

RHODES UNIVERSITY

MASTERS THESIS

---

**Leveraging LTSP to deploy a  
sustainable e-infrastructure for  
poor communities in South Africa**

---

*Author:*

Tichaona Manyara ZVIDZAYI

*Supervisor:*

Prof. Alfredo TERZOLI

*Co-supervisor:*

Prof. Nomusa DLODLO

*A thesis submitted in fulfillment of the requirements*

*for the degree of Master of Science*

*in the Department of Computer Science*

July 28, 2021

# Abstract

Poverty alleviation is one of the main challenges the South African government is facing. Information and knowledge are key strategic resources for both social and economic development, and nowadays they most often rely on Information and Communication Technologies (ICTs). Poor communities have limited or no access to functioning e-infrastructure, which underpins ICT. The Siyakhula Living Lab (SLL) is a joint project between the universities of Rhodes and Fort Hare that has been running for over 15 years now. The SLL solution is currently implemented in schools in the Eastern Cape's Dwesa-Mbhashe municipality as well as schools in Makhanda (formerly Grahamstown). Over the years, a number of blueprints for the meaningful connection of poor communities was developed.

The research reported in this thesis sought to review and improve the Siyakhula Living Lab (SLL) blueprint regarding fixed computing infrastructure (as opposed to networking and applications). The review confirmed the viability of the GNU/Linux Terminal Server Project (LTSP) based computing infrastructure deployed in schools to serve the surrounding community.

In 2019 LTSP was redesigned and rewritten to improve on the previous version. Amongst other improvements, LTSP19+ has a smaller memory footprint and supports a graphical way to prepare and maintain the client's image using virtual machines. These improvements increase the potential life of ICT projects implementing the SLL solution, increasing the participation of members of the community (especially teachers) to the maintenance of the computing installations.

The review recommends the switching from thin clients deployments to full ("thick") clients deployments, still booting from the network and mounting their file systems on a central server. The switch is motivated by reasons that go from cost-effectiveness to the ability to survive the sudden unavailability of the central server.

From experience in the previous deployment, electrical power surge protection should be mandatory. Also, UPS to protect the file system of the central server should be configured to start the shutdown immediately on electrical power loss in order to protect the life of the UPS battery (and make it possible to use cheaper UPS that report only on network power loss).

The research study contributed to one real-life computing infrastructure deployment in the Ntsika school in Makhanda and one re-deployment in the Ngwane school in the Dwesa-Mbhashe area. For about two years, the research also supported continuous maintenance for the Ntsika, Ngwane and Mpume schools.

# Acknowledgements

Firstly, I give all the glory, praise and honour to God for helping me do this work. I would have never been able to finish this without my faith in the unconditional love of God.

Secondly, I would like to thank Prof Alfredo Terzoli and Prof Nomusa Dlodlo for their supervision and for making the thesis possible, not forgetting the help I received from Dr Mosiuoa Tsietsi.

Thirdly, I want to thank my family and colleagues Kudzayi Michael Zvidzayi, Rejoice Zvidzayi, Thamsanqa Nqiwane, Itai Zvidzayi, Tapiwa Chindeka and my classmates for their support.

Lastly, I would like to appreciate the support and funding I received from the Telkom Centre of Excellence and the hardware equipment from SGT solutions.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.1.1 Siyakhula Living Lab . . . . .	2
1.1.2 Linux Terminal Server project . . . . .	2
1.1.3 Thin and thick client history . . . . .	2
1.2 Research context and aim . . . . .	3
1.3 Goals and Objectives . . . . .	3
1.4 Methodology . . . . .	4
1.5 Thesis organisation . . . . .	4
1.5.1 Thesis features and conventions . . . . .	5
<b>2 Related Work</b>	<b>6</b>
2.1 Background . . . . .	6
2.2 ICT for Development . . . . .	8
2.2.1 ICT4D strategic partnerships . . . . .	9
United Nations 2030 Agenda . . . . .	9
South Africa Connect 2030 Agenda . . . . .	9
2.2.2 ICTs in Education . . . . .	10
One Laptop Per Child . . . . .	10
Chrome-books . . . . .	13
2.3 An overview of client-server networks . . . . .	14
2.3.1 BeTwin ES . . . . .	15
2.3.2 OpenThinClient . . . . .	15
2.3.3 Diskless Remote Boot in Linux . . . . .	15
2.3.4 Linux Terminal Server Project . . . . .	15
2.4 The Greek school- switch to LTSP . . . . .	16
2.5 The Siyakhula Living Lab . . . . .	18

2.5.1	Siyakhula Living Lab Challenges . . . . .	21
2.6	Importance of ICT4D . . . . .	22
2.7	Challenges of ICT4D . . . . .	23
2.8	Conclusion . . . . .	24
<b>3</b>	<b>Theoretical background</b>	<b>25</b>
3.1	An overview of some of the protocols . . . . .	25
3.1.1	Dynamic Host Configuration Protocol . . . . .	25
3.1.2	Network File System . . . . .	25
3.1.3	TFTP . . . . .	26
3.1.4	PXE . . . . .	26
	iPXE . . . . .	26
3.1.5	X Windows system core protocols . . . . .	27
	X Windows architecture . . . . .	28
	XDMCP . . . . .	29
3.2	Linux Terminal Server Project . . . . .	29
3.2.1	LTSP booting sequence . . . . .	29
3.2.2	The LTSP19+ versus the old LTSP5+ . . . . .	31
	Support of new technologies . . . . .	32
	Reduced thin client support . . . . .	32
	Virtualisation player to maintain client's image . . . . .	32
	Easy NAT configuration . . . . .	32
	Introduction of iPXE . . . . .	32
	Reduced file sizes . . . . .	32
3.2.3	Benefits of LTSP . . . . .	33
3.2.4	Drawbacks of LTSP . . . . .	33
3.3	LTSP computer clients . . . . .	33
3.3.1	Thin Client . . . . .	34
	Thin clients and green technology . . . . .	36
3.3.2	Thick client . . . . .	37
3.4	Network configuration . . . . .	38
3.4.1	Single NIC configuration . . . . .	38
3.4.2	Dual NIC configuration . . . . .	39
3.4.3	A comparison a single and dual NIC configuration . . . . .	41
3.5	Maintaining client image . . . . .	42
	a) Chroot . . . . .	42
	b) Chrootless . . . . .	42
	c) LTSP19+: raw virtual image . . . . .	42

	Comparing chroot and virtual machine modes . . . . .	44
	Benefits of using a virtualisation player . . . . .	45
	Drawbacks of using a virtualisation player . . . . .	46
3.6	Conclusion . . . . .	46
<b>4</b>	<b>Laboratory experiments: LTSP5+</b>	<b>48</b>
4.1	Experiment 1: LTSP performance . . . . .	48
4.1.1	Experiment questions . . . . .	48
4.1.2	Hypotheses . . . . .	48
4.1.3	Hardware specifications . . . . .	48
	SuperMicro server . . . . .	49
	Mecer Xtreme Jupiter Mini PC client . . . . .	49
	Proline Nanoware client . . . . .	50
	HP T510 client . . . . .	50
	Ethernet switch . . . . .	50
4.1.4	Software specifications . . . . .	50
	Long Term Support . . . . .	51
	Supporting tools . . . . .	51
4.1.5	Experimental procedures . . . . .	52
4.2	LTSP5+ thin-client computing . . . . .	53
4.2.1	Server results . . . . .	53
4.2.2	30 seconds LAN benchmark: Thin clients . . . . .	54
	Observation: LTSP server usage . . . . .	55
4.3	LTSP5+ thick clients results . . . . .	56
4.3.1	Server results . . . . .	57
4.3.2	A 30 seconds LAN benchmark: Thick clients . . . . .	59
	Observation: Server usage . . . . .	59
4.4	An LTSP5+ computing infrastructure client hybrid . . . . .	61
4.4.1	Server usage . . . . .	63
4.4.2	A 30 seconds LAN benchmark: Hybrid clients . . . . .	64
4.4.3	Observation: LTSP computing infrastructure comparison . . . . .	65
4.5	NAT: GNU/Linux packet forwarding . . . . .	67
4.6	Analysis . . . . .	68
4.6.1	A comparison of thin and thick client deployment . . . . .	68
4.6.2	Challenges of the experiment . . . . .	70
	PCI bus errors . . . . .	70
	Scattered documentation . . . . .	72
	Distribution errors . . . . .	72

	Ethernet switch . . . . .	73
4.7	Conclusion . . . . .	73
<b>5</b>	<b>LTSP5+ deployment: Ntsika Secondary School</b>	<b>74</b>
5.0.1	Background . . . . .	74
5.1	Deployment . . . . .	75
5.1.1	Aim of the deployment . . . . .	75
5.1.2	Objectives . . . . .	75
5.1.3	Hardware specifications . . . . .	76
	LTSP server machine . . . . .	76
	Mecer Xtreme Jupiter Mini PC thick client . . . . .	76
	Ethernet switch . . . . .	76
5.1.4	Software specifications . . . . .	77
5.1.5	Deployment and testing methods . . . . .	77
5.2	Results . . . . .	79
5.2.1	A 30 seconds LAN benchmark . . . . .	80
5.3	Analysis . . . . .	81
5.3.1	A comparison of a small and full-blown deployment . . . . .	81
5.3.2	Server booting tests . . . . .	84
5.3.3	Troubleshooting the filesystem . . . . .	85
	Fixing broken Ethernet ports . . . . .	85
	Rebuilding the corrupted chroot environment . . . . .	85
	Possible causes . . . . .	85
	Attempting to solve the problem. . . . .	85
5.4	Conclusion . . . . .	87
<b>6</b>	<b>LTSP5+ deployment: Dwesa schools</b>	<b>89</b>
6.1	Background . . . . .	89
6.2	Ngwane Junior Secondary school . . . . .	91
6.2.1	Aim of intervention . . . . .	92
6.2.2	Objectives . . . . .	92
6.2.3	Hardware specifications . . . . .	92
	SuperMicro server . . . . .	92
	Proline Nanoware client . . . . .	93
	Ethernet switch . . . . .	93
6.2.4	Software specifications . . . . .	93
6.2.5	Procedures . . . . .	94
6.3	Ngwane results . . . . .	96
	A manual for cleaning resitting the RAM module . . . . .	96



6.4	Mpumbe Junior High school . . . . .	100
6.4.1	Aim of the intervention . . . . .	101
6.4.2	Objectives . . . . .	101
6.4.3	Hardware specifications . . . . .	102
	Dell server . . . . .	102
	HP T510 client . . . . .	102
	Ethernet switch . . . . .	102
6.4.4	Software specifications . . . . .	102
6.5	Mpumbe results . . . . .	103
6.6	Conclusion . . . . .	103
<b>7</b>	<b>Laboratory Experiments: LTSP19+</b>	<b>104</b>
7.1	LTSP19+ background . . . . .	104
7.2	Experiment 1: Using Mecer, HP, Proline clients . . . . .	104
7.2.1	Experiment 1 Question . . . . .	104
7.2.2	Hypothesis . . . . .	105
7.2.3	Hardware specifications . . . . .	105
	Dell server . . . . .	105
	Mecer Xtreme Jupiter Mini PC client . . . . .	105
	Proline Nanoware client . . . . .	106
	HP T510 client . . . . .	106
7.2.4	Software specifications . . . . .	106
7.2.5	Experiment 1 procedures . . . . .	106
7.2.6	LTSP19+ server results . . . . .	107
	A comparison of LTSP19+ versus LTSP5+ CPU and RAM usage . . . . .	107
	A comparison of LTSP19+ versus LTSP5+ LAN band- width . . . . .	109
7.2.7	Results . . . . .	111
7.3	Experiment 2: Using the Intel NUC clients . . . . .	111
7.3.1	Experiment 2 Questions . . . . .	111
7.3.2	Hypotheses . . . . .	111
	Dell server . . . . .	112
	Intel NUC client . . . . .	112
7.3.3	Software specifications . . . . .	112
7.3.4	Experiment 2 procedures . . . . .	112
7.3.5	LTSP19+ server results . . . . .	113
7.3.6	Results . . . . .	114

7.3.7	LTSP19+ Adjustments . . . . .	114
	Missing Kernel on image build . . . . .	114
	Users listed on client login screens . . . . .	115
7.4	Conclusion . . . . .	115
<b>8</b>	<b>Making deployment and maintenance easier</b>	<b>117</b>
8.1	Making deployment easier . . . . .	117
8.1.1	An LTSP ready ISO image . . . . .	117
	PinguyBuilder Ubuntu-based customiser . . . . .	117
	Benefits of an LTSP ready ISO image . . . . .	118
	Drawbacks of an LTSP ready ISO image . . . . .	118
8.1.2	Live USB sticks . . . . .	118
8.1.3	Offline application installation . . . . .	119
8.1.4	Supporting software tools . . . . .	120
8.1.5	LTSP19+ client image virtual machines . . . . .	120
	User credentials management . . . . .	121
	User session monitoring . . . . .	123
8.2	Maintenance . . . . .	125
	Potential LTSP server challenges . . . . .	125
8.2.1	Power surge protector . . . . .	126
8.2.2	Uninterruptible Power Supply . . . . .	126
	Network UPS tools . . . . .	127
	NUT installation . . . . .	130
	nut.conf . . . . .	130
	ups.conf . . . . .	130
	upsd.conf . . . . .	131
	upsd.users . . . . .	131
	upsmon.conf . . . . .	131
	upssched.conf . . . . .	131
	A Customised shell script . . . . .	131
8.2.3	Dust collection . . . . .	132
8.2.4	Redundant Array of Inexpensive Disks . . . . .	132
	RAID 1 mirroring . . . . .	132
8.2.5	Extra disk deployment . . . . .	133
8.2.6	Motherboard failures . . . . .	134
8.2.7	Remote system logging . . . . .	134
8.2.8	Remote access . . . . .	135
8.3	Other ways to improve the LTSP experience . . . . .	137

8.3.1	Solid State Drive . . . . .	137
8.3.2	LTSP configuration file . . . . .	137
	Avoiding user session collisions . . . . .	138
	Guest sessions . . . . .	138
	Single NIC over a dual NIC configuration . . . . .	138
8.4	Conclusion . . . . .	139
<b>9</b>	<b>Conclusion and future work</b>	<b>140</b>
9.1	Research summary . . . . .	140
9.2	Revisiting the research questions and goals . . . . .	141
	Research questions . . . . .	141
9.3	Research work contributions . . . . .	143
9.4	Future work . . . . .	143
9.5	Parting thoughts . . . . .	144
	<b>Bibliography</b>	<b>145</b>
<b>A</b>	<b>LTSP Documentation</b>	<b>156</b>
A.1	LTSP Background . . . . .	156
A.2	LTSP 5+ Installation procedures . . . . .	156
	A.2.1 Adding the LTSP5+ Personal Package Archive . . . . .	156
	A.2.2 Installing the LTSP5+ . . . . .	156
	A.2.3 Installing in chrootless mode . . . . .	157
	A.2.4 Installing in chroot mode . . . . .	157
A.3	Network configuration . . . . .	157
	A.3.1 Single NIC configuration . . . . .	157
	A.3.2 Dual NIC configuration . . . . .	157
	A.3.3 LTSP configuration file . . . . .	157
A.4	Thick client support . . . . .	158
	A.4.1 Configuring thick client settings . . . . .	158
A.5	LTSP 19+ Installation procedures . . . . .	159
	A.5.1 Adding the LTSP19+ Personal Package Archive . . . . .	159
	A.5.2 Installing LTSP19+ packages . . . . .	159
	A.5.3 Single NIC . . . . .	159
	A.5.4 Dual NIC . . . . .	159
A.6	Maintaining the client image . . . . .	159
	A.6.1 Chrootless mode . . . . .	159
	A.6.2 Chroot mode . . . . .	160
	A.6.3 Virtual machine . . . . .	160

A.7	Configuring iPXE . . . . .	160
A.8	Configuring NFS . . . . .	160
A.9	Generating ltsp.img . . . . .	160
<b>B</b>	<b>Appendix</b>	<b>161</b>
B.1	LTSP5+ Configuration file . . . . .	161
B.2	LTSP19+ Configuration file . . . . .	163

# List of Figures

2.1	South Africa, general household survey 2016. Source [23] . . .	7
2.2	A picture taken from India's Marathi medium school in 2008. Source [39] . . . . .	11
2.3	A client-server network. Adapted from [48] . . . . .	14
2.4	An example of Epopetes classroom management tool. Source [54]	17
2.5	The Siyakhula Living Lab ecosystem and relations. Source [58]	19
2.6	The Siyakhula Living Lab network. Source [58] . . . . .	20
3.1	A conceptual view of how PXE protocol works. Source [76] . .	27
3.2	An architectural view of the X Windows System. Source [80] .	28
3.3	An architectural view of LTSP. Adapted from [84] . . . . .	30
3.4	An architectural view of LTSP. Adapted from [84] . . . . .	31
3.5	Architectural view of the thin client architecture. Adapted from [84, 95, 96] . . . . .	35
3.6	Architectural view of the thick client architecture. Adapted from [84, 95, 96] . . . . .	37
3.7	A single NIC configuration . . . . .	39
3.8	A dual NIC configuration . . . . .	40
3.9	An example of two network configuration files for a dual NIC network. . . . .	41
3.10	An illustration of a type 2 hypervisor Source [104] . . . . .	43
3.11	Xubuntu Software Center [106] . . . . .	46
4.1	An example of the Glances application running on the server .	52
4.2	An LTSP computing infrastructure running four clients . . . .	53
4.3	A 30 seconds LAN bandwidth benchmark . . . . .	54
4.4	A graphical representation of the server's CPU and memory usage . . . . .	55
4.5	A graphical representation of the server's bandwidth consump- tion . . . . .	56
4.6	A 30 seconds LAN bandwidth benchmarking . . . . .	59

4.7	A graphical representation of the server's CPU and RAM usage . . . . .	60
4.8	A graphical representation of the server's bandwidth consumption . . . . .	60
4.9	An LTSP hybrid computing infrastructure . . . . .	62
4.10	Epoptes showing a hybrid LTSP computing infrastructure . .	63
4.11	A 30 seconds LAN benchmark: Hybrid clients . . . . .	65
4.12	A performance comparison of a thin vs thick vs hybrid client network . . . . .	66
4.13	An example of routing packets from thick clients to the Internet	68
4.14	A black screen showing a PCI bus error Source [116] . . . . .	71
5.1	A view of the Ntsika Secondary School LTSP thick clients . . .	75
5.2	Epoptes showing thick clients usage . . . . .	78
5.3	Ntsika secondary School Server Usage . . . . .	79
5.4	network benchmark . . . . .	80
5.5	A diagram showing how the server performed in the lab versus the Ntsika Secondary School . . . . .	81
5.6	SuperMicro vs Dell server benchmark. Source [114] . . . . .	82
6.1	The Siyakhula Living lab network. Source [58] . . . . .	90
6.2	Ngwane Junior Secondary school computer lab . . . . .	91
6.3	Unscrewing the server. . . . .	97
6.4	Pull the cover until it is free. . . . .	97
6.5	Lift the cover up and place it aside . . . . .	98
6.6	Apply pressure on the end clips in an outwards direction . .	98
6.7	Gently lift the RAM module by holding the ends . . . . .	99
6.8	Use a piece of clean cloth that is not too dry or wet and gently clean the RAM module. . . . .	99
6.9	Resitting the RAM module. . . . .	100
6.10	Mpumbe Junior Secondary School Computer lab . . . . .	101
7.1	A comparison of server CPU usage for LTSP5+ and LTSP 19+ .	107
7.2	SuperMicro vs Dell server benchmark. Source [114] . . . . .	108
7.3	A comparison of server's RAM usage for LTSP5+ and LTSP 19+	109
7.4	A comparison of packets transmitted to clients by the server for LTSP5+ and LTSP 19+ . . . . .	110
7.5	A comparison of LAN bandwidth usage to support clients for the server for LTSP5+ and LTSP 19+ . . . . .	110

7.6	Five Intel clients on a single NIC configuration. . . . .	111
7.7	Intel NUC vs Mecer clients. . . . .	113
8.1	A GUI-based PinguyBuilder. Source [126] . . . . .	118
8.2	CubeGet a portable package manager. Source[106] . . . . .	120
8.3	An example of LTSP-manager tool managing user credentials	122
8.4	An example of an educator assisting a student via Epointes . .	124
8.5	An architectural view of how the NUT tools communicates with the UPS. . . . .	128
8.6	A diagram showing the UPS metrics . . . . .	129
8.7	An example of the UPS configuration file. . . . .	130
8.8	An example of a script to shutdown the server . . . . .	132
8.9	RAID 1 mirror of drive 1 and drive 2 . . . . .	133
8.10	Logwatch postfix configuration menu . . . . .	134
8.11	An LTSP administrator using TeamViewer to update the LTSP server in Ntsika school remotely . . . . .	136
A.1	An example of a dual NIC configuration. . . . .	158

## List of Tables

3.1	Comparison of a single NIC and dual NIC . . . . .	41
3.2	TuxMath installation modes comparison . . . . .	44
3.3	A comparison of a virtual machine and chroot modes . . . . .	45
4.1	SuperMicro server hardware specifications . . . . .	49
4.2	Mecer Mini PC hardware specifications . . . . .	49
4.3	Proline Nanoware hardware specifications . . . . .	50
4.4	HP Client hardware specifications . . . . .	50
4.5	Thin-client computing server's results . . . . .	53
4.6	Thick-client computing server's results . . . . .	57
4.7	Thick clients 'local' computing results . . . . .	58
4.8	Server hybrid client computing results . . . . .	63
4.9	Hybrid clients local resources usage . . . . .	64
4.10	Comparison of a thin versus thick LTSP computing infrastructure . . . . .	70
5.1	Dell Server hardware specifications . . . . .	76
5.2	Mecer Mini PC hardware specifications . . . . .	76
6.1	SuperMicro Server hardware specifications . . . . .	93
6.2	Proline Nanoware hardware specifications . . . . .	93
6.3	Fixing the Proline Nanoware clients . . . . .	95
6.4	Dell Server hardware specifications . . . . .	102
6.5	HP client hardware specifications . . . . .	102
7.1	Dell Server hardware specifications . . . . .	105
7.2	Mecer Mini PC hardware specifications . . . . .	105
7.3	Proline Nanoware hardware specifications . . . . .	106
7.4	HP client hardware specifications . . . . .	106
7.5	Server thick clients computing . . . . .	107
7.6	Intel NUC J4005 Mini PC . . . . .	112
7.7	Server thick clients results . . . . .	113



# List of Abbreviations

<b>BOOTP</b>	<b>BOOT</b> strap Protocol
<b>CMOS</b>	Complementary Metal <b>O</b> xide Semiconductor
<b>DHCP</b>	Dynamic Host Configuration Protocol
<b>GNU</b>	<b>GNU</b> Not <b>U</b> nix
<b>ICT4D</b>	Information Communicaton <b>4</b> (for) <b>D</b> evelopment
<b>IP</b>	Internet Protocol
<b>(i)PXE</b>	Preboot <b>eX</b> ecution Environment
<b>LAN</b>	Local Area Network
<b>LTS</b>	Long Term Support
<b>LTSP</b>	Linux Terminal Server Project
<b>NAT</b>	Network Address Translation
<b>NFS</b>	Network File System
<b>NIC</b>	Network Interface Card
<b>OS</b>	Operating System
<b>RAID</b>	Redundant Array Independent Disk
<b>Rx</b>	Recieve
<b>SLL</b>	Siyakhula Living Lab
<b>Tx</b>	Transmit
<b>UPS</b>	Uninterrupted Power Supply
<b>X11</b>	X Window System

# 1 Introduction

This chapter serves to familiarise the reader with this research study by exploring the background, context, aim and methodology of this research. This chapter concludes with an outline of the structure of this thesis.

## 1.1 Background

Poverty alleviation is one of the top priorities the South African government seeks to accomplish. Poverty might be in part due to a lack of functional e-infrastructure [1, 2, 3]. The World Bank states that the knowledge about ICTs (Information and Communication Technology) leverage economic growth and the well-being of a society [4]. Therefore ICTs may assist the South African government alleviate poverty, probably because information and knowledge are vital to any development. Thus, ICTs are essential tools to ensure that information and knowledge reach poor communities, helping close the digital divide. The digital divide is the gap between communities included and communities excluded in the global information world [5]. Most of the poor population reside in rural or peri-urban areas. Poor communities often lack or have limited access to functional e-infrastructure. One of the reasons why most ICT projects do not last is due to a lack of adequate maintenance costs [6]. It is important to minimise both deployment and maintenance costs when possible.

Schools are active learning and convenient locations central to the community, and easier to deploy computing infrastructure. Most schools have a computer lab or at least have the means to convert a classroom into a computer lab. The Siyakhula Living Lab (as discussed in the next section) is one example of a blueprint that assists poor communities to participate in the global information economy.

### 1.1.1 Siyakhula Living Lab

The Siyakhula Living Lab (SLL) is a joint initiative between Rhodes and Fort Hare universities that focuses on the co-creation of ICT solutions to bring sustainable and meaningful e-infrastructure to poor communities in South Africa [7]. The SLL project (as discussed in Section 2.5) has progressed under the Telkom South Africa's Centres of Excellence hosted in the departments of Computer Science in both Rhodes and Fort Hare universities. Currently, each school has several terminals that rely on a server machine to do computing, using the Linux Terminal Server Project (discussed in the next section). This computing infrastructure, together with the associated network infrastructure, allows Internet access to the Dwesa-Mbhashe community and supports learning in schools.

### 1.1.2 Linux Terminal Server project

The Linux Terminal Server Project (LTSP) is a GNU/Linux add-on application that allows a computer to function as a server for computer clients [8]. LTSP is a component of the SLL blueprint and the main focus of this research. Computer clients can be thin or thick clients. A thin client is typically a low-powered computer that performs most of its computing on another computer [9]. A thick client performs most of its computing locally [10]. Section 3.3 discusses how thick and thin clients work together with LTSP on a technical level.

### 1.1.3 Thin and thick client history

The term 'thin client' surfaced in 1993 when Oracle wanted to move away from Microsoft personal computers [11]. Managing terminals separately was becoming complex, and thin clients offered cost-effective and centralised computing. Thin clients share a single installation of the operating system and its applications, reducing license costs and maintenance. Gordon Moore predicted that the number of transistors per silicon chip doubles every two years [12, 13]. Indeed technology has progressed almost as predicted by Moore's law, and as a result, computer and software licenses are dropping. Most GNU/Linux applications are free of cost. Thick-client computing has become more attractive. Currently, graphical applications are more popular than command-line applications, and they demand more system resources.

Thick clients offer much better performance that is not achievable with thin clients.

*"As technology advances, businesses and institutions will keep rotating between thin and thick client computing as a way to achieve better computing." ( Alfredo Terzoli, personal communication, February 18, 2021)*

Currently, thick-client computing is better than thin-client computing, but as technology advances, thin-client computing will most likely emerge as a better way of computing. Probably, for this reason, the new version of LTSP has limited the support of thin clients (as discussed in Section 3.2.2)

## 1.2 Research context and aim

Existing literature has revealed that most ICT for Development (ICT4D) projects fail to reach their potential, and in some cases, might cause harm to the communities they intended to develop [14, 15]. Currently, the SLL blueprint is implemented in the Eastern Cape's Dwesa-Mbhashe schools as well as local Makhanda schools [16, 17]. The SLL has been running for over a decade now, and just like all ICT4D projects, the SLL might eventually fail if it continues unchecked. Continuously updating and upgrading the SLL blueprint will help deliver sustainable ICT4D solutions.

The aim was to review the SLL blueprint primarily focusing on the LTSP computing infrastructure to confirm if the blueprint is still viable. The researcher attempted to achieve the task by attempting to answer the following question:

1. *Is LTSP still a good tool for the SLL blueprint?*
2. *What improvements should be made to have a more sustainable deployment?*

## 1.3 Goals and Objectives

This section explores the goals and objectives of the research to help the reader understand the rationale of the research study. The research aim as described in the previous section is achieved by the following goals:

1. *Assessment of the related work*
2. *LTSP computing infrastructure laboratory experiments.*

3. *Deployments of LTSP computing infrastructure in schools.*

## 1.4 Methodology

The research will review the SLL blueprint and answer the two research questions by performing lab deployments followed by real-life deployments. Lab deployments are smaller deployments that enable the researcher to test the LTSP computing infrastructure thoroughly, which is not easy to do in real-life deployments. The lab experiments tested server resource usages such as CPU, RAM and LAN bandwidth consumption. The lab experiments also compared the two versions of LTSP as well as the user experience of the LTSP clients. Lab experiments, together with real-life deployments and re-deployments, assisted the researcher to review the SLL blueprint and answer the research questions.

## 1.5 Thesis organisation

This section outlines the document structure as well as the document's features and conventions.

- **Chapter 1:** *Introduction*  
This chapter introduces the research study and outlines the research problem, goals and objectives, and the methodology.
- **Chapter 2:** *Related work*  
This chapter outlines the related work and similar initiatives taken to improve computing infrastructure.
- **Chapter 3:** *Theoretical background*  
This chapter describes the theory behind the main concepts of an LTSP computing infrastructure.
- **Chapter ??:** *Laboratory experiments: LTSP5+*  
This chapter performs a series of lab experiments to determine ways to best setup an LTSP5+ computing infrastructure on a smaller scale.
- **Chapter 5:** *LTSP5+ deployment: Ntsika Secondary School*  
This chapter outlines the deployment in the Ntsika school and complements chapter 4 lab experiments.

- **Chapter 6:** *LTSP5+ deployment: Dwesa-Mbhashe schools*  
The chapter outlines the deployment and maintenance work packages in the Ngwane and Mpume schools.
- **Chapter 7:** *Laboratory experiments: LTSP19+*  
This chapter performs a series of lab experiments using the LTSP19+ performed on a smaller scale.
- **Chapter 8:** *Making deployment and maintenance easier*  
This chapter explores ways to optimise and maintain an LTSP computing infrastructure so that it is more sustainable and easy to deploy and maintain.
- **Chapter 9:** *Conclusion and future work*  
This chapter concludes the research study and proposes the areas of future research that could not be included in the document.

### 1.5.1 Thesis features and conventions

The symbol '+' on naming the versions of LTSP was used to classify versions variations e.g. LTSP19+ includes are versions under LTSP19 e.g. LTSP19.04. To remove ambiguity the word *client*: refers to a computer terminal that is either thin client or thick client 'fat client' computers. This research study uses italics to emphasise CLI commands, file configurations parameters etc.

