server and boot over the network. All the software, user files and profiles are stored centrally on the server.

The choice of software used in the schools in the SLL is coupled with the choice of running thin client computer labs. The SLL employs Edubuntu Linux. The reasons for this are that firstly, Edubuntu is an operating system aimed at education and the computer labs are in the local schools; it includes educational applications, tools, content, and themes and was was developed in collaboration with teachers and technologists from multiple countries [108]. Secondly, Edubuntu ships with LTSP as standard which allows thin clients computers to boot off a central server running the LTSP service [108]. LTSP is a free and open source terminal server for Linux-based operating systems. It allows many people to simultaneously use the same computer/server. Applications run on the server with thin clients (terminals) handling the input and output [201]. This technology is becoming popular in schools as it allows the school to provide learners with access to computers without purchasing or upgrading expensive desktop machines.



Figure 9.1: The original design of a generic computing infrastructure (typically as a computer lab) in a DAN at a school in the SLL

Thirdly, Edubuntu Linux is open source software, which means that source code of the software is available to the user; affording them the opportunity to study and change the software and then redistribute to others [108]. This means that the software intrinsically supports the tenets of education, allowing users (both school learners and and local community members) to learn how software works, make changes or additions, and thereby create their own locally useful versions of various software packages. This, the SLL research team believes, can only help to foster the development of human capital within communities. Finally, the Edubuntu Linux operating system, together with a number of the user applications, such as the LibreOffice Suite and the Firefox web browser, are available in some of the South African local languages, such as isiXhosa, the dominant language in the Mbashe municipality.

From Figure 9.1 it can be seen that a LAN is deployed within the school, to which a standard central thin client server is connected as well as thin clients. Figure 9.1 refers to the clients being present in a laboratory configuration, but the clients can be available through out the school as well as within a traditional computer lab.

Access to the network was extended to mobile devices, such as laptops, cellphones and tablet PCs, by the addition of one or more WiFi access points throughout the school. This further facilitated teachers and learners (as well as local community members making use of the facilities) accessing the network and its resources from anywhere on the school property. These devices were, however, not provided by the SLL project and would be the private devices of the teachers, learners and community members. The LAN was in turn connected to the wider community network (and its shared resources; VoIP, local web services, etc) and the Internet via the network connection present at the school.

From the results of the SLL survey described in Chapter 6, we know that typically the teachers were unable to describe what kinds of computers they had, nor how they were configured at each of their schools.

Similarly, less than half of the teachers were able to recognize that their schools had a LAN, nor were they able to identify what type of Internet access their school had. As such, the report on the computing infrastructure in place at the six participant schools surveyed in this dissertation, is from observation data that I made during my visits to each school. I have consolidated the data about the computing infrastructure of the six participant schools into Table 9.1. (Note, that these six schools are just a subset of the 17 schools that form the SLL.) From the table it can be seen that prior to the field test reported in this dissertation three of the schools had computer laboratories, while two had computers in the staff rooms and one had one computer in one of the classrooms. After the field test, four schools had computer laboratories, all of which also had computers in other venues such as administration offices and staff rooms. The only school to have computers in the classroom, was FH3, which only had one computer and no communal space so by default had to place the computer in one of the classrooms. All the schools had a LAN within the premises, and all the schools were connected to the Internet in some manner through the broadband island in the SLL. Of the computer laboratories, all were based on Edubuntu, except for FH6, who insisted on thick clients running Windows XP or Windows 7. As a compromise the SLL was allowed to install Edubuntu on the thick clients in addition to the Windows operating system and the clients could dual boot. FH6 is the only school with a computer lab that has no server. FH1, FH2 and FH5 are all types of Edubuntu LTSP labs; FH2 and FH5 use thin clients while FH1 had fat/hybrid clients. The computers at FH4 are also all thick clients, one of which boots Edubuntu, the other three boot Windows 7. FH4 also does not have a server.

### Networking infrastructure

The SLL network can be described as a community-wide broadband network (the "broadband island") that connects a number of DANs, housed on school properties, within the various communities the SLL operates. The entire network spans approximately 20 square kilometers and uses WiMAX technologies from Alvarion to share/distribute Internet access (via VSAT solutions initially from Telkom, then Telkom and Internet Solutions and now currently YahClick/Vox and SAAB Grintek) as well as other services such as VoIP and shared local web content (for the purposes of education mainly, but with a view to increasing local services to other areas too, such as eHealth and eGovernance). The choice of VSAT was as a result of no fixed-line alternative in the area of the SLL in the Mbashe Municipality and VSAT offers greater through-put than the patchy 3G connectivity in the area.

Due to the mountainous terrain in the Mbashe Municipality and specifically the Dwesa-area and the lack of fixed-line infrastructure available the use of Wireless technologies was the only truly feasible option for connecting the various schools to one another and shared Internet access. I had significant experience in building wireless networks over long distances in order to connect school premises; my own masters thesis was heavily focused on building WiFi networks for providing Internet access to previously disadvantaged schools in peri-urban areas [100]. From my past experience I knew the potential pitfalls of using WiFi over long haul, specifically that WiFi is very prone to interference over long distance and that given enough interference the network can significantly degrade and be rendered useless. As a result I wanted to provide a network solution that was more robust, especially given that we were approximately 400 km away from the site of the SLL. Fortunately, the Telkom CoE at the University of Fort Hare had an industry sponsor, SAAB Grintek, who specialized and resold WiMAX equipment. SAAB Grintek are suppliers of Alvarion WiMAX technologies. Through their support and expertise we were able to acquire Alvarion WiMAX micro base stations and CPEs. Furthermore, SAAB Grintek provided us with training and support materials in order to configure and deploy WiMAX networks within previously disadvantaged communities and more specifically within the communities participating in the SLL. The broadband island in the SLL has experimented with both fixed WiMAX technology and mobile WiMAX technology. For these reasons WiMAX technolo-

School	Types of computers	Computer configuration	Number of com-	Location of computers	Operating systems	LAN	Internet connection
FH1	Pentium 4 PCs, a server, laptops	Before field test: none Post field test: Edubuntu fat clients with server	30 PCs; 1 server; 2 laptops	Computer lab; school office; ad- ministration office	Before field test: Windows 7 Post field test: Windows 7, Edubuntu fat clients Server: Edubuntu	Yes	Before field test: none Post field test: VSAT
FH2	Before field test: Pentium 4 thin clients and a server Post field test: HP thin clients and a server	Before field test: Thin clients with server Post field test: thin clients with server	Inital: 6 PCs and one server Post field test: 20 thin clients and one server	Intial: Staff room; school office Post field test: Computer lab; admin- istration office; staff room	Before field test: Edubuntu Post field test: Edubuntu; Windows	Yes	Before field test: WiMAX link to FH5 Post field test: VSAT
FH3	Atom PC	Thick client	1	Classroom	Edubuntu	Yes	WiMAX link to FH5
FH4	Atom PC and Core 2 desktops	Think clients	4	Staff room	Edubuntu; Windows 7	Yes	WiMAX link to FH5
FH5	Atom thin clients and server	Before field test: Thin clients with a server Post field test: thin clients with a cluster server	Before field test: 20 thin clients and a server Post field test: 20 thin clients and 5 servers as a cluster	Computer lab	Edubuntu	Yes	VSAT
FH6	Pentium 4s and Core 2 destops	Thick clients	20	Computer lab; admin- istration offices	Windows XP; Windows 7; Edubuntu	Yes	WiMAX link to FH5

Table 9.1: Computing infrastructure of the six SLL participant schools in this study

gies together with VSAT were used to interconnect and connect the participating communities within the SLL.

# **9.2.2** What current and future computing infrastructures (that are robust, easy to maintain, financially sustainable and pedagogically appropriate) are available for use in South African rural schools?

Current computing/technology trends facilitate ubiquitous access to applications and data [116, 114, 9, 117]. Computation is seen as moving into the "Cloud" making it possible for users to connect to resources from any device anywhere [118, 9]; "Powerful services and applications are being integrated and packaged on the Web in what the industry now calls 'Cloud Computing'" [114, p1]. This is often facilitated through multi-functional small devices, such as laptops and cellular phones that support numerous functions and connect to services, such as email, browsing, video playback and other online services that provide access to data files [116, 114, 9]. These online services are fast becoming supported by groups of servers working together in either clusters or grids [118, 114, 9]. The use of grids, clusters and clouds also have potential value in educational settings, providing high availability and increased computing power [119, 9]. In addition, they facilitate resource sharing and can readily be built using commodity, off-the-shelf equipment, which have cost reduction implications [119, 9, 117].

As such I experimented with technologies that would facilitate the use of both local clusters and grids within the computing infrastructure deployments in the SLL. The DNS round robin allowed me to experiment with the use of a server cluster within the school environment to support thin client computers – in much the same way that the Cloud supports Internet-enabled devices utilizing online services. In addition, I deployed a grid-like solution in the fat/hybrid clients, where the computational load was distributed across many computers in order to maximise resource utilization. Related to the concepts of the cluster, cloud and grid, is virtualisation. Desktop virtualization separates the personal computer desktop environment and applications from the underlying physical machine using the client–server model of computing [196]. The benefits of desktop virtualization are similar to the benefits of virtualization in general, in that the overall hardware needs and costs are reduced and the desktops are centrally stored making them easier to manage, control and backup [10]. While some of the virtualization methods investigated here were too expensive in terms of the hardware requirements, in the future this is less likely to be the case, and we could see more virtualization and specifically desktop virtualization taking place within the school environment.

# 9.2.3 What are the technological-based issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?

When asked about computing infrastructure or ICTs at the six schools, 326 comments were made by the teachers (see Figure 6.10 in Chapter 6). This was the most discussed topic across all the interviews and was discussed more than twice the number of times as the next highest category. Arguably, infrastructure is a large part of the focus of this project, as it seeks to investigate appropriate computing infrastructure for schools, and this would account some way for the high number of comments in this category. However, the fact that the comments were more than double the next highest category, is an indication of how important it is to get the infrastructure right in order to even begin to ask the question of how one integrates the use of computing infrastructure into teaching and learning. The infrastructure category was further subdivided into Hardware, Software, Networking/Internet, Physical (schools building, electricity, etc), Finance, Importance of technology and maintenance.

In terms of hardware, teachers largely discussed the limitedness of their resources, as a major technologicalbased issue that they were facing. Teachers talked about needing access to more technology, such as more stations in the school in order to better accommodate the large number of learners, and more portable facilities, such as laptops and data projectors in order to be able to take some of the computing infrastructure into the classroom on an ad-hoc basis. Furthermore, teachers expressed a need for computing infrastructure to be robust and reliable; faulty computing infrastructure limits use and hampers integration efforts.

In terms of software, teachers again highlighted their limited resources and stressed the need for more software which is appropriate to the South African curriculum. In addition, they highlighted their own limited knowledge about various software packages, expressing the need for training and support.

Finally, in terms of networking infrastructure, teachers noted that networking infrastructure needs to also be robust and reliable, as teachers have come to rely on access to the Internet in order to obtain much needed educational content. Furthermore, teachers wanted networking access to be available throughout the school, so that they and the learners could access the Internet and shared resources from any location on the school property and using any device.

Thus, the technological-based issues that the school teachers at the six participant schools highlighted around the use of technology for teaching, were predominantly linked to a lack of resources, both computing and software. Furthermore, they stressed that those resources need to be robust and reliable, so that they can always trust that the facilities will be in a working condition in order to promote integration into teaching practices.

## **9.2.4** What are the pedagogical issues that the school teachers face in introducing the use of technology into their teaching practices in the SLL?

From section 6.3.3 of Chapter 6, it can be seen that PK was not classified once during the course of the data analysis of the interviews/focus groups, meaning that no teacher talked about pedagogical knowledge (i.e. teaching strategies) during the course of the interviews/focus groups. That in and of itself is a very telling finding. It would seem that the teachers surveyed from the SLL do not take into consideration teaching strategies when attempting to make use of technology for teaching. This could indicate that the teachers are not carefully planning technology integration in their lessons and are either: tacking their use onto the lesson (for example, setting an assignment where the learner must use a computer to either write an essay or search for information) and leaving their teaching strategies completely unchanged; or only using technology as a method of presenting a lesson (the modern version of the chalkboard); or not attempting to make use of technology in the lesson at all. The teachers' inability or unwillingness to adapt their teaching strategies in order to meaningfully consider and include the use of technology is a pedagogical issue that they face in introducing the use of technology into their teaching practices.

Another pedagogical issue is the lack of training for teachers in the use of and specifically the integration of technology into teaching practices and could be a contributing factor to teachers not considering their teaching strategies when attempting to make use of technology in the classroom. Teacher\_G noted in her interview that the Department of Basic Education pushes teachers to use technology in the classroom, but does not provide any training or support in order to appropriately achieve that goal. She said, "They didn't teach us or show us how to use those computers but they want us to use them". Teachers need to be adequately trained as users of technology, in addition to appropriately trained in the integration of technology into teaching practices. Without these two training types taking place, there is nothing more than a "wing and a prayer" on which to pin hopes that computing infrastructure will be meaningfully utilized in teaching practices.

# **9.2.5** How do clustered LTSP servers and fat/hybrid client configurations compare (in terms of performance, reliability, cost and ease of maintenance) with the standard thin client configuration in the school environment in the SLL?

In order to answer this research question, I have broken down the comparison of the two alternative configurations (clustered LTSP servers and fat/hybrid clients) and the standard thin client configuration, to discuss the comparisons in terms of each of the stated variables, namely, performance, reliability, cost and ease of maintenance, for each of the solutions.

### 9.2.5.1 Clustered LTSP server

### Performance

The primary purpose of both the cluster server solution and the fat/hybrid client (grid-like) solution was to attempt to make available more computing resources to the client computers at schools in order to improve the overall experience for the end user, without significantly increasing the overall cost of the computing infrastructure. Furthermore, both alternatives were an attempt at eliminating the single point of failure that plagued the single, central server of the current computing infrastructure model deployed in the SLL.

I used the MRTG software to monitor the hardware of each of the configurations, both in the laboratory experiments and in the real world deployments. From the data gathered by in the laboratory experiments that simulated user interactions (using Sikuli) I found that the DNS Round Robin cluster was able to service almost twice the number of thin client connections as compared with a single quad-core server, which had 4 GB of RAM (the minimum requirement for a server to support 20 clients according to Edubuntu [290]). The Quad-core server could, however, service the incoming requests from the clients faster than the cluster server (see Table 7.2 vs Table 7.3 in Chapter 7). That said, it is of greater value to support more clients (and simultaneous users) than being able to service fewer users at a faster rate when providing general desktop computing services; as long as the user experience is not negatively impacted which would hinder effective use of the clients. In addition, if we consider the MRTG results from the laboratory testing in section 7.6.1 vs section 7.6.3, we can see that during the Sikuli user interactions on the standard thin client the CPU maxed at 100% and the free memory dropped to about 2 GB (from a total of 8 GB). The loads measured on the DNS Round Robin cluster during the same tests showed that the CPU and memory utilization on the file server node (Node1) were very low, as expected, as this node only serves files to the clients. While on the LTSP nodes, the CPUs were loaded between 30% and 90%, while free memory dropped from around 3 GB to approximately 1.4 GB (one of the nodes, the one that had the most clients, dropped at times to around 800 MB). Thus, the cluster was able to more than adequately supply computing resources to the clients, as compared to a single server, and with a few added benefits. These benefits are, firstly, if we assume that a single server had the same amount of resources (specifically memory and CPU) as a cluster of servers working together, the cluster of servers are going to cost less than the single server, because that single server will have to be high end in order to provide that much computing power in one box. Secondly, the cluster server, if designed properly, would not present the single point of failure that the single server does.

### Reliability

In the case studies conducted at FH5 and FH2, where a DNS Round Robin cluster Edubuntu LTSP server and a standard single Edubuntu LTSP server were respectively deployed, we saw that the DNS Round Robin cluster was less susceptible to the single point of failure from the perspective of providing the LTSP service to the clients, but that the configuration was flawed in that the file server (Node1) was still a single point of failure. However, if a back-up server was in place for the file server, as discussed in section 8.6

of Chapter 8, then the solution would be more robust and reliable with a much greater fault tolerance, that would eliminate the single point of failure that plagues the standard thin client configuration of FH2.

### Cost

The ability to use older, refurbished computers/servers in the cluster reduces the cost of obtaining nodes to join the cluster or replace faulty nodes in the cluster. Furthermore, single powerful servers (i.e. high end servers, which have a lot of memory and CPU power) tend to cost a lot of money. Using cheaper computers, working together to pool their hardware resources can produce cheaper, overall, hardware solutions.

### Ease of maintenance

In terms of ease of maintenance, the high availability of the overall system – if one or two nodes (servers) fail or are not running for whatever reason, there are still enough remaining nodes to service clients – is hoped to ease maintenance, as should a node fail there remain others to continue servicing clients. This provides the technicians with time to assess the faulty node and either make repairs or replacements and then return it into the cluster to continue servicing clients. Furthermore, if the back-up server is implemented as described in Chapter 8, the overall reliability of the computing infrastructure would be improved, which would help ease maintenance as there would be time for technicians to travel to the schools without the school and users experiencing long periods of downtime. That said, the additional complexity of multiple servers working together to act as though they were one, will be more complex to understand and maintain for novice or inexperienced users, as opposed to a single server.

### 9.2.5.2 Fat/Hybrid clients

### Performance

As I said above, the primary purpose of the alternative computing infrastructure configurations, to that of the standard thin client, was to make available more computing resources to the client computers. Furthermore, they were an attempt at eliminating the single point of failure that plagued the single, central server of the current computing infrastructure model deployed in the SLL. The fat/hybrid client configuration is essentially a grid-like solution, that distributed the computational load of servicing the desktops of the clients to the clients themselves. Thus the server was freed to be just a DHCP, TFTP, DNS and file server (and any other services that a school might need, such as proxy or web).

From the data gathered when performing the user simulation testing (in the laboratory, reported on section 7.6.2 of Chapter 7), using Sikuli, I found that the tests ran faster for both Firefox and LibreOffice on the fat/hybrid client as opposed to those running on the thin client server. Furthermore, the load on the server servicing fat/hybrid clients was negligible as all the processing of the Sikuli tests and simulated user interactions were being handled by the client directly. The load on the client was significantly less than compared to the load on the thin client server, because the client was handling the processing of one user, while the thin client server was supporting 21 simultaneous users. Thus, the performance of fat/hybrid clients was superior to that of a single server attempting to service multiple user sessions. Furthermore, when considering the data from the real world case study, conducted at FH1 and reported on in Chapter 8 (section 8.3), it was seen that the load on the central server was less as compared to the loads that had been recorded by standard thin client server configurations described in Chapter 3 (section 3.2.4.3). The fat/hybrid client server at FH1 had less load on the CPU, and more free memory than the results described in Chapter 3.

### Reliability

In the deployment of the fat/hybrid client solution to FH1, the server only ever experienced one failure in a three year window. I have surmised that that is due to the computation load being distributed to the clients; increasing the overall lifespan of the server hardware. This was a significant improvement over the number of failures of server hardware witnessed at schools in the SLL where the standard thin client configuration was in use, and even with respect to the DNS Round Robin solution implemented at FH5. That said the configuration of the fat/hybrid client model, still only uses one central server, which remains a single point of failure. In order to address this, a back-up server should be implemented, and the configuration significantly more reliable.

### Cost

The fat/hybrid client solution is probably the most cost effective solution of the three (cluster, standard thin client and fat/hybrid client) as the server requirements in this configuration are not high end and only two servers are required (the primary and the back-up) and the clients were going to be purchased anyway - regardless of which of the three solutions implemented. However, the resources of the clients will be utilized and not sit, essentially, ideal.

### Ease of maintenance

The fat/hybrid client configuration is as easy to maintain as a standard thin client configuration, as there is still only one computer that needs to be maintained, namely the server. Furthermore, implementation of a back-up server will make the configuration more robust and reliable, limiting downtime and providing technicians time to travel to the school to effect repairs when the primary server fails. Like the standard thin client configuration, fat clients do not store anything (operating systems or user data), so should they fail no user data is lost and all that needs to be repaired or replaced is the hardware of the client.

### 9.3 Conclusion and recommendation

The main research question that guided this project was:

"What efficient and effective, off-the-shelf computing infrastructure, that is easy to maintain, robust, pedagogically appropriate and financially sustainable, can support rural schools in South Africa in implementing their vision for integrating ICT use into teaching, learning and administration?"

Drawing from the findings of the SLL survey, the laboratory experiments and the real world deployment case studies within the SLL, it is my recommendation that the revised computing infrastructure model for the SLL consist of a single Edubuntu LTSP fat/hybrid client server, together with Edubuntu LTSP fat/hybrid clients and a back-up server in order to address the single point of failure. The fat/hybrid client configuration has shown in both the laboratory tests and the case studies to be the most reliable and effective method of providing cheap computing infrastructure, that maximizes the use of all available resources within the computing infrastructure configuration, provided that both central server and fat/hybrid clients are chosen at the correct price/performance point.

Furthermore, the server (and the back-up) should be configured to offer file sharing services to other non fat/hybrid clients (through the use of SAMBA for example) in order to facilitate users access to their files and folders regardless of the device from which they are operating within the school property. In addition, the server (and the back-up) can be configured to provide access to other services like a proxy service in order to conserve Internet bandwidth utilization, ensure equitable access to the Internet and allow some control on accessed content. In terms of routing functionality for the school, I would suggest the use of a Mikrotik router-board, which will more than adequately support the connection of the server and fixed Ethernet clients to the Internet, as well as facilitate the connection of WiFi Access Points to the Mikrotik to support wireless devices in and around the school property accessing the local services on the network (such as the server for serving files) and the Internet access.

The fat/hybrid client computing infrastructure model builds on and improves the functionality of the computing infrastructure model in the SLL, as all the computing resources are utilized, providing increased access to computing resources, and with the back-up server, the single point of failure eliminated, while still remaining cost effective, easy to maintain and retaining the use of the Edubuntu OS and free, open source software. This model makes the most sense as the primary computing solution for schools and communities in rural and marginalized areas in order to provide functional computing infrastructure that supports users as both producers and consumers of information. Furthermore, this model can easily be supplemented by use of more trendy mobile devices such as smart cellphones and tablet PCs.

### 9.4 Research contribution

This dissertation has attempted to document 10 years worth of ICT4D and ICT for education research conducted within the SLL. The long term nature of the work makes it unique in that the reader and research community can reflect on the multiple contributions implemented and corresponding changes that were experienced in communities and schools over time where this study has taken place. Furthermore, it allows the reader to fully understand the nuances and complexities in conducting research of this nature. The long term nature also reflects the difficulties of introducing technology into rural areas and rural schools and how over time, and through lessons learned, the technology deployments adapt and change and become better tailored to their particular environments. In addition, the long term nature of this project can support or facilitate planning for similar projects to be conducted in other areas, as the detailed description of events over time can contribute to the planning stages of similar projects in such a way that those projects are better off for the knowledge and experience gained here.

As a result of this research work, I am able to provide a computing infrastructure model for use in rural schools, that will hopefully help the integration of technology into teaching, while supporting the use of both fixed and mobile computing resources from any location within the school grounds. Furthermore, access to the facilities for the surrounding communities members will hopefully result, over time, in positive economic changes for rural South African communities, giving them a voice and a place within the Networked Society [27].

### 9.5 Future work

The long term nature of the SLL supports the notion that we have not learned all we can from the communities in Dwesa, nor about ICT4D and ICT for education. As such, there are a number of research areas that need to be investigated, following on from this work.