## 2 Related Work

This chapter introduces the reader to the background literature, research context and related work. The chapter's aim is to build a foundation that sets the scene for the research study. The contents of this chapter include ICT for Development, an overview of client-server networks, the Greek school switch to open source and the Siyakhula Living Lab.

The reader needs to note that Linux Terminal Server Project (LTSP), which is the main technology of this thesis, is a Github project. Github provides the hosting for software development and version control [18, 19]. Most of the resources come from blogs and Github community issues. Also, before 2019 LTSP project was poorly documented. This therefore creates a literature gap that limits the scope of this chapter.

### 2.1 Background

Poverty might be linked in part due to a lack of functioning e-infrastructure. Information and knowledge are key strategic resources for social and economic development [1, 2, 3]. In the modern-day, ICTs are essential to ensure that information and knowledge reach poor communities. ICT tools are the major enablers of sharing information and knowledge. In recent years educational and business institutions have increased their use of e-infrastructure to support their day to day activities. Currently, there is an increase in demand for remote learning due to the global COVID pandemic. Telkom, a South African technology company, reported a rise in the demand for connectivity and therefore emphasised the need for a reliable e-infrastructure to support consumers [20]. Remote activities demand an e-infrastructure, although the more impoverished people and communities can barely afford it. ICT tools help bridge the gap between those that can afford and the marginalised groups in access to global knowledge. The gap in access to ICT between the well-endowed and the marginalised groups is called the digital divide [21, 22]. According to [21], the term digital divide lacks conceptual elaboration

and definition. In this research study, the term digital divide is being restricted to a lack of access to computing infrastructure. Figure 2.1 shows the differences in access to basic between the metros, urban and rural areas.

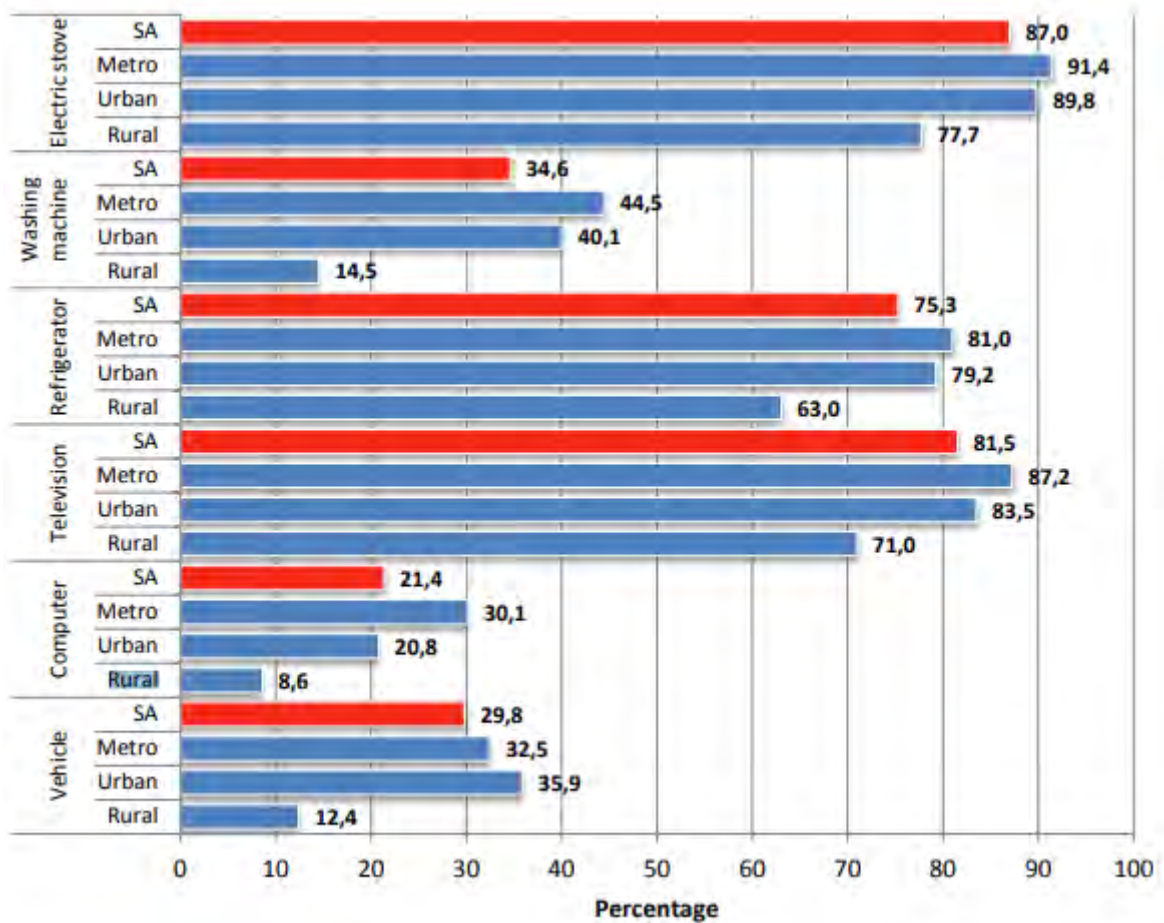


FIGURE 2.1: South Africa, general household survey 2016.  
Source [23]



Technology solutions leverage the integration of disadvantaged communities into the global knowledge community [17]. ICT tools play an important role in achieving sustainable development in communities [1, 2]. Successful ICT tools facilitate the reduction of poverty in developing countries by introducing new revenue streams. South Africa is one of Africa's developing countries that prioritises poverty eradication on the continent. As of 2019, South Africa recorded a gross expenditure on research and development of 8.5%, which marked a significant increase from 2017 [24]. Research and development are vital to identifying crucial areas that need investment.

This research study is focusing on poor communities or groups in both rural and peri-urban areas. A peri-urban (rurban) is a population that exists on the boundaries of both urban and rural areas [25, 26]. A marginalised community is a community that is excluded due to uneven political, economic, and cultural dimensions [27]. Marginalised communities do not participate in mainstream activities as compared to urban communities. Historically, the rural population has less developed e-infrastructure compared to the urban populations. The difference is due to the profit-oriented institutions and government targeting the urban population as their major consumers. This resulted in the stunted development of e-infrastructure in rural communities. Most of South Africa's poor population resides in rural areas. According to [28], South Africa's rural population accounts for 34% of the nation's population. Even though most poor communities are in rural areas, they also exist in parts of peri-urban and urban areas of different sizes. The urban population is more developed and demands more e-infrastructure compared to the peri-urban and rural population. Technology companies are, therefore, motivated to invest more in the urban areas where they benefit most. This is one of the challenges of ICT4D as described in Section 2.7.

## 2.2 ICT for Development

Information Communication Technology for Development (ICT4D) is the use of ICT technologies to leverage development in communities. According to [29], ICT4D is the use of technologies to aid the poor and marginalised people in developing communities. ICT4D makes use of ICTs to foster development by bridging the digital divide between population groups. ICT4D is mainly an applied field of study since it targets development. This subsection looks at ICT4D strategic partnerships, ICT in education and ICTs in communities.

### 2.2.1 ICT4D strategic partnerships

Since information and knowledge are key to development, it is vital for every community to participate in the global information economy. Stakeholders in strategic partnerships are independent but share risks and benefits of collaborative actions[30]. Partnerships can be amongst international agencies, donors, private sectors and civil societies. The following subsections discuss some of the existing partnerships.

#### United Nations 2030 Agenda

In 2015, the United Nations organisation set out a 2030 agenda of ending poverty in all its forms by supporting sustainable development[14]. Included in *Goal 2a* of the 17 goals under the 2030 Agenda is to improve rural infrastructure and technology development. *Goal 4b* emphasises the expansion of scholarships awarded to both developed and underdeveloped countries towards higher education, including ICT [14]. *Goal 9c* highlighted the need for under-developed countries to increase access to ICTs and affordable access to the Internet [14]. Furthermore, Goal 5b highlights the importance of ICTs to empower women in developing and underdeveloped countries.

The United Nations Conference on Trade and Development (UNCTAD) supports developing countries to fairly and more effectively benefit from the global economy [31]. UNCTAD assists developing countries by partnering with private organisations, donors, civil society, governments and international agencies [31]. UNCTAD has inclusive ICT strategies that put policies in place to improve research, trend analysis and development growth. UNCTAD offers the opportunity to share experiences and building consensus around key issues. UNCTAD also assists governments with sustainable competitive legal frameworks. In summary, the UN 2030 Agenda indicates the importance of ICTs to foster development.

#### South Africa Connect 2030 Agenda

ICTs have become part of strategies and policies for developing countries like South Africa. In 2013, the South African government introduced the 2030 Agenda South Africa Connect (SA Connect), which aims at connecting 100% of the population by 2030 [16, 32]. The initiative aims to put South Africa in a competitive position with other developed countries to participate in the global information economy. The World Economic Forum (WEF) reported

that South Africa has an overall improvement in ranking from 56 to 49 in terms of Global Competitive Index (GCI) [33]. The improved ranking results from increased usage of ICT tools such as increased bandwidth speeds and improved innovations. According to [34], the 2016 State of the Nation Address (SONA) recorded no real progress of the SA Connect's broadband implementation, but the government promised to fast track the process. The SONA reported highlights some of the challenges of ICT projects.

Strategies and partnerships on different levels are vital for successful ICT4D projects. Trust is necessary for ICT4D stakeholders to participate in the global information economy. For thriving and more inclusive ICTs, ICT infrastructures, institutional and legal frameworks, and the right policies should exist.

### **2.2.2 ICTs in Education**

ICT4D projects in communities are best delivered in schools since schools are cost-effective centres for ICT4D projects. Schools are now adopting an approach that is student-centred to improve the quality of education. ICTs are increasingly becoming unavoidable for use in educational institutions. According to [35], the quality of education relies on the ICTs in place. Education contributes to the well-being of a community through building an informed and knowledgeable society. Integrating ICTs with education motivates learners to participate more in educational materials and to think creatively. Typical examples of learning materials are educational games that train the brain to solve problems and enhance recall. ICTs in education can modify the roles of a teacher to a facilitator who guides students whilst using ICTs [35]. Students can participate in online learning at more flexible times. Computing infrastructure such as a computer laboratory can teach learners valuable skills that can open up career paths.

#### **One Laptop Per Child**

One Laptop Per Child (OLPC) was a non-profit initiative launched by Massachusetts Institute of Technology (MIT) in 2005 to provide quality education through distributing one laptop per child [36, 37]. Several companies helped fund the OLPC such as AMD, eBay, Google and Red Hat. The OLPC believed that by giving a laptop to a child promotes digital literacy, thereby changing the world [37]. OLPC had the potential to change the world and

supplied about 5 million units in the first year [38]. OLPC targeted disadvantaged children including those that live in areas with no electricity and harsh conditions. Figure 2.2 shows a diagram of the OLPC XO laptop.



FIGURE 2.2: A picture taken from India's Marathi medium school in 2008. Source [39]

The XO laptops were low-cost, low-powered and had a high resistance to damage from dust, water and could be used outdoors [40, 41]. XO laptops were durable and ideal for supporting sustainable development in poor communities. Below is a quote from Nicholas Negroponte the founder of the OLPC project.

*“We can drop it again and again and again and it will always work,”* Nicholas Negroponte [36]

Each XO laptop came with a solar panel for charging the battery, and the battery lasted for more than 12 hours navigating an E-book [40]. Solar panels reduced the dependence from electricity service providers. The XO laptops promoted collaboration, and children would interact and help each other via a camera or text [41]. The XO laptops had an extended WiFi range that allowed them to have broader coverage. The XO laptops allowed schools to communicate with each other schools via a mesh network [42]. XO laptops used the Internet for collaboration, and when the Internet was down, the XO laptops would create a mesh network to interact within a classroom. A mesh network is a network infrastructure with nodes (in our case, XO laptops) that are connected dynamically and non-hierarchically and can communicate with each other [43]. According to [37], the XO laptops implemented the following principles:

- *Children get to keep the laptops*  
When children keep their laptops, they get to spend more time interacting with the laptop and can collaborate with other children at flexible hours.
- *Focus on early education 6 - 12 years*  
OLPC believed that giving children access to laptops at an early age will improve their ICT skills.
- *Large deployment of laptops*  
OLPC believed in distributing or selling the XO laptops in large quantities rather than in small quantities.
- *Children should have a connection to the Internet*  
OLPC believed Internet connectivity would improve the experience and learning of the children.
- *Free and open source software*  
Free and open source will reduce the total cost of deploying the XO

laptops, especially in schools.

There is several OLPC literature on how the project started as a success and eventually failed. The target population criticised XO laptops for being tailored for the United States of America (USA), poorly maintained, lacked proper training for teachers, and being expensive [36, 39]. Probably XO laptops could not suit everyone's needs because of diversity in cultures and languages. Interface designs such as icons may portray something different in another setting such as culture and nation. Stakeholders of OLPC did not adequately train teachers to use the XO laptops, and as a result, the XO laptops did not help the children learn. Even though manufacturers cut costs resulting in noninterchangeable components, a price tag of \$100 was expensive [36]. Shortages of water and basic learning infrastructure convinced the target population (poor population) that XO laptops were of a lesser priority [36]. The OLPC struggled to raise funds even after the initiative 'Give One Get One', and also, in the year 2008, world economies were going into recession [38]. OLPC struggled to get hardware contracts, and some buyers withdrew their orders, e.g., the Libyan government cancelled an order for 1.2 million XO laptops and ordered PCs instead [44].

However, as of 2021 some OLPC projects are still active. The government of Rwanda has managed to distribute 27500 XO laptops to 1500 schools [44]. Also descendent of OLPC, Sugar Labs launched a live USB to support old computers during a LinuxTag conference in Berlin [45].

### **Chrome-books**

Chromebooks are Google laptops that run on Google's Chrome OS (Operating System). Google is changing the classroom by supplying low-cost laptops, and free cloud-hosted applications [46]. Chromebooks do not support desktop applications, but all computing happens within the Chrome browser. The Chrome OS comes bundled with free Google applications such as Google docs and Google sheets. Chrome-books run on their OS called Google OS. Chromebooks are ideal for low-cost laptops and free cloud-based applications. However, Chromebooks are limited to Google applications and rely on an active Internet connection. Since most poor communities lack or have limited access to the Internet, the Chromebook is an ideal solution.

## 2.3 An overview of client-server networks

A client-server network (model) where a network infrastructure whereby a central server serves clients with services and resources [47]. Figure 2.3 shows an example of a client-server network.

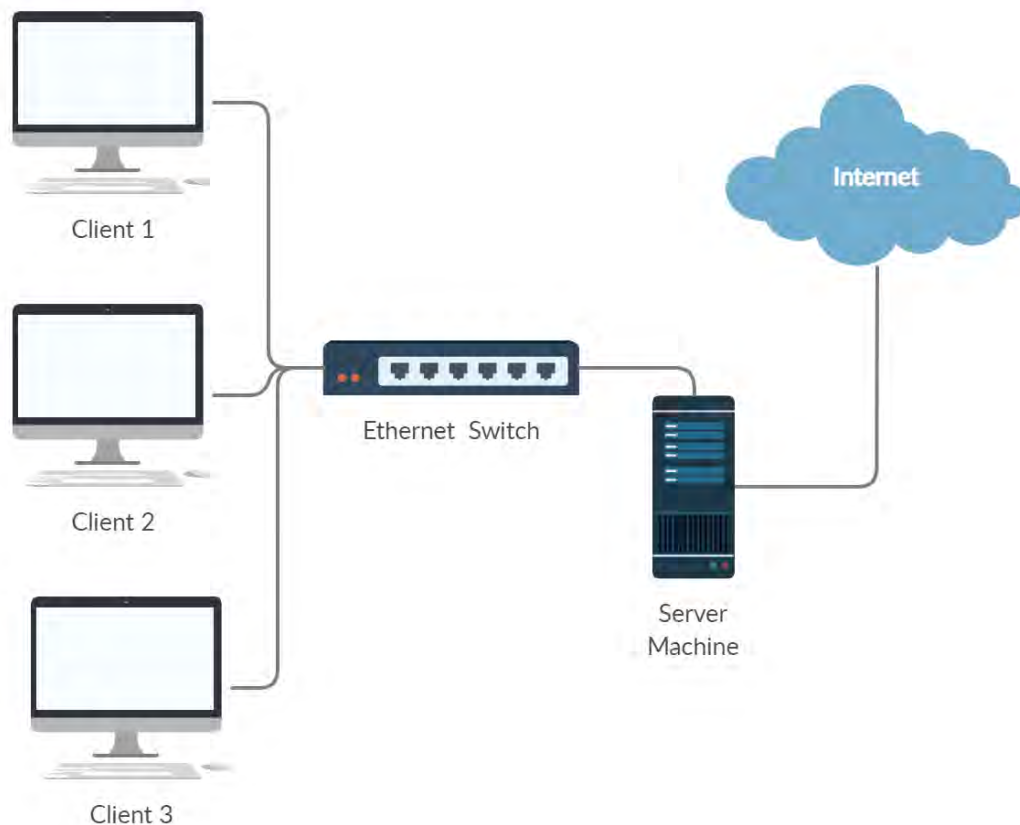


FIGURE 2.3: A client-server network. Adapted from [48]

Figure 2.3 shows a client-server model whereby computer clients are connected to the server machine via an Ethernet switch. Alternatively, the clients can also use a wireless medium. The following subsection covers low-cost technologies such as BeTwin ES, OpenThinClient, Diskless Remote Boot in Linux, and LTSP.

### 2.3.1 BeTwin ES

BeTwinES is a Windows OS-based application that allows nine computers to share an installation [49]. BeTwinES is not ideal for large deployments that involve more than nine computers. Large deployments will increase the costs of an extra server(s) and software licenses and possibly complicate the computing infrastructure's maintenance. However, nine BeTwin computer clients are cheaper than standard nine Windows OS-based computers.

### 2.3.2 OpenThinClient

OpenThinClient (written as a noun without spaces) is an open-source Linux OS-based and written in Java programming language application that boots client computers over a network [50]. A thin client is a low-powered computer that performs most of its computing on the server machine (see Section 3.5). OpenThinClient supports a large number of thin clients and effectively manage them via a graphical interface. Since OpenThinClient is Java-based, it runs on any platform that supports Java applications.

### 2.3.3 Diskless Remote Boot in Linux

Diskless Remote Boot in Linux (DRBL) is an open-source and free software that distributes a single GNU/Linux operating system installation across clients computers [51]. Diskless are computer clients without a physical disk for local storage. DRBL provides centralised computing for all clients connected to the DRBL network. DRBL uses PXE, NFS (see Section 3.1 and 3.1.2) and Network Information Service (NIS) to serve the client computers. A NIS allows a set of client computers to share common configuration files [52].

### 2.3.4 Linux Terminal Server Project

The Linux Terminal Server Project (LTSP) is an add-on application that helps terminals boot from a single server. This research study is going to focus on



LTSP as the primary technology. The following section discusses the deployment of LTSP in Greek schools, and Chapter 3 explains the LTSP at a more technical level.

## 2.4 The Greek school- switch to LTSP

This section reports the deployments in the Greek school by Alkis Georgopoulos. Georgopoulos is a teacher and one of the active contributors to LTSP, and the creator of the current version of LTSP (see Section 3.2.2). In 2007 the Greek school switched to open source software from using Windows applications [53]. The open-source applications were Debian-based distributions such as Edubuntu were bundled with the LTSP tool. The initial deployments sent out Edubuntu servers to 20 schools, and interestingly only one deployment from Georgopoulos school was successful [53]. In 2008 an IT support team documented the setting up of LTSP [53]. The LTSP Manager application initially sch-scripts simplified LTSP management. In 2009, The LTSP team added integrated Greek windows applications and created repositories to support more than 100 educational software [53]. In 2010 the LTSP team added the support of thick clients that allowed the LTSP server to be less expensive and powerful. In 2011, the team abandoned the iTALC classroom monitoring tool because of bugs and moved to Eoptes. Eoptes is an open-source application for the classroom. It allows remote command execution, message sending, screen locking etc [54]. Figure 2.4 shows a diagram of Eoptes managing a classroom.

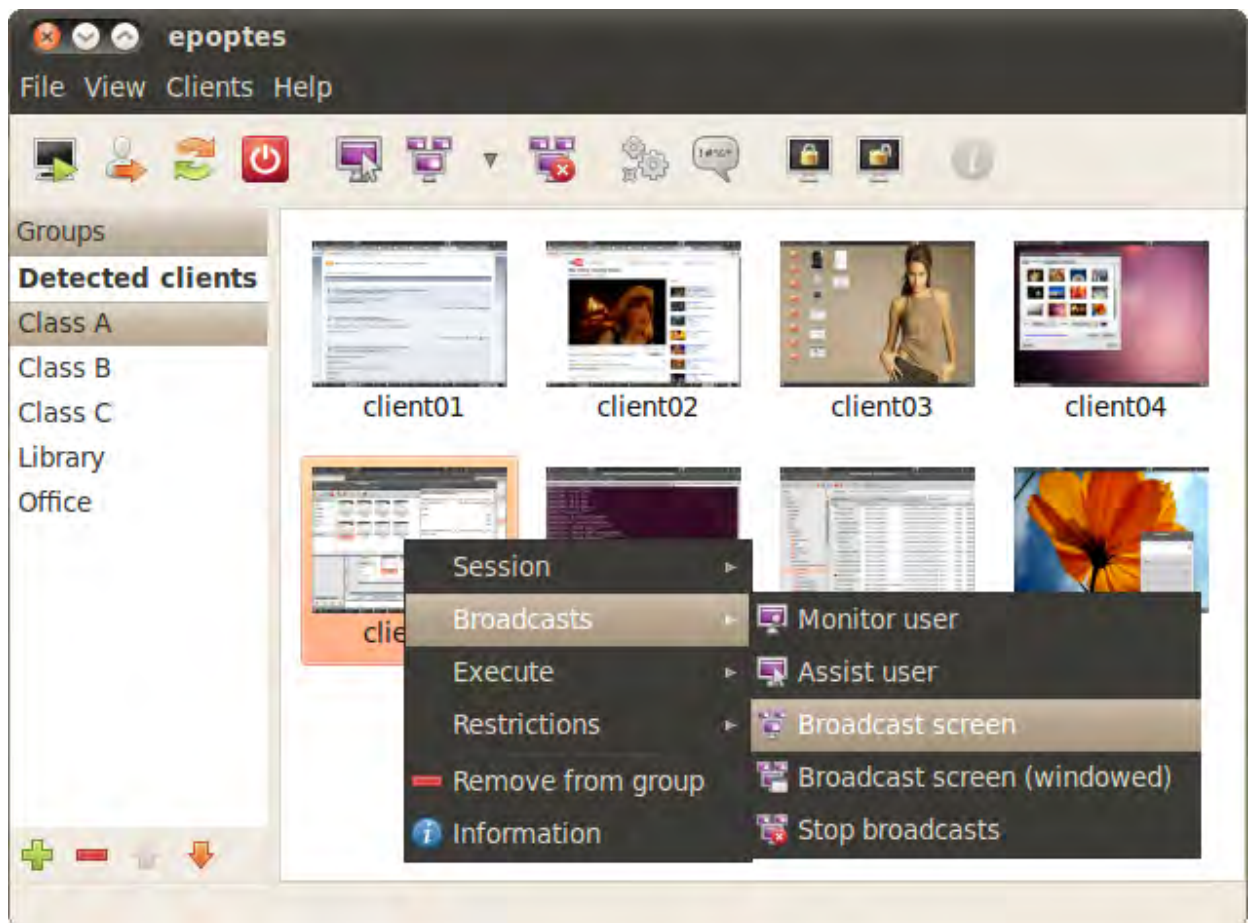


FIGURE 2.4: An example of Epoptes classroom management tool. Source [54]

Epoptes is a graphical application that is user-friendly and can conveniently manage users without using command-line tools. As seen later in this document, Epoptes will be widely used together with LTSP.

In 2012 the LTSP team introduced the classic LTSP and the LTSP-pnp methods for maintaining the client images. The classic LTSP (chroot mode) maintained the client image separate from the server installation. Chroot create a separate computing space that is independent from the host machine computing space (see Section 3.5). The LTSP-pnp (chrootless) allows the client computers to use the same installation image as the server (see Section 3.5). The LTSP team managed to distribute Live CDs for primary, secondary and high schools [53]. Also, the LTSP stakeholders managed to conduct seminars in 50 cities and to educate almost 2000 IT teachers [53]. The training and the active involvement of the Greek government contributed to the success of LTSP deployments in the Greek schools after their pilot project failed.

## 2.5 The Siyakhula Living Lab

A living lab is a user-driven open innovation in the real world and is guided by the government, business, citizens and academia partnerships [55, 56]. Stakeholder partnerships as highlighted on Section 2.2.1 is vital to achieving a sustainable product.

The Siyakhula Living Lab (SLL) was initiated in 2005 to focus on connecting South Africa's poor communities to the global information world by introducing sustainable and meaningful e-infrastructure [16, 7, 57]. The SLL blueprint focuses on poor communities located in rural and peri-urban areas. SLL uses the Research, Development and Innovation (RDI) methodology to co-create solutions and end-user involvement [58]. The RDI methodology uses research to develop a solution. The original and main site of SLL is located near the Dwesa-Cwebe wildlife reserve, and marine sanctuary under the municipality of Mbashe of the Eastern Cape province in South Africa [16]. SLL is a joint venture of the Telkom Centre of Excellence at Rhodes University and the University of Fort Hare, the rural communities of Dwesa and Nkwalini, and the government and industry partnerships [16, 17]. The SLL has developed under the Telkom South Africa's Centres of Excellence hosted in the departments of Computer Science in both universities. Figure 2.5 shows a diagram of the SLL's ecosystem and relations with components.

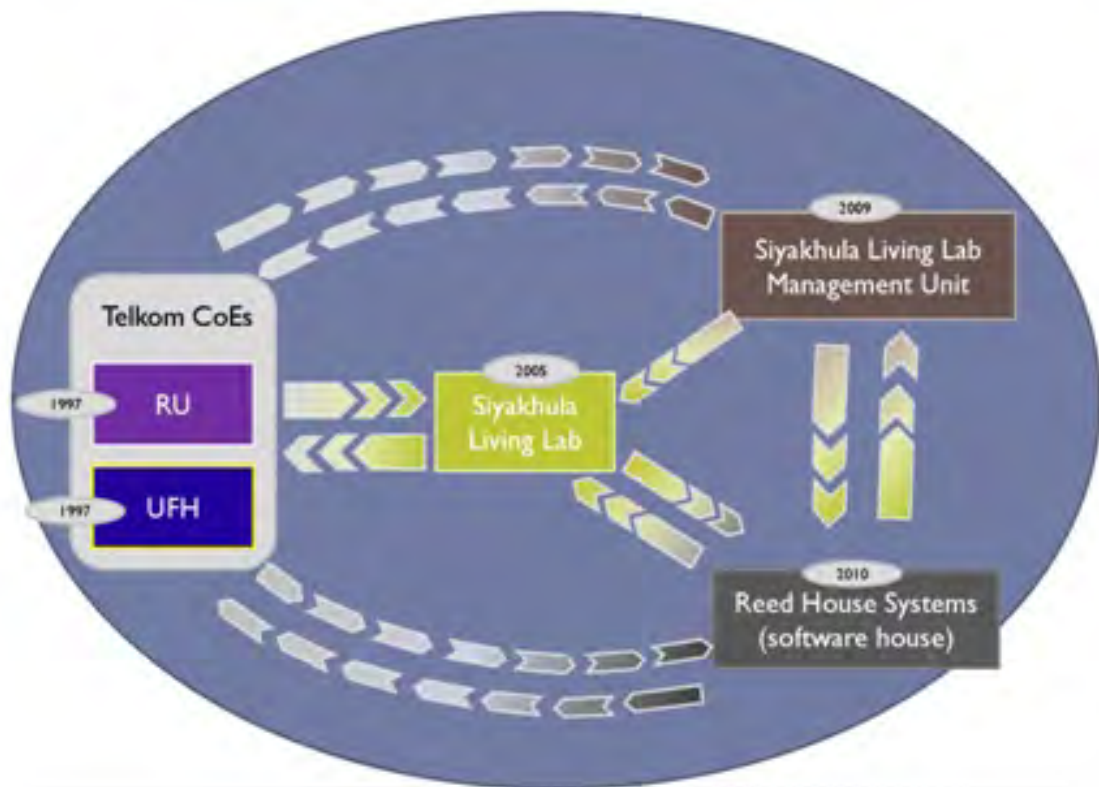


FIGURE 2.5: The Siyakhula Living Lab ecosystem and relations.  
Source [58]

The Reed House System (RHS) supports the development of software by providing development tools, and environment [58]. TeleWeaver is the main product of RHS that offers a middleware platform for services or application deployment [58]. The SLL Management Unit is responsible for overseeing the overall functions of the SLL ecosystems and relations, e.g. RHS, research, funding, external organisations. Figure 2.6 is a geographical diagram of the SLL network. The application layer design of the SLL will bring in revenue streams that will support the deployed infrastructures and improve the lives of the target community.

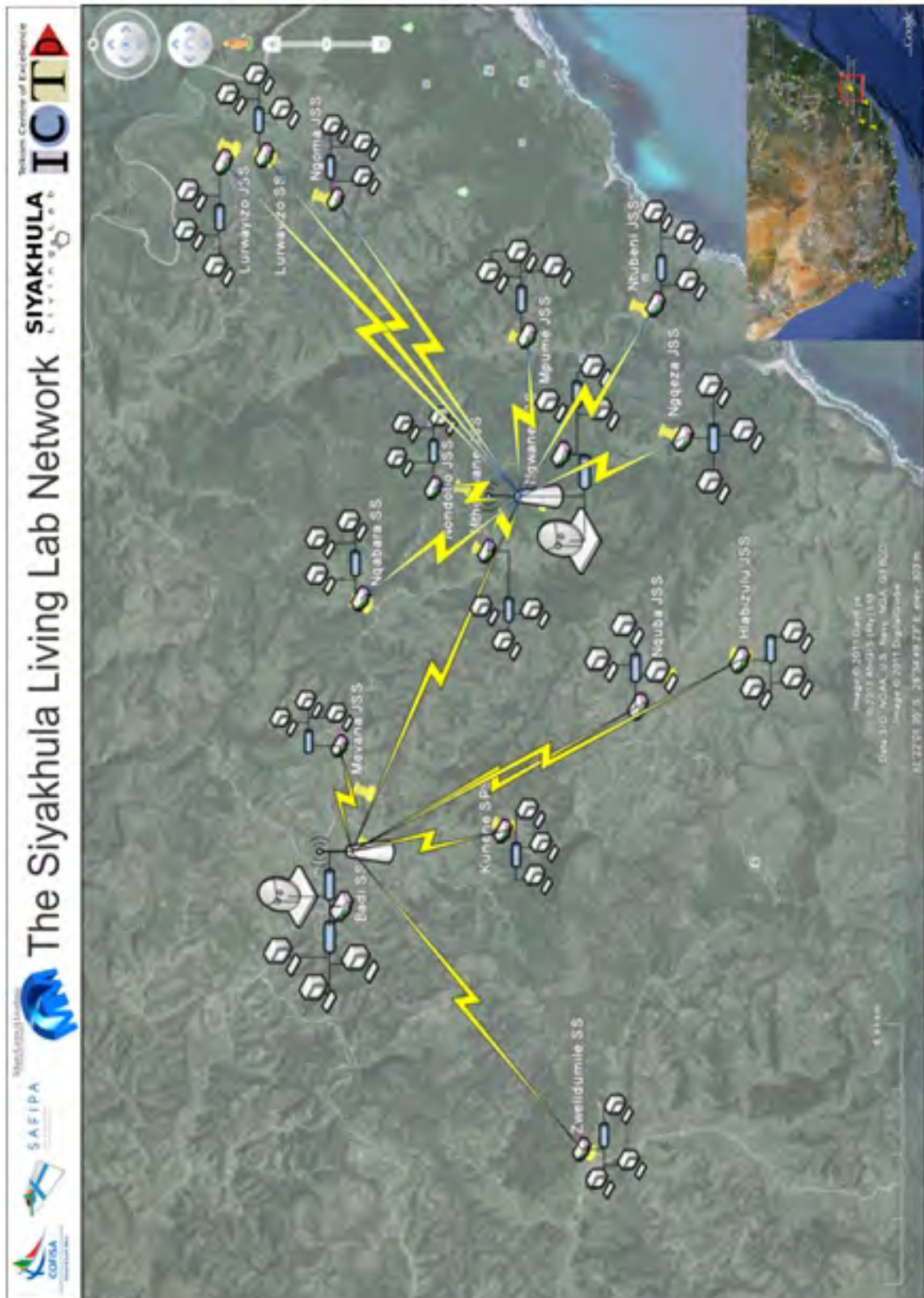


FIGURE 2.6: The Siyakhula Living Lab network. Source [58]

The SLL network infrastructure has high-speed wireless LAN (local area networks) called broadband islands that connect the SLL schools [58, 16]. The

network infrastructure caches data to minimise communication outside the broadband [16]. The SLL schools host Digital Access Nodes (DAN) that has the LTSP computing infrastructures. Each school has several terminals that rely on the LTSP server for computing. The LTSP computing infrastructure can independently function without Internet connectivity. However, in cases where the LTSP server uses one network interface card, if the network infrastructure fails (specifically DHCP service), the LTSP clients will not be able to boot (see Section 3.10). The DHCP, as discussed in Section 3.1.1 is responsible for assigning terminals network identities (IP addresses).

### 2.5.1 Siyakhula Living Lab Challenges

The existence of the SLL blueprint for over a decade proves that the initiative has been somewhat successful. However, like any ICT4D project, there are still challenges and barriers. Some of the challenges are as follows:

1. **Poor road infrastructure**

The Dwesa schools have poorly maintained road infrastructure. A poorly maintained road discourages and limits essential services, especially those that fixes a problem. Services such as repairing the lab will likely take more than necessary time to repair.

2. **Poor electricity service and e-infrastructure**

The availability of electricity is limited to some areas of the Dwesa community, such as the reserve, schools and health centres citeTechnology-Publication. The community has poor electricity service, and from experience, there are cases when the power company takes weeks to fix a faulty power line. As discussed in Section 6.2, the labs do not seem to have a readily available electrician to fix power failures. Attempts to install solar power by non-governmental organisations have failed due to theft, and vandalism [17]. The absence of electricity in some sections of Dwesa-Cwebe makes it challenging to deploy computing infrastructures. The Dwesa community also experiences inadequate network coverage and has few telephone lines, limiting the adoption of ADSL technology [17].

3. **E-skilling**

The SLL project focuses on poor communities, mainly in the rural and peri-urban areas. Most of the target population lacks e-skills, especially

IT teachers who should be teaching the skills. The shortage of IT teachers is out of our hands and depends on the Department of Education. Initiatives such as digital literacy courses should help equip the community with ICT skills.

#### 4. Funding and partners of ICT4D projects

Most ICT4D projects are non-profit-making projects, but they demand funding to meet expenses such as deploying and maintenance costs. The SLL implements a multidisciplinary approach with partners in different industries such as the Rhodes and Fort Hare universities, Telkom Centre of Excellence (CoE), SGT Solutions and Khula Technologies (local Makhanda company). A multidisciplinary approach faces the challenge of meeting the needs and agendas of individual stakeholders [59]. Different SLL partners may want to do things differently and may conflict with other partners. For this reason, governments should be the long term primary funders because profit-oriented organisations can easily lose interest in funding a non-profit initiative. An ideal ICT4D project should be able to generate its revenues. The SLL application layer is one example of generating revenues streams via advertisements.

#### 5. Research and frameworks modelling

Several frameworks for rural areas exist and currently the SLL uses the Research development Methodology (RDI). The existence of different rural frameworks and methodologies can be a challenge when making decisions. The whole project might fail because of an incompatible framework and methods that do not support the SLL environment.

ICT4D projects face challenges before, during and after implementation and SLL possibly faces more challenges than mentioned in this subsection some of the challenges like poor road infrastructure can not be solved by SLL team. However, efforts can be made to encourage e-skilling and advocate for quality service supply.

## 2.6 Importance of ICT4D

Everyone should participate in the evolving information economy. Information and knowledge are critical strategic resources for both economic and social development. ICT tools play an essential role to bring information and knowledge to the users. Since poverty is closely related to non-functioning e-infrastructure, it is important to introduce ICTs in poor communities. ICT4D

projects will help develop poor communities by potentially supporting e-learning, new careers opportunities, creating new businesses, and supporting existing businesses to grow. The poor communities economy primarily relies on farming, and farmers can use ICTs to advertise and make online payments.

## 2.7 Challenges of ICT4D

ICT projects tend to result in total or partial failure due to the incomplete interpretation of the problem and incorrect metrics for evaluating a solution [60]. The biggest challenge comes when the ICTs are not as transformative as perceived by stakeholders [61]. A successful ICT4D has to be sustainable and produce quality outcomes. An ICT4D project, as seen in previous sections, should be affordable whilst meeting its targets. The following are some of the challenges of ICT4D:

- **Poor maintenance**

An easy to deploy and maintain ICT4D project will potentially allow faster deployments and extend the lifetime of the deployment. ICT4D projects should be easy to deploy to encourage participation, especially from people with less technical skills. A well-maintained deployment will extend the lifetime of the project. Chapter 8 focuses on how the LTSP can be easily deployed and maintained. In most cases, maintenance relies on the availability of funding that usually comes from strategic partnerships, as discussed in the next subsection.

- **Lack of strategic partnerships**

ICT4D projects should have active partners that fund the deployments and maintenance of the deployment. As discussed earlier, governments should have an active presence in ICT4D projects. The active presence of the Rwandan government has probably seen the OLPC project roll-out XO laptops for almost a decade. Another example is the active presence of the Greek government has perhaps seen a successful switch of the Greek schools to LTSP.

- **Lack of e-skilling**

Twinomurinzi [61] argues that whilst ICTs are being integrated as part of national strategies by developing countries, there is still a lack of e-skilling. Inadequate e-skilling results in unused e-infrastructure that will eventually become obsolete. The stakeholders in ICT4D projects



should put strategies that enable the community to learn ICT skills. One example is deploying the ICT4D projects in communities centres that already support learning, such as schools.

- **Frameworks and policies**

Stakeholders should adequately research the right framework that suits the target population to increase the chances of success. One way is to partner with educational institutions such as universities. The government should implement the right policies and strategies that promote ICT4D projects. Right policies will also encourage other institutions to participate in the ICT4D projects.

Ideal ICT4D projects should have self revenue-generating strategies in place. A revenue-generating ICT4D project will potentially fix some, if not most, of the challenges of ICT4D challenges.

## **2.8 Conclusion**

This chapter reviews the ICT4D projects that are similar or related to the LTSP computing environment. The aim was to familiarise the reader with the existing ICT4D projects and establish how the LTSP computing infrastructure can contribute to sustainable computing infrastructure.

## 3 Theoretical background

This chapter takes a more theoretical approach to discuss the architectures and protocols involved in the LTSP-based computing. The aim is to give the reader a theoretical background necessary to solidify the understanding of chapters that follow.

### 3.1 An overview of some of the protocols

This section discusses a brief overview of the protocols that are involved in an LTSP-based computing environment. It is important to note that LTSP is not a protocol but an GNU/Linux add-on application.

#### 3.1.1 Dynamic Host Configuration Protocol

The DHCP (Dynamic Host Control Protocol) is a client-server service protocol that enables a server computer to automatically assign network configuration settings to an Internet Protocol (IP) host [62, 63]. The DHCP manages the network configurations such as network identities (IP) of the connected nodes (devices), default gateway, netmask and DNS addresses. A DNS (Domain Name System) is similar to a phonebook, and it translates website names to IP addresses that other connected nodes understand [64, 65]. For example, *'https://www.siyakhulall.org* translates to 34.254.68.101. The DHCP also automatically configures time and printer server. The DHCP automates and centralises the management of network configuration settings, and this makes it possible to support a large network [66]. A DHCP server is required by the PXE (see Section 3.1) to allow the server to know the source or destination of packets.

#### 3.1.2 Network File System

The NFS (Network File System) is a distributed file protocol that allows users to access files and directories on the same network domain remotely [67, 68].

The NFS is based on the Remote Procedure Control (RPC) protocol. The RPC protocol allows computer clients to work with the remote procedure as if there are local [69, 70]. The NFS centralises storage and reduces data redundancy, and NFS clients can access files from the NFS server the same way the client access local files.

### 3.1.3 TFTP

TFTP (Trivial File Transfer Protocol) is a simplified File Transfer Protocol (FTP) defined in *RFC2347* used to transfer small files over the User Datagram Protocol (UDP) [71]. A TFTP server is required to avail booting images on a network [72]. TFTP uses less memory than FTP, making it ideal for booting images on a network. Unlike FTP, TFTP does not require TCP (Transmission Control Protocol) authentication to send files and provides access to all the files in a TFTP folder. Less memory and lack of authentication and encryption make it ideal for sending large files via Ethernet, e.g. XDMCP traffic. However, TFTP can pose a security risk over the public network due to a lack of security.

### 3.1.4 PXE

PXE (Preboot Execution Environment) is a network protocol that facilitates the booting of a terminal from the server before receiving the operating system booting image [73]. The PXE is an open-source protocol that utilises the computer's Network Interface Card (NIC) to make it function as a booting device by allowing a server to send files [74]. Most existing hardware support PXE, and in our case, the terminals should be able to network boot. The following subsection discusses iPXE, a new version of PXE.

#### iPXE

iPXE is a network boot firmware licensed under the GNU General Public Licence (GNU GPL) that provides a full implementation of PXE with additional features [75]. Other features are to boot via HTTP, InfiniBand network, Fibre Channel SAN and Wireless network etc. PXE protocol incorporates the TFTP and DHCP protocols. Figure 3.1 is a conceptual view of how the protocols work.

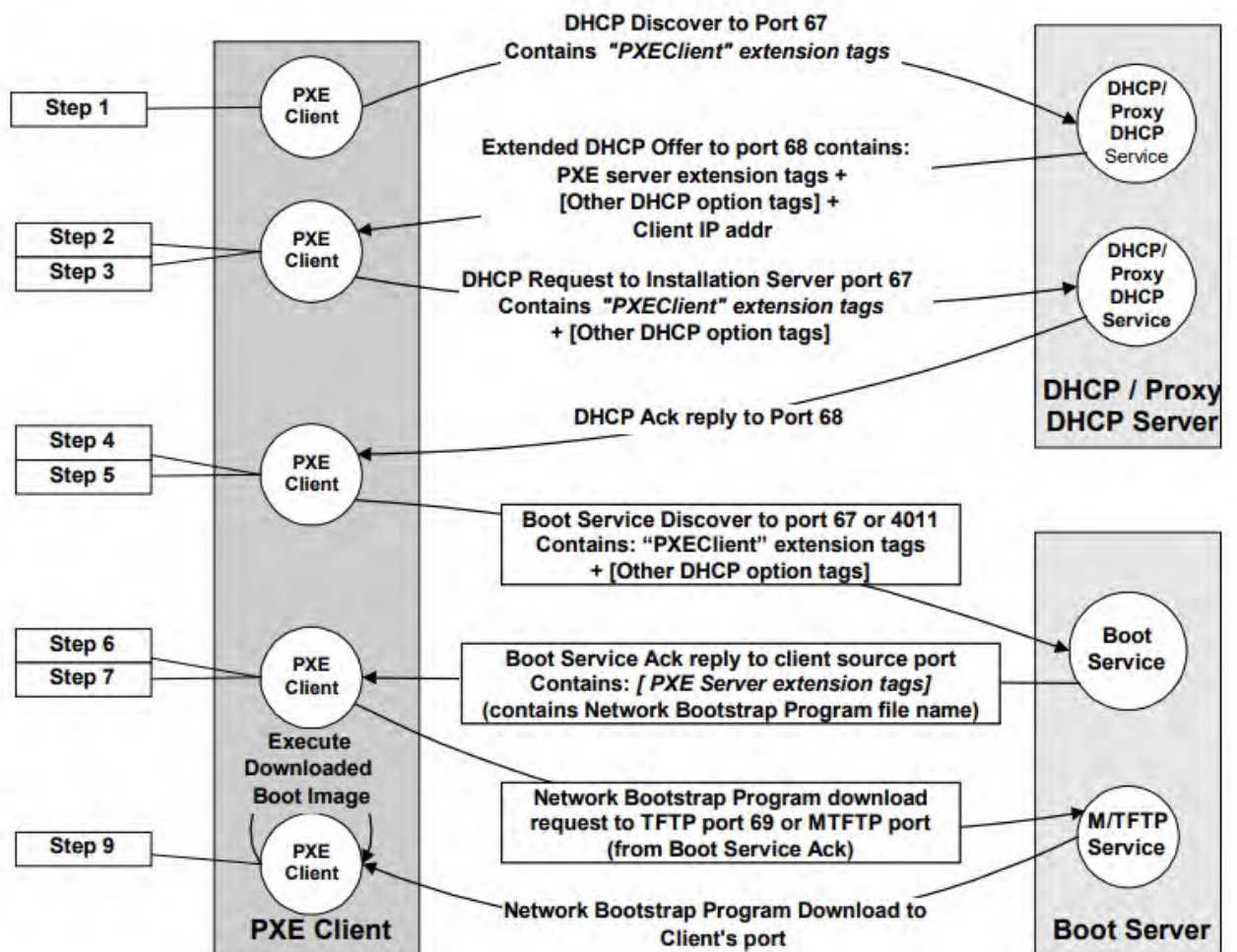


FIGURE 3.1: A conceptual view of how PXE protocol works.  
Source [76]

The PXE client (terminal NIC) sends out a broadcast request to the DHCP server. The DHCP server returns the IP address of the client along with the address of the TFTP server and the location of boot files on the TFTP server etc. The PXE client then boots from the PXE server. The LTSP server also supports Etherboot/gPXE, an older version of PXE to support NICs without PXE.

### 3.1.5 X Windows system core protocols

The X.Org Foundation provides the implementation of the X Window System, and the X Windows System (X11) follows the client-server model [77,

78]. The exchange of messages between the X-client and the X-server is guided by the X protocol (X11). The X11 was developed in the early 1980s by MIT (Massachusetts Institute of Technology) to provide a graphics rendering engine for Unix systems [79]. Since the terminals serve input such as keyboard and mouse, the X-server resides on the terminals. A remote application (X-client) sends output requests to the X-server, and multiple X-servers can access a single X-client application. The X Windows System allows only the X-server to access the hardware. Figure 3.2 shows the architecture of the X Windows System.

### X Windows architecture

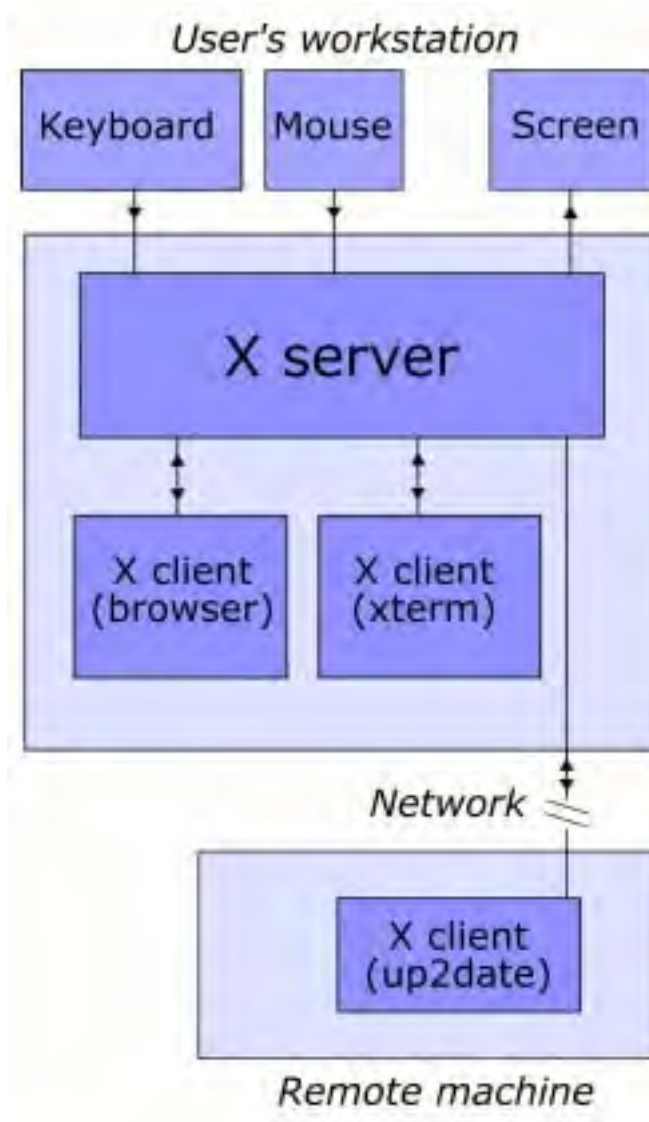


FIGURE 3.2: An architectural view of the X Windows System.  
Source [80]

## XDMCP

X Display Manager Control Protocol (XDMCP) is a remote desktop protocol that allows a client using X11 protocol to interact with another server [81]. The XDMCP allows the graphical login of the LTSP clients. The GNU/Linux display manager has to implement the graphical interface to allow a remote client to compute on a local machine. The lack of compression in the XDMCP promotes faster rendering of the video output. However, the XDMCP pose a security risk since it does not implement encryption [81].

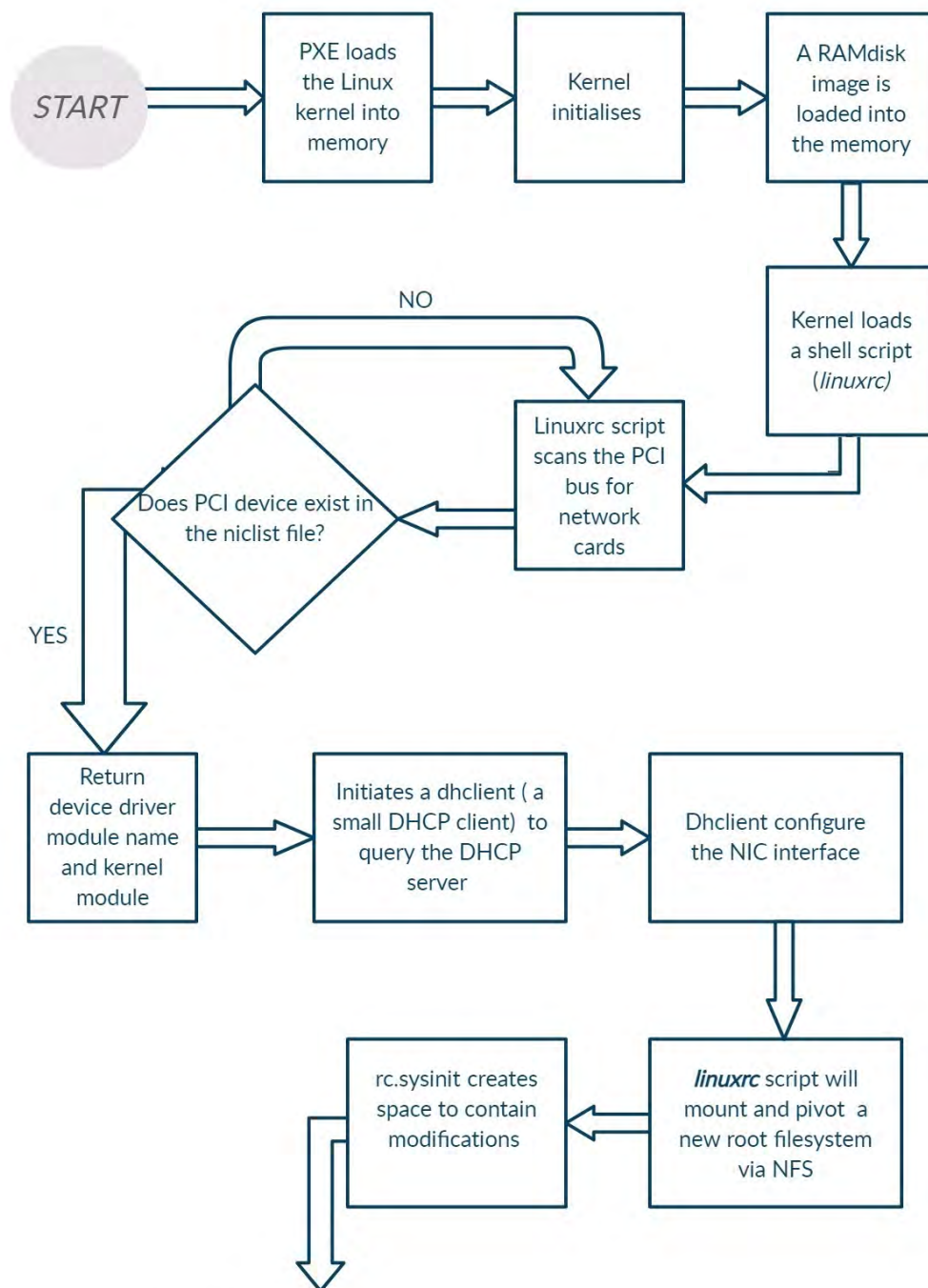
The following section discusses how the protocols described in this section work within an LTSP computing infrastructure.

## 3.2 Linux Terminal Server Project

The Linux Terminal Project (LTSP) is a GNU/Linux software package that enables computer clients to boot from another machine [82]. Most GNU/Linux OS distributions such as Ubuntu, Edubuntu, Arch-Linux and Open SUSE supports LTSP. The following subsection shows the LTSP booting process.

### 3.2.1 LTSP booting sequence

The LTSP computing infrastructure is currently limited to Ethernet cable because of high bandwidth consumption (as seen in the following chapters). Therefore, LTSP clients should have PXE network boot enabled to be able to receive files. Protocols such as TFTP and NFS (in some cases NBD) facilitates the transfer of files within an LTSP computing infrastructure. Network Block Device (NBD) is a protocol used to transfer files across computers in a network [83]. Figure 3.3 and 3.4 shows how the LTSP processes from loading the kernel into memory until the LTSP clients are ready to login.



The diagram continues  
on the next page.

FIGURE 3.3: An architectural view of LTSP. Adapted from [84]

The diagram continues from this point .

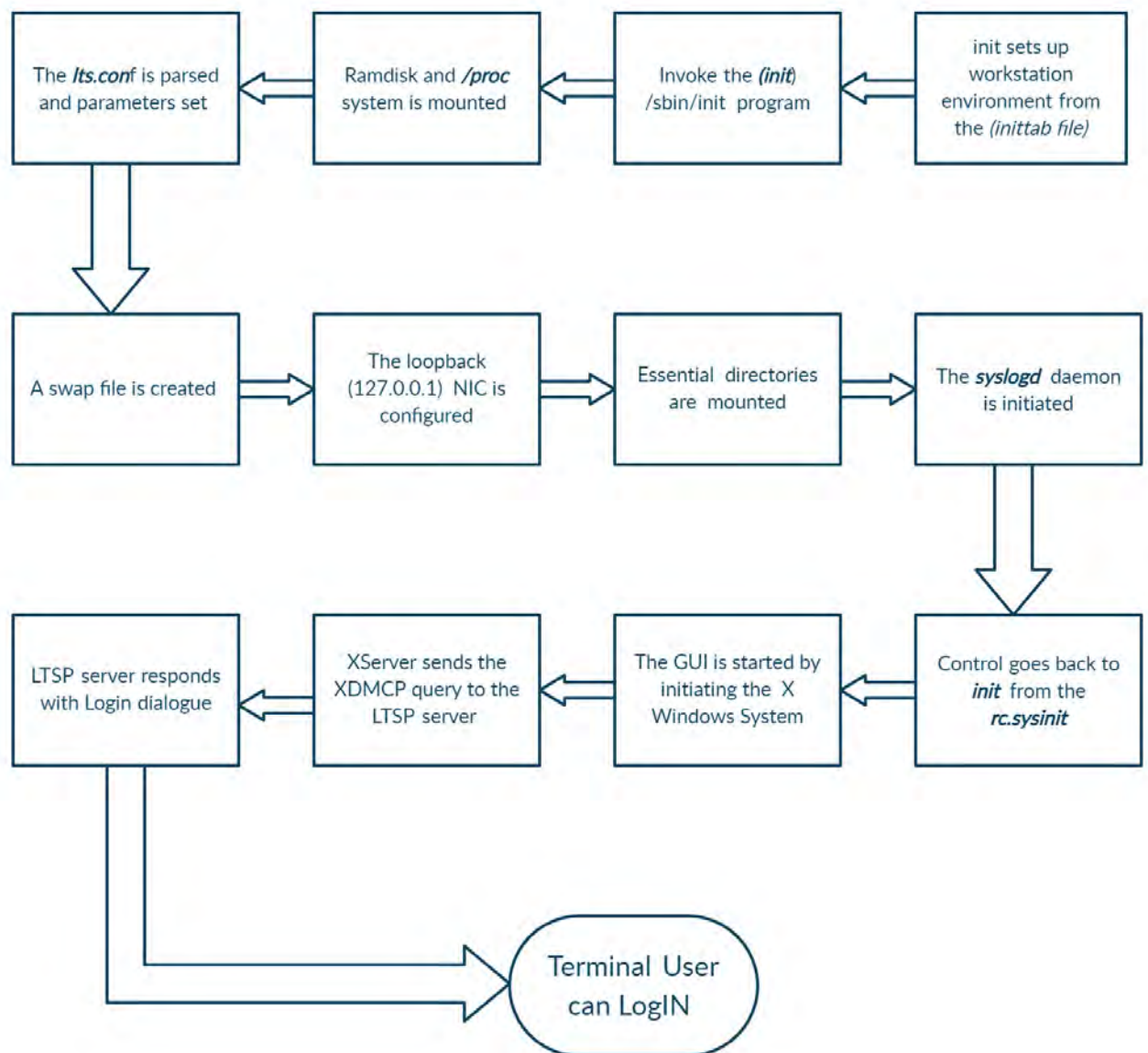


FIGURE 3.4: An architectural view of LTSP. Adapted from [84]

### 3.2.2 The LTSP19+ versus the old LTSP5+

The LTSP19+ follows LTSP 5.2 that was released in 2010 [85, 86]. In 2019 the LTSP5 project was redesigned from scratch to support new technologies [82]. The developers of LTSP19 have tested Ubuntu Mate, but any GNU/Linux distribution should support LTSP.



### **Support of new technologies**

LTSP19+ supports new technologies such as systemd, Wayland, UEFI, and other updated desktop environments. The UEFI (LTSP19+ also added an option of maintaining the client's image within a virtualisation player (see Section 3.5).

### **Reduced thin client support**

LTSP19+ reduced the support of thin clients to remote desktops protocols such as VNC, x2go and xfreerdp [87, 88] probably because modern clients are powerful enough to function as thick clients.

### **Virtualisation player to maintain client's image**

LTSP19+ added an option of maintaining the client's image using a virtualisation player. Virtualisation player isolates the client image from the host OS, and modifications do not affect the server's filesystem (see Section 3.5).

### **Easy NAT configuration**

Unlike the previous LTSP versions, the LTSP19+ simplified how NAT is configured (see Appendix B). A Network Address Translation (NAT) allows a private IP address to access an external network [89]. Section 4.5 discusses how an LTSP5+ server is modified to allow NAT.

### **Introduction of iPXE**

LTSP 19+ introduced the iPXE, an open-source network boot firmware that implements the PXE with additional features [75]. Other iPXE features include network boot from HTTP, wireless network and InfiniBand network etc. The iPXE boot menu, unlike the older version PXE, offers the user an option to choose a specific boot image. The administrator can avail several LTSP client's images, and LTSP users can pick from the PXE displayed options.

### **Reduced file sizes**

LTSP5+ used a meta-package 'ltsp-server-standalone' to download dependencies to simplify the installation process. The meta-package will then get server and client packages, e.g. ltsp-server, ldm-server, ltspfs, ltsp-client-core, ldm etc. These LTSP5 dependencies packages had a total of approximately 2000KB in size. The new LTSP19+ uses the 'ltsp' and 'ltsp-binaries'

that are 254KB and 1328KB in size, totalling 1582KB. LTSP19+ excludes previous services such as 'idminfod', a service that was responsible for outputting Login sessions and available locales [90]. LTSP19+ has a lesser memory footprint on the server than LTSP5+.

LTSP 19+ is also compatible with older versions of operating systems and can run in parallel with old LTSP5. The new version LTSP19+ offers an active support forum on GitHub [90].

### 3.2.3 Benefits of LTSP

LTSP has several benefits, and this subsection lists some of the LTSP benefits.