Firstly, the revised computing infrastructure model could and should be deployed to schools in the SLL in order to test the reliability and performance of the solution. In addition, a follow-up survey should be conducted with the teachers from the SLL in order to gauge their perceptions of the revised model and if they find it to be more reliable and an overall better user experience; going some way to address their needs as expressed in the last survey, covered in Chapter 6.

Secondly, there is so much more data that was generated during the SLL survey that I conducted, but because of time constraints and the delimitation of scope could not be adequately addressed and analysed

here. As such there is much more data analysis and corresponding knowledge to be unlocked from further analysis of the data collected in the SLL survey (from Chapter 6).

Thirdly, the frameworks of Affordances and TPACK could be utilized in order to produce guides for teachers, where technology affordances are mapped to classroom activities; producing a clear, step-by-step guide for teachers of how computing infrastructure could be used to support classroom activities in order to support the integration of technology into teaching practices.

Fourthly, the broadband island in the SLL would benefit from better fault tolerance. For example, implementing OSPF in order to simplify routing and better support redundant routes, which could facilitate back-up routes using, for example, WiFi in order to minimize network downtime. Furthermore, better monitoring software could be implemented to monitor links and servers in order to provide early notification to technicians at the universities of fault occurrences.

Finally, as highlighted in the body of this dissertation, the reasons for providing computing infrastructure via thin clients over thick clients or something more trendy, such as a tablet, were numerous: thin clients are cheaper than thick clients; historically the use of personal devices such as mobile phones and tablet PCs were limited; smart phones and tablet computers are personal devices, which are not typically shared devices; and rural communities will derive much more benefit from being producers of information and not just consumers of information. That said, computing infrastructure and resources are tending to move to "the Cloud" and end user devices are typically just portals to remote infrastructure. Research and experimentation could be done to investigate providing similar offline solutions that allow mobile devices, such as phones and tablets, to hand off computational load to a central, powerful shared computing resource for computation and results returned to the end user device, so that mobile devices might be more useful to the end user in the SLL.

### 9.6 Parting thoughts

By providing robust and reliable computing infrastructure and networking facilities at schools across the country, where teachers, learners and local community members can access the facilities, we can begin to bridge the digital divide in South Africa. To that end I have proposed the use of fat/hybrid clients together with a primary and back-up server as the primary method of network access in rural South African schools. In this manner, the role of education, namely, to develop higher-order thinking, inquiry, collaborative problem solving and transmit culture, values and lessons of the past while preparing individuals for the world of information and knowledge, is extended to the wider community. Bridging of the digital divide would further help realise the networked society and collective intelligence of which Cornu spoke [27]. Developing this collective intelligence would allow people to work together and solve problems and address questions in new and innovative ways [27]. Through our collective intelligences South Africans can build a better country for us all.
### **List of References**

- B. Bergvall-Kareborn and A. Stahlbrost, "Living lab: an open and citizen-centric approach for innovation," *International Journal of Innovation and Regional Development*, vol. 1, no. 4, pp. 356–370, 2009.
- [2] National Education Policy Investigation, Governance and Administration: Report of the NEPI Governance and Administration Research Group. A Project of the National Education Co-ordinating Committee. Oxford University Press and the National Education Co-ordinating Committee, Cape Town, 1992.
- [3] J. Harris, P. Mishra, and M. Koehler, "Teachers' technological pedagogical content knowledge and learning activity types: Curriculum-based technology integration reframed," *Journal of Research on Technology in Education*, vol. 41, no. 4, pp. 393–416, 2009.
- [4] P. Mishra and M. J. Koehler, "Technological pedagogical content knowledge: A framework for teacher knowledge," *Teachers College Record*, vol. 108, no. 6, pp. 1017–1054, 2006.
- [5] Wikipedia Foundation, Inc, "Wikipedia." Website, September 2016. https://en.wikipedia.org/wiki/Wikipedia Last Accessed: September 2016.
- [6] AntiClockwise, "One of Google's MASSIVE server farm," 2007. Last Accessed May 2008.
- [7] R. Buyya, T. Cortes, and H. Jin, "Single System Image (SSI)," *The International Journal of High Performance Computing Applications*, vol. 15, no. 2, pp. 124–135, 2001.
- [8] L. Youseff, M. Butrico, and D. D. Silva, "Toward a unified ontology of clound computing," in *Grid Computing Environments Workshop 2008 (GCE '08)*, pp. 1–10, 2008.
- [9] F. M. Aymerich, G. Fenu, and S. Survis, "An approach to a cloud computing network," in *The First International Conference on Applications of Digital Information and Web Technologies*, 2008 (ICADIWT 2008), pp. 113–118, 2008.
- [10] D. Vile, T. Lock, M. Atherton, and J. Collins, *Desktop Virtualization for Dummies*. Wiley Publishing Inc, 2010.
- [11] W. R. Stevens, *TCP/IP Illustrated*, *Volume 1: The Protocols*, vol. 1. Addison-Wesley Professional Computing Series, 1993.
- [12] B. Callaghan, NFS Illustrated. Addison-Wesley Longman Ltd, 2000.
- [13] T. Bao, "The architectural differences between lte and wimax," September 2014.
- [14] Revolution Linux, "Linux cluster: Technical introduction." Online, 2010. Last Accessed: November 2013.

- [15] Department of Communications, "Electronic communications act, 2005 (act no. 36 of 2005): South Africa Connect: Creating opportunities, ensuring inclusion South Africa's broadband policy." Government Gazette, December 2013.
- [16] Wikimedia Foundation, Inc, "Single system image." Online, February 2013. http://en.wikipedia.org/wiki/Single\_system\_image Last Access: Oct 2013.
- [17] Ubuntu documentation, "UbuntuLTSP/FatClients." Online, December 2012. https://help.ubuntu.com/community/UbuntuLTSP/FatClients Last Accessed: November 2013.
- [18] A. Molnar, "Computers in Education: A brief history," *Technological Horizons in Education Journal* (*T.H.E.*) *journal*, vol. 24, pp. 63–68, June 1997.
- [19] E. Rusten, "Using computers in schools." Digital Opportunities for development, 2003.
- [20] D. Watson, "Pedagogy before technology: re-thinking the relationship between ICTs and teaching," *Education and information technologies*, vol. 6, no. 4, pp. 251–266, 2001.
- [21] L. Dawes, "First connections: teacher and the National Grid for Learning," *Computers and Education*, vol. 33, pp. 235–252, 1999.
- [22] M. Warschauer, M. Knobel, and L. Stone, "Technology and equity in schooling: Deconstructing the Digital Divide," *Educational Policy*, vol. 18, no. 4, pp. 562–588, 2004.
- [23] D. Watson, "Understanding the relationship between ict and education means exploring innovation and change," *Education and Information Technologies*, vol. 11, pp. 199–216, 2006.
- [24] N. Selwyn, S. Gorard, and S. Williams, "Digital Divide or Digital Opportunity? The Role of Technology in Overcoming Social Exclusion in U.S. Education," *Educational Policy*, vol. 15, no. 2, pp. 258–277, 2001.
- [25] Department of Education, "White Paper on e-Education: Transforming learning and teaching through Information and Communication Technologies," Sept. 2004.
- [26] World Bank, "World Development Report 1998-1999: Knowledge for Development," 1998.
- [27] B. Cornu, World Year Book of Education 2004: Digital Technology, Communities and Education
   Chapter2: Networking and collective intelligence for teachers and learners. Routledge Falmer, London and New York, 2004.
- [28] C. Pade, I. Sieborger, H. Thinyane, and L. Dalvit, ICTs and Sustainable Solutions for the Digital Divide: Practical Approaches - Development Informatics and Regional Information Technologies: Theory, Practice and the Digital Divide, vol. 2, ch. The Siyakhula living lab: a holistic approach to rural development through ICT in rural South Africa. IGI Global, 2010.
- [29] M. Prensky, "Digital natives, digital immigrants," On the Horizon, vol. 9, no. 5, 2001.
- [30] D. Conradie, C. Morris, and S. Jacobs, "Using information and communication technologies (ICTs) for deep rural development in South Africa," *Communicatio*, vol. 29, no. 1, pp. 199 217, 2003.
- [31] I. Sieborger and A. Terzoli, "WiMAX for rural SA: The experience of the Siyakhula Living Lab," in Southern African Telecommunication Networks and Applications Conference, The Future - a Society Driven by Innovation and Applications, 2010, Sept.6–8, 2010.

- [32] Telkom Centre of Excellence, University of the Western Cape, "Centre of Excellence in Internet Computing." http://www.coe.uwc.ac.za/ Last accessed: May 2011.
- [33] UCT Centre in ICT for Development, "UCT Centre in ICT for Development." Online, 2017. http://ict4d.cs.uct.ac.za/ Last Accessed: Jan 2017.
- [34] V. van Reijswoud, "Appropriate ICT as a Tool to Increase Effectiveness in ICT4D: Theoretical Considerations and Illustrating Cases," *The Electronic Journal on Information Systems in Developing Countries*, vol. 38, no. 9, pp. 1 – 18, 2009.
- [35] I. Mulder, W. Bohle, S. Boshomane, C. Morris, H. Tempelman, and D. Velthausz, "Real world innovation in rural south africa," *The Electronic Journal for Virtual Organisations and Networks*, vol. 10, pp. 8–20, 2008.
- [36] J. Isaac, D. Johnson, and N. Ntlatlapa, "Eastern Cape Broadband Master Plan: Broadband Advantage Eastern Cape!." CSIR: Meraka Institute, January 2015.
- [37] R. B. Johnson and A. J. Onwuegbuzie, "Mixed methods research: A research paradigm whose time as come," *Educational Researcher*, vol. 33, no. 7, pp. 14–26, 2004.
- [38] R. B. Johnson, A. J. Onwuegbuzie, and L. A. Turner, "Towards a definition of mixed methods research," *Journal of Mixed Methods Research*, vol. 1, no. 2, pp. 112–133, 2007.
- [39] S. Laldaparsad, "Urban and rural trends in south africa," tech. rep., Statistics South Africa, Pretoria, South Africa, 2011. http://www.statistics.gov.hk/wsc/CPS101-P5-S.pdf Last Accessed: August 2016.
- [40] S. Townships: Mahajan, ed., **Economics** of South African Special fo-Washington, cus on Diepsloot. World Bank Group; D.C., 2014. http://documents.worldbank.org/curated/en/217211468302413395/pdf/899170PUB0978100Box 385216B00PUBLIC0.pdf Last Accessed: Jan 2017.
- [41] A. Sayer, Realism and social science. Sage: London, 2000.
- [42] B. Carter and C. New, eds., *Making realism work: Realist social theory and empirical research*. London: Routledge, 2004.
- [43] W. M. Trochim, "Research Methods Knowledge Base," 2002. http://www.socialresearchmethods.net/kb/positvsm.htm Last accessed Jan. 2006.
- [44] H. Patomaki and C. Wight, "After Postpositivism? The Promises of Critical Realism," *International studies quarterly*, vol. 44, no. 2, pp. 213–237, 2000.
- [45] P. Irwin, "A very brief Introduction to positivism," Education Department Rhodes University, 2004.
- [46] G. Walsham and S. Sahay, "Research on information systems in developing countries: Current landscape and furture prospects," *Information Technology for Development*, vol. 12, no. 1, pp. 7–24, 2006.
- [47] J. McNiff and J. Whitehead, All you need to know about Action Research. Sage Publications, Inc, California, 2nd ed., 2011.
- [48] Wikipedia Foundation, Inc, "Action research." Online, October 2016. https://en.wikipedia.org/wiki/Action\_research Last Accessed: November 2016.

- [49] C. MacDonald, "Understanding participatory action research: A qualitative research methodology option," *Canadian Journal of Action Research*, vol. 13, no. 2, pp. 34–50, 2012.
- [50] R. Baskerville, "Investigating information systems with action research," *Communications of AIS*, vol. 2, October 1999.
- [51] Wikipedia Foundation, Inc, "Participatory action research." online, September 2016. https://en.wikipedia.org/wiki/Participatory\_action\_research Last Accessed: October 2016.
- [52] A. R. Vollman, E. T. Anderson, and J. McFarlane, *Canadian community as partner*. Philadelphia, PA: Wolters Kluwer | Lippincott Williams & Wilkins, 2012.
- [53] A. Cornwall and R. Jewkes, "What is participatory research," *Social Science and Medicine*, vol. 41, pp. 1667–1676, December 1995.
- [54] C. Dell'Era and P. Landoni, "Living lab: A methodology between user-centred design and participatory design," *Living Lab*, vol. 23, no. 2, pp. 137–154, 2014.
- [55] Directorate-General for the Information Society and Media, *Living Labs for user-driven open inno-vation: an overview of the living labs methodology, activities and achievements*. Directorate-General for the Information Society and Media: Unit F4 New Infrastucture Paradigms and Experimental Facilities, Luxembourg: Office for Official Publications of the European Communities, January 2009.
- [56] A. Stahlbrost and M. Holst, *The living lab methodology handbook*. Centre for Distance-spanning Technology, Social Informatics, Lulea University of Technology and CDT, Sweden, Centre for Distance-spanning Technology, Social Informatics, Lulea University of Technology and CDT, Sweden, 2012.
- [57] Wikipedia Foundation Inc, "Long-term experiment." Online, June 2016. https://en.wikipedia.org/wiki/Long-term\_experiment Last Accessed: October 2016.
- [58] B. Owens, "Long-term research: Slow science," *Nature*, vol. 495, pp. 300–303, March 2013. http://www.nature.com/news/long-term-research-slow-science-1.12623 Last Accessed: October 2016.
- [59] Institute for Work & Health, "What researchers mean by... cross-sectional vs. longitudinal studies." online article, May 2015. https://www.iwh.on.ca/wrmb/cross-sectional-vs-longitudinal-studies Last Accessed: October 2016.
- [60] M. R. Pelton and F. T. van Manen, "Benefits and pitfalls of long-term research: a case study of black bears in great smokey mountains national park," *Wildlife Society Bulletin*, vol. 24, no. 3, pp. 443–450, 1996.
- [61] D. B. Lindenmayer and G. E. Likens, "Adaptive monitoring: a new paradigm for long-term research and monitoring," *Trends in Ecology and Evolution*, vol. 24, pp. 482–486, May 2009.
- [62] C. Robson, Real world research. Oxford: Blackwell, 1993.
- [63] R. K. Yin, Applications of Case Study Research. Sage Publications, Inc, California, 2012.
- [64] G. Wisker, *The postgraduate research handbook: Succeed with your MA, MPhil, EdD and PhD.* New York: Palgrave, 2001.

- [65] S. Rose, N. Spinks, and A. I. Canhoto, "Management research: Applying the principals case study research design." 2015.
- [66] R. Pawson, Making realism work: Realist social theory and empirical research, ch. Evidence based policy: A realist prespective. London: Routledge, 2004.
- [67] R. Palmer, H. Timmermans, and D. Fay, From Conflict to Negotiation: Nature-based Development on South Africa's Wild Coast. Pretoria: Human Sciences Research Council, 2002.
- [68] L. Dalvit, M. Thinyane, A. Terzoli, and H. Muyingi, "The deployment of an e-commerce platform and related projects in a rural area in South Africa," in *3rd Annual International Conference on Computing and ICT Rresearch - SREC07, Kampala, Uganda*, 2007.
- [69] Siyakhula Living Lab, "Siyakhula living lab." Website, 2014. http://www.siyakhulall.org Last Accessed; November 2014.
- [70] L. Dalvit, I. Sieborger, and H. Thinyane, "The expansion of the siyakhula living lab: a holistic perspective," in *AFRICOMM 2011* (R. Popescu-Zeletin et al, ed.), pp. 228–238, Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2012.
- [71] I. Sieborger and A. Terzoli, "The network society: A model for computing infrastructure in south african schools," in *IST-Africa 2016 Conference Proceedings* (P. Cunningham and M. Cunningham, eds.), 2016.
- [72] S. J. Howie, A. Muller, and A. Paterson, *Information and Communication Technologies in South African Secondary Schools*. HSRC Press, Cape Town, South Africa, 2005.
- [73] Wikipedia Foundation Inc, "Bantustan." Online, December 2016. https://en.wikipedia.org/wiki/Bantustan Last Accessed: December 2016.
- [74] E. B. Fiske and H. F. Ladd, "Racial equity in education: How far has south africa come?," *International Journal of Educational Development (IJED) on Education Policy and Practice in Post-Apartheid South Africa - Ten years of Democratic Change*, 2004.
- [75] M. S. Rakometsi, THE TRANSFORMATION OF BLACK SCHOOL EDUCATION IN SOUTH AFRICA, 1950-1994: A HISTORICAL PERSPECTIVE. PhD thesis, Department of History at the University of the Free State, November 2008.
- [76] P. Irwin, "Introduction to Surveys and Questionnaires," *Education Department Rhodes University*, 2004.
- [77] G. Euvrard, "An introduction to conducting interviews," 2003.
- [78] W.-C. Leung, "How to design a questionnaire," StudentBMJ, vol. 9, pp. 187–189, June 2001.
- [79] T. F. Burgess, *A general introduction to the design of questionnaires for survey research*. University of Leeds, 1.1 ed., May 2001.
- [80] S. McLeod, "The interview method." Online, 2014. http://www.simplypsychology.org/interviews.html Last Accessed: Dec 2016.
- [81] D. W. Turner, "Qualitative interview design: A practical guide for novice investigators," *The Qualitative Report*, vol. 15, no. 3, pp. 754–760, 2010.

- [82] T. Zorn, "Designing and conducting semi-structured interviews for research," August 2003.
- [83] S. McLeod, "Experimental method." Online, 2012. http://www.simplypsychology.org/experimentalmethod.html Last Accessed: December 2016.
- [84] Wikipedia Foundation, Inc, "Experiment." Online, November 2016. https://en.wikipedia.org/wiki/Experiment Last Accessed: December 2016.
- [85] W. L. Neuman, Social research methods : qualitative and quantitative approaches. Allyn and Bacon, Boston, 2003.
- [86] United States General Accounting Office, "Quantitative Data Analysis: An Introduction," May 1992. http://www.gao.gov/special.pubs/pe10111.pdf Last accessed Jan. 2006.
- [87] Rhodes University Department of Statistics, "Quantitative Data Analysis using Statistica: An Introduction." Course notes, Mar. 2005.
- [88] Wikipedia, the free encyclopedia, "Quantitative analysis." online encyclopedia, 2004. http://en.wikipedia.org/wiki/Quantitative\_Analysis Last accessed Oct. 2005.
- [89] S. J. Taylor and R. Bogdan, *Introduction to qualitative research methods: A guide and resource*. John Wiley and Sons, Inc, New York, third ed., 1998.
- [90] M. B. Miles and A. M. Huberman, *Qualitative data analysis: A sourcebook of new methods*. Sage Publications, Inc, California, second ed., 1984.
- [91] M. Bower, "Affordance analysis matching learning tasks with learning technologies," *Educational Media International*, vol. 45, no. 1, pp. 3–15, 2008.
- [92] A. Bryman and D. Cramer, *Quantitative data analysis for social scientists*. Routledge, London and New York, 1990.
- [93] C. H. Leung, *Quantitative analysis of computer systems*. John Wiley and Sons, Chichester, New York, Brisbane, Toronto and Singapore, 1988.
- [94] A. Goldstuck, "Internet Access in South Africa 2010: A Comprehensive study of the Internet Access Market in South Africa," tech. rep., World Wide Worx, 2010.
- [95] Statistics South Africa, "Census 2011: Key results." online, 2012. http://www.statssa.gov.za/census/census\_2011/census\_products/Census\_2011\_Key\_results.pdf Last Access: August 2016.
- [96] Statistics South Africa, "General household survey 2013." online, June 2014. http://beta2.statssa.gov.za/publications/P0318/P03182013.pdf Last Accessed: August 2016.
- [97] South African Consulate General, "Communications," 2003. http://www.southafricanewyork.net/consulate/telecom.htm Last accessed Jun. 2004.
- [98] D. Kravets, "U.N. Report Declares Internet Access a Human Right." Online, 2011. http://www.wired.com/threatlevel/2011/06/internet-a-human-right Last Accessed: Jan 2017.
- [99] R. Palmer, H. Timmermans, and D. Fay, *From Conflict to Negotiation Nature-based development on the South African Wild Coast*. Pretoria: HSRCPress, 2002.

- [100] I. Brandt, "Models of Internet connectivity for secondary schools in the Grahamstown Circuit," Master's thesis, Rhodes University, Jan. 2006.
- [101] M. Walton, "Is 'wi-fi on steroids' really the next big thing?." Online, March 2006. http://edition.cnn.com/2005/TECH/10/17/wireless.wimax/Last Accessed: Jan 2017.
- [102] T. Shaffer, "Wimax can it stand the test of time?," 2011. http://classes.lt.unt.edu/Summer\_10W\_2011/LTEC\_4550\_020/trs0109/wimaxpaper.pdf Last Accessed: Jan 2017.
- [103] Alvarion, "BreezeMAX 3500: System Manual," tech. rep., Jan. 2005.
- [104] Alvarion, "4motion system manual," tech. rep., Alvarion, February 2011.
- [105] C. Gibbs, "Court: Sprint can begin to kill wimax network this week." online article, February 2016. http://www.fiercewireless.com/wireless/court-sprint-can-begin-to-kill-wimax-network-week Last Accessed: September 2016.
- [106] P. Dampier, "Sprint will shut down clear/4g wimax network by 2015; td-lte upgrade for most cell sites." online article, April 2014. http://stopthecap.com/2014/04/09/sprint-will-shut-down-clear4gwimax-network-by-2015-td-lte-upgrade-for-most-cell-sites/ Last Accessed: September 2016.
- [107] Intel Corporation, "Preboot execution environment (pxe) specification version 2.1," Septmeber 1999.
- [108] Edubuntu Project, "About edubuntu." Online, 2012. http://www.edubuntu.org/about Last Accessed: Oct 2013.
- [109] J. Gall, "Rethinking the computer in education," *Journal of Educational Technology Systems*, vol. 30, no. 4, pp. 379–388, 2002.
- [110] S. Smith and B. Sherwood, "Educational uses of the plato computer system," *Science*, vol. 192, no. 4237, pp. 344–352, 1976.
- [111] P. Hepp, E. Hinostroza, E. Laval, and L. Rehbein, "Technology in Schools: Education, ICT and the Knowledge Society." World Links report, October 2004.
- [112] G. Farrell and S. Isaacs, "Survey of ICT and Education in Africa: A summary report based on 53 country surveys." Info Dev, World Bank, Washington DC, 2007.
- [113] S. Isaacs, "ICT in Education in South Africa, Survey of ICT in Education in Africa: South Africa country report." InfoDev, World Bank, Washington, DC, 2007.
- [114] A. Weiss, "Computing in the clouds," *netWorker Cloud Computing: PC functions move onto the web*, vol. 11, no. 4, pp. 16–25, 2007.
- [115] J. James, "Free software and the digital divide: opportunities and constraints for developing countries," *Journal of Information Science*, vol. 29, no. 1, pp. 25–33, 2003.
- [116] G. Kress and N. Pachler, "Thinking about 'm' in m-learning," in *Mobile learning towards a research agenda*, WLE Centre Occasional papers in Work-based learning 1, WLE Centre, London, 2007.
- [117] M. Armburst, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "Above the Clouds: A Berkeley view of Cloud Computing," Tech. Rep. UCB/EECS-2009-28, University of California Berkeley, 2009.