#### 1. Reduced hardware and software licenses costs

Typically old computer hardware has inferior specifications, and in most cases, they become obsolete. Since an LTSP server performs most of the computing for thin clients, LTSP can easily convert low-powered old computers to thin clients. Converting old computer hardware to thin clients reduces cost. Also, stakeholders can deploy cheaper and low-powered thin clients. Unlike Windows alternatives like BwtwinES, most GNU/Linux applications are free.

#### 2. Centralised maintenance

LTSP can maintain a single client image to serve multiple LAN clients, which reduce administration overheads. If one of the LTSP clients fails, the client is replaced without losing user data. LTSP users can also login from any of the terminals.

### 3.2.4 Drawbacks of LTSP

If the server fails the whole LTSP computing infrastructure will be downy. The LTSP transmits large packets of data that are currently only ideal if transmitted via an Ethernet eliminating the possibility of wireless transmission.

## 3.3 LTSP computer clients

This research study focuses on thin and thick clients and even though both thick and thin clients.

### 3.3.1 Thin Client

A thin client is a computer that performs most of its computing on another machine. Thin client computers use desktop virtualisation, whereby applications and resources reside on the server and are made available on request. Virtual Desktop Infrastructure (VDI) creates an end user's desktop environment by isolating the desktop environment and applications from the end user's underlying physical hardware [91, 92]. The thin client architecture uses the session virtualisation whereby multiple clients simultaneously use the same LTSP client image. All applications and data reside on the server, and a constant connection to the LTSP server is required. Since computation is done on another computer, a thin client can be low-powered and diskless. Modern thin-client systems provide OS features whilst providing centralised computing with reduced administration costs [93].

A user interacting with a thin client has the same impression as if they are interacting with a complete desktop computer. A normal desktop computer has a monitor, mouse, keyboard, and processor that includes a local hard disk. The main difference is that thin clients' inputs, and output video streams are transmitted between the client and the server. A thin client function by network booting from a server and does not require a hard drive for local storage as all data is stored on the server. Only RAM is needed to allow the loading of BIOS firmware into memory. The BIOS firmware will then allow the thin client to boot from the LTSP server. An LTSP computing infrastructure requires a PXE or iPXE, and it recommends using Ethernet as a transmission medium. Ethernet reduces data packet latency and simplifies how clients receive a DHCP offer. The LTSP server provides the sharing of computing resources among thin clients. Sharing server resources reduces administration costs. A single software package is shared among thin clients and can share the same license. The LTSP server efficiently utilizes its resources by dynamically allocating the available system resource, e.g. CPU, RAM. Since most computing is performed on the server machine, a powerful processor supported with adequate storage and RAM is required. The number of clients an LTSP server can support depends on how powerful the server machine is, including the switch link speed. For example, LTSP-Wiki Debian recommends a CPU Mark of 5000 for ten thin clients :

$$\text{CPU Score} = 2000 + (300 * \text{thin clients}) \text{ Source [94].}$$

Figure 3.5 shows how the thin client architecture works on an architectural level.

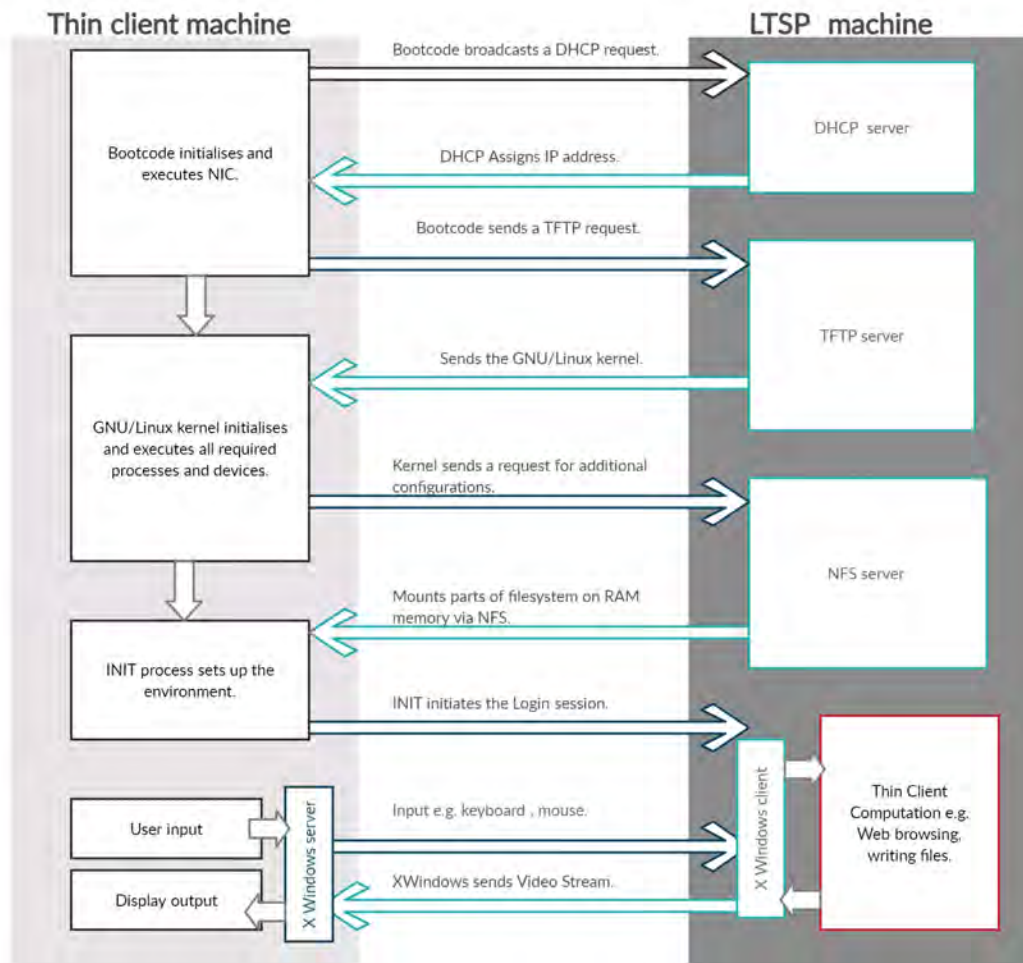


FIGURE 3.5: Architectural view of the thin client architecture.  
Adapted from [84, 95, 96]

Bootcode is a set of instructions that executes first when a client is turned on. Bootcode instructions require memory to execute instructions. RAM is used as a temporary location to load bootcode, and a thin client cannot network boot without the availability of RAM. After the bootcode is loaded into the RAM, it scans for available NICs. The bootcode initialises and executes the NICs by broadcasting a DHCP request. If the DHCP server is running, it assigns an IP address to the thin client. Each IP address is linked to the thin client's physical address (MAC address). The bootcode then sends a TFTP request to download the Linux kernel. After the kernel is successfully downloaded to the thin client RAM, the bootcode will pass operations to the kernel.

The kernel initialises and executes by initialising the entire system and peripheral devices. The bootcode does not pass information to the kernel, the kernel has to request the server for information it needs, and this includes network parameters [84]. The Linux kernel requests the NFS server for the filesystem. The NFS server replies by sending parts of the filesystem that are mounted in RAM.

The GNU/Linux kernel passes control to the init process. The init process initialises the environment needed for an OS to run. The init process sends an XDMCP request to the XWindows client to initiate a Login session. Here the user can log in using the credentials provided with an impression of using a typical desktop computer. All application and data changes computation happen on the LTSP machine. The thin client needs to maintain a constant connection with the LTSP machine. User input such as mouse and keyboard are transmitted over Ethernet as frames to the XWindows client. The input is processed and send back to the XWindows server as a video stream and is displayed on the user's monitor. As the user interacts with the thin client, the cycle from the XWindows server-client is repeated. Since thin clients do not compute locally, the server should be in constant communication with the client.

### **Thin clients and green technology**

IT can play an important role to support green IT in the product life cycle of manufacturing, operation and recycling hardware [50]. Thin clients contribute to green technology by being low-powered and manufactured using less resources compared to normal desktop computers.

### 3.3.2 Thick client

A thick client computer, sometimes called a fat client or a rich client, performs most of its computing locally and does not entirely rely on the server machine. Figure 3.6 is a diagram highlighting how the LTSP-thick client architecture.

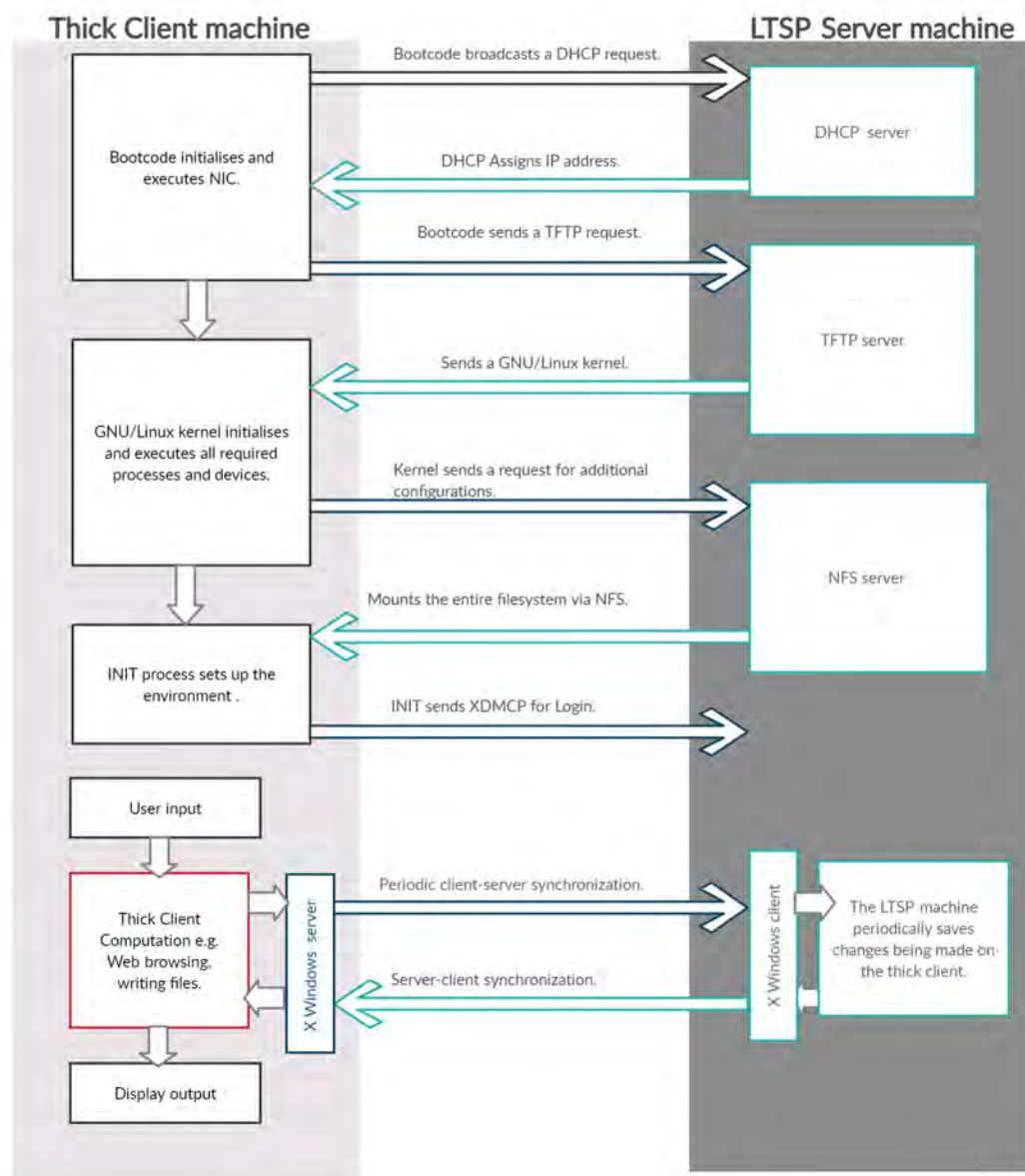


FIGURE 3.6: Architectural view of the thick client architecture.  
Adapted from [84, 95, 96]

Figure 3.6 is similar to Figure 3.5 the only difference is that thick clients perform most of their computing locally. The user's task such as web browsing and word processing is done locally without have to be sent to the server for processing. Hosting local applications locally has demands fewer resources from the LTSP machine. Thick clients connect periodically to the LTSP machine, and unlike thin clients, applications are run locally. The processes are the same, but applications are hosted locally, and there is no need to maintain a constant connection. Since the LTSP server manages the user accounts, including storing data files, the thick client periodically connects to the server to synchronize the changes.

## **3.4 Network configuration**

The network can be configured in two ways, single or dual NIC. The administrator has to choose if the LTSP server should route clients to the Internet or should an external DHCP be responsible for routing packets to the LTSP clients.

### **3.4.1 Single NIC configuration**

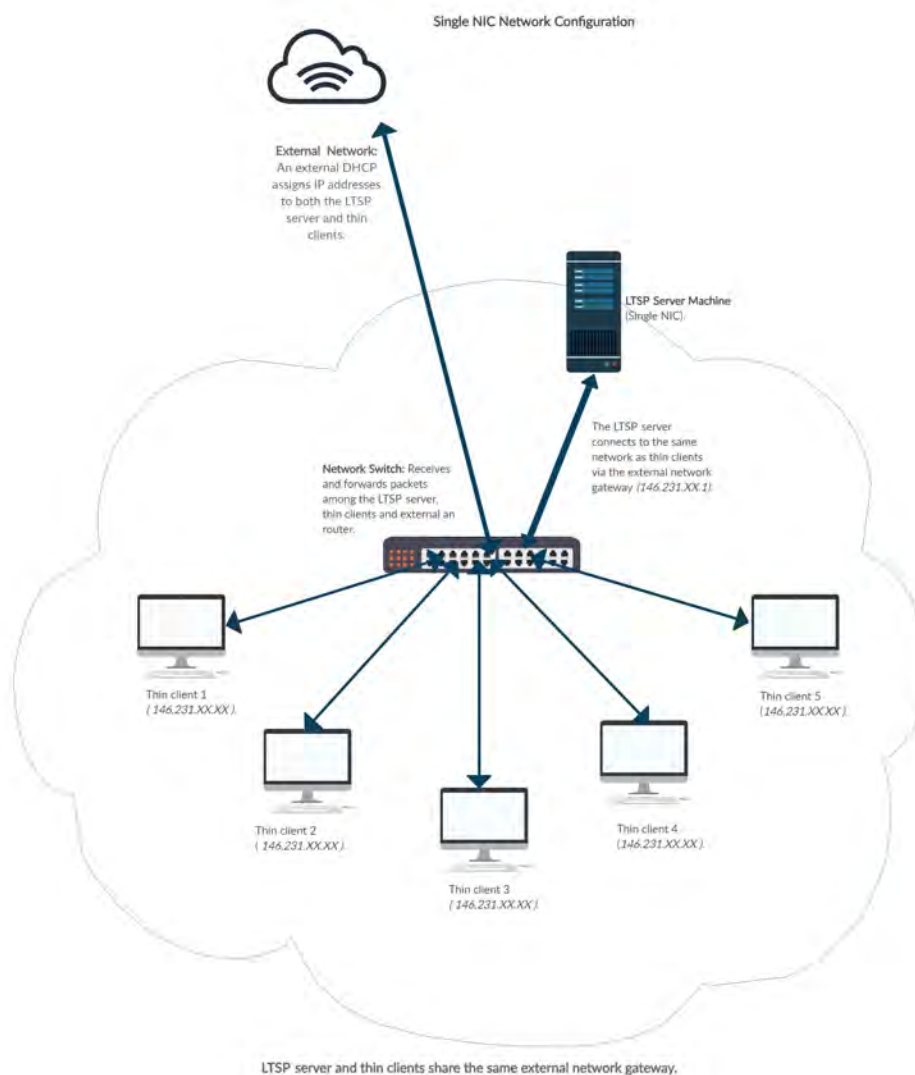


FIGURE 3.7: A single NIC configuration

A single NIC computing infrastructure configuration is the less complicated way to deploy the LTSP server. The network topology remains unchanged since both the LTSP server and thin clients are connected to the same switch(es) and DHCP server. An external DHCP server in this scenario can be an Internet Service Provider (ISP) router that is connected to the public Internet. The immediate drawback of this configuration is that when the external router's DHCP server is down, both the LTSP server and thin clients will not get assigned the IP addresses.

### 3.4.2 Dual NIC configuration



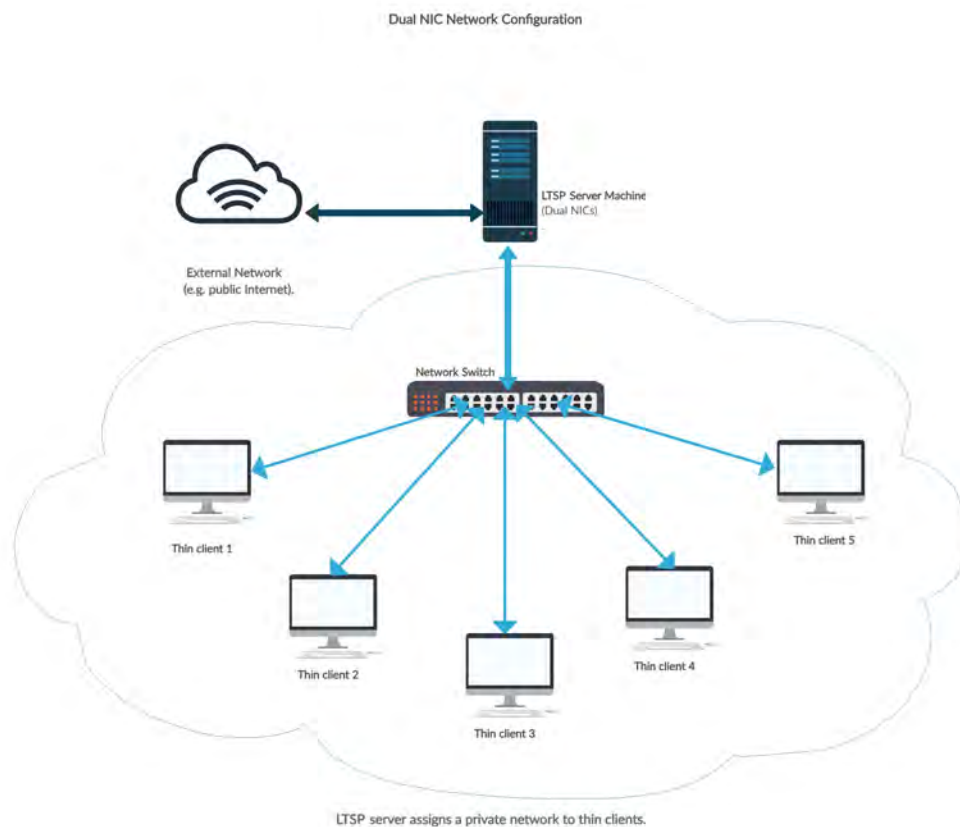


FIGURE 3.8: A dual NIC configuration

A dual NIC network configuration requires that one NIC connects to an external network and the other one connected to the LTSP clients. As shown in Figure 3.10, a dual NIC configuration does not rely on the external network DHCP to assign IP addresses. Here IP addresses are assigned by an internal lightweight DHCP server 'dnsmasq'. The dnsmasq can be modified to suit the user by editing the '/etc/dnsmasq.conf' file using an editor of choice. The dnsmasq is configured to support LTSP settings, and by default, the gateway is set to 192.168.67.1. Additional network settings are required to connect thin clients to the LTSP server and clients to an external network if required. The immediate advantage of this network topology is that the thin clients do not rely on an external DHCP but the LTSP server's internal DHCP server. The disadvantage is that it might be complicated to configure the network configurations. Figure 3.9 shows examples of a dual NIC configuration files using the Netplan and the interfaces network managers:

```

auto lo
iface lo inet loopback

auto eno1
iface eno1 inet auto

auto enp2s0
iface enp2s0 inet static
    address 192.168.67.1
    netmask 255.255.255.0
    network 192.168.67.0
    broadcast 192.168.67.255
    gateway 192.168.67.1
    dns-nameservers 192.168.67.1
    dns-search home

```

a) Interfaces configuration file

```

network:
  ethernets:
    eno1:
      addresses: []
      dhcp4: true
    enp2s0:
      addresses: [192.168.67.1/24]
      gateway4: 192.168.67.1
      dhcp4: false
  version: 2

```

b) Netplan configuration file

FIGURE 3.9: An example of two network configuration files for a dual NIC network.

### 3.4.3 A comparison a single and dual NIC configuration

Both single and dual NIC have their benefits and drawbacks, and it is up to the administrator to use what suits the deployment environment.

TABLE 3.1: Comparison of a single NIC and dual NIC

Single NIC	Dual NIC
An external DHCP server configures client's network settings.	An internal DHCP server usually lightweight to configure client's network settings e.g. dnsmasq.
No single point of failure. If the server fails the clients can continue working and accessing Internet using Live USB sticks. Note that Live USB sticks are ideal for thick clients.	Single point of failure and if the server fails the whole LTSP computing infrastructure fails.
Easier option as no additional network settings are required after configuring the dnsmasq.	More complicated to setup as more additional network settings are required after configuring the dnsmasq.
Website access is controlled by the Internet Service Provider.	The server machine can control and limit client's websites access e.g. the server can block access to non-educational or illegal sites.

## 3.5 Maintaining client image

LTSP provides and maintains the OS image (.img) file for the client either as a separate image or the same image as the server. This can be done using three methods of maintaining the client's image, namely chroot, chrootless and virtual image [97, 98].

### a) Chroot

Chroot (change root) is an operation that modifies the apparent root directory for the process, and their children [99]. A chroot is frequently used as Linux and mobile Android OS as the alternative to a virtual machine. A process or program in the modified directory 'chroot jail' cannot access files or run commands outside the modified directory. Here the administrator maintains the server OS and the client OS as separate images. All updates and changes performed on the server do not appear in the client's image.

### b) Chrootless

The chroot uses the server OS as the client image, i.e. both the server machine and clients use the same OS template. Here the administrator maintains the client image by maintaining the server OS. All updates and modifications done on the server will appear on the client image.

### c) LTSP19+: raw virtual image

The raw virtual machine mode was introduced in LTSP19+ (as discussed in Section 3.2.2). Here the client image is separately maintained within a virtualisation software, e.g., KVM, Virtual Box. A virtualisation player is software that creates an abstraction layer on top of computer hardware (host) and emulates one or more computer systems (virtual machines) by sharing the hosts computing resources, e.g. CPU, storage, RAM [100, 101]. A virtual machine (VM), alternatively called an instance, runs its operating system separately from the other virtual machines and the host. A hypervisor is a software or hardware component that separates the operating system and applications from the physical hardware [102]. Several virtualisation players are available for free use or at least free for non-commercial use. Examples are VMWare, KVM and Oracle VirtualBox virtualisation players. The independence from

the underlying hardware makes it possible to run multiple and different virtual machines (instances) on a single host computer. Virtualisation players utilise a hypervisor to emulate computer systems.

A hypervisor (virtual machine monitor) is a software, piece of hardware, or firmware that creates and run virtual machines by separating the OS and applications from the physical hardware [102, 103]. The hypervisor is responsible for managing and allocating system resources such as RAM and CPU to the virtual machines. A hypervisor allows the mobility of virtual machines since they are independent of the host machine. A user can run GNU/Linux Xubuntu on a Windows machine. A type 1 hypervisor is a layer of software installed directly on the physical host machine [100]. A type 2 hypervisor runs within the OS environment of the host machine [100]. LTSP uses a type 2 hypervisor to create a client's image virtual machines.

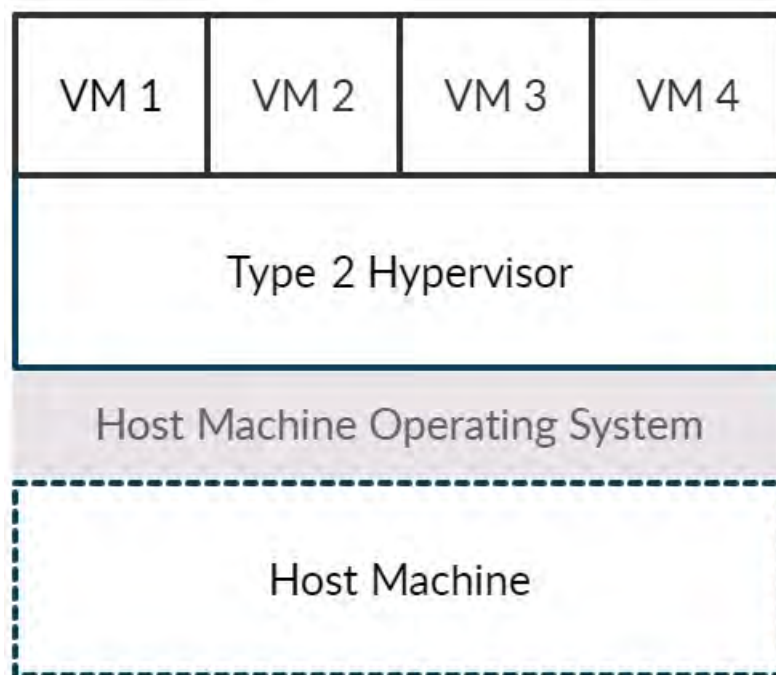


FIGURE 3.10: An illustration of a type 2 hypervisor Source [104]

All updates and modifications done on the server will appear on the client image. The administrator will graphically maintain and test the client image. An example of how the client image is maintained as a raw virtual machine is described in Section 8.1.5.

### An example of using Oracle VirtualBox to prepare an LTSP client's image virtual machines

The following are CLI commands to prepare an LTSP client's image using the Oracle VirtualBox.

- *ln -rs /VirtualBox VMs/Xubuntu/xubuntu-flat.vmdk /srv/ltsp/Xubuntu.img*
- *ltsp image Xubuntu*
- *ltsp ipxe*
- *ltsp nfs*
- *ltsp initrd*

The virtual machine is symbolic-linked before building the LTSP client's image. A symbolic-link (symlink) or 'soft link' is a link or shortcut that points to the original file [105]. If the original file gets deleted, the symlink becomes invalid since it now points to a non-existing file. Symlink creates a symbolic link between the 'xubuntu-flat.vmdk' and the LTSP image directory. Afterwards, the user generates the client's image, iPXE menu and compresses (as shown in Table 3.2). Table 3.2 shows how a user can install the TuxMath application using the chroot and virtualisation player modes.

TABLE 3.2: TuxMath installation modes comparison

Chroot mode	Virtual machine mode
Open the GNU/Linux CLI	Open the client's image using the player
<i>apt install tuxmath</i>	Open the Software Center
<i>ltsp image x86_64</i>	Search and install TuxMath application
<i>ltsp ipxe</i>	Click the custom bash script
<i>ltsp nfs</i>	LTSP clients can now access TuxMath
<i>ltsp initrd</i>	

### Comparing chroot and virtual machine modes

Table 3.3 compares the preparing the LTSP client's image as virtual machines and in chroot mode.

TABLE 3.3: A comparison of a virtual machine and chroot modes

Chroot Mode	Virtual Machine Mode
Runs in an isolated chroot jail environment.	Runs in an isolated environment within the virtualisation player.
Until LTSP19+ chroot image modification e.g. application installation affects the server.	VM modification affects do not affect the server machine.
The user has to know the chroot CLI commands.	The user can navigate and explore Runs in an isolated environment within the virtualisation player
Modifications are tested on the LTSP client.	Modifications can be tested in both LTSP client and within the virtualisation player.
The client's image can only be based on the host OS.	The client image is not forced to be based on the host OS e.g. a Xubuntu host can have a Linux Mint as a client image.
Runs on top of the host kernel whilst providing a separate filesystem.	Runs on an entire separate system with its own kernel outsourcing resources from the host OS.
Chroot uses less LTSP server space.	VM uses more host OS space.

### Benefits of using a virtualisation player

With a virtualisation player, one can run and test the client's image on the server without testing the client's image from the LTSP clients. Unlike the chroot (see Table 3.3), whereby CLI commands manage applications, the virtualisation player offers an easier to use GUI environment. A virtualisation player facilitates a graphical way for a user to install, uninstall and explore applications without using a CLI. Most GNU/Linux OS come with graphical application stores e.g. Snap, Ubuntu Store. Figure 3.11 example of the Xubuntu Software Center.

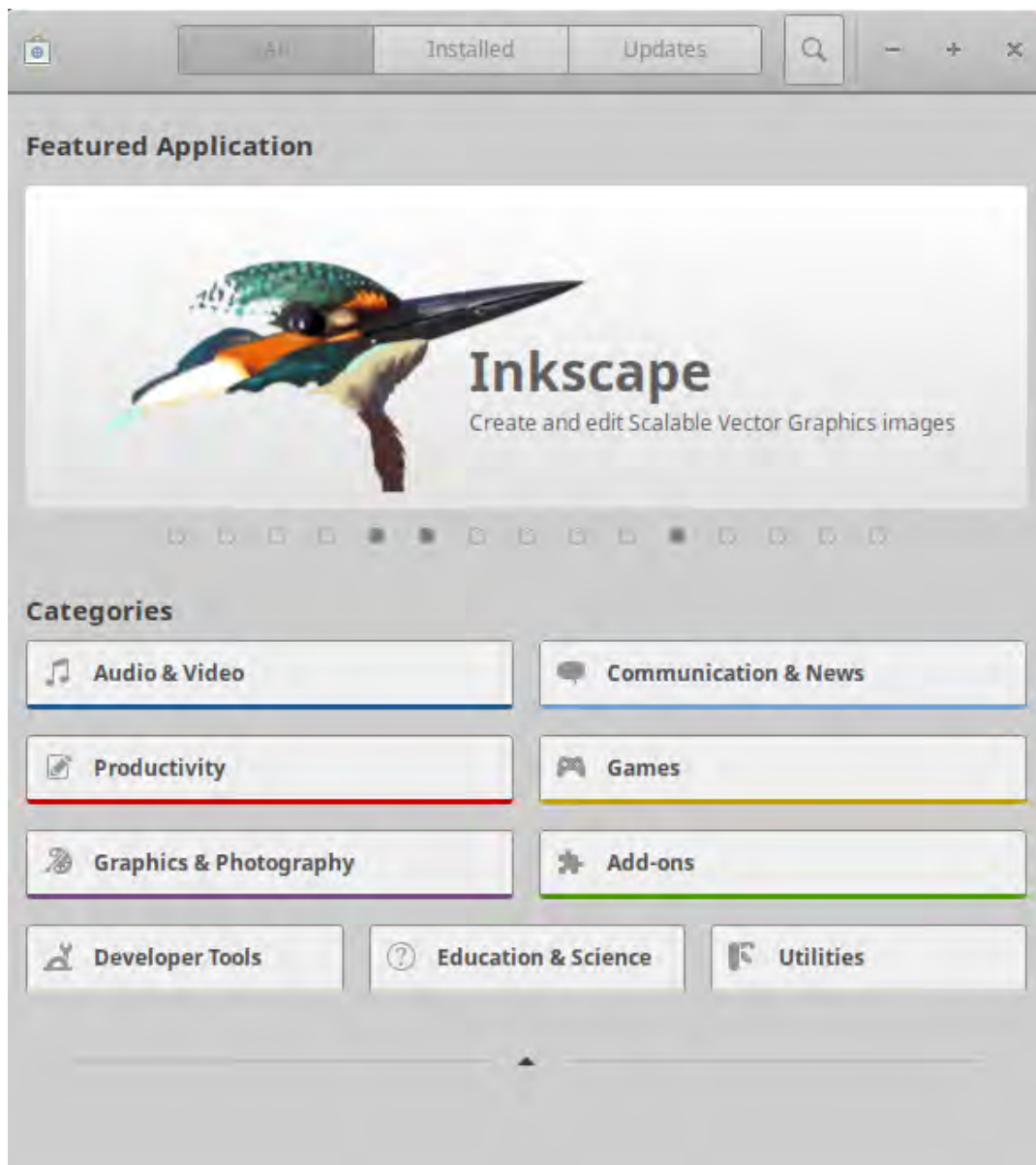


FIGURE 3.11: Xubuntu Software Center [106]

### Drawbacks of using a virtualisation player

As indicated in Table 3.3, since VM is a complete system with its own kernel it uses more space compared to chroot modes. Administrative rights are required to manage and prepare the LTSP client's image.

## 3.6 Conclusion

This chapter introduced the reader to the core concepts of how an overall LTSP computing infrastructure functions on a more technical level. A full

documentation of how to install LTSP 5.2 and LTSP19+ is attached in Appendix A.



## 4 Laboratory experiments: LTSP5+

This chapter introduces the initial laboratory procedures conducted using the then current version LTSP5+. A new version of LTSP, version 19+ was released in August 2019. The aim of this chapter is to find the best setup and tools before deploying a sustainable LTSP computing infrastructure.

### 4.1 Experiment 1: LTSP performance

This section asks the questions that the experiment procedures seek to answer and also justifies the experiment decisions taken.

#### 4.1.1 Experiment questions

The aim is to find out how the LTSP5+ based computing infrastructure performs with thin, thick and a hybrid of both thin and thick clients. In particular the experiment aims at answering the following questions:

1. When is it best to use a client as an LTSP thin or thick client?
2. How do the server machine and the network behave with four clients simultaneously using them?

#### 4.1.2 Hypotheses

1. Thin clients use more Ethernet bandwidth and server resources compared to thick clients.
2. Older hardware has an inferior computing power and should work better as thin clients.

#### 4.1.3 Hardware specifications

Hardware specifications are retrieved from the server machine or client computer by checking the BIOS and running the command *lshw*. This subsection

details the hardware specifications of the hardware involved in the LTSP installation. The experiment involved a SuperMicro server, Mecer, Proline and HP clients.

### SuperMicro server

TABLE 4.1: SuperMicro server hardware specifications

Description	SuperMicro
Product	X9SCL/X9SCM
Vendor	Supermicro
CPU	Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz
RAM	8GiB.
Disk	931GiB (1TB)
Free Memory	913.7 GB

The SuperMicro server machine supports four disk drives that can be hot-swapped during operation using the RAID system. The server machine specifications are adequate to support the LTSP clients lab experiment.

### Mecer Xtreme Jupiter Mini PC client

TABLE 4.2: Mecer Mini PC hardware specifications

Description	Mecer Xtreme Jupiter Mini PC
Model	Jupiter-h110
CPU	Support 6th and 7th Generation Core i7 Skylake and Kaby Lake CPU [i7-6700K / i7-7700K]
RAM	4GB DDR4-2400, (2 x 288PIN DDR4 DIMM Slot, Upgradable to 32GB).
Disk	0GiB (can be installed)

### Proline Nanoware client

TABLE 4.3: Proline Nanoware hardware specifications

Description	Proline Nanoware
CPU	Intel (R) Core(TM) 2 Duo CPU E7500 @ 2.93 GHz Intel 64 Architecture
RAM	2GB (Upgradeable)
Disk	320GiB

### HP T510 client

TABLE 4.4: HP Client hardware specifications

Description	Hewlett Packard T510
CPU	VIA Eden X2 U4200 @ 1.00 GHz
RAM	2GB
Disk	0GB (can be installed)

The Mercer Mini PC is more recent hardware with many functions of a modern computer such as WiFi support and USB 3.0 Super-speed connector. The other clients, Proline Nanoware and HP are older hardware and designed to function as thin clients.

### Ethernet switch

A 16-ports Gigabit and an 8-ports 10/100 Mbps switch connected LAN clients to the server. Both D-Link switches have a minimum of five ports. The LTSP computing infrastructure only requires one switch at a time, and both switches could support five Ethernet ports, i.e. four ports for clients and one port for the server machine. An Ethernet Gigabit switch implies that the switch can support 10/100/1000 speeds and will switch to the relevant one [107]. A Gigabit switch, therefore, works almost the same as a 10/100 Mbps switch. The Ethernet switch keeps track of client computer MAC addresses and forwards frames to ports without significant network delays [94]. A simple hub will allow all points of the LAN to see packets, and this slows traffic.

#### 4.1.4 Software specifications

Initial experiments used LTSP5.2 which was then the latest version before the release of LTSP19+ on 18 August 2019. LTSP5+ experiments were conducted using several GNU/Linux systems and supporting applications. Since LTSP

clients need a GUI environment to work with, it is recommended to install a GNU/Linux desktop. The server's desktop is used as a template to build the clients image as either chroot or chrootless see Appendix A.

A combination of Ubuntu Server and Xubuntu 18.04 LTS desktop was chosen because by default a GNU/Linux server uses fewer resources and has a smaller memory footprint. The Xubuntu desktop is lightweight compared to other Ubuntu-based systems like Mate, GNOME and Linux Mint.

### **Long Term Support**

For a sustainable deployment, an LTS (Long Term Support) OS version of Ubuntu was chosen. An LTS version of GNU/Linux is released every two years and is actively updated, patched and maintained throughout its lifetime to provide quality software [108]. LTS are thoroughly tested and involve years of improvements, and are more stable releases of Ubuntu-based systems. An LTS version allows a more stable computing environment that does not break due to updates or other modifications [109].

### **Supporting tools**

The Glances monitoring tool was used to measure the system resource utilisation such as CPU, memory and bandwidth. Glances is a tool for monitoring computing resources for GNU/Linux and Windows systems [110]. Figure 4.1 shows an example of a Glances diagram showing the system resources usage from the server machine whilst serving three LTSP clients.

```

Terminal - tichzvidz@ltspserver2: ~
File Edit View Terminal Tabs Help
ltspserver2 - IP 146.231.88.18/24 Pub 146.231.88.18 Uptime: 2:05:01
CPU [ 52.4%] CPU \ 52.4% MEM - 49.3% SWAP - 0.0% LOAD 4-core
MEM [ 49.3%] user: 41.2% total: 7.66G total: 4.00G 1 min: 2.18
SWAP [ 0.0%] system: 8.6% used: 3.77G used: 0 5 min: 2.16
idle: 47.5% free: 3.89G free: 4.00G 15 min: 2.68

NETWORK Rx/s Tx/s TASKS 429 (1199 thr), 2 run, 369 slp, 58 oth
enp0s31f6 2.48Mb 57Kb Systemd 12 Services loaded: 216 active: 216
enp4s0f0 0b 0b
enp4s0f1 0b 0b
enp4s0f2 0b 0b
enp4s0f3 22.9Mb 527Mb
lo 184Kb 184Kb
DefaultGateway 12ms

DISK I/O R/s W/s
sda1 0 0
sda2 0 28K
sr0 0 0

FILE SYS Used Total
2021-01-27 12:01:48 No warning or critical alert detected

CPU% MEM% PID USER NI S Command
19.8 1.5 18166 tux4 0 S /opt/brave.com/bra
32.9 3.5 18262 tux4 0 S /opt/brave.com/bra
36.9 1.7 8835 tux3 0 S /opt/brave.com/bra
17.7 1.6 19890 tux2 0 S /opt/brave.com/bra
5.6 0.6 9892 tichzvidz 0 R /usr/bin/python3 /
5.6 0.6 13842 tux3 0 R /usr/bin/python3 /
4.3 0.9 2170 root 0 S /usr/lib/xorg/Xorg
3.1 2.8 19848 tux2 0 S /opt/brave.com/bra

```

FIGURE 4.1: An example of the Glances application running on the server

Here the LTSP machine is using a dual NIC configuration with the NIC *enp4s0f3* connecting to the LTSP clients via an Ethernet switch and NIC *enp0s31f6* connecting to the Internet.

The *Epoptes* classroom management tool was used to perform a 30 seconds LAN bandwidth. A 30 seconds LAN (Local Area Network) benchmarking was used using the *Epoptes* tools. The LAN benchmark tool measures the upload and download speeds between two computer [111].

#### 4.1.5 Experimental procedures

The clients and the server were setup to emulate a full-blown lab on a smaller scale. A full-blown lab can have 20 or 30 clients for example. The goal was to push the available computing resources close to their limits to emulate a real life situation. This was guided by revisiting the experiment questions in Section 4.1.1. Therefore the computing resources might be higher compared to normal user usage. Figure 4.2 shows all hardware connected running LTSP5+ over a dual NIC network configuration.

The following applications were run simultaneously on four LAN clients:

1. Firefox browser simultaneously streaming two YouTube tabs

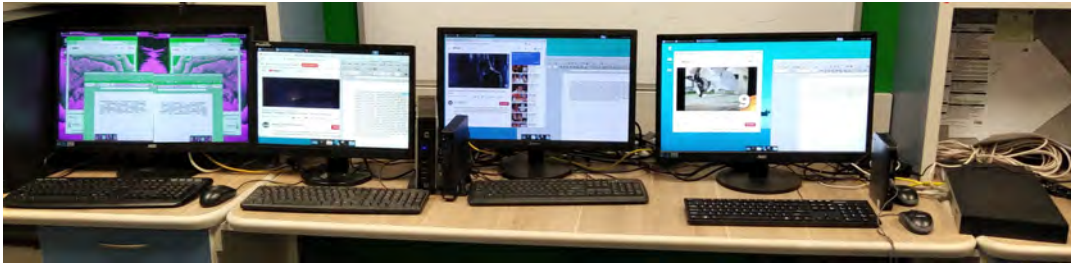


FIGURE 4.2: An LTSP computing infrastructure running four clients

## 2. LibreOffice writer

The CPU usage by applications such as streaming videos is volatile. As an attempt to be more precise, the recordings are averages of 3 sets of results.

The client computers connect to the server machine via an Ethernet Gigabit D-Link switch. The four LAN clients were both old and new hardware clients. This helped to determine how the LTSP computing infrastructure will perform with different old and recent hardware specification.

## 4.2 LTSP5+ thin-client computing

This section measures the performance of the LTSP5+ network by measuring the CPU, memory, transmission and receive bandwidth. The measurements were taken from the server machine and thin clients.

### 4.2.1 Server results

The values in the tables were averages of repeated three measurements per server machine. Table 4.5 shows the computing resources used by the server machine. The first row has zero number of clients so that the jump from zero to one clients is measured accurately.


TABLE 4.5: Thin-client computing server's results

Number of Thin Clients	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
0	26.40	20.20	296.00B	104.00B
1	32.70	38.90	315.1MB	15.80MB
2	37.10	48.10	418.3MB	16.60MB
3	52.40	49.30	527.50MB	22.70MB
4	64.20	48.10	516.00 MB	24.38MB

Table 4.5 shows the CPU and memory usage in percentages for two Mecer Mini PC and 2 Proline Nanoware thin clients. The thin clients were streaming two YouTube tabs and using LibreOffice writer. The MB (MegaByte) and B (Byte) represents the transmission or receive measured per second. Zero number of clients represents an almost idle server machine state when no client is using server resources.

### 4.2.2 30 seconds LAN benchmark: Thin clients

An Epopetes 30 seconds LAN (Local Area Network) benchmark tool measured the bandwidth consumption within 30 seconds. The LAN benchmark tool indicates the upload and download speeds between the server and thin clients.



The screenshot shows a window titled "Network benchmark results" with a table of bandwidth data for four thin clients. Below the table, there are overall statistics for the server's download and upload bandwidth.

Computer	Upload bandwidth	Download bandwidth
ltsp247	130.7 Mbps	61.9 Mbps
ltsp232	95.9 Mbps	72.1 Mbps
ltsp231	67.6 Mbps	130.5 Mbps
ltsp192	562.5 Mbps	140.2 Mbps

**Overall bandwidth statistics**  
Average server download: **101.2 Mbps**    Average server upload: **214.2 Mbps**  
Total server download: **404.7 Mbps**    Total server upload: **856.7 Mbps**

Close

FIGURE 4.3: A 30 seconds LAN bandwidth benchmark

The first column represents the LTSP thin clients. The second column represents the upload bandwidth of traffic being directed from the server machine to thin clients. The third column represents the download bandwidth of traffic coming from the clients to the server machine.

### Observation: LTSP server usage

The graph represents how thin clients use the server machine's CPU and Memory in percentages.

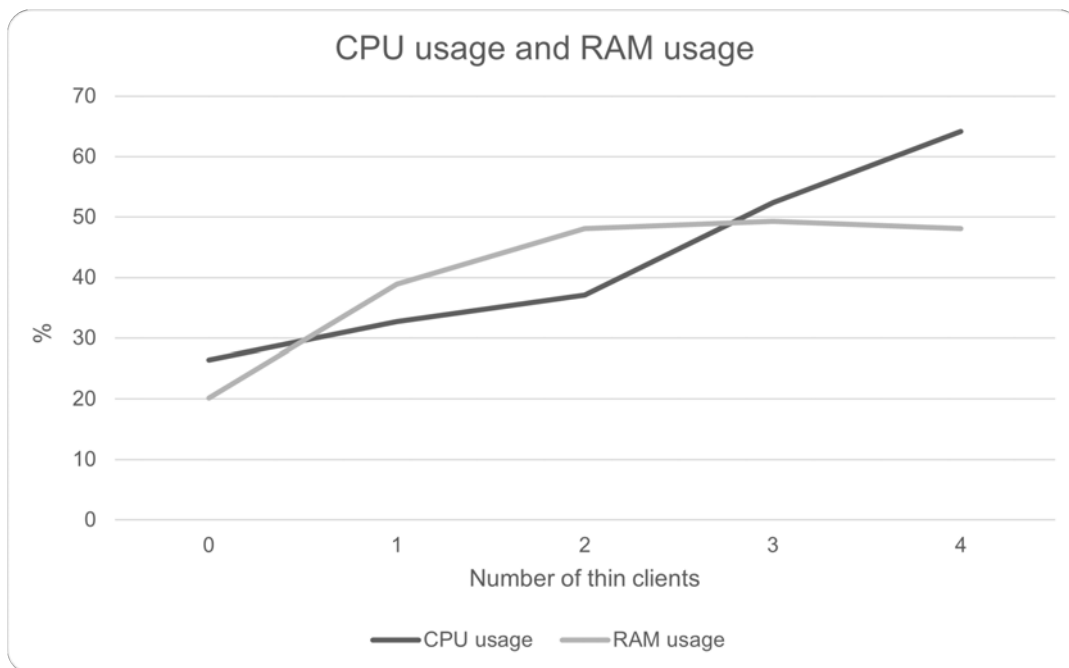


FIGURE 4.4: A graphical representation of the server's CPU and memory usage



As more clients were connected to the server machine the CPU and memory consumption increased. The CPU usage increased following a much steeper curve compared to memory. After adding two thin clients the Memory curve stabilises and showed signs of an asymptotic behaviour towards 50%.

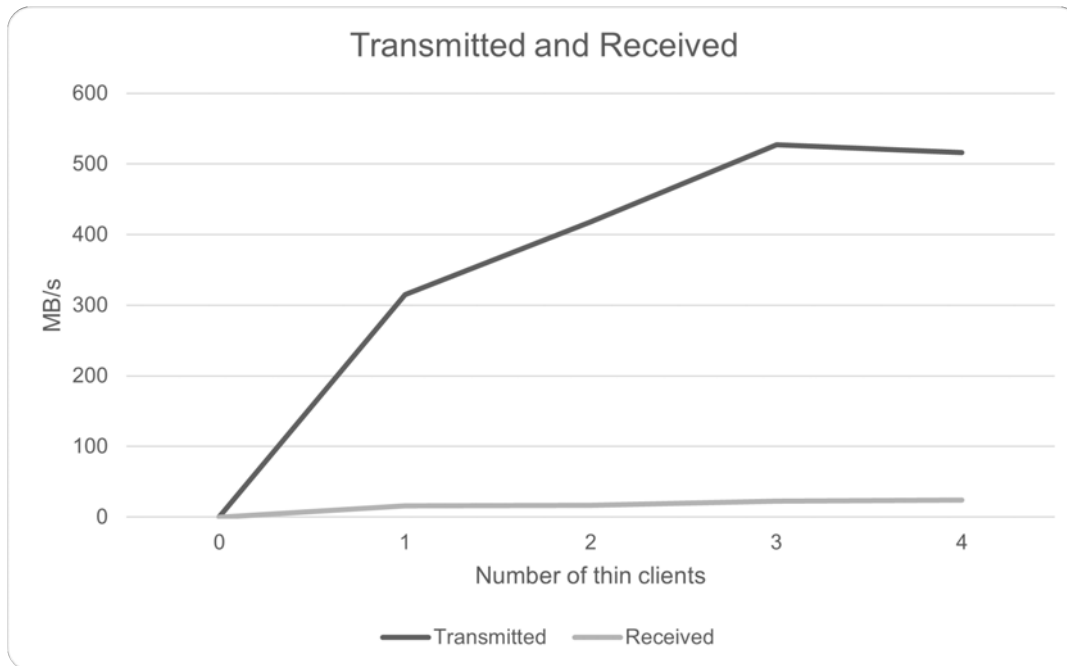


FIGURE 4.5: A graphical representation of the server's bandwidth consumption

The number of packets transmitted per second increased to just over 500MB/s as more clients were added to the LTSP computing infrastructure. The number of packets received remain almost stable ranging from 15 to 20MB/s.

### 4.3 LTSP5+ thick clients results

A similar experiment was conducted but the LTSP5+ server machine was working as a thick or fat client-server machine. Thick clients perform most of their computing locally and for this reason, a desktop GNU/Linux is needed to support local computing. An example of installing Xubuntu desktop is achieved by running the chroot CLI command below.

```
ltsp-chroot -m apt install xubuntu-desktop
```

### 4.3.1 Server results

Table 4.6 shows the results from the server machine when performing thick-client computing.

TABLE 4.6: Thick-client computing server's results

Number of thick clients	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
0	0.80	28.60	2.00B	208.00B
1	1.60	28.90	3.44MB	4.17MB
2	1.60	28.10	3.94MB	4.17MB
3	1.20	27.50	5.05MB	5.72MB
4	1.60	28.10	3.94MB	4.17MB

As expected, thick clients perform most of their computing locally. The thick-client LTSP computing infrastructure unlike a thin-client LTSP computing infrastructure utilises fewer server machine resources. On the other hand, there is high local usage of computing resources by thick clients.

Figure 4.7 shows how the individual thick clients performed using their local resources.

TABLE 4.7: Thick clients 'local' computing results

Thick Client	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
Proline Nanoware	97.10	75.20	4.25MB	9.40MB
HP T510	95.00	71.90	1.10MB	1.54MB
Mecer MiniPC_1	80.00	67.10	1.92MB	1.60MB
Mecer MiniPC_2	90.40	89.30	9.20MB	9.17MB

### 4.3.2 A 30 seconds LAN benchmark: Thick clients

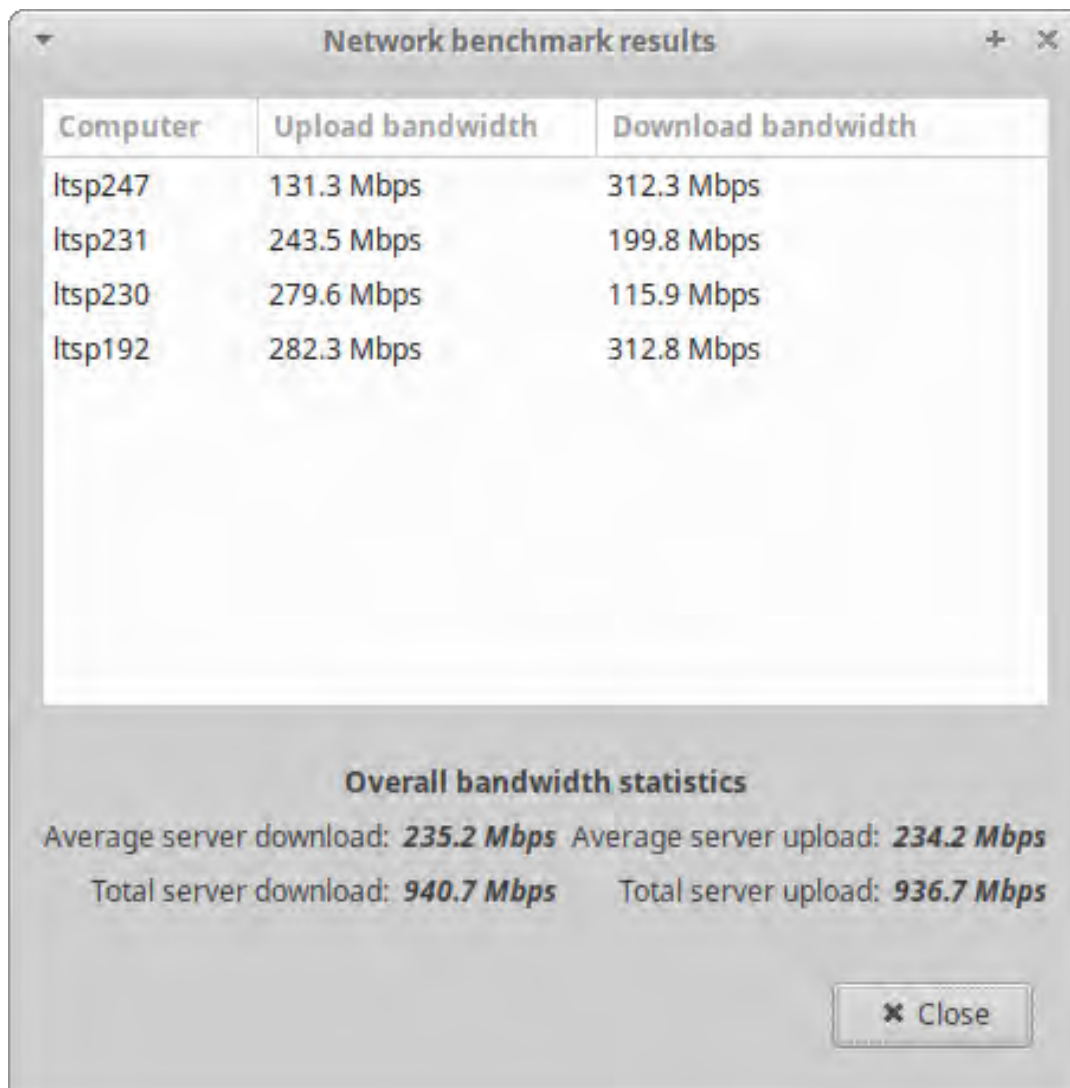


FIGURE 4.6: A 30 seconds LAN bandwidth benchmarking

A 30 seconds network benchmark results for Figure 4.6 are higher compare to thin clients benchmark in Figure 4.6. Notably, the total server download for thick client was 940.7 MBps compared to 404.7MBps of thin-clients network. This is because the server machine has to load a GNU/Linux system that can function almost independently from the LTSP server machine.

#### Observation: Server usage

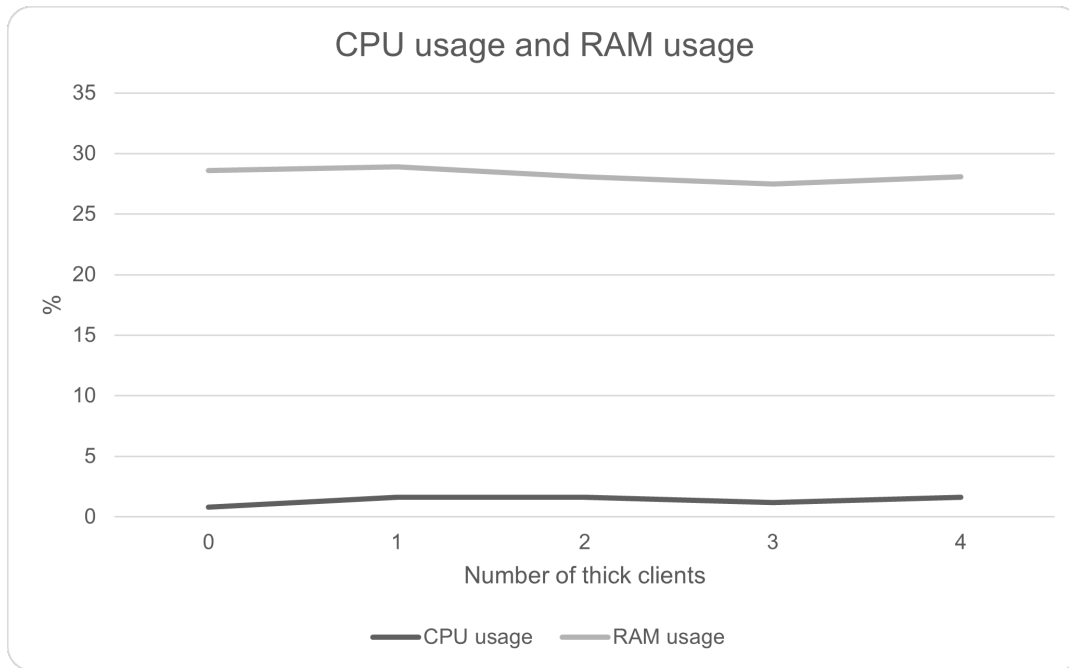


FIGURE 4.7: A graphical representation of the server's CPU and RAM usage

As expected Graph 4.7 shows relatively lower server machine's CPU and Memory usage compared to Graph 4.4. This is because the server machine is not performing all of the computing.

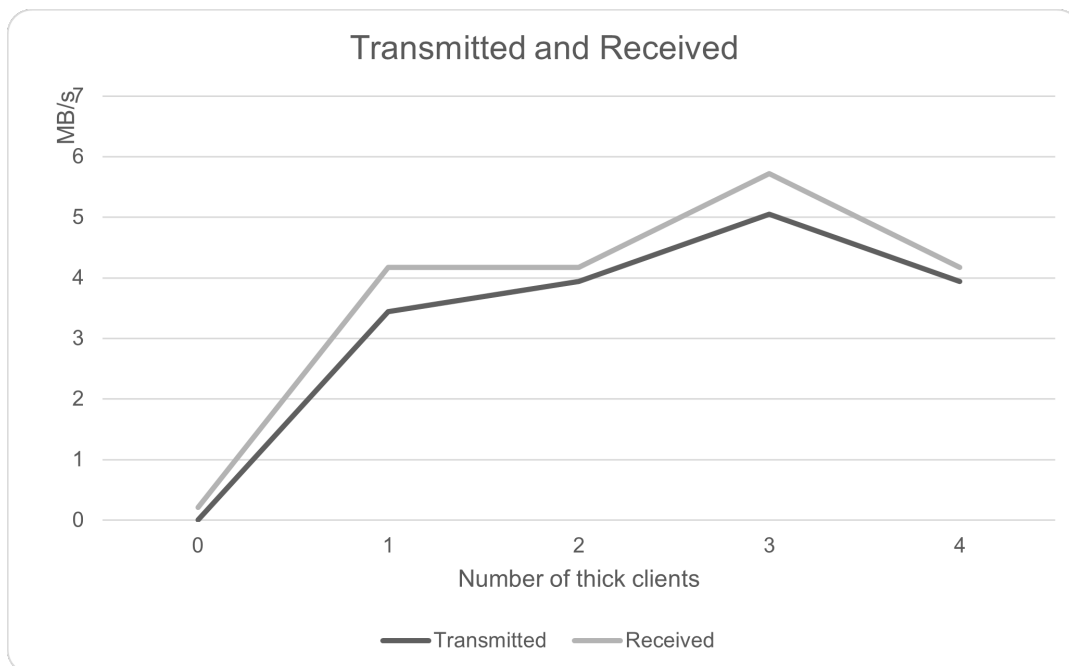


FIGURE 4.8: A graphical representation of the server's bandwidth consumption

Compared to Graph 4.5, thick clients Graph 4.8 showed lesser bandwidth consumption notably the transmitted traffic. This is because thick clients do not require to maintain a constant connection to the LTSP server machine as most resources are available locally.

The LTSP5+ computing infrastructure can simultaneously host thin and thick clients as a client hybrid network. The following section experiments with client hybrid LTSP5+ network.

## 4.4 An LTSP5+ computing infrastructure client hybrid

A hybrid LTSP5+ computing infrastructure is an LTSP computing infrastructure that consists of both thin and thick clients. Figure 4.9 shows an LTSP computing infrastructure with a hybrid of both thin and thick clients. In LTSP5+ this is attainable by adjusting the LTSP configuration file (*lts.conf*) file with the following parameters.

```
FAT_RAM_THRESHOLD=502
```

The value 502 is in MB and can be adjusted as needed. Also, comment out the fat client line by adding a # as below.

```
#LTSP_FATCLIENT=False
```

The clients can also be assigned to a thin or thick client by using the MAC address and adjusting the parameters under the client like below.

```
[ A1:B1:C2:D4:E5:F6 ]  
LTSP_FATCLIENT=False
```

The above parameters use the MAC addresses to turn the LAN client into a thin client.