- [118] RandomWire, "10 future trends of technology predicted." Online, September 2008. Last Accessed: November 2013.
- [119] T.-C. Yang and H.-C. Ho, "An e-learning platform on grid architecture," *Journal of Information Science and Engineering*, vol. 21, pp. 911–928, 2005.
- [120] M. Streicher-Porte, C. Marthaler, H. Boni, M. Schluep, A. Camacho, and L. Hilty, "One laptop per child, local refurbishment or overseas donations? sustainability assessment of computer supply scenarios for schools in colombia," *Journal of environmental management*, vol. 90, pp. 3498–3511, 2009.
- [121] The Shuttleworth Foundation, "The tuxlab project." Website, 2005. Last Accessed: July 2005.
- [122] L. Dalvit, M. Tsietsi, A. Terzoli, P. Maseko, M. Sam, and T. Mapi, "The localisation into isixhosa of the ilanga telephone system," *International Journal of Computing and ICT Research*, vol. 4, no. 3, pp. 236–242, 2008.
- [123] M. Pustakalaya, PAN Localisation: Guide to localisation of Open Source Software. Center for Research in Urdu Language Processing National University of Computer and Emerging Sciences Lahore, Pakistan, International Development Research Center: Walayatsons, Pakistan, 2007.
- [124] A. Souphavanh and T. Karoonboonyanan, *Free/Open Source Software Localisation*. United Nations Development Programme-Asia Pacific Development Information programme (UNDP-APDIP): EL-SEVIER, 2005.
- [125] G. G. Cossa and J. C. Cronje, "Computers for africa: lessons learnt from introducing computers to schools in mozambique," *International Journal of Learning Technology*, vol. 1, no. 1, pp. 84–99, 2004.
- [126] R. Patra, J. Pal, S. Nedevschi, M. Plauche, and U. Pawar, "Usage models of classroom computing in developing regions," in *Proceedings of the International Conference on Information and Communication Technologies and Development*, (Bangalore), pp. 1–10, 2007.
- [127] T. Power, "Using suites of free refurbished computers may cost over four time more than buying and using "state of the art" learning technologies," in *British Educational Research Association Annual Conference 2006*, (University of Warwick, Coventry, UK), September 2006.
- [128] K. L. Kraemer, J. Dedrick, and P. Sharma, "One laptop per child: Vision vs reality," *Communications of the ACM*, vol. 52, no. 6, pp. 66–73, 2009.
- [129] G. Conole and M. Dyke, "What are the aaffordance of information and communication technologies?," ALT-J, Research in Learning Technology, vol. 12, no. 2, pp. 113–124, 2004.
- [130] J. Gibson, The ecological approach to human perception. Houghton Mifflin: Boston, MA, 1979.
- [131] J. McGrenere and W. Ho, "Affordance: Clarifying and evolving a concept," in *Graphics Interface*, (Montral, Canada), May 2000.
- [132] M. Oliver, "The problem with affordance," *E-Learning*, vol. 2, no. 4, pp. 401–413, 2005.
- [133] D. Norman, The design of everyday things. Currency/Doubleday: New York, 1988.
- [134] H. Hartson, "Cognitive, physical, sensory and functional affordances in interactive design," *Behaviour and Information Technology*, vol. 22, no. 5, pp. 315–338, 2003.

- [135] W. Graver, "Technology affordances," in *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology*, pp. 79–84, ACM Press, 1991.
- [136] M. Hammond, "What is an affordance and can it help us understand the use of ict in education?," *Educational and Information Technologies*, vol. 15, no. 3, pp. 205–217, 2010.
- [137] B. Dalgarno and M. Lee, "What are the Affordance of 3-D virtual environments?," *British Journal of Education Technology*, vol. 41, no. 1, pp. 10–32, 2010.
- [138] S. Britain and O. Liber, "A framework for pedagogical evaluation of virtual learning environments." JiSC Technology Applications Programme, Report 41, 1999.
- [139] C. Bowers, *The cultural dimensions of educational computing: understanding the non-neutrality of technology*. Teacher College Press: New York, 1988.
- [140] M. Cox, M. Webb, C. Abbott, B. Blakeley, T. Beauchamp, and V. Rhodes, "Ict and pedagogy, a review of the research literature." ICT in Schools Research and Evaluation Series, 2003. Last Access: January 2008.
- [141] R. Owston, "Contextual factors that sustain innovative pedagogical practice using technology: an international study," *Journal of Educational Change*, vol. 8, no. 1, pp. 61–77, 2007.
- [142] D. Reynolds, D. Treharne, and H. Tripp, "ICT The hopes and the reality," British Journal of Education Technology, vol. 34, no. 2, pp. 151–167, 2003.
- [143] L. Beastall, "Enchanting a disenchanted child: revolutionising the means of education using information and communication technology and e-learning," *British Journal of Sociology of Education*, vol. 27, no. 1, pp. 97–110, 2006.
- [144] B. O'Bannon and S. Judge, "Implementing partnerships across the curriculum with technology," *Journal of Research on Technology in Education*, vol. 37, no. 2, pp. 197–213, 2004-2005.
- [145] T. Levin and R. Wadmany, "Listening to students' voices on learning with information technologies in a rich technology-based classroom," *Journal of Educational Computing Research*, vol. 34, no. 3, pp. 281–317, 2006.
- [146] The Department of Education (DoE), "The Revised National Curriculum Statement for Grades R-9 (Schools)," 2001. http://www.info.gov.za/view/DownloadFileAction?id=70257 Last Accessed: November 2013.
- [147] The Department of Education (DoE), "National Curriculum Statement for grades 10 - 12 (Schools): Computyping (Computer Applications Technology)," 2003. http://www.education.gov.za/LinkClick.aspx?fileticket=MLRz54Uw5tk%3d&tabid=246&mid=594 Last Accessed: November 2013.
- [148] The Department of Education (DoE), "National Curriculum Statement for grades 10
   12 (Schools): Computer Studies/Information Technology/Computer Science," 2003. http://www.education.gov.za/LinkClick.aspx?fileticket=kRQQWosZoLY%3d&tabid=246&mid=594
   Last Accessed: November 2013.
- [149] C. Hodgkinson-Williams, "Dust on the keyboards: Policy gaps in the integration of ict into the south african curriculum," in 8th IFIP World Conference on Computers in Education: What Works?, 2005.

- [150] C. Hodgkinson-Williams, I. Sieborger, and A. Terzoli, "Enabling and constraining ict practice in seconday schools: case studies in south africa," *Internal Journal of Knowledge and Learning*, vol. 3, no. 2/3, pp. 171–190, 2007.
- [151] M. Bialobrzeska and S. Cohen, "Managing ICTs in South African Schools: A Guide for School Principals." South African Institute for Distance Education (SAIDE), 2005.
- [152] N. Dabbagh, "Pedagogical models for e-learning: A theory-based design framework," *International Journal of Technology in Teaching and Learning*, vol. 1, no. 1, pp. 25–44, 2005.
- [153] M. Koehler and P. Mishra, "What is technological pedagogical content knowledge?," *Contemporary Issues in Technology and Teacher Education*, vol. 9, no. 1, pp. 60–70, 2009.
- [154] S. Howard and K. Maton, "Theorising knowledge practices: a missing piece of the educational technology puzzle," *Research in Learning Technology*, vol. 19, no. 3, pp. 191–206, 2011.
- [155] S. Papert, "A critique of technocentrism in thinking about the school of the future," 1987.
- [156] R. Glaser, "Education and thinking: The role of knowledge," American Psychology, vol. 39, no. 2, pp. 93–104, 1984.
- [157] R. Putman and H. Borko, "What do new views of knowledge and thinking have to say about research on teacher learning?," *Educational Researcher*, vol. 29, no. 1, pp. 4–15, 2000.
- [158] L. Shulman, "Those who understand: Knowledge growth in teaching," *Educational Researcher*, vol. 15, no. 2, pp. 4–14, 1986.
- [159] L. Shulman, "Knowledge and teaching: Foundations of the new reform," *Harvard Education Review*, vol. 57, no. 1, pp. 1–22, 1987.
- [160] C. Graham, J. Borup, and N. Smith, "Using TPACK as a framework to understand teacher candidates" technology integration decisions," *Journal of Computer Assisted Learning*, vol. 28, pp. 530–546, 2012.
- [161] J. Harris and M. Hofer, "Instruction planning activity types as vehicles for curriculum-based TPACK development," in *Research highlights in Technology and Teacher Education 2009* (C. Maddux, ed.), pp. 99–108, Society for Information Technology in Teacher Education (SITE) Chesapeake, VA, 2009.
- [162] J. Harris and M. Hofer, "Technological pedagogical content knowledge (tpack) in action: a descriptive study of secondary teachers' curriculum-based, technology-related instructional planning," *Journal of Research on Technology in Education*, vol. 43, pp. 211–229, 2011.
- [163] M. Manfra and T. Hammond, "Teachers' instructional choices with student-created digital documentaries?: case studies," *Journal of Research on Technology in Education*, vol. 41, pp. 223–245, 2008.
- [164] J. Polman, "Activity structures for project-based teaching and learning: Design and adaptation of cultural tools," in Annual meeting of the American Educational Research Association, San Diego, CA, April 1998.
- [165] M. Windschitl, "What types of knowledge do teachers us to engage learners in "doing science?"." Paper commissioned by the National Academy of Science. Washington, DC: Board of Science Education, 2004.

- [166] J. Lemke, "Social semiotics and science education," *The American Journal of Semiotics*, vol. 5, no. 2, pp. 217–232, 1987.
- [167] Equal Education, "School infrastructure." Website, 2014. http://old.equaleducation.org.za/campaigns/minimumnorms-and-standards Last Access: Nov 2016.
- [168] Department of Education, "South african schools act 84 of 1996: Regulations relating to minimum uniform norms and standards for public school infrastructure." Government Gazette 37081, November 2013.
- [169] S. Wheller and G. Goode, "tuxlab cookbook." PDF, January 2007. http://www.shuttleworthfoundation.org/architect/wpcontent/uploads/2010/06/tuXlab%20Cookbook.pdf Last Accessed: Oct 2013.
- [170] Wikipedia Foundation, Inc, "Fat client." Online, September 2013. http://en.wikipedia.org/wiki/Fat\_client Last Accessed: Oct 2013.
- [171] Webopedia, "The differences between thick and thin client hardware." Online, 2013. http://www.webopedia.com/DidYouKnow/Hardware\_Software/2006/thin\_client.asp Last Accessed: Oct 2013.
- [172] Wikipedia Foundation, Inc, "Thin client." Online, October 2013. http://en.wikipedia.org/wiki/Thin\_client Last Accessed: Oct 2013.
- [173] Wikipedia Foundation, Inc, "Client (computing)." Website, August 2016. https://en.wikipedia.org/wiki/Client\_(computing) Last Accessed: September 2016.
- [174] D. Colbert, "Hybrid computing, not cloud computing, is the future of technology." online article, March 2010. http://www.techrepublic.com/blog/techrepublic-out-loud/hybrid-computing-not-cloudcomputing-is-the-future-of-technology/ Last Accessed: September 2016.
- [175] The Ubuntu Documentatio Team, "UbuntuLTSP/FatClients." Online, October 2016. https://help.ubuntu.com/community/UbuntuLTSP/FatClients Last Accessed: Jan 2017.
- [176] C. Marthaler, "Computers for schools: Sustainability assessment of supply strategies in developing countries, a case study in colombia," master's thesis, Swiss Federal Institute of Technology, Zurich, April 2008. Last Accessed: Oct 2013.
- [177] OECD, "Understanding the digital divide." Paris, Organisation for Economic Co-operation and Development, 2001.
- [178] J. James, "Low-cost information technology in developing countries: current opportunities and emerging possibilities," *Habitat International*, vol. 26, pp. 21–31, 2002.
- [179] J. Evans, "One laptop per child: What went wrong." The Walrus magazine, January 2009. Last Access: Oct 2013.
- [180] M. Streicher-Portem, R. Widmer, A. Jain, H.-P. Bader, R. Scheidegger, and S. Kytzia, "Key drivers of the e-waste recycling system: Assessing and modelling e-waste processing in the informal sector in Delhi," *Environmental Impact Assessment Review*, vol. 25, pp. 472–491, 2005.
- [181] I. Foster and C. Kesselman, *The Grid 2: Blueprint for a new computing infrastructure*. Morgan Kaufman Publishers: Amsterdam, 2004.

- [182] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Linder, "A break in the clouds: Towards a cloud definition," ACM SIGCOMM Computer Communication Review, vol. 39, no. 1, pp. 50–55, 2009.
- [183] F. Berman, G. Fox, and A. Hey, *Grid Computing: Making the global infrastructure a reality*. Wiley: West Sussex, 2003.
- [184] D. Minoli, A networking approach to grid computing. John Wiley and Son, Inc: New Jersey, 2005.
- [185] "Teragrid." Online. https://www.teragrid.org/ Last Accessed: Oct 2013.
- [186] D. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer, "SETI@home: An experiment in public-resource computing," *Communications of the ACM*, vol. 45, no. 11, pp. 56–61, 2002.
- [187] J. Hennessy and D. Patterson, *Computer Architecture: A qualitative approach*. Morgan Kaufman Publishers: San Francisco, 2003.
- [188] M. J. Ma, C.-L. Wang, and F. C. Lau, "Jessica: Java-enabled single-system-image computing architecture," *Journal of Parallel and Distributed Computing*, vol. 60, pp. 1194–1222, 2000.
- [189] Wikimedia Foundation, Inc, "Round-Robin DNS." Online, October 2013. http://en.wikipedia.org/wiki/Round-robin\_DNS Last Accessed: Oct 2013.
- [190] L. AndrBarroso, J. Dean, and U. Hzle, "Web search for a planet: The google cluster architecture," *IEEE MICRO*, vol. 23, no. 2, pp. 22–28, 2003.
- [191] D. F. Carr, "How Google Works." Web article, 2006. http://www.baselinemag.com/c/a/Projects-Networks-and-Storage/How-Google-Works-[1]/ Last Accessed May 2008.
- [192] B. J. Walker, "Open Single System Image (openSSI) Linux Cluster Project," 2004. Last Accessed May 2008.
- [193] C. Morin, P. Gallard, R. Lottiaux, and G. Vallee, "Towards an efficient single single image cluster operating system," *Future Generation Computer Science Systems*, vol. 20, pp. 505–521, 2004.
- [194] D. Kusnetzky, "Single system image clusters: an idea whose time has come and gone?." Online article, May 2007. http://www.zdnet.com/article/single-system-image-clusters-an-idea-whose-timehas-come-and-gone/ Last Accessed: September 2016.
- [195] B. Hayes, "Cloud computing," Communications of the ACM, vol. 51, no. 7, pp. 9–11, 2008.
- [196] Wikipedia Foundation, Inc, "Desktop virtualization." Online, December 2016. https://en.wikipedia.org/wiki/Desktop\_virtualization Last Accessed: January 2017.
- [197] D. G. Chandra and M. D. Borah, "Cost benefit analysis of cloud computing in education," in 2012 International Conference on Computing, Communication and Applications (ICCCA), IEEE Xplore, IEEE Xplore, 2012. Last Accessed: Jan 2017.
- [198] A. Paventhan, H. Krishna, V. Gayathri, and N. Pahuja, "Towards educational school cloud for india: A case study with kendriya vidyalayas," in *Humanitarian Technology Conference (R10-HTC), 2014 IEEE Region 10*, IEEE Xplore, IEEE Xplore, 2014. Last Accessed: Jan 2017.
- [199] Wikimedia Foundation, Inc, "Edubuntu." Online, October 2013. http://en.wikipedia.org/wiki/Edubuntu Last Accessed: Oct 2013.

- [200] Wikimedia Foundation, Inc, "Linux terminal server project." Online, June 2013. http://en.wikipedia.org/wiki/Linux\_Terminal\_Server\_Project Last Accessed: Oct 2013.
- [201] LTSP.org, "Ltsp concepts." Online, January 2013. http://wiki.ltsp.org/wiki/Concepts Last Accessed: Oct 2013.
- [202] J. Pfaffman, "Transforming instruction without training: A case study of the k12 linux terminal server project," in *Proceedings of World Conference on Educational Multimedia*, Hypermedia and *Telecommunications 2007* (C. M. J. Seale, ed.), pp. 363–366, 2007.
- [203] Intel Corportation, "Wired for management baseline version 2.0 release," December 1998.
- [204] H. Stern, M. Eisler, and R. Labiaga, *Managing NFS and NIS*. O'Reilly Media, second ed., 2001.
- [205] K. Zeilenga, "Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map," 2006.
- [206] G. Carter, LDAP System Administration. O'Reilly Media, 2003.
- [207] T. A. Howers, M. C. Smith, and G. S. Good, Understanding and Deploying LDAP Directory Services. New Riders Publishing, 1998.
- [208] G. Good, "Request for comments: 2849, the ldap data interchange format (ldif) technical specification," rfc, The Internet Society, 2000.
- [209] JExplorer, "Jexplorer, java ldap browser." http://jxplorer.org/ Last Accessed: June 2016.
- [210] GONICUS GmbH, "GOsa2: Welcome to the GOsa Project." https://oss.gonicus.de/labs/gosa Last Accessed: June 2016.
- [211] php LDAP Admin, "php ldap admin: Web-based ldap browser to manage your ldap server." http://phpldapadmin.sourceforge.net/wiki/index.php/Main\_Page Last Accessed: June 2016.
- [212] Webmin, "Webmin," 2013. http://www.webmin.com/index.html Last Accessed: June 2016.
- [213] Wikimedia Foundation, Inc, "Domain Name System," 2011. http://en.wikipedia.org/wiki/Domain\_Name\_System Last accessed: May 2011.
- [214] C. Lui and P. Albitz, DNS and BIND. O'Reilly Media, third ed., 1998.
- [215] G. Lehey, The Complete FreeBSD. O'Reilly Media, Inc, fourth ed., 2003.
- [216] L. Hardesty, "Picture-driven computing: New research could enable computer programming based on screen shots, not just code." Online, January 2010. http://web.mit.edu/newsoffice/2010/screenshots-0120.html Last Accessed: November 2013.
- [217] T. Yeh, T.-H. Chang, and R. C. Miller, "Sikuli: Using GUI Screenshots for Search and Automation," in Proceedings of the 22nd annual ACM symposium on User interface software and technology, UIST'09, 2009.
- [218] T.-H. Chang, T. Yeh, and R. C. Miller, "GUI Testing Using Computer Vision," in *Proceedings of the 28th international conference on Human factors in computing systems, CHI '10*, 2010.

- [219] S. Khan, "Monitoring South Africa Connect: Creating Opportunities, Ensuring Inclusion," ResearchICTAfrica.net, 2015. https://www.itu.int/en/ITUtech. rep., D/Statistics/Documents/events/ethiopia2015/4-RIA\_SA\_Broadband\_Policy.pdf Last Accessed: October 2016.
- [220] S. Qoza, ""SA Connect" Implementing the Broadband Policy and Strategy." Website, April 2014. http://www.dtps.gov.za/mediaroom/media-statements/312-%E2%80%9Csa-connect%E2%80%9D-%E2%80%93-implementing-the-broadband-policy-and-strategy.html Last Accessed: September 2016.
- [221] A. Terzoli and S. Gumbo, "Siyakhula (we are growing) living lab project." May 2010.
- [222] A. M. Nungu, *Towards Sustainable Broadband Communication in Under-served Areas : A Case Study from Tanzania*. Doctoral thesis, Royal Institute of Technology, December 2011.
- [223] I. Sieborger and A. Terzoli, "Field testing the Alvarion BreezeMAX as a last mile access technology," in 10th Annual Southern Africa Telecommunication and Networking Applications Conference, Sugar Beach, Mauritius, Sept. 2007.
- [224] A. Terzoli and S. Gumbo, "The siyakhula living lab and reed house systems a brief overview," May 2012. http://siyakhulall.org/sites/default/files/SiyakhulaLast Accessed: Jan 2017.
- [225] Genesis Analytics: James Hodge, Sarah Truen, Brent Cloete, and Gilberto Biacuana, "South african telecommunications prices: An updated international price comparison, with regulatory recommendations." Occasional paper number 3, November 2007.
- [226] Genesis Analytics, "Telecommunications prices in South Africa: An international peer group comparison." Occastional Paper No 1/2005, Apr. 2005.
- [227] M. Jenson, "The African Internet A status report," 2002. http://www3.sn.apc.org/africa/afstat.htm#costs Last accessed Jun. 2004.
- [228] SAPA, "Copper theft sting nets 5," 2003. http://www.news24.com/News24/South\_Africa/News/0,,2-7-1442\_1386374,00.html Last accessed Sep. 2004.
- [229] SAPA, "Four jailed for copper wire theft," 2002. http://www.dispatch.co.za/2002/08/30/southafrica/ DCOPP.HTM Last accessed Sep. 2004.
- [230] "Mobile industry finds home in Africa," 2003. http://www.suntimes.co.za/2003/05/18/business/surveys/ survey08.asp Last accessed Sep. 2004.
- [231] R. Lowe and C. Arevalo-Lowe, "Accessing the Internet DSL, Cable Modem or Dialup," 2004. http://www.webhero.org/System/access.htm Last accessed Jun. 2004.
- [232] K. DeMartino, "ISDN and the Internet," Computer Networks, vol. 31, pp. 2325–2339, 1999.
- [233] J. Cioffi, P. Silverman, and T. Starr, "Digital subscriber lines," *Computer Networks*, vol. 31, pp. 283–311, 1999.
- [234] A. Gumaste and T. Antony, *First Mile Access Networks and Enabling Technologies*. 800 East 96th Street, Indianapolis, IN 46240 USA: Cisco Press, 2004.
- [235] W. Lehr and L. W. McKnight, "Wireless Internet access: 3G vs. WiFi?," *Telecommunications Policy*, vol. 27, pp. 351–370, 2003.

- [236] LAN/MAN Standards Committee of the IEEE Computer Society, "Information technology -Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," tech. rep., IEEE-SA Standards Board - The Institute of Electrical and Electronics Engineers, Inc., 1999.
- [237] F. Bellotti, A. D. Gloria, D. Grosso, and L. Noli, "WLESS-frame: a simulation-based development environment for 802.11 stations," *Computer Networks*, vol. 36, pp. 625–641, 2001.
- [238] N. Baghaei and R. Hunt, "Review of quality of service performance in wireless LANs and 3G multimedia application services," *Computer Communications*, vol. 27, pp. 1684–1692, 2004.
- [239] LAN/MAN Standards Committee of the IEEE Computer Society, "Supplement to IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Higher-Speed Physical Layer Extension in the 2.4 GHz Band," tech. rep., IEEE-SA Standards Board - The Institute of Electrical and Electronics Engineers, Inc., 1999.
- [240] LAN/MAN Standards Committee of the IEEE Computer Society, "IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz Band," tech. rep., IEEE-SA Standards Board - The Institute of Electrical and Electronics Engineers, Inc., 2003.
- [241] LAN/MAN Standards Committee, "Ieee standard for information technology telecommunications and information exchange between systems - local and metropolitan area networks - specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. amendment 5: Enhancements for higher throughput," tech. rep., IEEE Computer Society, October 2009.
- [242] Wireless First, "Advantages of Wireless Data Technology," June 1999. Last accessed Apr. 2005.
- [243] "Telecommunications Act, 1996." No. 103 of 1996, 2001. Last accessed Jun. 2004.
- [244] LAN/MAN Standards Committee of the IEEE Computer Society, "802.16.2: IEEE recommended practice for Local and Metropolitan area networks - Coexistence of fixed broadband wireless access systems," tech. rep., The IEEE Computer Society - The Institute of Electrical and Electronics Engineers, Inc., 2004.
- [245] C. Gabriel, "WiMAX: The Critical Wireless Standard." Blue Print WiFi Report I, Oct. 2003.
- [246] Alvarion, "Introducing WiMAX, the next broadband wireless revolution." white paper, 2004.
- [247] Worldwide Interoperability for Microwave Access Forum, "IEEE 802.16a Standard and WiMAX Igniting Broadband Wireless Access." white paper, 2003.
- [248] D. O'Shea, "A Standard argument: Why will WiMAX rule?," Telephony, pp. 4–9, May 2004.
- [249] D. Sweeny, WiMAX operators manual: Building 802.16 Wireless Networks. 2560 Ninth street, Suite 219, Berkley, CA 94710: Apress, 2004.

- [250] B. Deva, "Lte vs wimax next generation telecommunication networks," June 2011.
- [251] S. G. Muhammad Nadeem Khan, "The wimax 802.16e physical layer model," in *IET International Conference on Wireless, Mobile and Multimedia Networks*, pp. 117–120, January 2008.
- [252] Z. Abichar, Y. Peng, and J. M. Chang, "Wimax: The emergence of wireless broadband," *IT Professional*, vol. July/August, pp. 44–48, 2006.
- [253] D. Eberle, "Lte vs wimax: 4th generation telecommunication networks," January 2011.
- [254] Wikipedia, the free encyclopedia, "IEEE 802.16." online encyclopedia, 2016. http://en.wikipedia.org/wiki/IEEE\_802.16 Last accessed Sept. 2016.
- [255] LAN/MAN Standards Committee of the IEEE Computer Society and the IEEE Microwave Theory and Techniques Society, "IEEE Standard for Local and Metropolitan are networks, Part 16: Air Interface for Fixed and Braodband Wireless Access Systems," tech. rep., The IEEE Computer Society - The Institute of Electrical and Electronics Engineers, Inc., 2004.
- [256] LAN/MAN Standards Committee of the IEEE Computer Society and the IEEE Microwave Theory and Techniques Society, "IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems, Amendment 2: Physical and Medium Access Control Layers for combined fixed and mobile operation in Licenced bands," tech. rep., The IEEE Computer Society - The Institute of Electrical and Electronics Engineers, Inc, 2005.
- [257] S. J. Vaughan-Nichols, "Achieving wireless broadband with wimax," *Computer*, vol. 37, pp. 10–13, June 2004.
- [258] C. Hoymann, "Analysis and performance evaluation of the OFDM-based metropolitan area network IEEE 802.16," *Computer Networks*, vol. 49, pp. 341–363, 2005.
- [259] S. Paul, "Long term evolution (lte) and ultra-mobile broadband (umb) technologies for broadband wireless access," May 2008.
- [260] Wikipedia Foundation, Inc, "LTE Advanced." Online, October 2016. https://en.wikipedia.org/wiki/LTE\_Advanced Last Accessed: October 2016.
- [261] Wikipedia Foundation, Inc, "High speed packet access." Online, September 2016. https://en.wikipedia.org/wiki/High\_Speed\_Packet\_Access Last Accessed: October 2016.
- [262] S. Carew, "Qualcomm halts umb project, sees no major job cuts." online article, November 2008. http://www.reuters.com/article/qualcomm-umb-idUSN1335969420081113?rpc=401& Last Accessed: October 2016.
- [263] Z. Abichar, J. M. Chang, and C.-Y. Hsu, "Wimax vs lte: who will lead the broadband mobile internet?," *IT Professional*, vol. 12, pp. 26–32, July 2010.
- [264] S. Sesia, I. Toufik, and M. Baker, *LTE: The UMTS Long Term Evolution, from Theory to Practice*. Wiley: West Sussex, 2011.
- [265] M. Hadjitheodosiou, A. Ephremides, and D. Friedman, "Broadband access via Satellite," *Computers Networks*, vol. 31, pp. 353–378, 1999.
- [266] D. Meldrum, "Developments in Satellite Communication Systems," Oct. 2003.

- [267] M. Bever, J. Freitag, S. Linsky, J. Myers, R. Nuber, J. Prieto, and E. Wiswell, "Fast-Packet Vs. Circuit Switch and Bent Pipe Satellite Network Architectures," in *Fourth Ka-Band Utilization Conference*, (Venice, Italy), Nov. 1998.
- [268] N. Abramsom, "Internet access using VSATs," *IEEE Communications Magazine*, vol. 38, no. 7, pp. 60–69, 2000.
- [269] W. Sun, M. Sweeting, and A. da Silva Curiel, "LEO Satellite Constellation for regional Communications," *Surrey Satellite Technology, University of Surrey*, 1996. Last accessed Jun. 2004.
- [270] F. Daoud, "Hybrid Satellite/Terrestrial networks integration," *Computer Networks*, vol. 34, pp. 781–797, 2000.
- [271] L. Franck and G. Maral, "Signaling for inter-satellite link routing in broadband non-GEO satellite systems," *Computer Networks*, vol. 39, pp. 79–92, 2002.
- [272] R. Muller, "Satellite broadband pricing comparison." Online, January 2012. Last Accessed: Oct 2013.
- [273] Telkom SA Ltd, "Telkom SA Ltd." Webpage, 2005. http://www.telkom.co.za/ Last accessed Jun. 2005.
- [274] Telkom SA Ltd, "Telkom Internet powered by Satellite," 2004. http://www.telkom.co.za/ Last accessed Sep. 2004.
- [275] Sentech, "Sentech Connecting you," 2004. http://www.sentech.co.za/ Last accessed Jun. 2004.
- [276] Sentech, "Sentech Connecting you: VStarInternet." Webpage, 2005. http://www.sentech.co.za/index.php?module=ContentExpress&func=display&ceid=15&bid=22& btitle=ProductsAndServices&meid=25 Last accessed Jun. 2005.
- [277] MyBroadband, "Yahclick affordable satellite broadband: pricing and other details." Online, May 2012. http://mybroadband.co.za/news/broadband/50823-yahclick-affordable-satellite-broadbandpricing-and-other-details.html Last Accessed: Oct 2013.
- [278] Satellite Providers, "List of ISP in South Africa." Online, 2016. http://www.allisps.com/en/list-ofall-services/South-Africa,c-1 Last Accessed: October 2016.
- [279] Iridium Satellite LLC: C. Lloyd, "Iridium Issues 2005 Mid-Year Report." Press Release, 2005. Last accessed Jan. 2006.
- [280] Telkom SA, "Telkom Tariff List 2005." http://www.telkom.co.za/pls/portal/docs/page/downloads/tarifflist\_e\_05% Last accessed Mar. 2005.
- [281] BeginLinux.com, "LTSP on Ubuntu 8.04," 2010. http://beginlinux.com/server\_training/126-linuxterminal-server/1053-ltsp-on-ubuntu-804 Last accessed: May 2011.
- [282] Canonical, "Edubuntu," 2006. http://www.edubuntu.org/ Last Accessed: Nov 2016.
- [283] Kerlabs, INRIA, "Kerrighed: Linux clusters made easy." Web site, 2008. http://kerrighed.org/ Last Accessed May 2008.
- [284] Ulteo, "Ulteo." Online, 2016. http://www.ulteo.com/ Last Accessed: Jan 2017.

- [285] FOSS-Cloud, "Foss-cloud." Online, 2015. http://www.foss-cloud.org/en/wiki/FOSS-Cloud Last Accessed: Jan 2017.
- [286] J. Carter, "How do ltsp fat clients work?." online, November 2010. http://jonathancarter.org/2010/11/24/how-do-ltsp-fat-clients-work/ Last Accessed: November 2013.
- [287] Squid Cache, "Squid: Optimising web delivery." Online, May 2013. http://www.squid-cache.org/ Last Accessed: Jan 2017.
- [288] T. Oetiker, "What is MRTG." Online, January 2012. http://oss.oetiker.ch/mrtg/doc/mrtg.en.html Last Accessed: Jan 2017.
- [289] Canonical, "Edubuntu appguide." Online, October 2015. https://wiki.ubuntu.com/Edubuntu/AppGuide Last Accessed: Jan 2017.
- [290] Canonical, "Edubuntu 12.04 14.04 lts installation guide." Online, 2012. https://www.edubuntu.org/documentation/12.04/installation-guide Last Accessed: Jan 2017.
- [291] Wikimedia Foundation, Inc, "Network file system." Online, November 2013. http://en.wikipedia.org/wiki/Network\_File\_System Last Accessed: November 2013.
- [292] J. D. Sloan, *High Performance Linux Clusters with OSCAR, Rocks, openMosix and MPI.* O'Reilly Media, 2004.
- [293] Wikipedia Foundation, Inc, "Ulteo open virtual desktop." Online, August 2015. https://en.wikipedia.org/wiki/Ulteo\_Open\_Virtual\_Desktop Last Accessed: Jan 2017.

### Appendix A

# **SLL Survey: Permission letter to District office**

Below is a copy of the permission letter sent to the Department of Basic Education district office in Idutywa in order to ask permission to conduct the survey with the six selected schools and 12 teachers (two from each school); the letter explains the nature and purpose of the research and survey. The letter has been anonymised here in order to protect the identity of the school and teacher participants. In the original sent to the district offices the real names of the schools were included in order that the Education District Officer of the Idutywa office knew exactly which schools we were approaching.

Department of Computer Science Hamilton Building Rhodes University Grahamstown 6140

09 January 2017

Mrs Magwentshu Department of Education: Idutywa Office Private Bag X1203 Idutywa 5000

Dear Mrs Magwentshu

Permission to conduct research at schools in your district

Ingrid Siebörger, a member of the Siyakhula Living Lab and a PhD candidate, is undertaking an investigation into appropriate Information Communication Technologies (ICTs) for use in teaching and learning in South African schools as part of her degree and to support the growth and development of the Siyakhula Living Lab. The study hopes to identify appropriate ICT architectures or configurations that best support teachers (and consequently learners) in the process of teaching and learning. The study is being supervised by Prof Alfredo Terzoli of the Computer Science department at Rhodes University, and Prof Cheryl Hodgkinson-Williams of the Centre for Educational Technology at the University of Cape Town.

We would like to approach the following schools (via their principals) within your district:

- FH1FH2
- FH3
- FH4
- FH 5
- FH6

Please note that not all the schools listed here will necessarily take part in the study. Each principal will be approached and can opt to not have his or her school included in the study. Thus, I am requesting permission to approach more schools than I require for the study in order to ensure that I am able to work with at least four schools in your district.

These schools have been chosen as they meet the two necessary requirements of the project, namely, that they have some form of ICTs available in their school and they have at least one teacher who has some form of ICT literacy and teaching integration training.

From this study it is hoped to gain a better understanding of how and why teachers currently use ICTs for teaching and learning, how effective the ICT architectures under investigation

are at supporting the pedagogical needs of teachers, what these architectures offer or afford teachers in terms of their pedagogic needs. It will hopefully inform future work mapping these affordances to specific pedagogic activities in teaching and learning.

The research method adopted is a survey that will require:

- A **questionnaire** to be completed by the *person in charge of the IT equipment* at each school as well as a **one other teacher** who is not necessarily in charge of the facilities.
- A follow-up **interview** with both teachers who completed the questionnaire from every school participating in the study.
- An **audit** of the **ICT infrastructure** at each of the schools to ascertain technical specifications, to be conducted by Ingrid directly.

Further negotiations on how and when the research could best proceed will be done on my first visit to each of the schools.

The findings from this study will be sent to yourselves as representatives of the Department of Education in order to provide you with the findings of the study regarding appropriate ICT architectures for teaching and learning, as well as highlight any other pertinent issues teachers may be facing in terms of using ICTs appropriately for teaching and learning. In addition, I hope to run a post study workshop with teachers where I will discuss the findings with them and also begin to engage with them regarding how to choose appropriate ICTs to support their teaching and learning requirements. I hope that this workshop will culminate in teaching aids that I hope to develop in future research work as a direct result of this study. This guide/teaching aid would then be made available to the participating teachers and schools as well as yourselves for evaluation and distribution to other schools in your district.

Thanking you in anticipation for working together in the interests of improving the quality of education in your district. If you have any further questions about the project, please feel free to e-mail or phone me (Ingrid Siebörger) or either of the two supervisors, Prof Alfredo Terzoli or Prof Cheryl Hodgkinson-Williams. A list of our contact details is attached for your convenience.

Yours sincerely

Ingrid Siebörger

### Contact details of researchers

Research Supervisors:

#### Prof Cheryl Hodgkinson-Williams

Centre for Educational Technology University of Cape Town Rondebosch 7700

 Tel:
 +27 21 650 5030

 Fax:
 +27 21 650 5045

 e-mail:
 cheryl.hodgkinson-williams@uct.ac.za

Prof Alfredo Terzoli

Telkom Centre of Excellence Department of Computer Science Rhodes University Grahamstown 6140

 Tel:
 +27 46 603 8602

 Fax:
 +27 46 636 1915

 e-mail:
 a.terzoli@ru.ac.za

PhD Candidate researcher:

### **Mrs Ingrid Siebörger**

PhD Candidate Telkom Centre of Excellence Dept of Computer Science Rhodes University Grahamstown 6140

Tel: e-mail: 073 246 6950 i.sieborger@ru.ac.za

### **Appendix B**

# **SLL Survey: Permission letter to school Principal**

Below is a copy of the permission letter sent to the six school principals, for each of the selected schools, in order to explain the nature and purpose of the survey and request their permission to run the survey at their schools with two of their teachers.

Department of Computer Science Hamilton Building Rhodes University Grahamstown 6140

09 January 2017

Mr X Principal FH3

Dear Mr X

#### Permission to conduct research at your school

Ingrid Siebörger, a member of the Siyakhula Living Lab and a PhD candidate, is undertaking an investigation into appropriate Information Communication Technologies (ICTs) for use in teaching and learning in South African schools as part of her degree and to support the growth and development of the Siyakhula Living Lab. The study hopes to identify appropriate ICT architectures or configurations that best support teachers (and consequently learners) in the process of teaching and learning. The study is being supervised by Prof Alfredo Terzoli of the Computer Science department at Rhodes University, and Prof Cheryl Hodgkinson-Williams of the Centre for Educational Technology at the University of Cape Town.

From this study it is hoped to gain a better understanding of how and why teachers currently use ICTs for teaching and learning, how effective the ICT architectures under investigation are at supporting the pedagogical needs of teachers, what these architectures offer or afford teachers in terms of their pedagogic needs and possibly inform future work on mapping these affordances to specific pedagogic activities in teaching and learning.

The research method adopted is a survey that will require:

- A questionnaire to be completed by the *person in charge of the IT equipment* as well as a **one other teacher** who is not necessarily in charge of the facilities.
- o A follow-up interview with both teachers who filled in the questionnaire
- An audit of the ICT infrastructure at your school to ascertain technical specifications, to be conducted by Ingrid directly.

Further negotiations on how and when the research could best proceed will be done on my first visit to your school.

The findings from this study will be sent to the Department of Education to provide them with information regarding appropriate ICT architectures for teaching and learning, as well as highlight any other pertinent issues teachers are facing in terms of using ICTs appropriately for teaching and learning. I also hope to run a post research study workshop with teachers from all the schools who participated in order to provide feedback from this study, as well as to begin to engage around the issues of how to choose appropriate ICTs to meet the teaching and learning needs of the classroom. This workshop will hopefully culminate in a guide for teachers that will be made available to your school as well as the DoE. In addition, the findings and any other teaching aids that may result from this research can be made available to each school that participated in the study, should it be required or requested.

Thanking you in anticipation for working together in the interests of improving the quality of education. If you have any further questions about the project, please feel free to e-mail or phone me (Ingrid Siebörger) or either of the two supervisors, Prof Alfredo Terzoli or Prof Cheryl Hodgkinson-Williams. A list of our contact details is attached for your convenience.

Yours sincerely

Ingrid Siebörger

### Contact details of researchers

Research Supervisors:

#### Prof Cheryl Hodgkinson-Williams

Centre for Educational Technology University of Cape Town Rondebosch 7700

 Tel:
 +27 21 650 5030

 Fax:
 +27 21 650 5045

 e-mail:
 cheryl.hodgkinson-williams@uct.ac.za

Prof Alfredo Terzoli

Telkom Centre of Excellence Department of Computer Science Rhodes University Grahamstown 6140

 Tel:
 +27 46 603 8602

 Fax:
 +27 46 636 1915

 e-mail:
 a.terzoli@ru.ac.za

PhD Candidate researcher:

### **Mrs Ingrid Siebörger**

PhD Candidate Telkom Centre of Excellence Dept of Computer Science Rhodes University Grahamstown 6140

Tel: e-mail: 073 246 6950 i.sieborger@ru.ac.za

## Appendix C

# **SLL Survey: Consent forms**

Below are copies of the consent forms that were presented to the school principals (section C.1) and to the school teachers (section C.2).

### C.1 School Principal consent form

### CONSENT FORM

Project Title	An investigation into appropriate ICT architectures for teaching and learning in South African schools			
Project Description	In this project the researcher hopes to identify suitable ICT architectures (configurations) for teaching and learning in South African schools. In so doing, the researcher will hopefully highlight appropriate ICTs for different classroom activities with the aim of producing or contributing to a guide for teachers in the future.			
Researcher	Mrs Ingrid Siebörger			
• I have re	eceived information about this research project.			
• I unders involver	tand the purpose of the research project and mine and my school's nent in it.			
<ul> <li>I unders any stag</li> </ul>	• I understand that I may withdraw my school from the research project at any stage.			
• I understand that participation in this user study is done on a voluntary basis.				
• I understand that while information gained during the study may be published, neither I nor my school will be identified and my personal and the school's details will remain confidential.				
Name of Principa	l			
Signed Date				
I have provided information about the research to the school principal and believe that he/she understands what is involved.				
Researcher's signature and date				

### C.2 Teachers' consent form

### **CONSENT FORM**

Project Title         An investigation into appropriate ICT architectures for teach and learning in South African schools		
Project Description	In this project the researcher hopes to identify suitable ICT architectures (configurations) for teaching and learning in South African schools. In so doing, the researcher will hopefully highlight appropriate ICTs for different classroom activities with the aim of producing or contributing to a guide for teachers in the future.	
Researcher	Mrs Ingrid Siebörger	
• I have red	ceived information about this research project.	
• I underst	and the purpose of the research project and my involvement in it.	
• I underst	and that I may withdraw from the research project at any stage.	
• I understa basis.	and that participation in this user study is done on a voluntary	
• To the be me from	est of my knowledge I have no physical impediments that will stop completing this study.	
• I underst published confident	and that while information gained during the study may be d, I will not be identified and my personal details will remain tial.	
Name of participa	nt	
Signed	Date	

I have provided information about the research to the research participant and believe that he/she understands what is involved.

Researcher's signature and date

### **Appendix D**

## **SLL Survey: Questionnaire**

Below is a copy of the questionnaire that was sent out to the 12 participating teachers during the SLL survey, which is reported on Chapter 6.

#### Questionnaire: ICT Architectures used in schools

This research aims to understand which ICT architectures will best support the educational (pedagogical) needs/vision of teachers in South African schools, while being constrained by costs and requiring ease of maintenance. It is hoped that this questionnaire will provide insight into the current configuration of ICT facilities within schools, while highlighting some of the uses of those facilities by teachers. In addition, we hope to identify those factors which both negatively and positively impact on the use of ICT facilities for the purposes of teaching and learning. Based on the responses provided in this questionnaire, we may approach you for follow up interviews in order to clarify points and explore deeper ideas raised here.

Please note: Your participation is completely voluntary and you can refuse to participate in answering any of the questions in this questionnaire. In addition, all information retrieved during the course of this study /questionnaire will be treated as strictly confidential. Data that may be reported on in the research report will not include information that identifies you or your school as respondents in the study.

#### Section A: Personal information

Previous research findings in the literature suggest that personal demographics and social and economic backgrounds are factors that influence the use of ICT. For this reason you are kindly requested to provide the following personal information. Please note that this information will remain confidential and used purely for comparative purposes.

1.	Name and Surname:				
2.	Age:	□ 20 - 24 □ 40 - 44 □ 60 - 65	□ 25 - 29 □ 45 - 49 □ > 65	□ 30 – 34 □ 50 – 54	□ 35 - 39 □ 55 - 59
3.	Gender:	🗆 Male	🗆 Female		
4.	Please specify the name of the school at which you teach:				
5.	How many years have you been teaching?	□ < 5 years □ 16 - 20 ye □ 26 - 30 ye □35 - 40 ye	□ 5 – 10 ears □ ears □ ears □	years □ 10 21 – 25 years 31 – 35 years > 40 years	– 15 years

Page 1 of 7

6.	Please list your highest qualification:				
7.	Subject areas that you teach? (You can list more than one subject)				
8.	Which grades do you teach? (You can choose more than one option)	<ul> <li>□ Grade R</li> <li>□ Grade 4</li> <li>□ Grade 8</li> <li>□ Grade 12</li> </ul>	□ Grade 1 □ Grade 5 □ Grade 9	□ Grade 2 □ Grade 6 □ Grade 10	□ Grade 3 □ Grade 7 □ Grade 11

9. Have you received any computer training (Computer Literacy or the like)?

🗆 Yes 🛛 🗆 No

If you ticked "Yes", please specify what training you have received and when (year/s)?

10. Have you received any training that focuses on the use of Computers as a tool for teaching and learning?

🗆 Yes 🛛 🗆 No

If you ticked "Yes", please specify what training you received and when (year/s)?

### Section B: Current ICT infrastructure

This section of the questionnaire is designed to help the researchers understand the current computer configuration at your school. If you do not readily know the answers to some of the questions below, please feel free to obtain this information from either your school Principal or IT technician.

11. What kinds of computers does your school have (you can tick more than one option)?

Pentium I (or equivalent)	🗆 Pentium II	🗆 Pentium III (or equivalent)
Pentium IV (or equivalent)	🗆 Core 2	🗆 Quad Core
🗆 Atom	Dual Core Atom	🗆 Unsure
Other (Please Specify		

12. Do you know the type of computer configuration your school has?

Think clients	Thin clients	A hybrid of thick and thin clients
🗆 Unsure	Other (Please specify	)

13. How many computers in total do you have at your school?

More than 50	Between 50 and 25	Between 25 and 10
Between 10 and 5	🗆 Less than 5	

 Where are the computers in your school located? How many computers are available at each location? Tick the locations where computers are available at your school and then specify in the next column how many computers are available at that venue, e.g. I Computer lab
 30

Computer Location	Number of Computers in location
🗆 Computer Lab	
🗆 Individual classrooms	
🗆 School library	
School office	
Administration offices (Principal, HODs)	
🗆 Multimedia centre	
Other (Please specify):	

15. Which operating systems are used on the computers at your school (you can tick more than one option)?

Windows 2000	Windows XP	🗆 Windows Vista
□ Windows 7	🗆 Ubuntu	🗆 Edubuntu
□ Unsure	Other (Please specify	)

16.	What other software does your school have access to and do/are you able to make use of that
	software? (Please list as many of the Software programs available at your school (that you know
	of) and tick appropriately to show whether or not you are using these applications.)