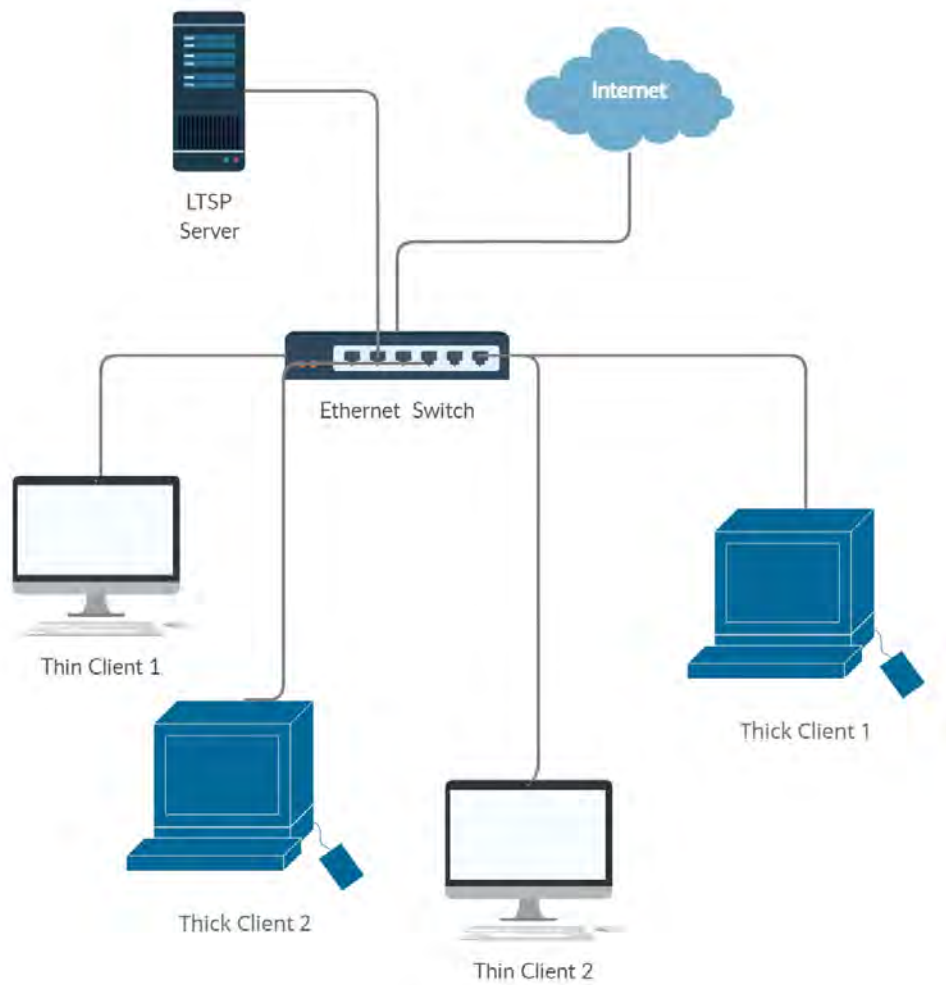


FIGURE 4.9: An LTSP hybrid computing infrastructure

The hybrid LTSP computing infrastructure is useful when there is a mix of old and new client hardware. Normally old clients hardware has inferior hardware capabilities and they perform better as thin clients. On the other hand new hardware can handle computing locally and do not require a server as powerful as a thin-client server. Figure 4.10 shows how a hybrid LTSP computing infrastructure looks like from the Eoptes perspective.

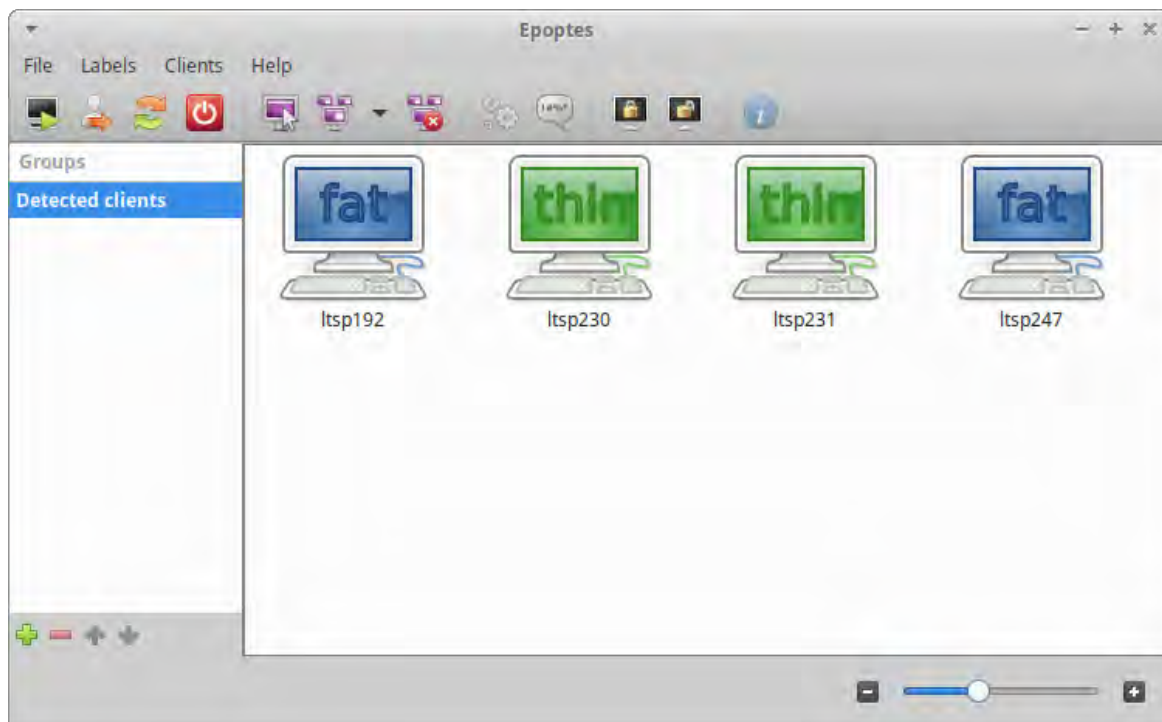


FIGURE 4.10: Epoptes showing a hybrid LTSP computing infrastructure

As indicated in Figure 4.10 the green represents thin clients whilst blue represents fat clients. Packet forwarding has to be enabled in the GNU/Linux kernel so that thick clients' packets are routed to an external network (see Section 4.5).

#### 4.4.1 Server usage

Table 4.8 shows the results of different combinations of a hybrid LTSP computing infrastructure.

TABLE 4.8: Server hybrid client computing results

Number of Clients	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
0	4.10	25.40	1.00KB	2.00B
1 (1 thick client)	1.9	25.40	1.29MB	1.41MB
2 (1 thick clients + 1 thin client)	41.40	72.04	322.00MB	3.24MB
3 (1 thick clients + 2 thin clients)	38.20	60.10	435.00MB	17.39MB
4 (2 thick clients + 2 thin clients)	51.10	65.30	443.00MB	20.00MB



Table 4.8 indicates a steep increase in transmitted traffic, CPU and Memory usage when a thin client was added. However, when a second thick client was added the margin of the increase was relatively lower than when a thin client was added.

TABLE 4.9: Hybrid clients local resources usage

Thick-Thin Client	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
Proline Nanoware	44.60	62.10	435.00MB	20.30MB
HP T510	95.00	71.90	435.00MB	21.00MB
Mecer MiniPC_1	26.20	52.00	10.00MB	9.00MB
Mecer MiniPC_2	25.40	42.10	3.38MB	3.39MB

Table 4.9, first two rows indicate thin clients followed by two Mecer Mini PC thick clients. The two Mecer Mini PC results show the local thin client computing resource utilisation. The Proline Nanoware and HPT510 results show computing resource utilisation on the server machine. As expected the two thin clients use more computing resources compared to the two Mecer Mini PCs.

#### 4.4.2 A 30 seconds LAN benchmark: Hybrid clients

A 30 seconds network benchmark was performed on the thin clients and thick clients connected to the LTSP computing infrastructure.

Computer	Upload bandwidth	Download bandwidth
ltsp247	141.4 Mbps	314.5 Mbps
ltsp231	252.3 Mbps	206.8 Mbps
ltsp230	267.4 Mbps	308.5 Mbps
ltsp192	276.9 Mbps	109.9 Mbps

**Overall bandwidth statistics**

Average server download: **234.9 Mbps**    Average server upload: **234.5 Mbps**  
 Total server download: **939.7 Mbps**    Total server upload: **937.9 Mbps**

Close

FIGURE 4.11: A 30 seconds LAN benchmark: Hybrid clients

The 30 seconds network benchmark closely represent the results for thick clients Figure 4.6. The results for an LTSP thin-client in Figure 4.3 network total server upload were almost half of both thick and hybrid client network. It could have been expected that the hybrid network benchmark results should be lower than the thick-client network results in Figure 4.6. A 50% thick client network(hybrid client network) had almost the same (total or average) upload and downloads as a 100% thick-client network.

#### 4.4.3 Observation: LTSP computing infrastructure comparison

This section uses Table 4.5, 4.6 and 4.8 performance results to construct a bar chart. A thin, thick and hybrid client LTSP computing infrastructure results were compared with each other. The three setups all had four clients running, and the CPU and RAM were measured from the server machine. Figure 4.12

is a bar chart showing a comparison of the different networks.

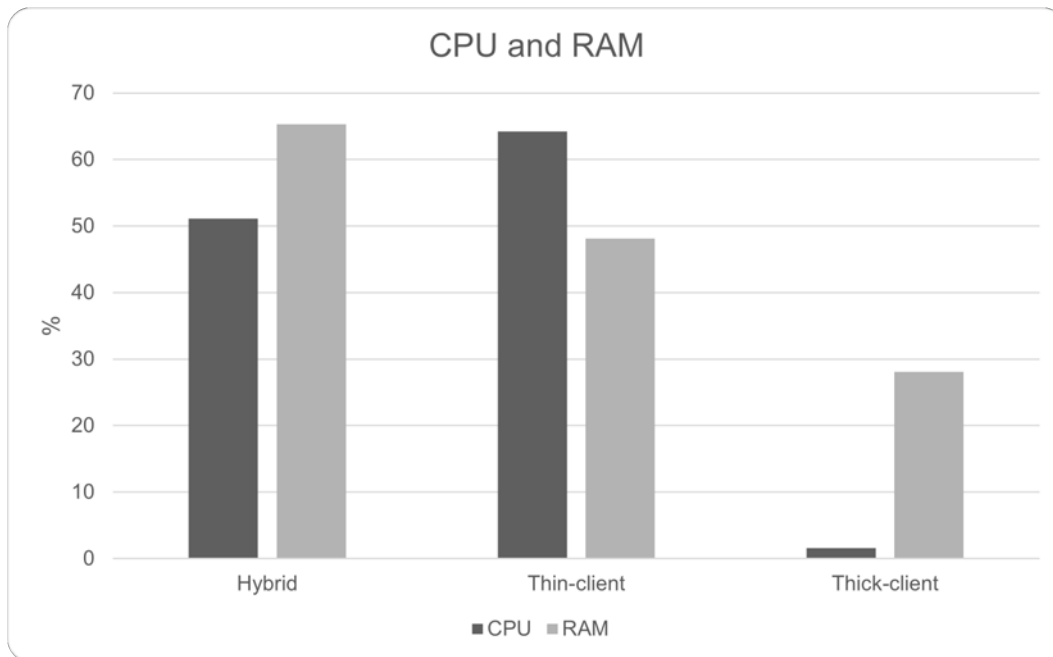


FIGURE 4.12: A performance comparison of a thin vs thick vs hybrid client network

Figure 4.12 shows that thick clients use less CPU and memory computing resources compared to thin and hybrid clients.

## 4.5 NAT: GNU/Linux packet forwarding

Packet forwarding is the ability of the GNU/Linux kernel to accept incoming packets that are not meant for the system and forward them to another network node [112]. Packets can be forwarded among NIC interfaces on different computers.

Since on an LTSP dual NIC configuration (see Section 3.4.2) the server machine manages clients IP addresses, it is important for the server machine to forward incoming and outgoing packets. Forwarding of packets is achieved using the NAT (Network Address Translation) protocol.

The NAT is a method of mapping IP addresses from computers to another single IP address, thereby enabling unregistered computers in a private network to connect to a public network [89, 113]. NAT is also a way to conserve IP addresses and creating private networks that are possibly safe from outside threats.

Thin client shares the same NICs with the server, and there is no need to forward packets. However, thick clients have a separate computing space, and all traffic is directed to the server machine. IP address forwarding can be achieved by creating a bash script that runs on system boot with the following lines.

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE  
sudo iptables -A FORWARD -i eth0 -o eth1 -m state --state  
RELATED,ESTABLISHED -j ACCEPT  
sudo iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Interface *etho 1* connects to the public network and interface *etho2* connects to the LTSP clients. The NIC interface can be adjusted to set route priorities via metrics. For example in the Netplan network manager the network settings can be set as follows:

```

network:
  ethernets:
    enp0s316f:      #Connects to the public network
      dhcp4: true
      dhcp6: true
    enp4s0f1:      #Connects to the LTSP network
      dhcp4: false
      dhcp6: false
      addresses: [192.168.67.1/26]
      routes:
        - to : 0.0.0.0/0
          via: 192.168.67.1
          metric: 150
  version: 2

```

FIGURE 4.13: An example of routing packets from thick clients to the Internet

Interface *etho 1* connects to the public network and interface Figure 2.6 shows how the Netplan routes all packets from the LTSP clients to a gateway *192.168.67.1*. By default the NIC *enp0s316f* is assigned a higher metric number than NIC (*enp4s0f1*) therefore *enp4s0f1* needs to be adjusted. The *enp4s0f1* is assigned a higher priority to allow packets from LTSP clients to be forwarded to an external network e.g Internet. The NICs metrics can easily be checked by running a GNU/Linux CLI *route -n*.

## 4.6 Analysis

This section analyses the results of the experiments and revisits the objectives of the experiment aims.

### 4.6.1 A comparison of thin and thick client deployment

The results indicated that modern computers such as Mecer Mini PC clients perform better as thick clients than as thin clients.

Since thin clients perform most of their computing on the server, the server should have more processing power than a server for thick clients. The LTSP-Wiki Debian [94] recommends a CPU Score of 3200 for four thin clients. According to [114], the SuperMicro had a CPU Score of 3760, and it was enough o experiment with four thin clients.

Thick clients perform most of their computing locally, and a server with lesser processing power compared to a server for thin clients can be used. The LTSP-Wiki Debian [94] recommends a CPU Score of 2120 for four thin clients.

Since thin clients perform most of their computing on the server, LTSP-Wiki Debian [94] recommends at very least 256MB of RAM. Most computer hardware comes with more than 256MB of RAM, which is not difficult to get nowadays. Although LTSP-Wiki Debian [94] suggested that thick clients should have a minimum RAM of 2GB nowadays, it is not enough. Most users now use more web applications that use intensive memory. Experiments in this chapter have shown that 4GB RAM is ideal for accommodating memory-intensive applications.

However, old clients such as Proline Nanoware performed better as thin clients. A client-hybrid network is best used when the lab consists of both old and new client computers. In a normal situation, two servers are used to support thin clients and the other to support thick clients. A client hybrid server machine will reduce the cost of buying another machine. The Bar-graph 4.12 indicates similar resource utilisation for a hybrid and thin client network. Therefore, unless there is a relatively low number of thin clients, the server machine should have almost the same computing power as a thin client-based server. Poor communities in South Africa normally have more old hardware and most probably a less powerful server machine. It is therefore recommended to use a powerful server that will support old and recycled thin client computers. Table 4.10 compares the deployment the LTSP computing infrastructure as thin vs thick clients.

TABLE 4.10: Comparison of a thin versus thick LTSP computing infrastructure

Thin client deployment	Thick client deployment
Can be deployed using low-powered hardware with a recommended 2GB RAM	Requires more recent hardware with a recommended 4GB RAM.
Requires a more expensive and powerful server machine e.g. a minimum of 11000 CPU Mark for 30 thin clients.	Requires a less expensive and powerful server e.g. a minimum CPU Mark of 2900 supports 30 thick clients.
Thin client deployment allows the recycling of old hardware	Thick client deployment requires hardware capable of performing their own computing without relying on the LTSP machine.

Table 4.10 shows that the choice of deployment depends on the hardware specifications of the LTSP clients or in other terms the budget.

### 4.6.2 Challenges of the experiment

This section explores the challenges and recommendations to to deploy an LTSP5+ computing infrastructure.

#### PCI bus errors

Peripheral Component Interconnect (PCI) bus connects the devices such as a mouse, keyboard, speakers etc with the computer motherboard [115]. Figure 4.14 shows an error generated by the PCI bus whilst trying to boot the GNU/Linux system.

```

Ubuntu 16.04.2 LTS ubuntu tty1
ubuntu login: [ 29.984023] pciport 0000:80:03.0: PCIe Bus Error: severity=Corrected, type=Data Link Layer, id=8018(Transmitter ID)
[ 29.984053] pciport 0000:80:03.0: device [8086:6f08] error status/mask=00001100/00002000
[ 29.984071] pciport 0000:80:03.0: [ 81 RELAY_NUM Rollover
[ 29.984085] pciport 0000:80:03.0: [121 Replay Timer Timeout
[ 29.984099] pciport 0000:80:03.0: Error of this Agent(8018) is reported first
[ 29.984122] pciport 0000:85:00.0: PCIe Bus Error: severity=Corrected, type=Physical Layer, id=8500(Transmitter ID)
[ 29.984144] pciport 0000:85:00.0: device [10b5:8747] error status/mask=00001101/0000e000
[ 29.984162] pciport 0000:85:00.0: [ 0] Receiver Error
[ 29.984175] pciport 0000:85:00.0: [ 81 RELAY_NUM Rollover
[ 29.984189] pciport 0000:85:00.0: [121 Replay Timer Timeout
[ 83.108935] pciport 0000:80:03.0: PCIe Bus Error: severity=Corrected, type=Data Link Layer, id=8018(Transmitter ID)
[ 83.109023] pciport 0000:80:03.0: device [8086:6f08] error status/mask=00001100/00002000
[ 83.109043] pciport 0000:80:03.0: [ 81 RELAY_NUM Rollover
[ 83.109058] pciport 0000:80:03.0: [121 Replay Timer Timeout
[ 113.651536] pciport 0000:80:03.0: PCIe Bus Error: severity=Corrected, type=Data Link Layer, id=8018(Transmitter ID)
[ 113.651641] pciport 0000:80:03.0: device [8086:6f08] error status/mask=00001100/00002000
[ 113.651699] pciport 0000:80:03.0: [ 81 RELAY_NUM Rollover
[ 113.651764] pciport 0000:80:03.0: [121 Replay Timer Timeout
[ 114.665528] pciport 0000:80:03.0: PCIe Bus Error: severity=Corrected, type=Data Link Layer, id=8018(Transmitter ID)
[ 114.665602] pciport 0000:80:03.0: device [8086:6f08] error status/mask=00001100/00002000
[ 114.665671] pciport 0000:80:03.0: [ 81 RELAY_NUM Rollover
[ 114.665698] pciport 0000:80:03.0: [121 Replay Timer Timeout

```

FIGURE 4.14: A black screen showing a PCI bus error Source [116]

The Mecer hardware could not handle interrupt messages properly and this resulted in PCI bus errors. The PCI Error is generated by the GNU/Linux



kernel reporting the hardware issue. As a result, the kernel will keep generating interrupt error messages up to a point where the log size reaches the limited root space. The system is then stuck normally at a black screen displaying the error message prevent the system from booting. The guides from [116, 117], suggested to adjust the following kernel parameter by editing the grub file (*/etc/default/grub*).

1. Turn off the Advanced Error Reporting (AER) so that the log size does not exceed the root space.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash pci=noaer"
```

2. Turn off the Message Signalled Interrupts (MSI), some hardware give PCI errors if the MSI was badly implemented.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash pci=noms"i"
```

3. Turn off memory mapping parameter since some BIOS may lead to PCI errors.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash pci=nommmconf"
```

4. Modern computer hardware has the Active State Power Management (ASPM) set to true to save power for the machine. This can make the hardware generate interrupts that exceed the root space. The PCI bus error is can be fixed by adjusting the ASPM and kernel parameters.

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash pcie_aspm=off"
```

5. Furthermore, if the system cannot boot because the log file is full. One can boot the machine in recovery mode and delete the log file in the CLI terminal.

### Scattered documentation

At the time (early 2019), LTSP documentation was scattered and it was a challenge to find a comprehensive source. Some sources were drafted for previous versions and could not lead to a successful installation.

### Distribution errors

Some OS even though they shared the same GNU/Linux kernel they behaved differently. A Mecer Mini PC will run into PCI bus errors in Ubuntu Mate but work perfectly fine with Xubuntu desktop (XFCE4). The Linux Lite

system was more lightweight but did not support LTSP as an add-on application.

### **Ethernet switch**

The Gigabit switch could not be tested to the limit with only four clients. This made it difficult to answer the third question in Section 4.1.1. The question then had to wait for a full-blown LTSP deployment with a higher number of clients. The opportunity presented itself with a full blown deployment in the Ntsika school (see Chapter 5).

## **4.7 Conclusion**

This chapter provided a walk-through of a small scale deployment of the LTSP5+ along with the possible deployment challenges. This chapter works as a preparation for full-blown deployments covering most of the scenarios that may arise. Therefore the aim of Chapter 5, 6 is to conduct a more conclusive deployment that covers grey areas.

## 5 LTSP5+ deployment: Ntsika Secondary School

This chapter extends the initial LTSP5+ lab experiments in Chapter 4 by deploying an LTSP5+ thick-client network at a larger scale. The chapter eliminates grey areas that are often overlooked by lab experiments.

### 5.0.1 Background

Ntsika Secondary School is a local Makhanda school located in a peri-urban area of the Eastern Cape. Makhanda is the largest town of Makana Municipality. The town unlike the SLL Dwesa-Cwebe location, Makhanda is inclined to the lower boundary of the urban population 'peri-urban'. In particular, the Ntsika Secondary School is located in a highly populated and impoverished area of Makhanda often code-named 'the location' or 'the ghetto'. This makes the school ideal for ICT4D projects and community e-skilling projects.

After the deployment the Ntsika school now has a dedicated computer lab available for students to use. The lab has a capacity of 30 students and functions under the supervision of a teacher or lab assistants. Before the installation of the LTSP computing infrastructure, there were a few laptops and tablets available for use. Interestingly and due to their mobility most of the laptops were stolen sometime after the lab was deployed. Without the LTSP lab the school will have to rely on tablets rapidly becoming obsolete. The lab used the Internet from an MTN and local Khula Tech Solutions service providers. Figure 5.1 shows several LTSP5+ thick clients in the Ntsika Secondary School computer lab.



FIGURE 5.1: A view of the Ntsika Secondary School LTSP thick clients

## 5.1 Deployment

As discussed earlier the deployment was supported by the lab experiments conducted in Chapter 4. The experiments helped choose an initial architecture for the real-life deployment. The hardware came from SGT Solutions, a turnkey solutions ICT company specialising in supply, deployment among other things [118].

### 5.1.1 Aim of the deployment

The deployment aimed at making available a cost-effective and sustainable computing infrastructure that can support the community.

### 5.1.2 Objectives

The objectives of the deployment is as follows.

1. Setup an LTSP server machine with the right specification to support thirty thick clients.
2. Integrate the LTSP computing infrastructure with the existing network from other stakeholders.
3. Use Epointes to run applications and measure the server performance and confirming that thick clients are running reasonably.

### 5.1.3 Hardware specifications

#### LTSP server machine

A Dell Mini Tower Server machine with a 3.30 GHz CPU, 16GB RAM and a Terabyte of hard disk was installed. Table 5.1 summarises the server's hardware specifications.

TABLE 5.1: Dell Server hardware specifications

Description	Mini Tower Computer
Product	PowerEdge T30 (07C5)
Vendor	Dell Inc.
CPU	Intel(R) Xeon(R) CPU E3-1225 v5 @ 3.30GHz.
RAM	16GiB.
Disk	D931GiB (1TB)
Free Memory	913.7 GB

The server machine was almost identical to the one used to run experiments in Chapter 4.

#### Mecer Xtreme Jupiter Mini PC thick client

According to [94], a thick client should have at least 1000MB of RAM and the Mecer Mini PC had more than enough RAM to function as thick clients computers. It should be noted that in reality nowadays the minimum size for RAM should be at least 2GB.

TABLE 5.2: Mecer Mini PC hardware specifications

Description	Mecer Xtreme Jupiter Mini PC
Model	Jupiter-h110
CPU	Support 6th and 7th Generation Core i7 Skylake and Kaby Lake CPU [i7-6700K / i7-7700K]
RAM	4GB DDR4-2400, (2 x 288PIN DDR4 DIMM Slot, Upgradable to 32GB).
Disk	0GiB (can be installed)

A total of 27 thick clients were assigned to be deployed at the Ntsika Secondary School.

#### Ethernet switch

A Gigabit Ethernet switch was used to connect the server machine and the 27 clients. The switch also connected the school's Physics lab, Administration building etc.

### **5.1.4 Software specifications**

The machine server was using a combination of Ubuntu Server 18.04 LTS with Xubuntu 18.04 LTS to provide the GUI environment to thick clients. Additional educational GNU/Linux applications such as Tux applications, Kanagram, GCompris, KWord quiz etc were also included.

### **5.1.5 Deployment and testing methods**

The server was mounted in a room separate from the actual lab. This helps protect the server from damage and potential vandalism. A UPS (Uninterrupted Power Supply) was installed to protect the server machine from unplanned power cuts by allowing the server machine to shutdown normally. The server machine created the LTSP computing environment via a mounted Gigabit switch. The Mecer thick clients were securely mounted and nailed to avoid vandalism and theft.

The application Epoptes was used to remote login into thick clients simulate normal usage with LibreOffice writer and browser sessions such Wikipedia and Google applications.. The test was setup to simulate normal browser

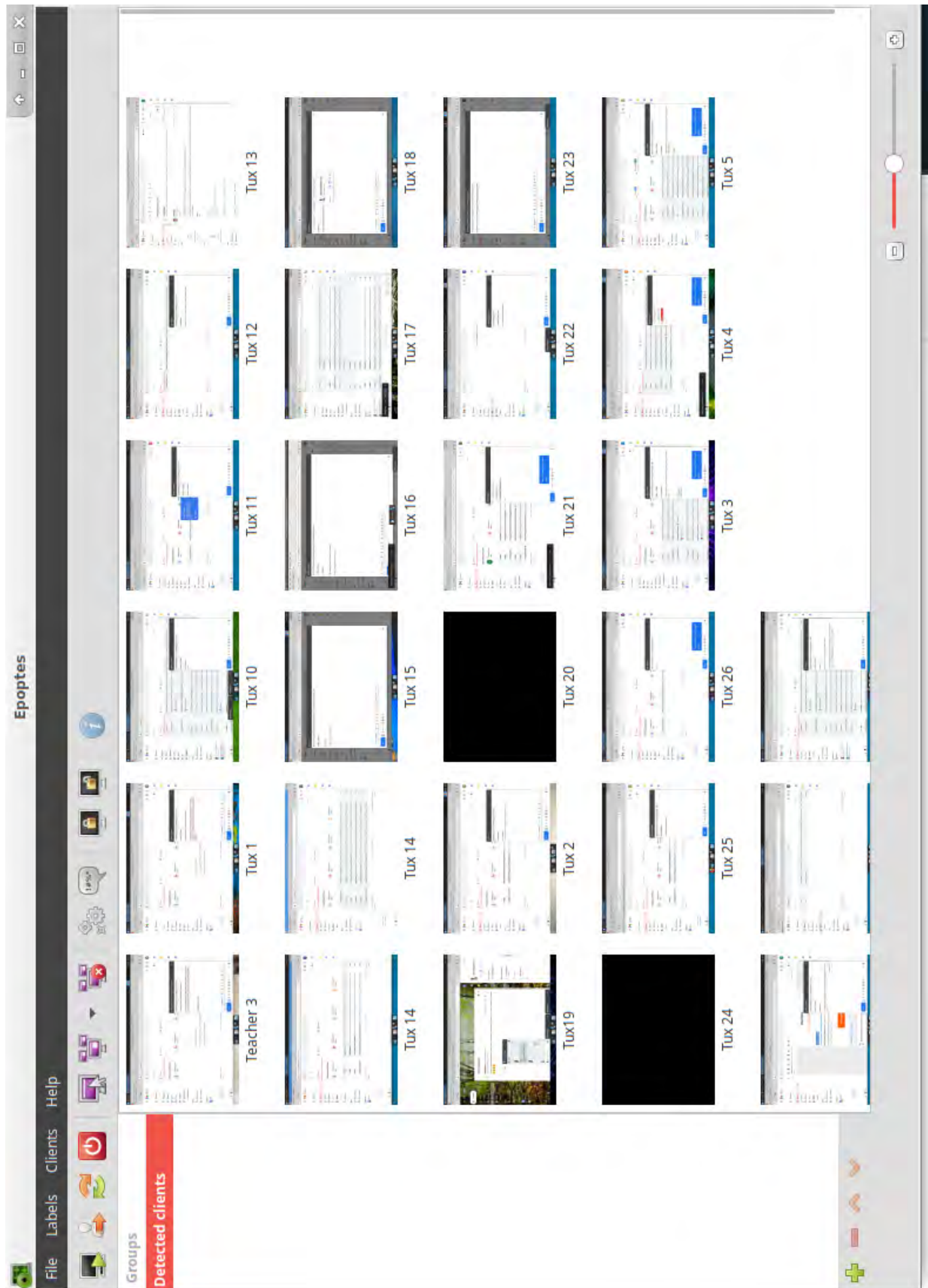


FIGURE 5.2: Epoptes showing thick clients usage

usage such as playing YouTube in the background and browsing web pages such as Google and Wikipedia.

## 5.2 Results

This section gives an overview of how the LTSP installation behaved after deployment. Figure 5.3 shows the total number of 27 Mecer Mini PC clients in usage via Epopetes.

```

Terminal - root@ltspsrvr1: ~
File Edit View Terminal Tabs Help
ltspsrvr1 (Ubuntu 18.04 64bit / Linux 4.15.0-130-generic) - IP 192.168.67.1/24 Uptime: 3:02:03
CPU [ | 12.8%] CPU 12.8% nice: 0.0% ctx sw: 4K MEM 21.3% active: 2.08G SWAP 0.0% LOAD 4-core
MEM [ | 21.3%] user: 9.9% irq: 0.0% inter: 3664 total: 7.66G inactive: 2.16G total: 4.00G 1 min: 0.42
SWAP [ | 0.0%] system: 2.6% iowait: 0.0% sw_int: 4047 used: 1.63G buffers: 447M used: 6 5 min: 0.35
idle: 87.2% steal: 0.0% free: 6.03G cached: 3.47G free: 4.00G 15 min: 0.16

NETWORK Rx/s Tx/s TASKS 370 (594 thr), 1 run, 313 slp, 56 oth sorted automatically by cpu_percent, flat view
enp0s31f6 5.80Mb 1.41Mb Systemd 33 Services loaded: 306 active: 305 failed: 1
enp4s0f0 0b 0b
enp4s0f1 0b 0b
enp4s0f2 0b 0b
enp4s0f3 4.32Mb 5.06Mb
lo 3Kb 3Kb
DefaultGateway 3ms
DISK I/O R/s W/s CPU% MEM% VIRT RES PID USER NI S TIME+ R/s W/s Command
sda1 0 0 11.8 8.9 299M 69.9M 7847 root 0 S 0:15.29 0 0 /usr/lib/xorg/Xorg -core :0 -seat seat0 -
sda2 0 0 5.9 0.6 695M 43.3M 8353 root 0 R 0:02.99 0 0 /usr/bin/python3 /usr/bin/glances
sdb 0 0 3.4 8.3 179M 21.2M 7683 tichzvidz 0 S 0:01.68 0 0 xfwm4 --replace
sr0 0 0 2.2 0.5 556M 37.1M 7827 tichzvidz 0 S 0:00.50 0 0 xfce4-terminal
1.9 2.5 819M 192M 7890 tichzvidz 0 S 0:10.83 0 11K /opt/google/chrome/chrome
sda1 0 0 1.6 0.6 139M 47.7M 1398 root 0 S 0:43.61 0 0 /usr/bin/python3 /usr/bin/twistd3 --nodae
sda2 0 0 1.2 1.2 614M 95.7M 7922 tichzvidz 0 S 0:09.34 0 0 /opt/google/chrome/chrome
sdb 0 0 0.3 0.1 69.9M 6.64M 1046 systemd-r 0 S 0:11.48 0 0 /lib/systemd/systemd-resolved
sr0 0 0 0.3 0.2 1.27G 13.8M 1669 root 0 S 0:17.70 0 0 /opt/teamviewer/tv_bin/teamviewerd -d
0.3 0.1 109M 7.57M 4019 tux23 0 S 0:05.36 0 0 3
0.3 0.1 108M 6.72M 5427 tux10 0 S 0:29.90 0 0 14
/boot/efi 6.04M 511M 0.3 0.0 0 0 7581 root 0 7 0:00.16 0 0 kworker/3:1
e18/1944 55.5M 55.5M 0.3 0.4 318M 28.9M 7687 tichzvidz 0 S 0:00.85 0 0 xfce4-panel
1804/145 163M 163M 0.3 0.5 471M 35.3M 7691 tichzvidz 0 S 0:01.21 0 0 xfdesktop
mes/1514 64.9M 64.9M 0.3 1.2 1.76G 96.8M 7695 tichzvidz 0 S 0:00.76 0 0 /opt/teamviewer/tv_bin/TeamViewer
pd/10707 31.1M 31.1M 0.3 8.4 236M 30.2M 7790 tichzvidz 0 S 0:00.15 0 0 /usr/lib/x86_64-linux-gnu/xfce4/panel/wra
rizer/12 157M 157M 0.0 0.1 221M 9.24M 1 root 0 S 0:32.34 0 0 /sbin/init
0.0 0.0 0 0 2 root 0 S 0:00.00 0 0 kthreadd
0.0 0.0 0 0 4 root -20 ? 0:00.00 0 0 kworker/0:0H
0.0 0.0 0 0 6 root -20 ? 0:00.00 0 0 mm_percpu_wq

SENSORS
acpitz 1 27C
acpitz 2 29C
2021-01-23 09:24:13 No warning or critical alert detected

```

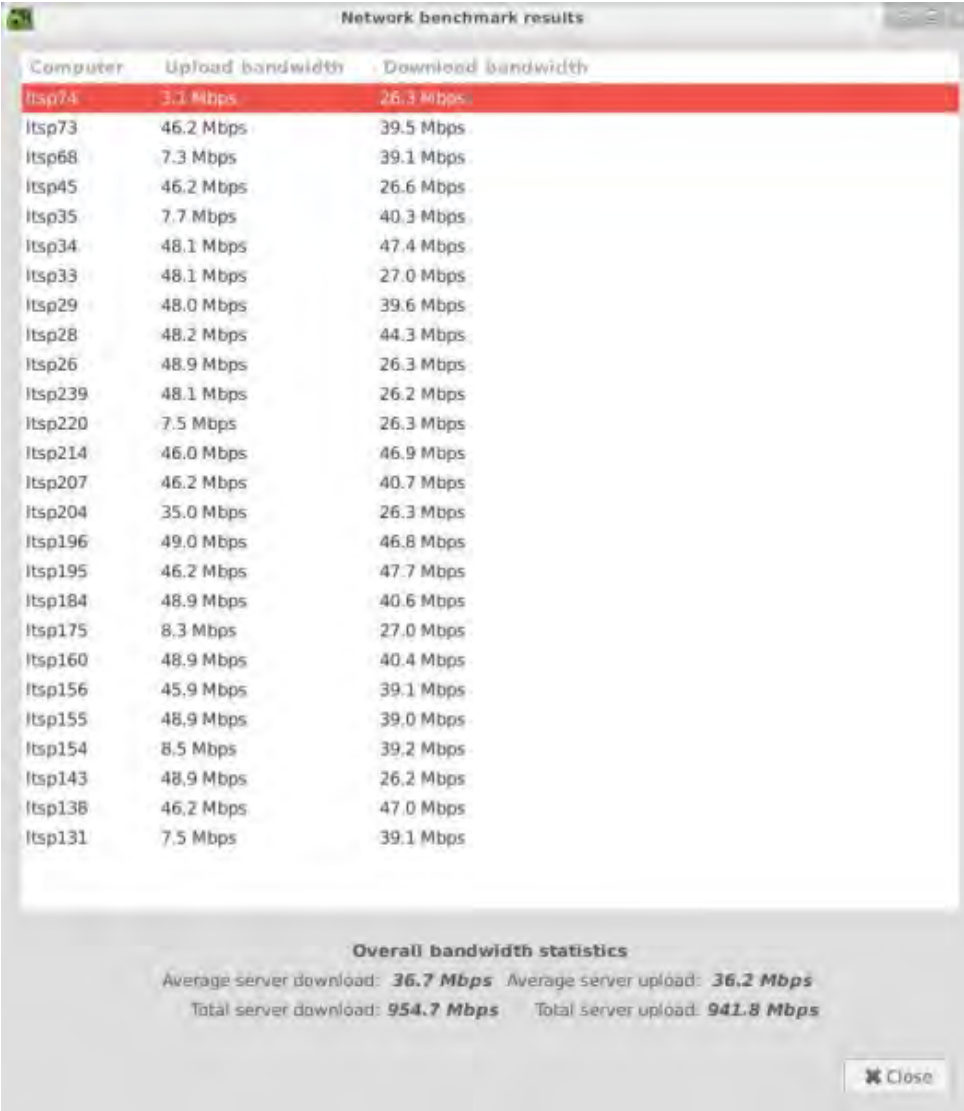
FIGURE 5.3: Ntsika secondary School Server Usage



Figure 5.4 illustrates the lab performance as shown by the Glances tool. A 12.80% CPU usage and 21.30% of memory consumption indicate a sustainable server performance. In this snapshot the server operated at 4.32MB per second and transmitted 5.06MB per second on the interface *enp4s0f3* which was connected to the local LAN and so to the clients. The network interface *enp0s31f6* handled the external traffic for the Internet.

### 5.2.1 A 30 seconds LAN benchmark

A 30 seconds network benchmark was performed on the 30 clients and the results recorded below. The clients were running the Browser and LibreOffice as used earlier in the lab test.



Computer	Upload bandwidth	Download bandwidth
Itsp74	3.1 Mbps	26.3 Mbps
Itsp73	46.2 Mbps	39.5 Mbps
Itsp68	7.3 Mbps	39.1 Mbps
Itsp45	46.2 Mbps	26.6 Mbps
Itsp35	7.7 Mbps	40.3 Mbps
Itsp34	48.1 Mbps	47.4 Mbps
Itsp33	48.1 Mbps	27.0 Mbps
Itsp29	48.0 Mbps	39.6 Mbps
Itsp28	48.2 Mbps	44.3 Mbps
Itsp26	48.9 Mbps	26.3 Mbps
Itsp239	48.1 Mbps	26.2 Mbps
Itsp220	7.5 Mbps	26.3 Mbps
Itsp214	46.0 Mbps	46.9 Mbps
Itsp207	46.2 Mbps	40.7 Mbps
Itsp204	35.0 Mbps	26.3 Mbps
Itsp196	49.0 Mbps	46.8 Mbps
Itsp195	46.2 Mbps	47.7 Mbps
Itsp184	48.9 Mbps	40.6 Mbps
Itsp175	8.3 Mbps	27.0 Mbps
Itsp160	48.9 Mbps	40.4 Mbps
Itsp156	45.9 Mbps	39.1 Mbps
Itsp155	48.9 Mbps	39.0 Mbps
Itsp154	8.5 Mbps	39.2 Mbps
Itsp143	48.9 Mbps	26.2 Mbps
Itsp138	46.2 Mbps	47.0 Mbps
Itsp131	7.5 Mbps	39.1 Mbps

**Overall bandwidth statistics**

Average server download: **36.7 Mbps**    Average server upload: **36.2 Mbps**  
 Total server download: **954.7 Mbps**    Total server upload: **941.8 Mbps**

Close

FIGURE 5.4: network benchmark

The total server download was 954.7 MB per second whilst the total server upload was at 941.8 MB per second. For this experiment it was useful to use the Eoptes tool because it can remote manage and load the applications on the thick clients.

## 5.3 Analysis

This section analyses the results of the deployment and extends experiments from Chapter 4 by performing similar experimental procedures on a larger scale.

### 5.3.1 A comparison of a small and full-blown deployment

Figure 5.5 below shows how LTSP5+ 4 lab thick clients compare to 27 clients deployed in a real-life setting.



FIGURE 5.5: A diagram showing how the server performed in the lab versus the Ntsika Secondary School

Figure 5.5 shows a comparison of 27 thick clients versus 4 thick clients. The 4 thick clients were tested earlier on before a full-blown deployment deployment as part of experiments in Chapter 4. The experiments had almost similar application running.

- **CPU**

The SuperMicro and Dell server machines belonged to the same class i.e Intel(R) Xeon(R) CPU E3-1220 and Intel(R) Xeon(R) CPU E3-1225 respectively. LTSP-Wiki Debian [94], recommends a minimum of 2120 CPU Mark (CPU Score) for 4 thick clients and a CPU Score of 2810 for 27 thick clients.

$$\text{CPU Benchmark score} = 2000 + (30 * \text{number of thick clients}) + (300 * \text{number of thin clients}) \text{ Source [94].}$$

The SuperMicro had a CPU score of 3760 and the Dell server had 3771 which was more than enough to sustain 27 thick clients. The Dell server machine was the most recent and had a higher CPU score and in reality it should perform better than the SuperMicro server regardless of being in the same class.

FIGURE 5.6: SuperMicro vs Dell server benchmark. Source [114]

	Intel Xeon E3-1220 @ 3.10GHz	Intel Xeon E3-1225 @ 3.10GHz
Price	\$59.99 <b>BUY NOW!</b>	\$213.19 <b>BUY NOW!</b>
Socket Type	LGA1155	LGA1155
CPU Class	Server	Server
Clockspeed	3.1 GHz	3.1 GHz
Turbo Speed	Up to 3.4 GHz	Up to 3.4 GHz
# of Physical Cores	4 (Threads: 4)	4 (Threads: 4)
Max TDP	80W	95W
Yearly Running Cost	\$14.60	\$17.34
First Seen on Chart	Q2 2011	Q1 2011
# of Samples	193	262
Cross-Platform Rating	9353	8749
Single Thread Rating	1563	1560
CPU Mark	<b>3760</b>	<b>3771</b>

The CPU usage for 27 thick clients as expected was higher compared to thick clients tested in the lab (as seen in Chapter 4). The 27 clients had a factor of 8 ( $12.8/1.6 = 8$ ) compared to the expected factor of 7.5 ( $30/4 = 7.5$ ). The 0.5 difference was mostly due to the fact that Epoptes was used to activate applications and to continuously monitor the 27 thick clients. Also, the server machines background processes have a varying consumption of both CPU and RAM resources.

- **RAM**

According to LTSPWiki-Debian [94], a calculation with 27 thick clients suggest that 2810MB is needed to sustain the LTSP computing environment.

$$\text{RAM} = 2000 + (30 * \text{number of thick clients}) + (300 * \text{number of thin clients})$$

MB Source [94].

Experiments however have shown that nowadays most applications have an intensive memory requirements. To accommodate applications with intensive memory requirements, the Dell server RAM was doubled to 16GB. This created a safe margin that prevents user experience degradation when additional services are added to the Dell server.

The memory consumption for 4 thick clients was lower than the memory consumption of 27 thick clients. However the RAM consumption for 4 clients was higher than expected. This is due to the following reasons.

1. The Dell server for 4 thick clients used 8GB RAM whilst the 27 thick clients used 16GB RAM.
2. For 27 thick clients, the Dell server was running Epoptes tool to activate applications and monitor the thick clients consuming approximately 400MB of RAM. Whilst for the 4 thick clients there was no Epoptes running and applications were activated manually.

- **Bandwidth**

The LAN bandwidth consumption was almost the same for both 30 and 4 thick clients. Both experiments used different ISP providers. The 4 thick clients had more bandwidth available that was approximately 100MB/s and the 27 thick clients used throttled network at approximately 8MB/s download speeds. Even though there cannot be a more

accurate comparison the bandwidth consumption in both experiments was satisfactory.

### 5.3.2 Server booting tests

After deployment, a lab test was run to test the time (seconds) it takes for a single machine to boot.

Mecer1	Mecer2	Mecer3	Mecer4	Mecer5	Mecer6	Mecer 7
40	23	40	22	20	32	39

Mecer8	Mecer9	Mecer10	Mecer11	Mecer12	Mecer13
37	20	21	23	24	40

The Mercer Mini PC averaged 29.31 seconds to boot each client individually. After testing each Mecer client individually the LTSP computing infrastructure was rebooted, and all clients were rebooted at the same time using the Epointes tool.

The second test was to determine the time it takes for the first and last Mercer client to boot. when 30 clients are booted simultaneously. The procedures were repeated three times to increase the precision of the data collected.

	Experiment1	Experiment2	Experiment3	Average
First Mecer to boot	30	34	29	31
Last Mecer to boot	60	50	58	56

The server machine was transmitting exactly 956MB of the client image via an Ethernet Gigabit switch. In this case, the server machine transmitted 28.68 GB (956MB \* 30) of data within 60 seconds. Booting the LTSP clients in 60 seconds is acceptable to run a computer lab. The results above shows that when 30 clients are booted simultaneously it takes an average of 31 seconds for the first client to boot and an average of 56 for the last client to boot. It is therefore can be assumed that LTSP5+ can simultaneously boot 30 clients in a minute.

### 5.3.3 Troubleshooting the filesystem

#### Fixing broken Ethernet ports

Some of the Ethernet ports were broken and needed to be replaced. This reduced the initial target of deploying 30 thick clients. After fixing all the ports the target of 30 thick clients was reached.

#### Rebuilding the corrupted chroot environment

The LTSP computing environment was running normally but the LibreOffice suite was broken. The LibreOffice application could not be accessed from the LTSP clients. However, from the server machine, LibreOffice was visible. The LTSP server machine maintained the LTSP client's image in a chroot environment. A chroot environment isolates the LTSP client image processes and their children from the LTSP server machine filesystem. Therefore, the LibreOffice and its dependencies files within the chroot were corrupted. The first attempt to solve the problem was to purge and reinstall a new copy of LibreOffice.

#### Possible causes

A system update or the server machine was not switched off normally due to power cuts corrupted the LibreOffice files. The OpenJDK updates may have broken the LibreOffice files and made them inaccessible to the LTSP clients.

#### Attempting to solve the problem.

##### First attempt: Purge and reinstall a new copy of LibreOffice.

The apt's purge command unlike the remove command uninstalls a package together with any configurations and data files of the package. LibreOffice relies on Java's OpenJDK files and it was ideal to reinstall a copy of both the LibreOffice and OpenJDK as highlighted below.

```
ltsp-chroot apt purge libreoffice && ltsp-chroot apt purge openjdk ltsp-chroot apt  
install libreoffice && ltsp-chroot apt install openjdk
```

After performing the above commands, the reinstallation was not successful, and the same errors were returned.

- ca-certificates-java
- openjdk-11-jre-headless:amd64

- openjdk-11-jdk-headless:amd64
- openjdk-11-jdk:amd64
- default-jdk
- default-jdk-headless
- openjdk-11-jre:amd64
- default-jre
- libreoffice-wiki-publisher
- libreoffice-sdbc-hsqldb
- libreoffice-script-provider-bsh
- libreoffice-nlpsolver
- libreoffice-script-provider-js
- Sub-process /usr/bin/dpkg returned an error code (1)

**Second attempt: Fix broken installation and manually delete all LibreOffice and OpenJDK files.**

The first attempt saw the LibreOffice and OpenJDK package being installed partially. The APT command to fix a broken installation was run followed by the 'autoremove' command.

```
dpkg --configure -a ltsp-chroot apt install -f
```

The dpkg command re-configures or repairs a package database if it has been corrupted. It does not configure the installed packages but re-configures unpacked packages useful when repairing the interrupted installation. The 'apt install -f' repairs the broken dependencies for the LibreOffice in this case the OpenJDK dependencies. Unfortunately, both of the above attempts were unsuccessful, and the same errors were returned.

After the second attempt failed the following commands were run to remove and clean all files that were not fully installed, and the files left behind.

```
ltsp-chroot apt autoremove ltsp-chroot apt clean
```

The 'autoremove' command removes all outdated packages and unused packages and dependencies. The 'clean' removes the unnecessary files left behind after a package installation in (/var/cache/apt/archives/ ).

**Third attempt: Manually delete all LibreOffice and OpenJDK files and re-install the packages.**

An attempt to manually delete all files relating to LibreOffice and OpenJDK was made using the following g commands.

```
ltsp-chroot mv /var/lib/dpkg/info/openjdk. /tmp  
ltsp-chroot mv /var/lib/dpkg/info/libreoffice. /tmp
```

The manual deleting of files relating to LibreOffice and OpenJDK was successful, however after an attempt to reinstall LibreOffice the same dpkg errors were returned.

**Fourth attempt: Rebuilding the chroot environment and install a fresh Xubuntu desktop.**

At the fourth attempt was to rebuild the chroot environment and install a new Xubuntu desktop environment. The Xubuntu desktop comes bundled with the LibreOffice applications, Mozilla browser etc. The following command was used to rebuild the chroot and regenerate an LTSP client's image.

```
ltsp-chroot -m apt purge xubuntu-desktop  
ltsp-chroot -m apt install xubuntu-desktop  
ltsp-update-image
```

The fourth attempt was successful, and the LibreOffice applications were now available for LTSP clients to use. It was necessary to approach the problem in various options to try and understand the root cause of the problem to avoid and troubleshoot future problems. The best way to avoid problems is to always backup a working version of the LTSP client's image and store them on a separate location that is not overwritten when generating LTSP client's images. This way in the future if there is a problem with the client's image a working image can easily replace to current one. The exercise also showed the benefits of GNU/Linux chroot as it saves time to rebuild the chroot rather than rebuilding the whole LTSP machine.

## 5.4 Conclusion

The experiments in Chapter 4 built a foundation and contributed to the success of the deployment of LTSP computing infrastructure in the Ntsika school. The introduction of the LTSP computing infrastructure increased the number



of available devices by 27 clients. The LTSP-based computing infrastructure now offers the benefits of free and open-source. For example, there is no need to buy licenses for applications. Students can now use one copy of LTSP client image for computing compared to having separate 27 OS installations. All updates and modification by the LTSP administrator are performed on one LTSP client image. If one of the thick clients in the Ntsika school fail it can be replaced without risking data loss or bringing in another OS. The system has been successfully running for almost two years and can be used as a blueprint to deployment computing infrastructure.

## 6 LTSP5+ deployment: Dwesa schools

This chapter extends the experiments in Chapter 4. Unlike the Ntsika school deployment this chapter focuses on a full scale deployment of thin clients using LTSP5+ computing infrastructure. An LTSP machine was deployed in the Ngwane school and maintenance procedures were done in the Mpume.

### 6.1 Background

The Ngwane Junior Secondary and Mpume Junior High schools are the original and main site of the Siyakhula Living lab project [7]. The two schools are located nearby the Dwesa-Cwebe wildlife reserve under the Mbhashe Local Municipality of the Eastern Cape. The Dwesa-Cwebe community is rural community that has both poor e-infrastructure and road infrastructure. Figure 6.1 shows the geographical location of the two schools highlighted using red dots.

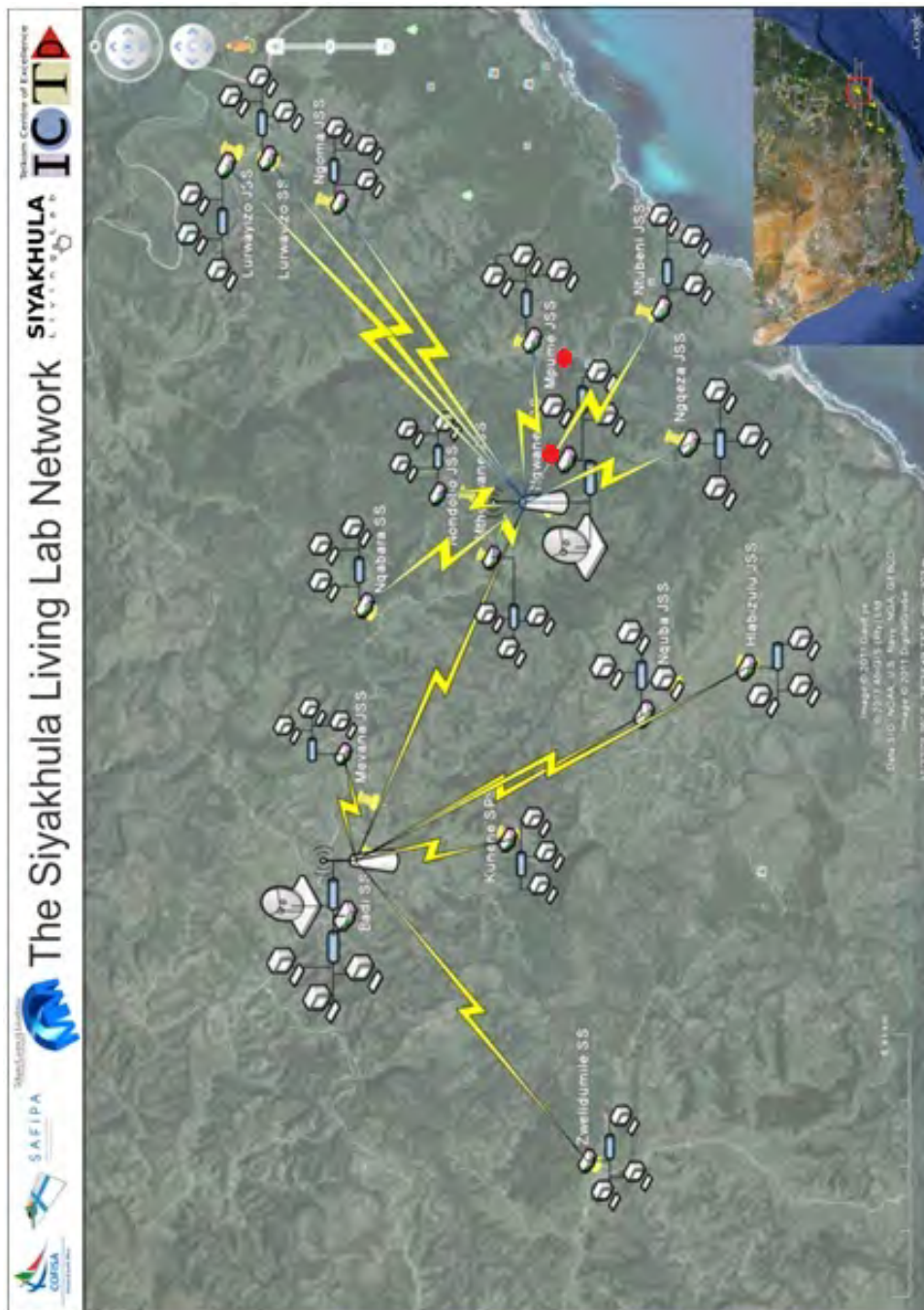


FIGURE 6.1: The Siyakhula Living lab network. Source [58]

Currently, each school has an LTSP server with several thin clients to it. The computing infrastructure together with the installed network infrastructure supports learning in schools, Internet access and services for the overall community.

## 6.2 Ngwane Junior Secondary school

Ngwane Junior Secondary School has a secure computer lab that accommodates about 20 thin clients. Below is a picture showing the Ngwane Junior High School computer lab.



FIGURE 6.2: Ngwane Junior Secondary school computer lab

During the 2019 visit to the Ngwane school, the computing infrastructure was not functioning because of the following problems specifically the following.

1. **The SuperMicro server was down**

The Ngwane SuperMicro machine was down and could not boot into BIOS. The first step was to open the SuperMicro server and clean the dust and resit the RAM the procedures are outlined in Section 6.3. Unfortunately, after several attempts the server still could not boot and needed to be replaced with another identical SuperMicro server.

2. **Several Proline clients were down**

In terms of components, the Proline thin client components can be moved and replaced which is useful when repairing the hardware. Some of the Proline clients were not used for a while and due to challenging environment they collected dust. Among other issues there were a few dead motherboards, power supplies, video output, NIC, and the CMOS

(Complementary Metal Oxide Semiconductor) batteries were fully discharged. Table 6.3 outlines the approach taken to fix the Proline clients problems.

### 3. Rewiring power cables

Some sections of the lab had faulty power connections and there was a need to rewire the lab to function at full capacity.

### 4. Power faults

Dwesa-Cwebe area is remote and it takes a long time for the power service providers to identify and fix a power failure. During periods of bad weather, like heavy rains or strong winds the power lines often fail and frequently it takes days or weeks to fix the problem.

The above procedures fixed several clients and more clients were added to the Ngwane school computer lab.

## 6.2.1 Aim of intervention

The aim of the intervention was to get the Ngwane computer lab fully functional again.

## 6.2.2 Objectives

1. Replace the existing SuperMicro server with an identical SuperMicro server.
2. Fix as many as possible Proline clients using the steps outlined in Table 6.3.
3. Rewiring the lab to allow the lab to function at full capacity.

## 6.2.3 Hardware specifications

The Ngwane school uses the Proline Nanoware thin clients and a SuperMicro server. The SuperMicro server was the same server as the one used in the experiments in Chapter 4. Also, the thin clients were identical to the ones experimented with in Chapter 4. Tables 6.1 and 6.2 details the hardware specifications of the SuperMicro server and Proline Nanoware thin clients.

### SuperMicro server

TABLE 6.1: SuperMicro Server hardware specifications

Description	SuperMicro
Product	X9SCL/X9SCM
Vendor	Supermicro
CPU	Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz
RAM	16GiB.
Disk	931GiB (1TB)
Free Memory	913.7 GB

The major difference was that the amount of RAM which was doubled to 16GB. This accommodated applications with intensive memory requirements and avoided user experience degradation.

### Proline Nanoware client

TABLE 6.2: Proline Nanoware hardware specifications

Description	Proline Nanoware
CPU	Intel (R) Core(TM) 2 Duo CPU E7500 @ 2.93 GHz Intel 64 Architecture
RAM	1GB, 2GB
Disk	320GiB

The Proline clients had 1GB or 2GB of RAM but it did not make much difference because the computing is performed on the server. The thin clients connected to a Supermicro server machine via a Gigabit Ethernet switch.

### Ethernet switch

A Gigabit Ethernet switch was used to connect the Proline thin clients to the SuperMicro server machine.

## 6.2.4 Software specifications

The desktop environment was a combination of Xubuntu 18.04 LTS desktop and Ubuntu Server 18.04 LTS identical to the ones deployed and used on throughout this research. Additional educational GNU/Linux applications such as Tux applications, Kanagram, GCompris, KWord quiz etc were also included.

### 6.2.5 Procedures

1. *Server replacement:*

The server machine was the same one that was tested in Chapter 4. The difference was that it had addition 8GB RAM to compensate for intensive memory applications.

2. *Fixing thin clients:*

Fixing of Proline clients was achieved by following the following steps.

TABLE 6.3: Fixing the Proline Nanoware clients

Problem	Approach
Dead power supply	If only the power supply is not working switch or replace the power supply with the ones with a dead motherboard.
Dead RAM	The first attempt was to clean and re-sit the RAM using a soft cloth (as seen in Section 6.3). The procedure requires avoiding static electricity so as to not corrupt the RAM chip.
If this does not work the RAM chip needs to be replaced or switched.	
No video output	Lack of video output was difficult to replace since the interface is attached to the motherboard. The solution was to test the VGA (Video Graphics Array), HDMI (High Definition Multimedia Interface) and DVI (Digital Visual Interface) video inputs. If all of the video inputs were not working the Proline client's components were used as parts for other Proline clients.
Flat CMOS battery	Some Proline clients were not in use for a while and the CMOS batteries went flat so the Proline thin client could not load the BIOS.
Dead NIC	The first step was to check if the network boot is enabled in BIOS. In case it was enabled but could not boot the client's components were used as parts for other Proline clients.

3. *Rewiring the lab:* Some power cables were rewired to allow the lab to function at full capacity.
4. *Power faults:* The distribution of electricity depends on the electricity providers. However, the use of power surge protector and UPS (Uninterruptible Power Supply) as discussed in Section 8.2.2 and 8.2.1. The



power surge protector protects the LTSP server from voltage spikes. The UPS allows the proper shutdown of the server machine. Ideally the LTSP server should shutdown immediately to protect the UPS battery.