Software	Do you make use of	this software?
🗆 Microsoft Word	🗆 Yes	🗆 No
Microsoft Excel	🗆 Yes	🗆 No
Microsoft PowerPoint	🗆 Yes	🗆 No
Microsoft Access	🗆 Yes	🗆 No
Image: Microsoft Outlook	🗆 Yes	🗆 No
🗆 Internet Explorer	🗆 Yes	🗆 No
Firefox	🗆 Yes	🗆 No
OpenOffice.org/Libre Writer	🗆 Yes	🗆 No
OpenOffice.org/Libre Calc	🗆 Yes	🗆 No
OpenOffice.org/Libre Impress	🗆 Yes	🗆 No
OpenOffice.org/Libre Base	🗆 Yes	🗆 No
	🗆 Yes	🗆 No
🗆 Cami Maths	🗆 Yes	🗆 No
🗆 Adobe Reader	🗆 Yes	□ No
Google Chrome	🗆 Yes	🗆 No
🗆 Google Earth	🗆 Yes	🗆 No
🗆 KHangman	🗆 Yes	□ No
🗆 Tux paint	🗆 Yes	□ No
🗆 Tux Math	🗆 Yes	🗆 No
🗆 KStars	🗆 Yes	□ No
Other (Please Specify):	🗆 Yes	□ No
Other (Please Specify):	🗆 Yes	□ No
Other (Please Specify):	🗆 Yes	□ No
□ Other (Please Specify):	🗆 Yes	□ No
Other (Please Specify):	🗆 Yes	□ No
□ Other (Please Specify):	🗆 Yes	□ No

17. Does the school have an internal computer network connecting the local computers at your school? (If you can print to a central printer you have a internal network in place.)

🗆 Yes 🗆 No

18. Does your school have a connection to the Internet?

🗆 Yes 🗆 No

Page 4 of 7

19 What type of Internet connection do you have at your school?

19. What type of interr	het connection do you ha	ave at your school?
<ul> <li>Digital Subscriber line</li> <li>Wireless: VSAT</li> <li>Wireless: WiMAX</li> </ul>	≥ (DSL) □ Dial- □ Wire □ Othe	up (via telephone line) less: WiFi r (Please Specify)
🗆 Unsure		
20. Does your school h	ave a computer technici	an who fixes your school computers?
🗆 Yes	□ No	
21. Does your school h area?	ave an IT teacher – who	teaches CAT or Information Technology as a subject
□ Yes	□ No	
22. If you have no IT te the computer facili	acher at school, has the ties at school?	school chosen one of the other teachers to manage
□ Yes	□ No	
23. Who do you go to f at school? (e.g. Sch	or computer related sup ool IT technician, anothe	port/help when you have problems with computers er teacher, principal, HOD, spouse, your children, etc.)
24. Approximately how current IT facilities cards, etc. In other computer facilities	<ul> <li>much money is availab</li> <li>(These include purchas</li> <li>words, anything that new</li> <li>working properly.)</li> </ul>	e within the school's budget for maintaining your ing toner cartridges, paper, keyboards, mice, network eds to be purchased or replaced in order to keep the
🗆 More than R50 000		Between R50 000 and R25 000

Nore than R50 000	□ B
Between R25 000 and R10 000	🗆 B

- 🗆 Less than R5 000 Reliant on donations

- Between R10 000 and R5 000
- No money is available for maintenance
- 🗆 Unsure
- 25. Approximately how much money is available within the school's budget for purchasing new IT equipment from time to time?

🗆 More than R50 000	Between R50 000 and R25 000
Between R25 000 and R10 000	Between R10 000 and R5 000
🗆 Less than R5 000	No money is available for maintenance

- Less than R5 000
- □ Reliant on donations

- 🗆 Unsure

Page 5 of 7

#### Section C: Computer use

This section attempts to better understand the current use of computers in the school (where they are available) and specifically in the realm of teaching and learning.

26. Do you make use of the computers in your school?

🗆 Yes 🛛 🗆 No

If "Yes" then how are you using them (you may choose more than one option)?

□ For classroom administration (marks, worksheets, etc.)

□ For school related administration (related to school management, emails, minutes, etc.)

□ For teaching (Teachers preparing worksheets, tests, presentations, etc.)

□ For learning (Learners typing essays, searching on the Internet, etc.)

Other (Please specify) \_

27. How many teachers at your school are using computers as part of their lessons for teaching purposes?

28. How often are the computer facilities at your school in a working order?

□ Always □ Sometimes □ Rarely □ Never

29. Do the computers available at school meet your ICT needs for teaching and learning?

□ Always □ Sometimes □ Rarely □ Never

30. Do the learners at your school have access to the computers?

🗆 Yes 🔅 No

If "Yes" when do the learners have access to the computers (you may choose more than one option)?

During scheduled class time	During specific IT related training
After school hours?	Other (Please specify)

31. Please rank how often learners at school have access to the computer facilities available, where 4 is Very little access, 3 is little access, 2 is some access and 1 is often have access. Please circle your choice:

Very little access 4 3 2 1 Often have access

32. Please rank the ease with which learners are able to gain access to the computer labs, where 4 is Very Difficult, 3 is Difficult, 2 is Easy and 1 is Very Easy. Please circle your choice:

Very difficult 4 3 2 1 Very Easy

Page 6 of 7

33. Please rank the ease with which you as a teacher are able to gain access to the computer labs for the purposes of teaching and learning, where 4 is Very Difficult, 3 is Difficult, 2 is Easy and 1 is Very Easy. Please circle your choice:

Very difficult 4 3 2 1 Very Easy

#### Section D: IT facility maintenance

This section of the questionnaire seeks to better understand the complexities in IT facility maintenance and is to be answered by the teacher or technician in charge of the computer facilities at your school. If that is NOT you, then please just ignore this section.

34. How would you rank the maintenance of your facilities at school?

🗆 Difficult 🛛 🗆 Manageable 🔅 🗆 Easy

- 35. How would you rate the utilization of the computer facilities at your school by teachers, i.e. how often do the teachers make use of the computers available at school?
- □ Heavily utilized □ Moderately utilized □ Rarely utilized □ Never utilized
- 36. How would you rate the utilization of the computer facilities at your school by learners, i.e. how often do the learners make use of the computers available at school?

□ Heavily utilized □ Moderately utilized □ Rarely utilized □ Never utilized

37. Is there anyone in your school, perhaps even yourself, who fills a support role of helping teachers plan how they might make use of the computer facilities within their teaching practices?

🗆 Yes 🛛 🗆 No

38. Does your school have any policies that govern the use, maintenance and procurement of IT related equipment?

□ Yes □ No
# **Appendix E**

# **SLL Survey: Interview Schedule**

Below is a copy of the interview schedule that was used in the interviews/focus groups with the teachers. The data and analysis generated are reported upon in Chapter 6.

#### Proposed Interview questions for teachers

Please note: Your participation is completely voluntary and you can refuse to participate in answering any of the questions in this interview. In addition, all information gathered during the course of this study will be treated as strictly confidential. Data that may be reported on in the research report will not include information that identifies you or your school as participants in the study.

- Why do you think it's important to have computer facilities at schools for use in teaching and learning?
- I noticed in your questionnaire that you said you do use the computers at school. Can you
  describe a typical instance of how you are using the computers at your school? Prompt: In
  other words, what are you using the school computers to do?

#### OR

I noticed in your questionnaire that you said you don't make use of the computers available at your school, can you tell me why that is?

3. I noticed in your questionnaire that you do use computers in the classroom for teaching and learning purposes. Can you describe a typical classroom instance of when you do use computers for teaching purposes. Prompt: what do you do with computers in the classroom?

OR

I noticed from your questionnaire that you answered "No" to using computers in the classroom for teaching and learning purposes, can you tell me why that is?

- 4. If you are using computers in the classroom for teaching and learning purposes, what do the learners use the computers to do in your lessons? Give some examples of how and when they make use of computers in the classroom.
- 5. I see you noted that the computers at your school work reliably/ unreliably. What are the difficulties that you are facing in using computers in the classroom for teaching and learning purposes? Prompt: What are the typical problems that you face when using computers –

what often goes wrong? I.e. Are the computers not working? Or is the Internet connection often down? Or are the computers too slow?

- 6. I noted from your questionnaire that the computers at school adequately/insufficiently meet your needs. What things about the computers at your school do/don't support you using them in the classroom? Prompt: Do you struggle to gain access to the facilities, or is there a particular software product that you need for your learning areas that you are unable to gain access to?
- 7. What discourages or encourages you when it comes to using computers at school? Prompt: What prevents you from using computers for teaching and learning? What inspires you to use computers for teaching and learning?
- 8. You listed the software available at your school and indicated what you were using in your questionnaire response. Can you tell me what you are using the software to do at school? Prompt: Are you using the software for classroom preparation and management, like worksheets and recording marks, or are you using it during teaching for presenting lesson using a presentation package, or for learning activities in the classroom, like writing essays or solving trigonometric equations?
- 9. Can you tell me why you are not using the other available applications at school? Prompt: Do you have no direct use for these software applications, or do you not know how to use these applications, or do you not know of any ways in which these applications would be useful in your teaching and learning areas?
- 10. How easy do you find it to associate classroom teaching activities with the computers available at your school? Prompt: In other words, can you explain how you link classroom activities to computer use or is this another constraint you face in using the computers at school?
- 11. What do you want to use computers to do that you currently are not doing? Prompt: In other words, do you wish you could ask the learners to work together on a project but have no way for them to each work on the same report simultaneously? Or is there a particular software product you wish you had access to for teaching? Basically, what teaching tasks do you wish you could do on a computer that you currently can't do?
- 12. What are your feelings about the placement of the ICT facilities? Prompt: Should they be in a lab only, in each classroom with a data projector or a combination of both, etc?
- 13. Do you know why some of the teachers don't want to use the computers for teaching? Prompt: Is it because they don't know how to use the computers, or they don't like to use computers/not comfortable using computers, or perhaps there is nothing useful that the computer offers them in terms of their subject area?
- 14. What do you think would make it easier to use the computers more in your teaching? Prompt: What would better help you to be able to integrate the use of technology more in your teaching practices? For example, if you could change something about the computers available at your school so that they were easier for you to make use of what would you change?
- 15. Are there any barriers or difficulties that the learners need to overcome in gaining access to the computer facilities, for instance do they have to pay to use the facilities, or are the computers always locked away, or is the computer laboratory locked after hours?

- 16. I see from your questionnaire you said that your school has X amount of money available for maintenance and buying new equipment. What are you feelings about finding sustainable IT solutions for your school?
- 17. Do you think that formalized support for teaching in terms of helping teachers relate what computers offer to specific classroom activities would be useful to teachers?

The following questions then would also be asked of the ICT teacher or specialist if available, because I don't think that the average teacher who does not tend to the facilities will really be in a position to answer these:

- 1. I see you ranked the maintenance at your school as X, can you tell me a little bit about some of the computer maintenance that you find needs to take place most often? What are the typical problems that you see with the computer facilities at your school?
- 2. I noticed that you ranked the computer utilization at your school as X, can you tell me a little bit about the utilization? How, when and how often are teachers in your school typically using computers at your school? How, when and how often do learners have access to using the computers at school?
- 3. What would you suggest could help or encourage teachers to make more effective use of the available facilities?
- 4. I noticed that you answered "Yes" in the questionnaire to a teacher who acts in a supportive role for teachers in terms of supporting them when they use computers for teaching and learning. Can you tell me a little bit about what that teacher (or yourself, if it's you) is doing in order to support the other teachers?

#### OR

I noticed that you answered "No" in the questionnaire to a teacher who acts in a supportive role for other teachers in terms of the use of computers for teaching and learning. Can you tell me why this is?

### **Appendix F**

# Standard Edubuntu LTSP Thin client configuration

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

#### F.1 Installation and configuration

The standard Edubuntu LTSP thin client configuration consists of a number of thin client computers (i.e. computers without hard disks and local operating systems or locally stored files) that boot off a central server over a LAN. The thin clients are not usable if the server is not running; they are wholly reliant on the server in order to run their operating systems and access their files. These clients retrieve their operating system from the central server which they then boot. Once they are fully booted they present a log in screen, which upon successful log in creates an ssh tunnel to the central server and the users run all their applications and interact with their files on the central server. The thin clients are essentially portals to using the central server and do no real work other than capture input from the user and relay it to the server and then display the output from the server to the user [172, 169].

Setting up a standard thin client server is relatively trivial in comparison to the Cluster and Fat client configurations. All you need is a single server of at least 4GB of RAM (8GB as a minimum is however preferable) onto which you install the Edubuntu distribution. I recommend installing the Long Term release (LTR) of the operating system; at the time of writing the latest long term release was Edubuntu 14.04 (also, as it happens the final Edubuntu release). In addition, install the 64-bit version in order to make maximum use of all of your RAM. Note, when installing make sure that you choose the LTSP server option during the installation process. If you miss this step you will be creating an Edubuntu desktop (and not an Edubuntu server), which can't support booting clients. In addition, I like to choose the option of using the old Gnome 3 desktop instead of Unity. This is because the schools in the SLL have become familiar with the the Gnome interface and sometimes changing the interface can result in confusion later as teachers are unsure of where to find their favourite programs and files.

The installation is straight forward and once done you have a nearly completed server image, with only a few more configurations remaining. Firstly you need to configure the network interface and the DHCP service of the server to work for your particular network needs. These changes are standard and straight forward; similar to the changes implemented in both the Fat client and cluster configurations. Next you need to fix the technical glitch that some of the critical services required for the LTSP services are started before the network interface comes up (this is only a problem in the current LTR of the Edubuntu Operating

system and could be fixed in later releases). To fix this you need to add two small scripts to the network interface start up sequences, found in /etc/network/if-up.d/

I created two scripts, one to start the DHCP service and the other to start the TFTP service, both are essential to booting the thin clients. This step is also necessary in configuring the Fat client server and the cluster servers. The scripts look as follows:

#### FOR THE DHCP SERVICE

#! /bin/bash
/etc/init.d/dhcp3-server start

#### FOR THE TFTP SERVICE

#! /bin/bash
/etc/init.d/tftp-hpa start

Both scripts need to be made executable by chmoding them to +x. Once this is in place you should be able to boot thin clients off your central single Edubuntu server. All that remains is to create some users so that users can log into the system from the thin clients.

While this configuration is simpler to maintain when compared with thick clients, because there is just one machine to maintain, namely the server, the major disadvantage is that the single central server is a single point of failure. That means that should the server not be working for any reason, none of the clients in the school will be able to boot, rendering all thin clients completely useless. This could potentially make all the computers (if they are all thin clients) unusable for the duration of time that the server is faulty. In addition, the central server needs to be a powerful machine with adequate RAM (>8GB) and a relatively useful processor(s) in order to support a large number of clients (typically most schools would like to have a computer lab of at least 20 computers as well as a few computers distributed throughout the school for administration and possibly teaching). This can make the server a very expensive computer to purchase for a school. Alternative configurations are thus to either have a fat client configuration (if you have decent computer hardware for the clients) in order to reduce the hardware needs and associated cost of the server or to make use of a cluster of servers in order to both reduce the hardware costs of the server (as these cluster machines need not be as powerful as a single machine) and also to eliminate or reduce the single point of failure aspect of the single central server.

#### F.2 How LTSP thin clients work in practice in Edubuntu

This section details how the Edubuntu LTSP thin clients boot up and work. I have tried to be as detailed as possible, hopefully without being too tedious to read, but rather providing a deeper understanding of how thin client computing works with LTSP and specifically Edubuntu's implementation thereof.

Step 1, the PXE [107] client (the primary loader) running on the thin client's PXE-enabled network card sends out a DHCP request. The DHCP server responds supplying the contents of the DHCP server config, which includes a number of things, but mainly the IP for the client and the name of the file that the PXE client needs to request from the server, via TFTP, in order to begin booting. The client receives its IP and the file name. It then TFTPs pxelinux.0 from the TFTP server running on the server. The file is transferred over the network to the client and the client loads it into memory and executes it.

The now running pxelinux (a secondary loader), loads its configuration file by fetching it from the TFTP server (these are the files found in pxelinux.cfg in the TFTP server's root directory). The configuration file contains the name of the kernel and the initram disk that need to be TFTP'ed from the TFTP server to the

client, as well as the necessary kernel arguments. Once retrieved by the client they are loaded into memory and the kernel then begins executing.

The now running kernel detects and initialises all the hardware. Because no root is specified the kernel then mounts the ramdisk image as the root file system. Next the kernel runs the init script which sets up environment variables and creates necessary files and ultimately loads a new root file system.

What init does more specifically seems to be that the scrip ltsp\_nbd gets run, which sends out another DHCP request and saves the response to a file. Next the nbd\_root\_server variable is set depending on either the value of the nbdroot (possibly set in the initram disk) or nbd\_root\_server which is set as the IP of the DHCP server, from the DHCP response. In addition, the nbd\_root\_port is set either based on the nbdroot or to 2000. Then the nbd\_swap variable is set which influences the values of the nbd\_swap\_server and port. If the client doesn't have enough memory (< 48M) then it uses nbdswapd to create extra swap on the server for use by the client.

Next, /cow (copy on write) gets created and has a tmpfs file system mounted. nbd client sets up a link to the nbd server and associates /dev/nbd0 with the link. On the server side /etc/inetd.conf has a line that allows nbdrootd to receive communications on port 2000 and then serve the nbd image file (blocks needed at the time). Next on the client side /dev/nbd0 gets mounted at /rofs (read only file system) (which was created earlier along with /cow) with a squashfs file system. Then /cow and /rofs are mounted together at /root with a unionfs file system. /cow is then mounted into the new root as /root/cow and similarly /rofs is mounted at the new root, /root/rofs

Next, the lts.conf file gets TFTP'ed to the client (if it exists, its only created if clients need to be different from the basic image). A script called run\_init is then executed which changes /root to / and then calls /sbin/init. All of this results in a proper operating system and /sbin/init begins to execute. /sbin/init then starts all the other services of the thin client machine (ssh, networking, etc), much like a normal Linux operating system booting.

### Appendix G

# Edubuntu LTSP Fat (hybrid) client configuration

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

Similarly to thin clients, fat/hybrid clients use the central server as a network disk for /home, for the root file system, and to authenticate via ldm/ssh. But in contrast with thin clients, fat clients use their own RAM and CPU power to run the applications. Of course this also means that their hardware requirements are the same as for running Ubuntu/Edubuntu on normal thick client workstations. This is a similar to the functionality in DRBL (Diskless Remote Boot in Linux) and Netlive (an alternative to LTSP developed in Italy), with the exception that LTSP fat/hybrid clients use NBD (Network Block Device) for offering the client's root file-system to the client, which is faster than NFS (Network File System) [175].

To set up an Edubuntu LTSP fat/hybrid client configuration begin by installing the Ubuntu server edition, 64-bit operating system. Once installed and fully up to date, install the Edubuntu-desktop. Next install the ltsp-server-standalone package (this will use the built in DHCP service that comes with LTSP. If you want to use a separate DHCP server then you would install the ltsp-server package instead and install isc-dhcp-server/dhcp3-server.). Once installed you can configure the DHCP service in /etc/ltsp/dhcpd.conf follows:

```
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;
subnet 192.168.1.0 netmask 255.255.255.0 {
option domain-name "ubuntu.example.com";
option domain-name-servers 192.168.1.1;
option routers 192.168.1.1;
next-server 192.168.1.1;
filename "ltsp/i386/pxelinux.0";
```

}

The file will already be configured with a default subnet of 192.168.0.0/24. Therefore all you need to do is change the subnet info and the corresponding broadcast and gateway information to suit your network. In addition, the Ethernet interface will be configured for 192.168.0.254/24 (more than likely eth0) and you will need to change this too to match your network.

You can now build the Fat Client image. In order to build the image you need to type the following command in:

ltsp-build-client — fat-client — fat-client-desktop edubuntu-desktop — arch i386 — prompt-rootpass

It is important to note that "Edubuntu-desktop" can also be replaced with "Kubuntu-desktop", "Ubuntudesktop" or "Xubuntu-desktop" but for schools Edubuntu-desktop makes the most sense as we are providing a system for an educational institution. The architecture is recommended as i386 as not all thin clients are necessarily able to boot 64-bit architectures, such as Intel Atom based clients. The "prompt-rootpass" option allows you set a root password for the image that the clients boot which is very useful when you are attempting to troubleshoot any issues between the clients and the central server.

Once the fat image has completed its build, if you wish to install any additional software onto the image, then you need to chroot into the ltsp environment and install the necessary applications. So for example, if you wanted to install vlc, sikuli-ide, apache, mrtg, snmpd and snmp-mibs-downloader into the client's image, then you would need to do the following:

ltsp-chroot -c p

You will now be in the LTSP chroot. Now you can install any additional applications that you might want. For example:

aptitude install vlc sikuli-ide apache mrtg snmpd snmp-mibs-downloader

Once you are done installing any additional software, or if you are happy with the defaults that are installed by Edubuntu-desktop, you should then sync the ssh keys and update the ltsp image. This is done outside of the LTSP chroot, so first exit the chroot. To update the ssh keys:

ltsp-update-sshkey

And update the ltsp image:

ltsp-update-image ---arch i386

Lastly, you should create some user accounts to allow users to log in and use the system. This configuration makes better use of all the available resources that the school has as the clients can use their own RAM and CPUs, reducing the load on the server and resulting in an improved user experience. However, this solution does not solve the problem of the single point of failure, specifically the central server, as the fat clients are still reliant on the server in order to boot their operating systems, retrieve and store their files and user authentication.

## **Appendix H**

# **DNS Round Robin Cluster configuration**

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

In order to configure a DNS round robin cluster server you need to configure a group of Edubuntu LTSP servers to share the same DNS name and file storage location. In order to do this I used a number of protocols and services, namely, DNS (bind), LDAP (OpenLDAP) and NFS (NFS-kernel-server and NFS-common) to support thin clients booting from a group of Edubuntu LTSP servers. In the following subsections I discuss the reasons for the choice of protocols and services used as well as their configurations in order to support the desired outcome of a DNS round robin LTSP cluster.

#### H.1 The NFS installation/configuration, server-side

First install nfs-kernel-server:

aptitude install nfs-kernel-server

Once installed the most important step to get the service running is to configure the exports file so that the correct machines can access the correct files. To export our the directories to a local network 192.168.2.0/24 (change the IP address to the appropriate subnet for your the network) – this is the subnet that is being used at Ngwane, my test site. I added the following line to /etc/exports

/home 192.168.2.0/24(rw, nohide, insecure, no\_subtree\_check, async)

What this says is that the home directory is being exported to any machine that falls in the IP range of 192.168.2.0/24. So if one has an IP address in that subnet one can then access the exported files. The options listed after the IP range in brackets apply to that IP range (hence no space left between the range and the list of options) and specify the type of access the computers in the IP range will have to the exported files. Finally, the NFS service needs to be restarted:

```
/etc/init.d/nfs-kernel-server restart
```

#### H.2 The NFS installation/configuration, client-side

The first step is to install the nfs-common package on the servers that are going to mount the NFS root to their file systems. This has to be done on all the other LTSP servers that are not the primary file server machine.

aptitude install nfs-common

On the client I can now mount the exported tree with one command:

mount -t nfs -o proto=tcp, port=2049 nfs-server:/home /home

To save one from retyping this after every reboot add the following line to /etc/fstab:

parentNode:/home /home nfs \_netdev, auto 0 0

The auto option mounts on startup and the \_netdev option waits until network devices are loaded.

One now has a working NFS server and clients, but user permissions need to be appropriate set so that users are able to access, edit, create and delete their files no matter where they are logged in and have the same rights to them from different servers within the cluster. Currently files can be created on the NFS server and seen by the NFS clients, but clients cannot delete or create new files. This is as a result of users not being identified by the NFS server as a valid user who has permission to be accessing and editing files. This is a problem. NFS user permissions are based on user ID (UID). UIDs of any users on the client must match those on the server in order for the users to have access. The typical ways of doing this are:

- Manual password file synchronization
- Use of NIS
- Use of LDAP

Manual password synchronisation is open to error. It's not ideal to be creating accounts in one place and then having them copied from one server in the cluster to another. A more central method of authentication is necessary, such as those provided by NIS or LDAP. While NIS is simpler to administer it is rather limited when compared with other directory services. At the time that "Managing NFS and NIS" was written, 2001, LDAP was in its relative infancy but still authors of [204] claimed that LDAP was the future of directory services on all operating systems [204]. LDAP is now considered an industry standard when implementing directory services; administrators and network designers are encouraged to replace their NIS services with LDAP [206]. As such I have chosen to use LDAP in order to authenticate users centrally so that both the client and the server nodes have the same permissions, users and groups.

#### H.3 OpenLDAP Installation/Configuration, server-side

First, install the OpenLDAP server daemon slapd and ldap-utils, a package containing LDAP management utilities:

By default slapd is configured with minimal options needed to run the slapd daemon.

OpenLDAP uses a separate directory which contains the cn=config Directory Information Tree (DIT). The cn=config DIT is used to dynamically configure the slapd daemon, allowing the modification of schema definitions, indexes, ACLs, etc without stopping the service.

The back-end cn=config directory has only a minimal configuration and will need additional configuration options in order to populate the front-end directory. The front-end will be populated with a "classical" scheme that will be compatible with address book applications and with Unix Posix accounts. Posix accounts will allow authentication to various applications, such as web applications, email Mail Transfer Agent (MTA) applications, etc.

First, some additional schema files need to be loaded. In a terminal enter:

```
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/cosine.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/nis.ldif
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/ldap/schema/inetorgperson.ldif
```

Next, copy the example LDIF file below, naming it back-end.example.com.ldif, onto your system:

```
# Load dynamic backend modules
dn: cn=module, cn=config
objectClass: olcModuleList
cn: module
olcModulepath: /usr/lib/ldap
olcModuleload: back_hdb
# Database settings
dn: olcDatabase=hdb, cn=config
objectClass: olcDatabaseConfig
objectClass: olcHdbConfig
olcDatabase: {1}hdb
olcSuffix: dc=example, dc=com
olcDbDirectory: /var/lib/ldap
olcRootDN: cn=admin, dc=example, dc=com
olcRootPW: secret
olcDbConfig: set_cachesize 0 2097152 0
olcDbConfig: set_1k_max_objects 1500
olcDbConfig: set_lk_max_locks 1500
olcDbConfig: set_lk_max_lockers 1500
olcDbIndex: objectClass eq
olcLastMod: TRUE
olcDbCheckpoint: 512 30
olcAccess: to attrs=userPassword by dn="cn=admin,dc=example,dc=com" \
                         write by anonymous auth by self write by * none
olcAccess: to attrs=shadowLastChange by self write by * read
olcAccess: to dn.base="" by * read
olcAccess: to * by dn="cn=admin,dc=example,dc=com" write by * read
Change olcRootPW: secret to a password of your choosing.
Now add the LDIF to the directory:
sudo ldapadd -Y EXTERNAL -H ldapi:/// -f backend.example.com.ldif
```

The front-end directory is now ready to be populated. Create a front-end.example.com.ldif with the contents found in the example below:

```
# Create top-level object in domain
dn: dc=example, dc=com
objectClass: top
objectClass: dcObject
objectclass: organization
o: Example Organization
dc: Example
description: LDAP Example
# Admin user.
dn: cn=admin, dc=example, dc=com
objectClass: simpleSecurityObject
objectClass: organizationalRole
cn: admin
description: LDAP administrator userPassword: secret
dn: ou=people, dc=example, dc=com objectClass: organizationalUnit ou: people
dn: ou=groups, dc=example, dc=com objectClass: organizationalUnit ou: groups
dn: uid=john, ou=people, dc=example, dc=com
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: john
sn: Doe
givenName: John
cn: John Doe
displayName: John Doe
uidNumber: 1000
gidNumber: 10000
userPassword: password
gecos: John Doe
loginShell: /bin/bash
homeDirectory: /home/john
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: john.doe@example.com
postalCode: 31000
1: Toulouse
o: Example
mobile: +33 (0)6 xx xx xx xx
homePhone: +33 (0)5 xx xx xx xx
title: System Administrator
```

```
postalAddress:
initials: JD
dn: cn=example,ou=groups,
dc=example,dc=com
objectClass: posixGroup
cn: example
gidNumber: 10000
```

In this example the directory structure, a user, and a group have been setup. In other examples you might see the objectClass: top added in every entry, but that is the default behaviour so you do not have to add it explicitly.

To add the entries to the LDAP directory:

```
sudo ldapadd -x -D cn=admin, dc=example, dc=com -W -f frontend.example.com.ldif
```

Assuming one didn't reset the password for the admin account created by the back-end ldif file then the password one would need to provide when it asks is secret.

One can check that the content has been correctly added with the ldapsearch utility. Execute a search of the LDAP directory:

```
ldapsearch -xLLL -b "dc=example, dc=com" uid=john sn givenName cn
```

Assuming the LDAP server is working correctly it should result in the following output:

```
dn: uid=john,ou=people,dc=example,dc=com
cn: John Doe
sn: Doe
givenName: John
```

#### H.4 OpenLDAP Server GUI applications

In order to avoid adding new users and groups to the LDAP directory by editing text files and then manually using the ldapadd utility, it would be preferable to add new users and groups through a graphical interface or a web based interface. Particularly when recalling that a local school teacher within each school's environment is more than likely going to be the administrator of the school computer laboratory and consequently its servers – particularly so in the disadvantaged schools that cannot afford to employ a system administrator or technical expert.

There seem to be a number of choices in terms of LDAP graphical or web interfaces. Those that I identified were: Jxplorer – a Java based LDAP browser [209]; GOsa2 – GPL'ed framework for managing accounts and systems in LDAP databases [210]; phpLDAPadmin – is a web-based LDAP client which provides easy, anywhere-accessible, multi-language administration for an LDAP server [211]; and Webmin - a web-based interface for system administration for Unix servers, which has an LDAP plugin for administering an LDAP server [212]. There are many more options available, however, I limited to applications that run well in Linux, particularly Edubuntu (as it is the operating system that I have chosen to use as the most appropriate for Education purposes in South African schools, particularly formerly disadvantaged schools), and to use open source and free (as in cost) software.

#### H.4.1 Jxplorer

Jxplorer is available in the Ubuntu repository and thus can be installed using apt or aptitude,

#### aptitude install jxplorer

Once installed one will find it available on the gnome menu under Applications and then Programming. Click on JXplorer to start and a dialogue box appears. One will then need to click on the little connect icon in the top left hand corner of the dialogue box in order to connect the application to your DLAP server. Alternatively click on File and then connect. An "Open LDAP connection" dialogue box pops up, allowing one to fill in information regarding the LDAP server one wishes to connect to. Under Host one will need to fill in the name or IP of one's host, in my case it was 192.168.2.1 and then one would need to fill in the port number that one's LDAP service is listening on. In the case of most Ubuntu (or Ubuntu based systems) this is 389 (and not the default 19389 listed). Under base dn one would need to fill in the need to change the security level to User + Password and fill in the dn of one's administrative user. In my example this was cn=admin, dc=local, dc=org and then type in the password of that user. one will then be connected to the LDAP server and one can begin editing/adding users and groups.

The application is less powerful than GOsa, but perhaps less intuitive than phpLDAPadmin. It could be easy for a teacher to become confused using JXplorer as it uses a lot of the LDAP terminology and does provide the administrator with the ability to be more specific about the types of classes that new Objects will possess. Objects and classes, while common in Computer Science are terms that are complicated and not understand by teachers who have had no Computer Science education. For a teacher with Computer Science training, specifically, programming, where they would have learned about the terms Objects and Classes, JXplorer is a very useful, easy to install and configure, application to use for LDAP administration. However, for local teachers in generally disadvantaged schools with no programming experience, I believe it is too complicated and confusing for teachers to use.

#### H.4.2 GOsa<sup>2</sup>

GOsa<sup>2</sup>installation is a bit tricky. Normal installation from your regular Ubuntu mirror does not have a recent enough version – 2.6.X – but rather provide 2.5.X. In which case, post installation you will probably encounter the error "fatal error: call to undefined function print\_array() in /usr/share/gosa/include/class\_dhcpPlugin.inc on line 206" or something similar. The solution to which, provided on launchpad here (https://bugs.launchpad.net/ubuntu/+source/gosa/+bug/370505) is to install a more recent version from the gosa developers themselves, by editing your sources.list to include the following:

deb http://oss.gonicus.de/pub/gosa/debian-etch/

Once installed, you need to connect to http://localhost/gosa where it will ask you to begin the set up process. The chances are however, that this will fail because the schemas and corresponding ldif files that you need for the set up to complete successfully and consequently the application to run properly are not installed when installing gosa – for some unknown reason to me. Hence you need to install "gosa-schema". Unfortunately this only installs the schema files and so you need to convert them ldif files yourself and then add them to the LDAP directory. This you do by following the instructions at https://bugs.launchpad.net/ubuntu/+source/gosa/+bug/319544 and http://www.

linuxquestions.org/questions/linux-server-73/how-to-add-a-new-schema-toopenldap-2-4-11-a-700452/. First creating the schema\_convert.conf file and then using slaptest to convert them to ldif files. You then need to edit those ldif files removing the numbers in the name at the top and adding the context (cn=schema, cn=config) as well as remove the last lines in the file from "structuralObjectClass" down. Once this is complete you can then manually add the ldif files to your LDAP config using the ldapadd command: ldapadd -Y EXTERNAL -H ldapi:/// -f `filename.ldif'. Only now can you begin the set up process of gosa on the service's website.

GOsa is a really powerful tool for LDAP administration, providing many tools to help an administrator, but I think for the purposes of school computer networks, being run by local teachers, it could be too powerful. The interface (tour of the interface available at http://oss.gonicus.de/screencasts/add-user.html) while clear and easy to understand by any experienced user is exhaustive and rich and I think that local teachers who are relatively novice users and often at a language disadvantage (second and third language English speakers) will get caught up in the many options and become confused as to what they need to do in order to create accounts. With such a powerful system and so many options they could conceivably perform complicated actions that they don't understand the full implications of and potentially crash their LDAP server. In a school environment (or any environment really) with fully qualified system administrators I think that GOsa would be a very useful tool for administrating LDAP servers, but for this particular environment I think it is too complex.

#### H.4.3 phpLDAPadmin

The installation of phpLDAPadmin is not too bad. You need to simply use apt or aptitude to install the application as per normal, ie:

#### aptitude install phpldapadmin

Once installed you can configure the interface by making changes to the config.php file found in /etc/ phpldapadmin/ The defaults are generally fine though. I only made two changes, I set the time to South African Standard Time and I forced the language to English. Otherwise I left the defaults as they were. The only other change that I made was to one of the template files (/etc/phpldapadmin/templates/creation) called posixAccount. xml. The reason being that the account types listed by default in the user account creation scripts do not provide an option for /bin/bash nor do they allow for adding a type on the fly. So I changed the xml document to include /bin/bash in the account type list of options so that I could assign the correct account type for Edubuntu users. In addition, I also made a change to the home directory creation. By default it creates the home directory at /home/users/userid. I would prefer it to match typical user account creation in UNIX, and so have changed it to create the home directories at /home/userid. To do this I just deleted the word "users" from the UID attribute in the posixAccount.xml file. Once this is complete you can then connect to http:localhost/phpldapadmin/ where you are presented with a login screen, and you need only click on login and then type in your admin account password. After that you are able to easily create groups and user accounts. The interface is relatively straightforward and easy to use. With simple instructions teachers should be able to create new accounts by just clicking on the appropriate options and then filling in the information that they are asked.

#### H.4.4 Webmin

Webmin, too, was relatively easy to install on an ubuntu based system. There are, however, some issues with depreciated Perl packages that Webmin uses and that ubuntu don't have in their repositories. I found a help-

ful site online (http://www.kelvinwong.ca/2010/05/22/installing-webmin-on-ubuntuserver-10-04-lts-lucid/) that helped me install a working version of Webmin. The OpenLDAP Webmin module is standard in the basic Webmin installation, so there is no need for any other installation once Webmin is installed. To connect to Webmin, you then need to open up your browser and then go to https://localhost:10000 From there you can log in using your credentials and then click on the servers link on the left hand side and choose LDAP from the list. You can then administer your LDAP server in the main window on the right hand side. The Webmin interface is very basic, so basic that it doesn't even provide the fields of the user and group entries but rather leaves them blank for the administrator to fill in not only the values of the variables but also the variables themselves. This level of control could be beneficial to some administrators, but for local teachers with very little computer experience this will make the entire process of administering the LDAP service very complex and confusing. Not only will they have to remember which steps are necessary in creating new accounts and groups but they will also need to remember all the variable names and remember them correctly in order to have a properly configured LDAP directory. For this reason I don't believe that Webmin is an appropriate solution as an LDAP administration tool for the context in which we are working.

Having tried all four options, JXplorer, GOsa, phpLDAPadmin and Webmin, I feel that the most appropriate solution for local, inexperienced school teachers administering an LDAP server is phpLDAPadmin.

#### H.5 OpenLDAP installation/configuration, client-side

This is to enable an ubuntu client to authenticate to an existing OpenLDAP server (which we created above). Begin by installing the following package: libnss-ldap During the installation a menu box will ask you to provide connection details about your LDAP server, as well as other configuration options regarding your server which you will need to answer. Should you make a mistake with any of the information you provide it you can reconfigure it by typing:

```
sudo dpkg-reconfigure ldap-auth-config
```

The results of the dialog can be seen in /etc/ldap.conf. If your server requires options not covered in the menu edit this file accordingly. You then need to enable the auth-client-config LDAP profile by entering:

sudo auth-client-config -t nss -p lac\_ldap

- -t: only modifies /etc/nsswitch.conf.
- -p: name of the profile to enable, disable, etc.
- lac\_ldap: the auth-client-config profile that is part of the ldap-auth-config package.

Finally, use the pam-auth-update utility to configure the system to use LDAP authentication:

sudo pam-auth-update

From the pam-auth-update menu choose LDAP and any other mechanisms that you need (in our example we just choose LDAP).

#### H.6 Bind installation/configuration

I chose to use the BIND implementation of DNS as it is the most popular of DNS implementations used worldwide today. A DNS service itself was necessary in order to allow multiple servers to respond to the same host name – necessary in achieving the round robin nature of the cluster that allows work to be evenly distributed and appear as one machine to all clients involved on the network (abstractly the complexities from the clients). In addition, DNS is useful in Edubuntu's thin client computing implementation as the communication between the clients and the servers tend to need name resolution services – which can be fudged using /etc/hosts but a DNS server is a neater solution.

In Ubuntu you can install BIND using apt or aptitude. I did the following:

aptitude install bind

All configuration files are found in /etc/bind Once BIND has installed change directory to /etc/bind and begin configuring your system by adding record information for your zone to your newly created db file. In the domain for the LTSP cluster server I decided to call my domain local.org so I created a file called db.local.org The contents of that file can be seen below:

```
; Definition of zone .local.org.
;SOA record portion
$TTL 1d
local.org. IN SOA dnsnode.local.org. root.local.org. (
2009041601 ; serial
1d ; refresh
2h ; retry
100d ; expire
1h); negative cache
:name servers
IN NS ns
ns IN A 192.168.2.2
: Hosts
localhost IN A 127.0.0.1
parentnode IN A 192.168.2.1
dnsnode IN A 192.168.2.2
gennode1 IN A 192.168.2.3
gennode2 IN A 192.168.2.4
gennode3 IN A 192.168.2.5
ltspserver IN 30 A 192.168.2.2
IN 30 A 192.168.2.3
IN 30 A 192.168.2.4
IN 30 A 192.168.2.5
ngrouter IN A 192.168.2.6
client250 IN A 192.168.2.250
client249 IN A 192.168.2.249
client248 IN A 192.168.2.248
client247 IN A 192.168.2.247
```

```
client246 IN A 192.168.2.246
client245 IN A 192.168.2.245
client244 IN A 192.168.2.244
client243 IN A 192.168.2.243
client242 IN A 192.168.2.242
client241 IN A 192.168.2.241
client240 IN A 192.168.2.240
client239 IN A 192.168.2.239
client238 IN A 192.168.2.238
client237 IN A 192.168.2.237
client236 IN A 192.168.2.236
client235 IN A 192.168.2.235
client234 IN A 192.168.2.234
. . .
client25 IN A 192.168.2.25
client24 IN A 192.168.2.24
client23 IN A 192.168.2.23
client22 IN A 192.168.2.22
client21 IN A 192.168.2.21
client20 IN A 192.168.2.20
```

The first portion of the file, before the name servers, forms the SOA record. The next section, forms the NS record for the domain, and the last section, is all the A records for the domain.

In the SOA record, the \$TTL variable is the "Time to live" variable which specifies the length of time that remote name servers should cache records for this zone [215]. Following that line in the file, the name on the left is the name of the zone (local.org) and the keyword IN mean Internet. Next the keyword SOA defines the type of record, an SOA record, while the domain name following it specifies the name of the master name server (dnsnode.local.org) [215]. The field which follows the master name server is the email address of the DNS administrator (root.local.org). The "@" is missing from the email address because the symbol is used by DNS and so would result in an error. All that one needs to remember is that when emailing replace the first dot with an @ symbol and you have root@local.org [215]. The remaining fields in the SOA record have comments next to them to make them more meaningful. The first field is the serial number of the zone configuration which helps to identify the version. It consists of the last date that the zone was altered followed by a two digit version of the day, i.e. if it's the first version of the day its 01, the second, its 02, seventh its 07, etc [215]. The refresh time is the time after which a remote name server will check to see if the zone configuration has changed. Next the retry time is the time to wait if an attempt to load the zone fails. The expire time is the time after which a salve name server will drop the information about the zone if it has not been able to reload it from the master name server. And lastly, the negative cache expiry time refers to the time that must be waited in between queries if the DNS server returns that a particular host does not exist [215].

The NS record section of the file specifies where the name servers of the domain can be found. In the local.org domain the name server is 192.168.2.2 (ns.local.org) [215]. The final section is the A records. The A records map the domain names of the hosts on the network to their associated IP addresses. This section is the longest as there are a possible 256 names and associated IPs in a single /24 network (which is what I had in the LTSP cluster server configuration) [215].

There is no real need to create more than two name servers on your network (two build in redundancy should one of them fail; in the case of my cluster I have only configured one machine to be the DNS server).

All non DNS servers can then be configured to use the DNS servers by setting their /etc/resolv.conf files to point to the IP addresses of the DNS server. A /etc/resolv.conf file can be configured as:

nameserver 192.168.2.2 search local.org

The first line specifies the IP address of the machine that is providing name services to the domain, while the second tells the resolver that in the case of a relative domain name query that the fully qualified domain name contains the domain name of local.org

Next, I provided for reverse look-up within the domain. Reverse look-up might not seem necessary, but it can be very important to services within networks (for example mail servers) but also to users who know the IP of a machine but wish to find the corresponding domain name. In order to resolve the reverse names we need another db file, which I called local-reverse. It looked as per the example below:

```
; reverse dns
$TTL 1d
@ IN SOA dnsnode.local.org. root.local.org. (
2009041601 ; serial
1d ; refresh
2h ; retry
100d ; expire
2h); negative cache
IN NS ns.local.org.
1 IN PTR parentnode.local.org.
2 IN PTR dnsnode.local.org.
IN PTR ltspserver.local.org.
3 IN PTR gennode1.local.org.
IN PTR ltspserver.local.org.
4 IN PTRgennode2.local.org.
IN PTR ltspserver.local.org.
5 IN PTR gennode3.local.org.
IN PTR ltspserver.local.org.
6 IN PTR ngrouter.local.org.
20 IN PTR client20.local.org.
21 IN PTR client21.local.org.
22 IN PTR client22.local.org.
23 IN PTR client23.local.org.
24 IN PTR client24.local.org.
25 IN PTR client25.local.org.
26 IN PTR client26.local.org.
. . .
245 IN PTR client245.local.org.
246 IN PTR client246.local.org.
247 IN PTR client247.local.org.
248 IN PTR client248.local.org.
249 IN PTR client249.local.org.
250 IN PTR client250.local.org.
```

The SOA record at the top is identical to that found in db.local.org with one exception, the @ symbol instead of the zone name at the beginning of line 2. The @ symbol represents the name of the zone, in this case 2.168.192.in-addr.arpa Below the SOA record are the PTR (reverse lookup) records. They specify only the last digit of the host IP address which is prepended to the zone name represented by the @. The hosts for each reverse lookup are fully qualified because reverse lookup is considered to be a separate zone from the forward lookup of the domain. Then I edited the file /etc/bind/named.conf and added the following lines:

zone "local.org" {
type master;
file "/etc/bind/db.local.org";
};
zone "2.168.192.in-addr.arpa" {
type master;
file "/etc/bind/local-reverse";
};

These two lines specified the zone files that are necessary for forward and reverse lookups for the local.org. domain. These are important so that the DNS server knows about the zones that is it authoritative for. Once I had made all the above changes I restarted BIND:

/etc/init.d/bind9 restart

Then check that its working by querying the DNS server. A good example query would be:

host dnsnode.local.org.

which in my configuration produces the following result:

dnsnode.local.org has address 192.168.2.2

Then to check that the reverse lookup is also working as expected, try:

Host 192.168.2.2

You should see the following returned:

2.2.168.192.in-addr.arpa domain name pointer ltspserver.local.org. 2.2.168.192.in-addr.arpa domain name pointer dnsnode.local.org.

#### H.7 Putting it all together

Once the DNS service is running there are only three important changes that need to be made in order to implement the round robin nature of the cluster.

The first is that you need to have configured your A records (both forward and reverse) such that there is one domain name that has multiple IP addresses associated with it. In the example above that domain is ltspserver.local.org. In both the forward and reverse zone files the name ltspserver.local.org has the IPs 192.168.2.2, 192.168.2.3, 192.168.2.4, and 192.168.2.5 associated with it. This means that when I query the DNS service about the domain name ltspserver.local.org. I receive the following responses:

ltspserver.local.org	has	address	192.168.2.4
ltspserver.local.org	has	address	192.168.2.5
ltspserver.local.org	has	address	192.168.2.2
ltspserver.local.org	has	address	192.168.2.3

All four IPs are provided as responses to the query. The IPs are listed at random, so if I ask the DNS service the same query a second and third time I get different orders. For example:

```
ings@DNSNode:/etc/bind$ host ltspserver.local.org
ltspserver.local.org has address 192.168.2.3
ltspserver.local.org has address 192.168.2.4
ltspserver.local.org has address 192.168.2.5
ltspserver.local.org has address 192.168.2.2
ings@DNSNode:/etc/bind$ host ltspserver.local.org
ltspserver.local.org has address 192.168.2.2
ltspserver.local.org has address 192.168.2.3
ltspserver.local.org has address 192.168.2.3
ltspserver.local.org has address 192.168.2.4
ltspserver.local.org has address 192.168.2.4
```

The second change is to the DHCP configuration so that any one of the LTSP servers can be used by the PXE booting client to TFTP its kernel and initram disk. This is done by setting the next-server option to point to the ltspserver.local.org domain name – which in turn resolves to be a list of the four IPs corresponding to the four LTSP servers. The PXE booting client will then randomly choose one of the options from the list of IPs and because the list is randomly ordered there is an approximate equal chance of choosing any one of the IPs. The dhcpd.conf file in /etc/ltsp/dhcpd.conf containing the next-server option looks as follows:

next-server ltspserver.local.org;

Note: only one of the LTSP servers must be configured to be a DHCP server. If more than one is a DHCP server there will be conflicts if both reply to a DHCP request with different IP addresses.

The third and final configuration setting that is required is to set the LDM\_SERVER variable in the lts.conf file. lts.conf can be found in /var/lib/tftpboot/lts.conf (where you create it).

The LDM\_SERVER variable sets the LDM (LTSP Display manager) server that the thin clients should be connecting to in order to access their files and applications. If you specify a domain name like lt-spserver.local.org for the LDM\_SERVER the clients will first resolve the name using the DNS services available on the network. They will then again be presented with a randomised list of the associated IPs and will randomly choose one of them. This results in an equal chance of any one of the LTSP servers acting as the LDM Server for any one thin client. These measures should allow you to comfortably have as many LTSP servers as you would like and the thin clients will be none the wiser as they will connect at random each time to one of them in order to access their files and applications. And because each of the LTSP servers has mounted the NFS server's disk in order to access the user files each user will experience the same functionality regardless of which LTSP server they are accessing.

# **Appendix I**

# **Ubuntu LTSP Cluster configuration**

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

In order to configure a DNS round robin cluster server you need to configure a group of Edubuntu LTSP servers to share the same DNS name and file storage location. In order to do this I used a number of protocols and services, namely, DNS (bind), LDAP (OpenLDAP) and NFS (NFS-kernel-server and NFS-common) to support thin clients booting from a group of Edubuntu LTSP servers. In the following subsections I discuss the configurations in order to support the desired outcome of a DNS round robin LTSP cluster.

#### I.1 Ubuntu LTSP Cluster installation and configuration

The implementation of the LTSP cluster in Ubuntu requires one root server (that provides services to the clients such as DHCP, NFS, TFTP, PXE booting, DNS, routing to the Internet, LDAP authentication, etc.) and one or more application servers which facilitate the clients access to applications (working similarly to a remote terminal service – rdesktop – or VNC). Each thin client must connect to an application server in order to access the applications available and run them. This process if often memory resource intensive and so having multiple application servers to choose from allows the system to better support more clients. This working together process of the root server and the applications servers in order to serve more thin clients is the cluster concept.

The first step is to set up the root server and its associated services. Once it is up and running we can progress to installing one or many (preferable) application servers.

#### I.1.1 Root server for LTSP-cluster

It's recommended that you install a 64-bit operating system for the root server. If you intend to make available more than 4GB of RAM, or if you know you will need more than 4GB, then this is sound advice. The operating system to install is the Ubuntu server edition. I installed the latest version, which at the time was Ubuntu Oneiric. Once the installation is complete make sure that your server's software is up to date by setting appropriate sources in sources.list and then running (as root):

apt-get update aptitude upgrade If aptitude does not work, first install it, with "apt-get install aptitude" Next, still as root, install the following packages: openssh-server, dhcp3-server and ltsp-server

```
aptitude install openssh-server dhcp3-server ltsp-server
```

This installation will take some time – particularly the installation of the ltsp-server. Once it is completed though the DHCP service can be configured. Modify /etc/dhcp/dhcp.conf file for your network, this is what mine looks like the example below:

```
#/etc/dhcp/dhcpd.conf
ddns-update-style none;
default-lease-time 600;
max-lease-time 7200;
authoritative;
log-facility local7;
subnet 192.168.1.0 netmask 255.255.255.0 {
option domain-name "ubuntu-ltsp-cluster";
option domain-name-servers 192.168.1.101;
option routers 192.168.1.101;
range 192.168.1.200 192.168.1.250;
next-server 192.168.1.101;
filename "ltsp/i386/pxelinux.0";
}
```

To restart the DHCP service run:

/etc/init.d/isc-dhcp-server restart

Once DHCP has now been configured and restarted you can proceed to building the chroot for the thin clients. According to the Ubuntu notes the clients require 32-bit images in order to work properly and can be done by running the following command:

sudo ltsp-build-client — arch i386 — ltsp-cluster — prompt-rootpass

During the build of the chroot you will be asked a series of questions pertaining to the image and the server. The Ubuntu documentation recommends that you answer these questions as follows (bearing in mind that you will need to change the server's name to whatever you are using):

Configuration of LTSP-Cluster NOTE: booleans must be answered as uppercase Y or N Server name: 192.168.1.101 Port (default: 80): 80 Use SSL [y/N]: N Enable hardware inventory [Y/n]: Y Request timeout (default: 2): 2 You will also be asked to enter a root user password for the chroot. Your answers to the above will be configured in /opt/ltsp/i386/etc/ltsp/getltscfg-cluster.conf should you wish to make any changes later. The next step in configuring the root server will be to configure the ltsp-cluster-control centre. You start by installing the web based admin program for the thin clients on the root server:

#### aptitude install ltsp-cluster-control postgresql

Once installed modify the programs configuration file (/etc/ltsp/ltsp-cluster-control.config.php) being sure to not leave any new line spaces before the php-tags (<?php / ?>) or the php will not run. Modify the file to look similar to (noting all the database related information) the example below:

```
<?php
$CONFIG['save'] = "Save";
$CONFIG['lang'] = "en"; #Language for the interface (en and fr are supported"
CONFIG['charset'] = "UTF-8";
$CONFIG['use_https'] = "false"; #Force https
$CONFIG['terminal_auth'] = "false";
$CONFIG['db_server'] = "localhost"; #Hostname of the database server
$CONFIG['db_user'] = "ltsp"; #Username to access the database
$CONFIG['db_password'] = "ltsp"; #Password to access the database
$CONFIG['db_name'] = "ltsp"; #Database name
$CONFIG['db_type'] = "postgres"; #Database type (only postgres is supported)
$CONFIG['auth_name'] = "EmptyAuth";
$CONFIG['loadbalancer'] = "192.168.1.101"; #Hostname of the loadbalancer
$CONFIG['first_setup_lock '] = "TRUE";
$CONFIG['printer_servers '] = array ("cups.yourdomain.com"); #Printserver hostname
$CONFIG['rootInstall'] = "/usr/share/ltsp-cluster-control/Admin/";
?>
```

Next you need to create the user referred to in the configuration file for the database. Make sure that you use the same password that you set in the configuration file (shown above) when you create the new account:

sudo —u postgres createuser —SDRIP ltsp Enter password for new role: Enter it again:

Now also create the necessary database:

sudo -u postgres createdb ltsp -O ltsp

then move to the new db directory (/usr/share/ltsp-cluster-control/DB/) and create the db tables:

cat schema.sql functions.sql | psql -h localhost ltsp ltsp Password for user ltsp: Change to the root's directory, /root and make sure you sudo to root (act as root) and then fetch the following two configuration files using wget:

wget http://bazaar.launchpad.net/%7Eltsp-cluster-team/ltsp-cluster/ ltsp-cluster-control /download/head%3A/controlcenter.py-20090118065910-j5inpmeqapsuuepd-3/control-center.py wget http://bazaar.launchpad.net/%7Eltsp-cluster-team/ltsp-cluster/ ltsp-cluster-control /download/head%3A/rdpldm.config-20090430131602-g0xccqrcx91oxs10-1/rdp%2Bldm.config

Once the files have been saved to root's directory you can then modify the first file that was downloaded, namely control-centre.py. Change it according to the example below:

```
#/usr/bin/python
import pgdb, os, sys
#FIXME: This should be a configuration file
db_user="ltsp"
db_password="ltsp"
db_host="localhost"
db_database="ltsp"
```

Now, install one last package, the python postgres package, python-postgres. Then you need to stop the apache server daemon and install the two files you downloaded to root's directory by running:

python control-center.py rdp+ldm.config

This will produce the following output to your screen:

```
Cleaned status table
Cleaned log table
Cleaned computershw table
Cleaned status table Cleaned log table
Cleaned computershw table
Regenerated tree
```

You can now restart the apache service. You can now see the control centre interface via the web at the following address (access this via your browser): http://localhost/ltsp-cluster-control/Admin/ admin.php Once you have loaded the page, make the following configuration changes via the website – use the first page ("Configuration") to make the changes:

LANG = en\_EN.UTF-8 LDM\_DIRECTX = True LDM\_SERVER = %LOADBALANCER% LOCAL\_APPS\_MENU = True SCREEN\_07 = 1dm TIMESERVER = ntp.ubuntu.com XKBLAYOUT = en

You are now in a position to create new nodes via the web interface. To create a new node:

- Press the create child button
- Type the name of your node in the name field and save
- Additional configuration information can be seen on the Ubuntu website look for the configuration screenshots: https://help.ubuntu.com/community/UbuntuLTSP/LTSP-Cluster

The final step in installing the root server involves installing and configuring the "loadbalancer". In order to do this install the following package:

aptitude install ltsp-cluster-lbserver

Once installed you need to modify the configuration file of the loadbalancer (found at /etc/ltsp/lbsconfig.xml) as per the example below:

```
<?xml version = "1.0"?>
<lbsconfig >
lbservice listen = "*:8008" max-threads = "1" refresh -delay = "60" returns = "$IP"/>
<lbslave is-slave="false"/>
<mgmtservice enabled="true" listen="*:8001"/>
<nodes>
<group default="true" name="karmic">
<node address=http://192.168.1.102:8000 name="1tsp-appserv01"/>
</group>
</nodes>
<rules >
<variable name="LOADAVG" weight="50">
<rule capacity =".7"/>
</variable>
<variable name="NBX11SESS" weight="25">
<rule capacity = "$CPUFREQ * $CPUCOUNT * $CPUCOUNT / 120" \
                                  critical = "$CPUFREQ*$CPUCOUNT*$CPUCOUNT/100"/>
</variable >
<variable name="MEMUSED" weight="25">
<rule capacity = "$MEMTOTAL-100000"/>
</variable>
</rules>
</lbsconfig>
```

You need to specifically focus on the group name, "karmic" and either leave it as such or change it to something you would prefer. I changed mine to "oneiric". You then also need to focus on the node information. Currently there is only one node in the "karmic" group, namely, ltsp-appserv01. If you have more than one application server that you intend to configure then you need to add additional nodes to the file. In my case, I have two application servers and so I added the following line below the first one:

If you have more than two application servers, list all of them here within your group. Once you have added all of their details here, you can save and exit the file. Your root server is now ready. We can move on to installing and configuring the application servers.

#### I.1.2 Application server(s) for LTSP-cluster

The LTSP-cluster application servers serve the thin clients in terms of their access to applications. The thin clients, once booted, connect to the application servers via ssh tunnels in order to access their desktop environments and run applications. This process can be thought of as a similar experience as using rdesktop or VNC to access a remote desktop and run applications.

To set up one or many application servers (more is better as it allows you to serve more thin clients) you need to install another computer(s) with the 64-bit alternative Ubuntu CD. However, do NOT install the LTSP server option available on the alternative CD. Once you have installed the OS then make sure that it is up to date by first editing your sources list (/etc/apt/sources.list) to point at appropriate sources that you have access to and then running the following:

sudo apt-get update sudo apt-get dist-upgrade

Once up to date you can begin installing and configuring the necessary software for the application server component of the LTSP-cluster. Install the following:

aptitude install edubuntu-desktop ltsp-server ltsp-cluster-lbagent ltsp-cluster-accountmanager

This will take some time to complete, especially the installation of the Edubuntu-desktop. This will require either a reboot or at least restarting X in order to make use of the new Edubuntu desktop. Once complete, you now need to create the following file: /etc/xdg/autostart/pulseaudio-module-suspend-onidle.desktop (see the example below).

```
[Desktop Entry]
Version=1.0
Encoding=UTF-8
Name=PulseAudio Session Management
Comment=Load module-suspend-on-idle into PulseAudio
Exec=pact1 load-module module-suspend-on-idle
Terminal=false
Type=Application
Categories=
GenericName=
```

Next create a test user and add it to the following groups:

```
sudo adduser ltsp001
sudo adduser ltsp001 fuse
sudo adduser ltsp001 audio
sudo adduser ltsp001 video
```

You now have a basic installation and configuration of the Ubuntu LTSP-cluster.

#### I.1.3 LTSP-cluster and NAT

If you want to have one or more of your servers act as gateways to the Internet, you can do that by having at least one of the servers with more than one network card. The simplest is probably to have two network cards within the LTSP-cluster root server. The server can then additionally act as a gateway for all the clients and servers on the internal network. The root server is configured to have two network cards, in separate subnets. The other clients and server then have one network card which binds an IP in the internal subnet and is configured such that the root server are their default gateway – either manually or via DHCP.

#### I.1.4 LTSP-cluster and OpenLDAP

As discussed in my implementation of a DNS round robin LTSP cluster, LDAP allows administrators to centrally manage users and groups (amongst a variety of other things) [207]. Centralised management of users and groups is of particular interest here as it provides a means for managing the users and their associated groups across all the application servers within the cluster. As a result regardless of which application server a thin client is connected to the same user account details apply [207]. Central user management is important when your services are handled by a cluster of computers, as a distributed user management system would require synchronisation and additional work, adding to levels of complexity and creating a lot opportunities for errors to creep in. In addition, central authentication supports access to the users' home directories (files and folders) that are centrally stored on the root server and accessible via NFS [207].

In order to make use of the advantages offered by LDAP the LDAP service needs to be installed on the root server. Start by installing OpenLDAP on the root server:

sudo apt-get install slapd ldap-utils

The installation guide at https://help.ubuntu.com/community/UbuntuLTSP/LTSP-Cluster \_NAT\_OpenLDAP recommends that the next step is to load the default schemas that are shipped with OpenLDAP (cosine.ldif, nis.ldif, inetorgperson.ldif), however, if you are running the latest version of Ubuntu then you need not complete this step as they are loaded now by default (see https://help.ubuntu.com/11.10/ serverguide/C/openldap-server.html ). That basic installation also assume that your domain name information is dc=example, dc=com. If you want to change this you need to make sure that in the domain name for the local host is set in /etc/hosts. For example:

127.0.0.1 localhost 127.0.1.1 ltsp-root01.local.org ltsp-root01 192.168.1.101 ltsp-root01.local.org ltsp-root01 192.168.1.102 ltsp-appserv01.local.org ltsp-appserv01 192.168.1.103 ltsp-appserv02.local.org ltsp-appserv02

```
# The following lines are desirable for IPv6 capable
# hosts::1ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

In addition, you need not add the back-end and front-end ldif files that they mention, Ubuntu is doing that all automatically now during the installation process. All you need to do is start adding the user information – populating your LDAP database – that you want to have. In my case I added the following content to the database by creating a file call add\_content.ldif which contained the contents seen in the example below:

```
# First level hierarchy - people
dn: ou=people, dc=local, dc=org
objectClass: organizationalUnit
ou: people
# First level hierarchy - groups
dn: ou=groups, dc=local, dc=org
objectClass: organizationalUnit
ou: groups
# Second level hierarchy – Example user
dn: uid=ings, ou=people, dc=local, dc=org
objectClass: inetOrgPerson
objectClass: posixAccount
objectClass: shadowAccount
uid: ings
sn: Sieborger
givenName: Ingrid
cn: Ingrid Sieborger
displayName: Ingrid sieborger
uidNumber: 1000
gidNumber: 1000
userPassword: phdproject
gecos: Ingrid Sieborger
loginShell: /bin/bash
homeDirectory: /home/ings
shadowExpire: -1
shadowFlag: 0
shadowWarning: 7
shadowMin: 8
shadowMax: 999999
shadowLastChange: 10877
mail: ings@ds9.org.za
initials: IS
# Second level hierarchy – Example group
```

```
dn: cn=ings,ou=groups,dc=local,dc=org
objectClass: posixGroup
cn: ings
gidNumber: 1000
```

To add the content to the database you need to run the following command:

```
ldapadd -x -D cn=admin, dc=local, dc=org -W -f add_content.ldif
```

Please note: you will probably be asked for the LDAP password – which you should have added during installation of LDAP.

You can add many new users through this method of creating ldif files and manually adding them. Alternatively you can use a graphical LDAP manager – like phpLDAPadmin which I discussed in the implementation of the DNS round robin LTSP cluster. I chose to install and use phpLDAPadmin for this installation of LDAP in the Ubuntu LTSP-cluster solution. It allows you to add groups and users in a more user-friendly web environment (for more information read the section titled "OpenLDAP GUI app for database population" in the DNS round robin LTSP cluster configuration).

Now the application servers need to be configured to use LDAP for authentication. In order to do that you need to install a few new packages on the application servers and make some changes to pam.d configuration files. The packages that need to be installed are: libnss-ldap and ecryptfs-utils:

sudo apt-get install libnss-ldap sudo apt-get install ecryptfs-utils

You will be asked to configure your LDAP client during installation. However, should you wish to change any of the configuration options that you set at that time you can do so by running the following command (which will ask you the same configuration questions as those at installation time in order to allow you to reconfigure your client should you need to):

sudo dpkg-reconfigure ldap-auth-config

The results of the dialog can be seen in /etc/ldap.conf. If your server requires options not covered in the menu edit this file accordingly. You then need to enable the auth-client-config LDAP profile by entering:

sudo auth-client-config -t nss -p lac\_ldap

A detailed explanation of what this command does can be read in the section pertaining to the DNS round robin LTSP cluster configuration. Finally, use the pam-auth-update utility to configure the system to use LDAP authentication:

sudo pam-auth-update

From the pam-auth-update menu choose LDAP and any other mechanisms that you need (in our example we just choose LDAP). Once the above has been completed the application servers will authenticate all clients against the LDAP service running on the root server.

At this stage, the way to create a user is a bit messy, it would involve creating a new user via the LDAP php web interface as well as creating a user using the standard user and group creation tools in (Ed)Ubuntu. This is obviously messy. In order to have the creation of the home directory and permissions automatically we can use pam, however, it crucially requires the LDAP server to trust itself for authentication – this is important otherwise NFS will not be able to create the home directories with the correct user permissions. The first step is then to follow the LDAP client instructions here on the LDAP server itself. It will point to itself in term of its ldap server (in my case ldap://192.168.1.101) but crucially will use both Unix AND LDAP authentication methods (options provided during pam-auth-update)! Once this completed, check that it works correctly by creating a new LDAP user and group and then using the id program in UNIX query the id of the new user, if you get a response your system is using LDAP for authentication. If it tells you there is no such user you will need to investigate why it is not querying the LDAP server.

Once this is successful you can now configure the application servers to create the home directories of new accounts automatically the first time that they are used. This is done through via PAM on the application servers. Edit the file /etc/pam.d/common-session on all of the application servers you have running and add the following line to the file:

session required pam\_mkhomedir.so skel=/etc/skel/ umask=0022

You should now find that you can create new users and groups via the php LDAP admin web interface and that their home directories will be created the first time that they log in via a thin client, with all the correct permissions and owner information preset.

#### I.1.5 LTSP-cluster and NFS

As previous discussed in the DNS round robin LTSP cluster section, an Network File System (NFS) is a network or distributed file system that allows users on client computers to access files over a network (on a remote computer/server) in a manner similar to how local storage (files on the local hard disk) is accessed [291]; the file system resides physically on one computer, the file server, which shares it's files over the network to other computers, the client computers [292] providing transparent access to remote disks [204]. For the client machines connecting to an NFS service, the remote file system is indistinguishable from the local file systems on the local hard disk [204].

The Ubuntu LTSP-cluster implementation recommends using NFS to access centrally stored user accounts. This means that instead of having multiple user accounts across the numerous application servers, in their model, and thus having to find some way of syncing those accounts in order to maintain integrity they can centrally store all accounts which are accessed at login from any client. The root server is recommended as the NFS server which will store all the user home directories and files. The application servers are then to be configured as NFS clients which mount those home directories as various users log into the system. Thin clients then connecting to the application servers will be authenticated via LDAP and then once authenticated and logged into the application server their home directories on the NFS server will be accessible via the application server (which mounts the exported home directories at boot up). This functionality is essentially identical to my DNS round robin LTSP cluster server implementation. The installation and configuration of the system is simple. First, on the root server you need to install the NFS kernel server: aptitude install nfs-kernel-server

Next you need to edit the file /etc/exports to look similar to:

/home 192.168.0.0/16(rw, no\_root\_squash, insecure, no\_subtree\_check, async)

Once done some recommend running exportfs with the arguments -avr in order to synchronise and update what NFS is exporting. Alternatively you can just restart the NFS kernel server:

/etc/init.d/nfs-kernel-server restart

On the client side, in this case on the application server(s) you need to install nfs-common and configure /etc/fstab:

aptitude install nfs-common

Once installed add the following lines to /etc/fstab:

#NFS home directories mounted from NFS server
192.168.1.101:/home /home nfs \_netdev, auto 0 0

If you are running NFS v4 then you will also need to edit the file /etc/default/nfs-common and set the variables NEED\_STATD and NEED\_IDMAPD as follows:

NEED\_STATD=no NEED\_IDMAPD=yes

Not doing so results in the wrong UID and GID mapping between the clients and will cause a lot of errors when users try to log in. Now if you run mount -a as root (alternatively rebooting your application server will achieve this too but will be a bit more disruptive while you wait for it to reboot) you should mount the exported home directories from the NFS kernel server.

# I.2 The difference between the DNS Round Robin and Ubuntu LTSP-cluster solutions

Essentially the difference between the solution implemented by the LTSP-cluster group and the DNS round robin solution that I proposed is the mechanisms by which the load distribution is handled. In my solution the load balancing is handled by the relative randomness in DNS resolution by the clients. The solution assumes that each of the servers, in the cluster, are identical (capable of the same level of service to clients) in that they can handle the same level of load. While the LTSP-cluster solution uses SNMP to ascertain the level of load and capabilities of each server in the cluster and then "intelligently" assign clients to an

application server such that they will get the best possible service from the servers. It's more carefully assigning work load in order to achieve a better distribution and utilization of the available hardware. In circumstances where the servers in the cluster all vary this solution would be a better option than the DNS round robin. However, in such circumstances where the hardware is identical the simplicity of the DNS round robin solution would be adequate.

# Appendix J

# Single System Image using Kerrighed configuration

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

A single system image (SSI) was implemented using Kerrighed. Kerrighed is an open source SSI project that provides the view of a single SMP machine on top of a cluster. The Kerrighed solution consists of a set of kernel distributed services that are in charge of the global management of the cluster resources [283, 193]. According to the project web page, the goals of Kerrighed are "ease of use, high performance of applications, high availability of the cluster, efficient resources management, and high customizability of the operating system" [283], while the implementation is an extension of the Linux operating system [193]. One of the Linux distributions that Kerrighed has been ported to is Debian, which is very similar to the Ubuntu and Ubuntu-related flavours like Edubuntu. We used the Edubuntu Linux distribution together with Kerrighed to build a SSI thin client cluster server. The SSI cluster had one master server from which the common file system was shared via NFS (Network File System). The hard disk of the master server was divided into two partitions. The Ubuntu operating system was installed in the second partition. The Edubuntu partition, patched with the Kerrighed extensions, was then mounted by the Ubuntu system and exported to the network. This meant that the thin client computers were connecting to Edubuntu Kerrighed Linux nodes.

#### J.1 Setting up NFS

Firstly, instead of the recommended Kerrighed installation where you install one operating system on the disk in the "conventional" manner, i.e. to install from an ISO, and then to chroot into a directory of that file system and then build the second operating system there using boostrapping principals, I installed two full operating systems side by side in the "conventional" manner, both from ISOs. The one operating system was regular Ubuntu, while the second was Edubuntu. The Ubuntu operating system operates as the NFS server, TFTP server and DHCP server. The Edubuntu partition is the root of the nodes that are booted via NFS, resulting in an Edubuntu Kerrighed SSI cluster.

Once both the operating systems were installed side by side, I began to follow the instructions at https://wiki.ubuntu.com/EasyUbuntuClustering/UbuntuKerrighedClusterGuide. Follow the instructions to install and configure the DHCP service, TFTP service and NFS service on the Ubuntu server. I didn't copy the kernel and initrd.img files however to the tftpd-hpa directory as I have not created an NFS bootable kernel or initrd.img. These will be done as soon as I boot back into the Edubuntu partition and are explained below.

I skipped configuring the bootable file system in section 1.4 as we installed a second operating system, Edubuntu, onto the hard disk already, but did install dhcp3-common nfs-common nfsbooted openssh-server. The first two are already installed as part of a standard Edubuntu installation. The second are not. Openssh is easy to install from the regular ubuntu repositories, but nfsbooted needs to be installed from the Debian repositories. For this I needed to add a line to my sources.list (/etc/apt/sources.list) like

#For nfsbooted deb http://debian.mirror.ac.za/debian lenny main

I then also created the Kerrighed directory and set up fstab in the Ubuntu partition to mount the Edubuntu partition into that directory (nfsroot/kerrighed/). You then need to configure /etc/fstab, /etc/hosts and the network interfaces for your chrooted partition (in my case Edubuntu). Then you need a Linux kernel image that can be booted. In the TFTP service section they just recommend you use a kernel that you have downloaded, or one that you have already available on your computer. For Kerrighed 3.0.0 you have to use the Linux kernel version 2.6.30. The version of Ubuntu which used the closest kernel was Ubuntu 9.10, which used 2.6.31. So hopefully the slight downgrade of the kernel will not negatively affect the rest of the operating system. Kernels can be downloaded from The Linux Kernel Archives (www.kernel.org).

In order to get it to boot properly I had to make a few changes to the kernel in order to support NFS and PXE booting. The variables that I set, using make menuconfig, were:

- Networking support
  - Networking options
    - \* IP: Kernel level autoconfiguration
      - · IP: DHCP support
- File Systems
  - Network File Systems
    - \* All the NFS client support options bar the developmental ones
    - \* All the NFS server support options
    - \* Root File system on NFS

Once you have set those, you might also need to build in your network interface drivers, for vmware those were under

- Device Drivers
  - network device support
    - \* there are some vmware specific drivers that can be chosen, as well as under Ethernet 10M or 100Mbit options choose the one called AMD PCnet32 PCI support.

Now, save the changes and then build a new kernel. To build the kernel I followed the instructions at https://help.ubuntu.com/community/Kernel/Compile (make sure you install all the necessary packages in order to build custom kernels – I followed the karmic suggestions (Ubuntu 9.10) as that is the OS version I am using).

Specifically, once you have finished saving the config you need to run the following:
```
make-kpkg clean
fakeroot make-kpkg ---initrd ---append-to-version=-some-string-here
kernel-image kernel-headers
```

That will build the image and its corresponding headers as .deb packages that can be added using dpkg -i package\_name.deb. Make sure you add the image before you add the headers.

Once you have added the packages you will then find the vmlinux, System and Config files in the /boot directory. You only need to then create the initram image. Edit the initramfs.conf file (found at /etc/initramfs-tools/initramfs.conf) and change the Device to the Ethernet interface you intend to use, and the boot device from "local" to "nfs". Typically you would also change the modules from "most" to "netboot", but I found that when I did that not enough of the kernel module were loaded in order to boot properly and so it's best to use "most"; that way all the kernel modules necessary for boot are loaded. You build an initrd image for the new kernel you just made by typing

mkinitramfs o name\_of\_your\_initrd.img kernel\_version\_of\_init

Once you have a kernel and an initrd you can then copy those from your Edubuntu chroot to the TFTP folder in your Ubuntu operating system - /var/lib/tftpboot/ - and edit the default file in pxelinux.cfg so that it points to those two files. The clients will then TFTP those files and boot your new nfs kernel image. Do this in order to check that the NFS booted image works as expected on the four proposed kerrighed nodes.

### J.2 Setting up Kerrighed

You are now in a position to start the kerrighed portion of the kernel patch. While you may have created the NFS bootable kernel in the Edubuntu partition of your server, it would appear that for the kerrighed portion you must have it installed on your ubuntu partition, as it is the container that runs the kerrighed system for you. So build your kerrighed kernel in the ubuntu partition.

Start by downloading the kerrighed source files from the kerrighed website (they will be a gzipped tar). And unpack that in /usr/src/. Next make sure that you have the following packages installed (some of which you will already have from having to build kernels earlier):

- automake
- autoconf
- libtool
- pkg-config
- gawk
- rsync
- bzip2
- gcc-4.4

- libncurses5-dev
- wget
- lsb-release
- xmlto
- patchutils
- xutils-dev
- build-essential
- openssh-server
- ntp

Once you have unpacked the Kerrighed source, copy a tarball of the Linux source (linux-2.6.30.tar.bz2) that you are using to the patches directory within the kerrighed parent directory. If you don't the subsequent steps you follow from the kerrighed documentation site will fail. From here on out we have to ignore the instructions at https://wiki.ubuntu.com/EasyUbuntuClustering/UbuntuKerrighedClusterGuide as they apply to kerrighed 2.4 and instead we apply the principles together with the instructions from kerrighed 3.0 found at http://www.kerrighed.org/docs/releases/3.0/INSTALL.

There is a known memory remap bug in the kerrighed-3.0.0 patch which is fixed by patching mm/mremap.c file with the patch found at http://git-externe.kerlabs.com/?p=kerrighed-kernel.git;a=patch;h=c5d39f7f6fe8977c157d42c9d2d41e07404b8847 You must do this before running make (as seen in the next paragraph) otherwise is will fail. To run the patch type the following inside the \_kernel/mm directory of your kerrighed extraction:

patch mremap.c < .../.../kerrighedfix.patch

You can only apply the patch once you have configured the kerrighed kernel which you do by using the kernel config of the nfs kernel that was built successfully. In order to configure the kernel with the .config we spent time working on previously for NFS we need to configure kerrighed with the following parameters

```
./configure ---with-kernel-config-file=/home/ings/file_name -sysconfdir=/etc
```

Now apply the patch. Once this has completed you can then run make and finally make install in order to build your new kernel. You will find a copy of the vmlinux, System and Config files in the /boot directory. You can then using initramfs tools make the initrd.img file for your kernel.

Next check to make sure that everything was installed as expected. The Ubuntu Kerrighed guide at https://wiki.ubuntu.com/EasyUbuntuClustering/UbuntuKerrighedClusterGuide lists software that should have been installed to your machine. Make sure that it's all been installed. Once you are sure that everything is there you need to configure your Kerrighed installation. The Ubuntu Kerrighed guide (link above) as well as the Kerrighed 3.0.0 installation notes are useful here to help you work through what still needs to be done in order to configure the service properly. Additional useful resources include:

- http://www.debianadmin.com/how-to-set-up-a-high-performance-cluster-hpcusing-debian-lenny-and-kerrighed.html
- http://kerrighed.sourceforge.net/wiki/index.php/Kerrighed\_on\_NFSROOT
- http://kerrighed.sourceforge.net/wiki/index.php/SchedConfig

These should help you understand the various configurations and give you a clearer idea of how the various portions work together. This involves creating a /config directory in the root directory and adding the relevant mount options to /etc/fstab

#### configfs /config configfs defaults 0 0

Once this is done, when it came to configuring the Kerrighed installation I followed the install notes for Kerrighed 3.0.0 as the installation instructions at https://wiki.ubuntu.com/EasyUbuntuClustering/ UbuntuKerrighedClusterGuide did not refer to files that were in existence in the new version of Kerrighed. For example, /etc/kerrighed\_nodes. Alternative installation/configuration notes are available on the Kerrighed wiki at http://kerrighed. sourceforge.net/wiki/index.php/ UserDoc#Quick\_usage\_guide There are 3 files referred to that are important for configuration, /etc/ default/kerrighed-host,/etc/default/kerrighed, and/etc/kerrighed/krginit\_ helper.conf.

Then reboot into the ubuntu nfs server partition, copy the vmlinuz and initrd.img files for the new kerrighed kernel into your /var/lib/tftpboot directory. You should now edit the default file in pxelinux.cfg of your TFTP server so that it uses the new kerrighed kernel and initrd.img files for booting your kerrighed patched nfs nodes.

I tried to boot the clients using pxelinux – the same way in which LTSP boots clients and the same way in which the original kerrighed booted nodes. However, this did not appear to be working, the node was crashing immediately at the point it tried to load the kernel. This error is linked to VMWare, (error message: \*\*\* VMware Workstation internal monitor error \*\*\* vcpu-0:NOT\_IMPLEMENTED vmcore/vmm/intr/apic.c:1903) and most of the online information around the error message suggested that this could be simply fixed by using the latest version of VMWare workstation 8. However, the problem seems to persist with the latest version installed.

As a possible method to circumvent this issue and having a booted system the kerrighed NFS notes at http://www.kerrighed.org/wiki/index.php/ Kerrighed\_on\_NFSROOT\_Old recommend the use of grub instead. This was achieved by providing the nodes with a very small hard disk, installing grub and then configuring it to load the kernel and initram image on the disk and then use the central NFS node as the hard disk for each node. Therefore this just removes the TFTP process from the normal NFS boot process recommended. I think it might be worthwhile however to ghost the server image onto a real machine and then see if this problem persists on physical hardware. The steps in configuring the nodes to use grub on their local disks are:

- Give node a 1G hard disk
- Boot a live CD on node
- Fdisk /dev/sda
  - Create a new primary partition
  - Save

- Give partition a file system
  - Mkfs ext3 /dev/sda1
- Mount /dev/sdal to /mnt
- Install grub to /dev/sda1
  - Grub-install -root-directory=/mnt /dev/sda
- Make your own grub.cfg inside /boot/grub
- Copy kernels to /boot on /dev/sda1
- Then shutdown and remove live CD
- Reboot with hard-disk and choose to boot the grub option for Kerrighed

This allowed me to boot the kernels and having a working operating system. However I struggled to start the kernighed service with this installation. For some reason the software was not convinced that the kernel that was running was kerrighed compliant. I am not sure if I was just repeatedly making the same mistake when configuring the kernel. While looking on-line for possible sources of help with the errors I was encountering I found a hidden Debian repository for Kerrighed, with pre-compiled kernels! I immediately set up a new virtual machine and installed Edubuntu LTSP 9.10 and Ubuntu 10.10 (as the NFS) component. I set up both systems as necessary but included the Kerrighed repositories that I had found in my sources file in order to install the pre-compiled Kerrighed packaged for an Ubuntu/Debian OS. This meant adding the following lines to my /etc/apt/sources.list file in the Edubuntu partition:

deb http://www.kerrighed.org/debian sid main deb-src http://www.kerrighed.org/debian sid main

You then need to update your sources tree by running apt-get update and then install Kerrighed and Kerrighed-dev by running

#### aptitude install kerrighed kerrighed-dev

This should install the kerrighed kernel for you and you should find the config, vmlinux, system and initram files in your /boot directory. If there is no initrd file you can create one in the usual manner. These can be copied to your NFS server in order to serve these files to the NFS nodes at boot time. You need to configure your Kerrighed system then in order to run, specifying the number of nodes in the cluster, the session, etc. Much the same as you would with the manual installation. For more information of using the pre-compiled Debian packages see the following two sites:

- http://www.estrellateyarde.org/discover/virtualizacion/clusters/clusters-kerrighed/ instalar-kerrighed
- http://www.estrellateyarde.org/discover/virtualizacion/clusters/clusters-kerrighed/ manejar-kerrighed

Both sites are in Spanish (I think), but Google Translate does an excellent job of translating them into English so that you can follow the how-to guide.

This required a pxegrub file, which I could not find on-line and so had to build one from the grub source. I downloaded the original grub source from ftp://alpha.gnu.org/gnu/grub/ and un-tarred them. They required the package "bison" being installed in order to configure the source. Run ./configure in order to configure the source, from within the untarred directory. Once completed run make.

Now boot your kerrighed nodes, checking firstly that the system boots and then once booted, that it works as expected.

### **Appendix K**

# Virtual Desktop using Edubuntu and VirtualBox configuration

Note: This appendix is not written in an academic style, but rather as instructions for system administrators who are wishing to reproduce the configuration (i.e. as a "How-to" document).

The system design that I am proposing in this section is a central server or servers that will serve an image for the thin clients to boot – much like LTSP. A server that responds to DHCP requests and serves TFTP files to booting thin clients is required. Once the clients are booted instead of connecting to an X server via ssh, which is the standard LTSP installation/configuration, they will connect using VNC to a virtual desktop running on the central server(s) using Virtualbox. We can balance the required services across multiple machines. One server could for example provide the DHCP, TFTP and file services (NFS) to the clients and virtual server(s). Then one or more machines could host the required virtual desktops. Each virtual desktop will require a minimum of 512M of RAM, 1GB however, would be preferable. Thus depending on the number of virtual desktops that could connect at any given time this solution has the greatest memory requirements when compared to the other cluster solutions discussed thus far. Assuming 20 clients, you need a minimum of 10G of RAM, but 20G would be better. Thus 64-bit servers with access to 10G of RAM or more, making the cluster hardware requirements a little more high-end than those of the LTSP solutions.

### K.1 Desktop virtualization: Installation and configuration

Begin by installing Ubuntu server 11.10 on a PC/Server. Once installed, I updated the image to the latest packages and installed the following:

- Ubuntu desktop
- LTSP-server and built the LTSP-client-image (similar to the LTSP-Cluster installation in Ubuntu)

aptitude install ltsp-server ltsp-build-client arch i386 prompt-rootpass

• Virtualbox (normal installation through Ubuntu channels at the time was a bit buggy and so I used the instructions from Sun: http://codefat.com/2011/11/installing-a-virtualbox-environment-on-ubuntu-server-11-04/)

In addition, in order to set up the Guest Additions that Virtualbox offers which consist of device drivers and system applications that optimize the guest operating system for better performance and usability, you need to install the following once Virtualbox has been installed on your server:

• virtualbox-guest-additions-iso (again, doable via normal Ubuntu source channels, but also done through downloading the ISO directly from Sun, available at: http://download. virtualbox.org/virtualbox/4.1.4/VBoxGuestAdditions\_4.1.4.iso)

Next you need to configure Virtualbox and install an Edubuntu desktop Virtual Machine (VM) on Virtualbox. To do this I read the following websites to learn more about Ubuntu and Virtualbox:

- https://help.ubuntu.com/community/VirtualBox
- https://help.ubuntu.com/community/VirtualBox/Installation
- https://help.ubuntu.com/community/VirtualBox/FirstVM
- https://help.ubuntu.com/community/VirtualBox/Networking

I specifically focused on the page: https://help.ubuntu.com/community/VirtualBox/FirstVM . It contains a number of useful screen shots to help you create your own VMs and how to install the guest operating system once the virtual machine has been configured. Follow the instructions there carefully and you should comfortably create your VM. Once created you need to install the operating system of the guest. Again, the instructions are there to help you do so, but you need to first make sure that you have the ISO file of the operating system you wish to install available on the server. I chose to install the Ubuntu Desktop i386 operating system. This means that you need to add it to the CD/DVD Rom drive. Do this its a bit different from the website's instructions. Click on the "Settings" menu in the VirtualBox Manager Window (Figure K.1).

8 🗇 🕘 Oracle VM VirtualBox Manage	er
<ul> <li>Image: A state of the state of</li></ul>	😨 Details) 💷 Snapshots
Running	OS Type: Obunco
	System
	Base Memory: 1024 MB Boot Order: Floppy, CD/DVD- ROM, Hard Disk
	📴 Display
	Video Memory: 12 MB Remote Desktop Server: Disabled
	Storage
	IDE Controller IDE Secondary Master (CD/DVD): Empty SATA Controller SATA Port 0: Edubuntu Desktop.vdi (Normal, 10,00 GB)
	De Audio
	Host Driver: PulseAudio

Figure K.1: Virtualbox Manager Window

This will open up the Setting Window for you to change the configuration of your VM. Here choose Storage, then click on Empty under CD/DVD Rom and then on the little CD image on the right, click on the drop-down arrow and choose to select an iso from your hard disk (Figure K.2). This will open a file manager window for you to locate your ISO and select it in order to add it to the drive. Once that is done you should be able to boot the machine and boot off the CD/DVD Rom drive and install your guest operating system of choice.



Figure K.2: Virtualbox Settings Window

Installation is then normal to installing on any physical computer or virtual machine using VMWare. Once the installation of the Ubuntu Desktop had completed, I then did the usual update of sources and upgrade to the latest version of the operating system. When that was completed I installed the Edubuntu-desktop; building an Edubuntu Desktop Virtual Machine. Alternatively, you could install the Edubuntu desktop image from the Edubuntu ISO image. In order to improve the mouse and keyboard interactions of my VM I then followed the Virtualbox User Manual (available at: http://www.virtualbox.org/manual/ UserManual.html) and specifically Chapter 4 which focuses on Guest Additions (http://www.virtualbox. org/manual/ch04.html). Once the virtualbox-guest-additions-iso package is installed in Ubuntu the rest is relatively simple. You need to just add the ISO to your CD/DVD Rom (in much the same way you did early when installing the operating system) which you can find at: /usr/share/virtualbox/ VBoxGuestAdditions.iso All you then need to do is open a terminal on your VM, change directory to the where the CD is mounted (usually in /media for Ubuntu) and run the following as root:

### sh ./ VBoxLinuxAdditions.run

Once it has completed you can reboot your VM in order to have the Guest Additions take effect. Now you have one complete Edubuntu Desktop.

The next step after installing a working VM is to be able to remotely access that VM from another host somewhere on the network. VirtualBox can display virtual machines remotely, meaning that a virtual machine can execute on one machine even though the machine will be displayed on a second computer, and the machine will be controlled from there as well, as if the virtual machine was running on that second computer. This is what we wish to do with the thin clients that the learners and teachers would have access to, namely, use them to access and control the VMs running on the central server(s). In order to make use of the VirtualBox Remote Display Protocol (VRDP) support you need to install the VirtualBox extension package. According the VirtualBox User Manual (at http://www.virtualbox.org/manual/ch01.html#intro-installing) the extension package can be downloaded from: https://www.virtualbox.org/wiki/Downloads

Once you have clicked on the link to download the extension package, you will be asked if you wish to open this package with VirtualBox, choose yes. VirtualBox will then require the root password from you in order to complete the installation process for you. Please note: the password of a user with administrator permissions will not suffice, it needs the password of the root user. If you have not set one yet, then using a terminal, while root to type:

passwd root

You will then be asked for the new root password and to confirm it, type the password of your choice. Then when VirtualBox asks you for a password in order to install the extension package type in the new root password you have just set. The extension package will be installed and you can now make use of the VRDP support for VirtualBox in order to run headless VMs and connect to them remotely. In order to start a VM headless you need to type the following into a terminal:

VBoxHeadless — startvm "Edubuntu Desktop" — vrdeproperty \ "TCP/Ports=4000" — vrdeproperty "TCP/Address=192.168.1.1"

- VBoxHeadless starts your VM headless, with no display
- startvm the parameter that accompanies VBoxHeadless that tells the VBoxHeadless command to start a VM
- "Edubuntu Desktop" the name of the VM that I wish to start, an argument to the startym parameter
- vrdeproperty the parameter that allows you to specify Virtual Remote Desktop environment variables. Allows you to specify two arguments, "TCP/Ports" and "TCP/Address". These specify on which port to listen to for remote connections and which IP address.

You can now test your ability to connect to your VM by typing the following in another terminal:

rdesktop -a 16 -N 192.168.1.1:4000

- -a 16 sets the colour depth for the connection
- -N enables numlock synchronization between the X-server and the remote RDP session.

A window should open up in which the remote VM is running and with which you can now interact and control. The next step is to integrate the rdesktop access to the remote VM via the thin client image built on the LTSP server. To do this is relatively simple. You can specify the connection type to another server using the lts.conf file that you need to create at /var/lib/tftpboot/ltsp/i386/lts.conf. You can specify how each client should connect based on the mac address of their network cards – allowing you to specify that each of the clients in a school connects to a separate/unique VM running on the server. For example:

[18:A9:05:A2:DB:33] SCREEN\_07="rdesktop -a 16 -N 192.168.1.1:4000" Changing the Macs and changing the ports you can connect each machine to a different VM – you would need to start each VM on a different port too and keep a list somewhere of which VM you wish to map to which port. The only problem with this configuration is that you would connect one thin client to one virtual machine only and not be able to offer the choice of connecting to another virtual machine. So what we need is some sort of menu that allows the user to choose a virtual machine that they want to connect to. An extremely simple way of doing this is to use the menu bash script (using Whiptail menu (http://en.wikibooks.org/wiki/Bash\_Shell\_Scripting/Whiptail)) created by the LTSP developers. This can be found on Launchpad at: http://bazaar.launchpad.net/~ltspupstream/ltsp/ltsp-trunk/files/head:/client/screen.d/

The file named menu is what you need. Download it and save it to the usr/lib/ltsp/screen.d directory of the thin client chroot (/opt/ltsp/i386/usr/lib/ltsp/screen.d). You can then edit the lts.conf file accordingly to point to the use of the menu file by doing the following:

[18:A9:05:A2:DB:33] SCREEN\_04=menu MENU\_TITLE="Please choose a desktop: " MENU\_ITEM\_01="My Virtual Desktop" MENU\_COMMAND\_01="rdesktop -a 16 -N 192.168.1.1:4000"

This is still rather limited as its difficult to specify the machine you would like to connect to without listing every possibility. In addition, it would mean that at start up the VM server(s) need to start every single possible VM, which is an inefficient use of resources. So it would be better to only start the generic VMs for use by students at the boot up of the VM server(s) and begin personal desktops (for teachers) only when they request to use them. In order to do this myself would mean writing custom code to create a front end menu system that allows users to connect to the VM of their choice. However, the long term maintenance of that would be in question, as there is no guarantee that I would be able to maintain it over time. It seemed a better idea to investigate the use of already available open source software that implements virtual desktop infrastructure. There are two well known and well utilized options, Ulteo [284] and FOSS-Cloud [285]. Ulteo is an open source project that provides "Application Delivery and Virtual Desktop infrastructure" that facilitates the delivery of applications or a desktop which are hosted on either Windows or Linux servers to the end users [293]. FOSS-Cloud, similarly, is an open source project that provides integrated and redundant server infrastructure in order to provide: virtualization and cloud-based services; Windows or Linux based Software as a service; Terminal services; virtual desktop infrastructure (Desktop as a Service); or virtual server environments [285]. I personally chose Ulteo as it has been developed for use on the Ubuntu operating system and so would more readily work with the Edubuntu OS. There are detailed instructions for installing Ulteo on Ubuntu at http://doc.ulteo.com/3.0/Easy\_Installation.html and http://doc.ulteo.com/3.0/Support\_Ubuntu\_Lucid.html. I chose not to implement a VDI solution in the SLL because of the cost associated with the hardware requirements in supporting VDI at the moment, as well as the fledgling nature of the technology at the time of the implementation work. That said though there is growing argument that the time for VDI is ripe and there are examples of it being used to support education initiatives in India [197, 198].

### **Appendix L**

## **Accompanying dissertation DVD**

This Appendix provides a list of the contents of this dissertation's accompanying DVD, that can be perused in order to view the data collected and analysed, as well as copies of the web-based references used during the writing of the thesis. The DVD contains the following:

- Soft copy (PDF) of this thesis
- A repository of all online and digitally available references
- Spreadsheets of the questionnaire and interview/focus group data with analysis
- MRTG data from each of the case studies
- MRTG data from the laboratory tests
- Sikuli data from the laboratory tests

Copies of the server images for the various computing infrastructure configurations are too large to be included on the CD, but are available upon request