### 6.3 Ngwane results

After the intervention the Ngwane computer lab was functional at full capacity. However, later on after a few months the server machine accumulated dust due to the challenging environment. This needed immediate attention and we were not in a position to come back to the Ngwane school. A solution was to help a person on site walk-through the process of cleaning and resitting the RAM module. The person on site had less technical skills, therefore a comprehensive manual was drafted to assist a person with less technical skills.

#### **A manual for cleaning resitting the RAM module**

This section intentionally uses trivial descriptions to suit a non-technical audience. In most cases, if the RAM is not properly placed the server produces a beeping sound during the booting process.

Before unscrewing the server, firstly, switch off the server machine properly and unplug it from the power sockets including other connected peripherals such as USB sticks. Secondly, get rid of static electricity by rubbing hands on conductive materials such as metal surfaces. This will prevent static electricity from damaging the RAM module. The following steps may slightly differ depending on the machine.

#### **1. Step 1**

Unscrew the server to be able to remove the lid.



FIGURE 6.3: Unscrewing the server.

**2. Step 2**

Gently pull the cover of the server.



FIGURE 6.4: Pull the cover until it is free.

### 3. Step 3

Remove the server's side panel.



FIGURE 6.5: Lift the cover up and place it aside

### 4. Step 4

Unplug the RAM module(s) from the slot(s).

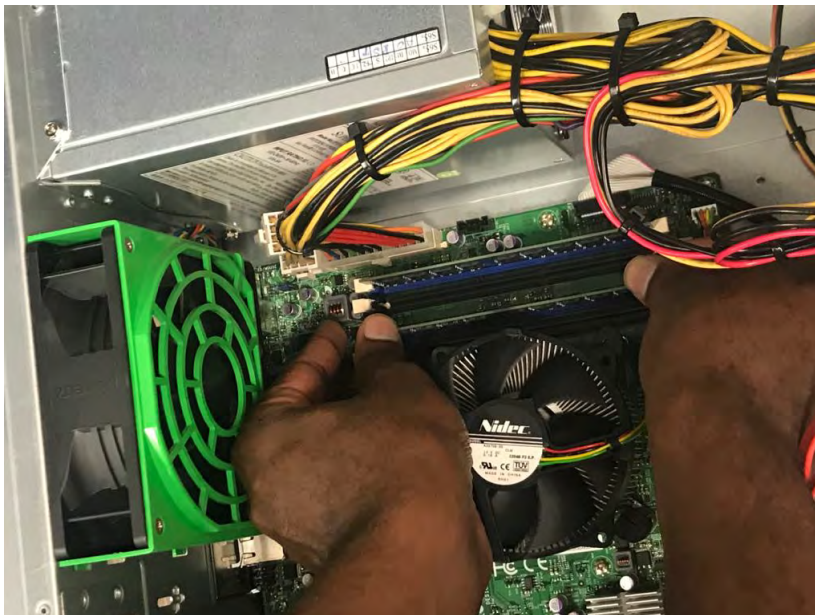


FIGURE 6.6: Apply pressure on the end clips in an outwards direction

**5. Step 5**

Take out the RAM module



FIGURE 6.7: Gently lift the RAM module by holding the ends

**6. Step 6**

Repeat Step 4, 5 and 6 for any available RAM modules.

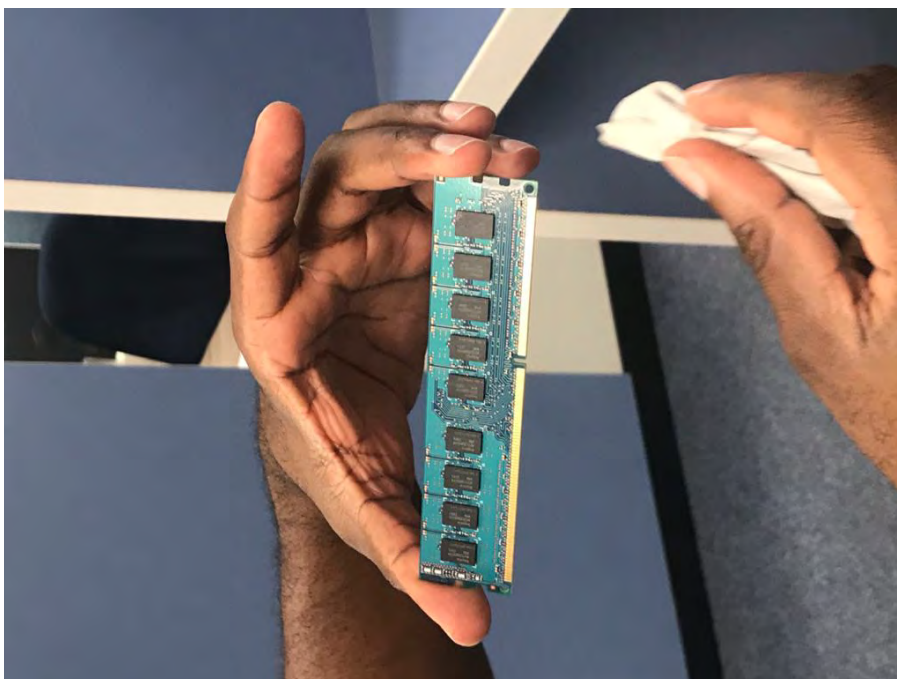


FIGURE 6.8: Use a piece of clean cloth that is not too dry or wet and gently clean the RAM module.

### 7. Step 7

The RAM module should be returned to the designated position where it properly fits.



FIGURE 6.9: Resitting the RAM module.

In case there are more than one RAM module, ensure that the RAM modules are inserted where the colour codes matches the RAM slots [119]. For example all RAM modules in blue first and then white or vice versa. This helps avoid the server downgrading the RAM modules to the lowest specifications available. Ensure the clips are open and gently push the RAM module down until it is properly inserted and close the clips going inwards. After you are done place the lid and push it inwards making sure it fits into the hinges.

### 8. Step 8

After you are done screw the lead , connect to power and test if RAM is working.

## 6.4 Mpume Junior High school

Mpume school has a secure computer lab that can accommodate over twenty thin clients. Figure 6.10 shows how the Mpume school lab is setup.



FIGURE 6.10: Mpume Junior Secondary School Computer lab

During the 2019 visit to the Ngwane school, the computing infrastructure was functional. However, power faults were interrupting the normal use of the lab and we had no control over it.

#### 6.4.1 Aim of the intervention

The aim was to evaluate and confirm that the computing infrastructure in the Mpume school was still functional and sustainable.

#### 6.4.2 Objectives

1. Test that all 23 thin clients are functional:  
This was achieved by powering on all the HP clients including checking that all the hardware peripherals such as LAN cables, keyboards and mice are properly connected.
2. Verify the health of the server:  
This was achieved by checking that the server boots and runs smoothly without errors. This includes checking that the applications like Libre-Office are functional.

### 6.4.3 Hardware specifications

The Mpume computing infrastructure had the following Dell server, HP clients and an Ethernet switch connecting the clients to the server. Table 6.4 and 6.5 details the hardware specifications.

#### Dell server

TABLE 6.4: Dell Server hardware specifications

Description	Dell Server
Product	X9SCL/X9SCM
Vendor	Dell
CPU	Intel(R) Xeon(R) CPU E3-1220 V2 @ 3.10GHz
RAM	24 GiB.
Disk	931GiB (1TB)
Free Memory	913.7 GB

#### HP T510 client

TABLE 6.5: HP client hardware specifications

Description	Hewlett Packard T510
CPU	VIA Eden X2 U4200 @ 1.00 GHz
RAM	1GB, 2GB
Disk	0GB (can be installed)

The Mpume HP thin clients had a clockspeed of 1 GHZ and 1GB of RAM. The HP thin clients were identical to the one experimented with in Section 4.4.

#### Ethernet switch

A Gigabit Ethernet switch was used to connect the HP clients to the Dell server.

### 6.4.4 Software specifications

The Mpume server uses Edubuntu 14.04 desktop environment. The Edubuntu desktop has an old version of LTSP that support only thin clients to boot from the server.

## 6.5 Mpume results

The Mpume lab was functional, all of the HP clients were working and the health of the Dell server was good. However, after a few months due to power cuts the server filesystem got corrupted. The solution was achieved by remotely assisting a person on site to run a file system consistency check CLI command *fsck*. The *fsck* CLI is used to check and attempt to repair the GNU/Linux filesystem [120]. The *fsck*, returns the exit code 0 if no errors were found, 1 if filesystem errors are corrected and 4 when the errors are not corrected [120].

## 6.6 Conclusion

The work packages in this chapter mainly were successful and reestablished the e-infrastructure in the Mpume and Ngwane schools. Both the Ngwane and Mpume schools did not need to be checked for CPU and RAM performance as they were already successful deployments. Most of the work packages were in Ngwane school and a few in the Mpume school. There still work to be done to protect the server machine better. It is vital to implement a UPS that automatically shuts down without human interaction. The methods and procedures to protect the server are discussed in Section 8.2.2. This chapter also further confirms that computer clients with inferior hardware or specifications work best as thin clients. In particular, the HP and Proline clients were specifically manufactured to be thin clients. This chapter also revealed the need to implement a sustainable strategy that allows remote assistance and logging of the LTSP machine. The methods include installing remote logging and monitoring tools such as Logwatch and Teamviewer. This depends on the availability of Internet connectivity, of course, and it can not be taken for granted. The sponsors of the SLL had a reorganisation, and as a result, the Ngwane and Mpume schools lost connectivity. This reveals the issues faced when the government does not run ICT4D projects. In particular, the Ministry of Education should be directly involved in ICT projects in schools. Chapter 8, will explore ways to improve LTSP computing infrastructure such as the ones visited in this chapter.



# 7 Laboratory Experiments:

## LTSP19+

After a successful deployment of LTSP5+ as reported in Chapter 5 and 6 it was necessary to compare the new version LTSP19+ versus LTSP5+. This chapter experiments with LTSP19+ and, most notably with its major novelty, the introduction of preparing and maintaining LTSP client images as virtual machines.

### 7.1 LTSP19+ background

In 2019, LTSP was redesigned and rewritten from scratch by Alkis Georgopoulos to improve on the previous version, LTSP5+ [82, 87]. Unlike the earlier versions of LTSP, LTSP19+ prioritised ease of maintenance rather than the possibility of using old hardware. This, together with other technological changes that currently favour thick client deployment, led to reduced thin clients support. Besides that, LTSP19+ supports new technologies like Wayland display server protocol, Systemd and UEFI firmware. The LTSP's CLI commands were redefined and simplified (shortened). Several CLI commands got deprecated, e.g. *ltsp-chroot*, *ltspfs\_mount*. Most important, perhaps, LTSP19+ now supports the maintenance of the LTSP client image through a virtual desktop environment (see Section 8.1.5).

### 7.2 Experiment 1: Using Mecer, HP, Proline clients

This section measures the performance of the LTSP19+ server by measuring the CPU, RAM and bandwidth usage.

#### 7.2.1 Experiment 1 Question

The experiment aimed to compare LTSP19+ and LTSP5+ and assess if future deployments can benefit from the LTSP19+. The comparison was made by

attempting to answer the following question:

- How does LTSP19+ perform compared to LTSP5+ ?

## 7.2.2 Hypothesis

1. An LTSP19+ thick-client computing infrastructure offers better performance than the older LTSP5+ thick -client computing infrastructure.

## 7.2.3 Hardware specifications

A dell server used LTSP19+ and a SuperMicro server used LTSP5+. Both servers were in the same class. In particular, a CPU Score comparison from [114], showed that the SuperMicro had a CPU score of 3760, and the Dell server had 3771 CPU score. The clients, excluding the Intel NUC, were the same used tested in Chapter 4. The experiments used the following hardware Dell server, Mecer, Proline Nanoware, HP client and Intel NUC clients. Table 7.1 - 7.6 outlines the hardware specifications in details.

### Dell server

TABLE 7.1: Dell Server hardware specifications

Product	PowerEdge T30 (07C5)
Vendor	Dell Inc.
CPU	Intel(R) Xeon(R) CPU E3-1225 v5 @ 3.30GHz.
RAM	8GiB.
Disk	D931GiB (1TB)
Free Memory	913.7 GB

### Mecer Xtreme Jupiter Mini PC client

TABLE 7.2: Mecer Mini PC hardware specifications

Description	Mecer Xtreme Jupiter Mini PC
Model	Jupiter-h110
CPU	Support 6th and 7th Generation Core i7 Skylake and Kaby Lake CPU [i7-6700K / i7-7700K]
RAM	4GB DDR4-2400, (2 x 288PIN DDR4 DIMM Slot, Upgradable to 32GB).
Disk	0GiB (can be installed)

## Proline Nanoware client

TABLE 7.3: Proline Nanoware hardware specifications

Description	Proline Nanoware
CPU	Intel (R) Core(TM) 2 Duo CPU E7500 @ 2.93 GHz Intel 64 Architecture
RAM	1GB, 2GB
Disk	320GiB

## HP T510 client

TABLE 7.4: HP client hardware specifications

Description	Hewlett Packard T510
CPU	VIA Eden X2 U4200 @ 1.00 GHz
RAM	2GB
Disk	0GB (can be installed)

## 7.2.4 Software specifications

The experiment used a combination of stable versions of Ubuntu Server and Xubuntu, version 20.04 LTS. The Glances tool recorded the server performance as well as the load on the thick clients.

## 7.2.5 Experiment 1 procedures

The LTSP19+ computing infrastructure was setup to emulate a lab but on a smaller scale while pushing the network to its limits as it may occur in real life. The experiment used HP T510, Proline Nanoware and two Mecer Mini PC. Even though Chapter 4 and 6 established that the HP and Proline clients work best as thin clients. This chapter focused on designing an almost identical to the LTSP5+ experiments in Chapter 4 that use the same clients. The client computers connected to the server machine via an Ethernet Gigabit D-Link switch. Experiment 1 used a dual NIC network configuration. The experiments procedures activated the following application:

1. Firefox Browser was streaming two videos simultaneously.
2. LibreOffice writer

The CPU usage by applications such as streaming videos is volatile. As an attempt to be more precise, the recordings are averages of 3 sets of results.

## 7.2.6 LTSP19+ server results

Table 7.5 shows the results of computing resources usage by the Dell server.

TABLE 7.5: Server thick clients computing

Number of Clients	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
0	4.10	25.40	1.00KB	2.00B
1	4.30	26.10	178.00KB	4.27KB
2	4.60	26.10	968.00KB	1.05MB
3	5.10	28.30	3.32MB	3.62MB
4	7.3	27.00	1.52MB	2.98MB

### A comparison of LTSP19+ versus LTSP5+ CPU and RAM usage

This section makes a direct comparison of LTSP19+ and LTSP5+ (see Table 4.6) results.

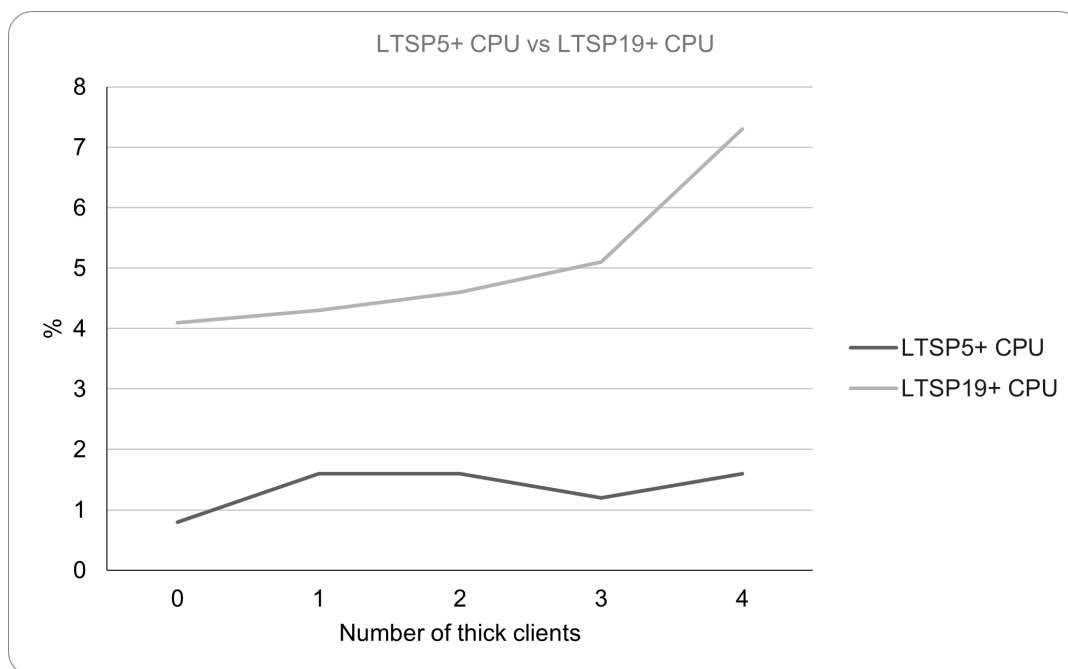


FIGURE 7.1: A comparison of server CPU usage for LTSP5+ and LTSP 19+

The CPU usage for zero clients was 0.8% for LTSP5+ and 4.1% for LTSP19+. By factoring out the difference in zero client CPU usage, we can observe that the CPU demand rose with almost the same factor. However, the LTSP19+ CPU spiked higher than expected when the fourth client was added. This probably due to the differences of the two CPU processors. The SuperMicro and Dell server machines belonged to the same class, i.e. Intel(R) Xeon(R)

CPU E3-1220 and Intel(R) Xeon(R) CPU E3-1225, respectively. Figure 7.2 gives a detailed comparison of the servers used.

FIGURE 7.2: SuperMicro vs Dell server benchmark. Source [114]

	Intel Xeon E3-1220 @ 3.10GHz	Intel Xeon E3-1225 @ 3.10GHz
Price	\$59.99 <b>BUY NOW!</b>	\$213.19 <b>BUY NOW!</b>
Socket Type	LGA1155	LGA1155
CPU Class	Server	Server
Clockspeed	3.1 GHz	3.1 GHz
Turbo Speed	Up to 3.4 GHz	Up to 3.4 GHz
# of Physical Cores	4 (Threads: 4)	4 (Threads: 4)
Max TDP	80W	95W
Yearly Running Cost	\$14.60	\$17.34
First Seen on Chart	Q2 2011	Q1 2011
# of Samples	193	262
Cross-Platform Rating	9353	8749
Single Thread Rating	1563	1560
CPU Mark	<b>3760</b>	<b>3771</b>

LTSP-Wiki Debian [94], recommends a minimum of 2120 CPU Mark (CPU Score) for four thick clients. The SuperMicro had a CPU score of 3760, and the Dell server had 3771, which was more than enough. Even though there is no much difference in CPU performance, the Dell server was expected to perform a bit better than the SuperMicro. Factoring out the initial server loads, we suggest that both LTSP19+ and LTSP5+ use roughly the same CPU.

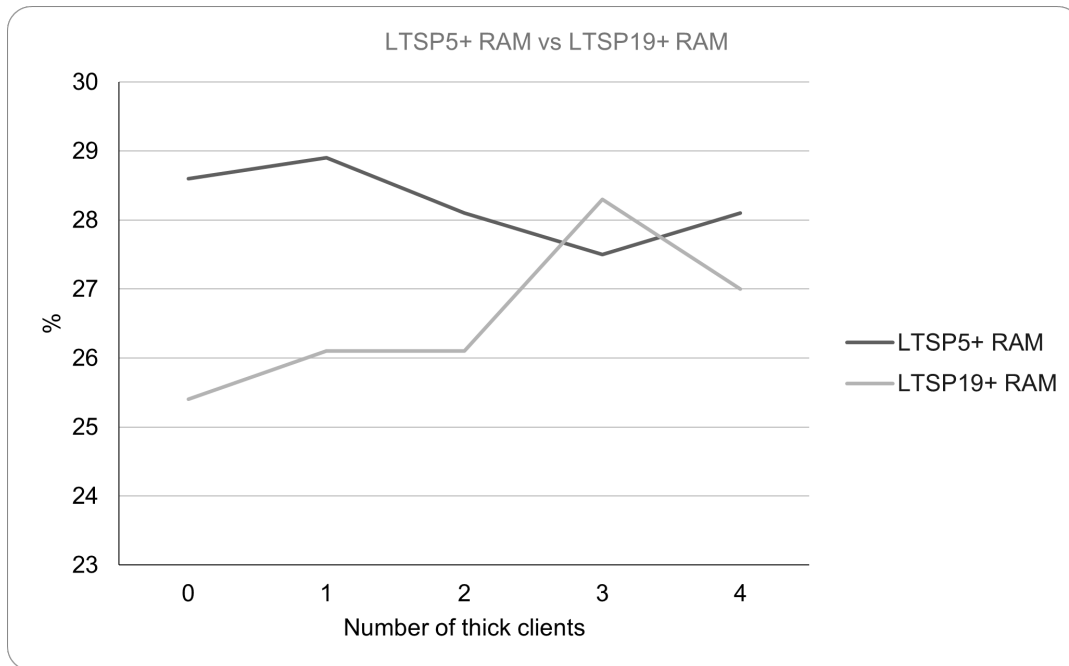


FIGURE 7.3: A comparison of server's RAM usage for LTSP5+ and LTSP 19+

Both LTSP5+ and LTSP19+ servers used 8GB of RAM. Most often CPU usage is directly proportional to memory consumption. Interestingly the LTSP19+ server used less RAM than LTSP5+ regardless of the LTSP19+ server having higher CPU usage than LTSP+. As discussed earlier in Section 3.2.2, LTSP19+ reduced its memory footprint. Probably because LTSP19+ used recent technologies with better implementation and a smaller footprint. LTSP19+ performed way better than LTSP5+ in terms of RAM consumption. This improvement may potentially reduce the RAM requirements for future LTSP deployments, especially for thin-client computing.

#### A comparison of LTSP19+ versus LTSP5+ LAN bandwidth

Figure 7.4 and 7.5 compares the LAN bandwidth usage for LTSP5+ and LTSP19+ servers.

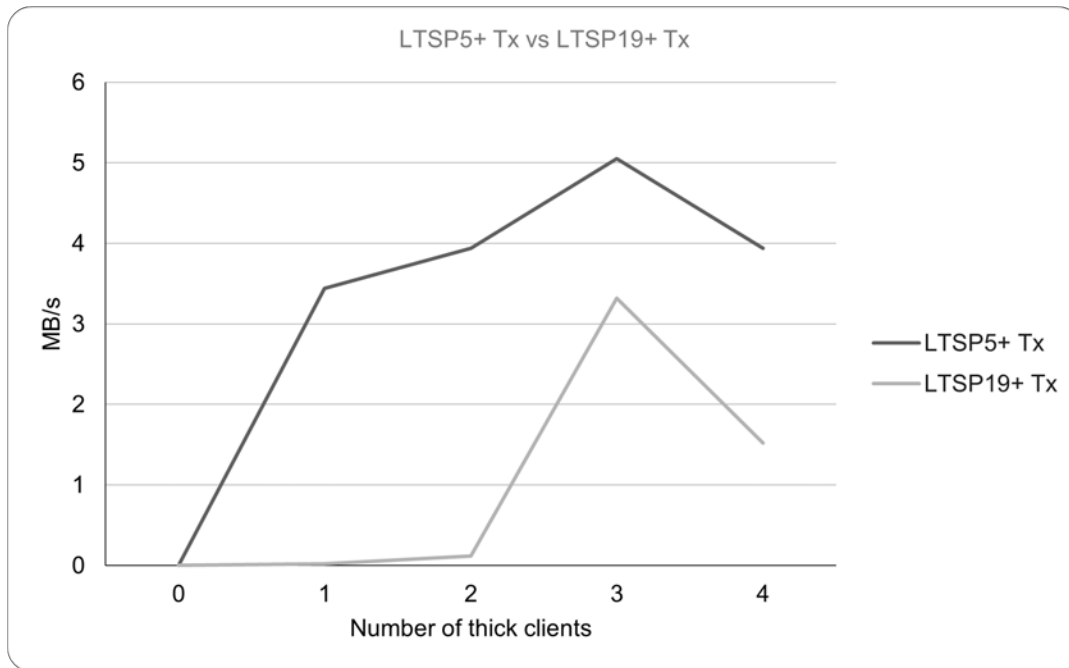


FIGURE 7.4: A comparison of packets transmitted to clients by the server for LTSP5+ and LTSP 19+

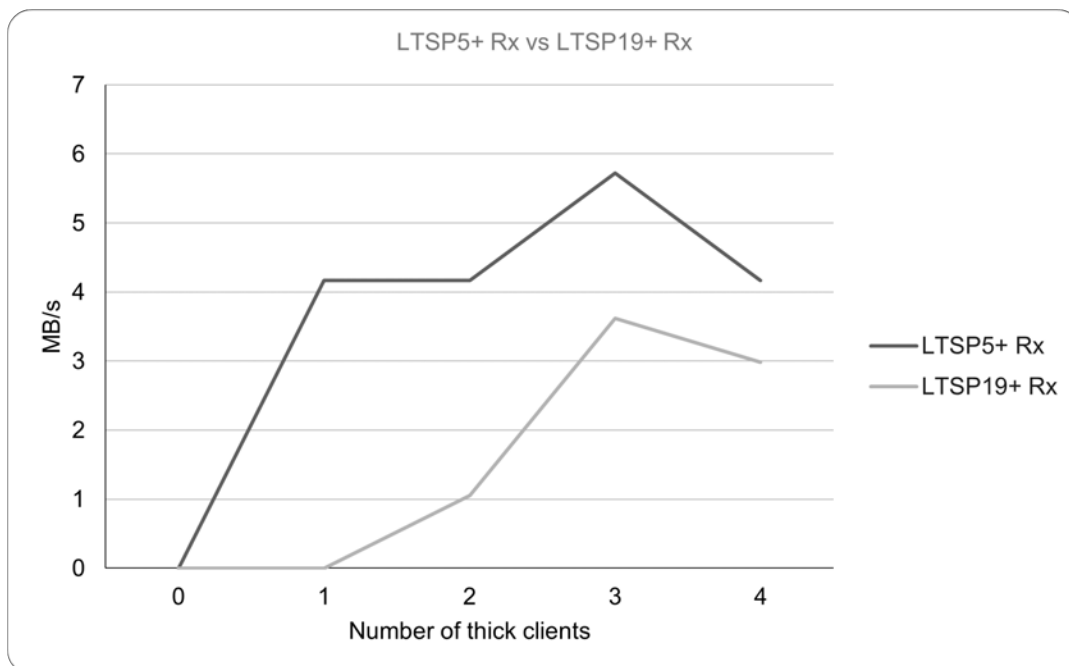


FIGURE 7.5: A comparison of LAN bandwidth usage to support clients for the server for LTSP5+ and LTSP 19+

The results depends on the following factors A comparison of bandwidth usage received and transmitted depends on several factors. In general, LTSP19+ used less bandwidth than LTSP5+.

### 7.2.7 Results

The overall LTSP19+ performance was better than LTSP5+. Notably, LTSP19+ RAM usage is lower than LTSP5+. However, the CPU usage is slightly higher considering that the LTSP19+ server's CPU had a higher CPU Score.

## 7.3 Experiment 2: Using the Intel NUC clients

Experiment 2 followed the same experimental procedures as Experiment 1. Figure 7.6 shows an LTSP19+ computing infrastructure supporting five Intel NUC thick clients using a single NIC configuration.

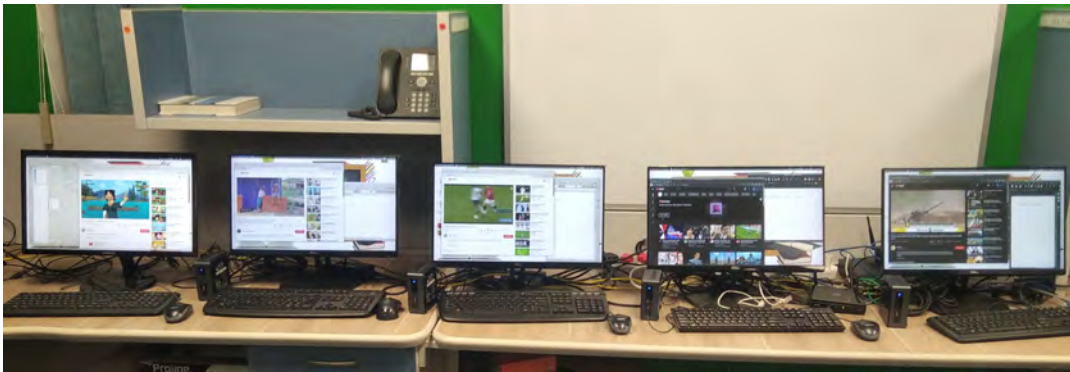


FIGURE 7.6: Five Intel clients on a single NIC configuration.

### 7.3.1 Experiment 2 Questions

The two questions the experiment attempted to answer were:

1. What is the impact of the Intel NUC clients on the server?
2. How do the Intel NUC clients impact on the server compare to Mecer clients deployed in the Ntsika school?

### 7.3.2 Hypotheses

1. The Intel NUC clients perform well on LTSP19+ thick-client computing infrastructure.
2. The Intel NUC performs similarly to Mecer clients deployed in the Ntsika school.



### Dell server

The Dell server was identical to the server used in Experiment 1 (see Table 7.1).

### Intel NUC client

The Intel NUC clients are more recent hardware and the SLL intends to deploy them in upcoming deployments.

TABLE 7.6: Intel NUC J4005 Mini PC

Description	Intel NUC Mini PC
Model	J4005
CPU	Intel Celeron Processor 2.00GHz
RAM	4GB
Disk	0GiB (can be installed)

### 7.3.3 Software specifications

The experiment used a combination of stable versions of Ubuntu Server and Xubuntu, version 20.04 LTS. The Glances tool recorded the server performance as well as the load on the thick clients.

### 7.3.4 Experiment 2 procedures

The experiment used five Intel NUC clients donated by SGT Solutions. The Dell server and the Intel NUC hardware are part of SLL upcoming deployments. Experiment 1 and 2 used the same procedures except that Experiment 1 used a dual NIC configuration and Experiment 2 used a single NIC configuration. The client computers connect to the server machine via an Ethernet Gigabit D-Link switch. The experiments procedure activated the following application:

1. Firefox Browser was streaming two videos simultaneously.
2. LibreOffice writer

The CPU usage by applications such as streaming videos is highly variable so the recordings are averages of 3 sets of results.

TABLE 7.7: Server thick clients results

Number of thick clients	CPU%	Memory%	Transmitted <sub>/s</sub>	Received <sub>/s</sub>
0	0.80	24.50	336.10B	24.08B
1	1.00	24.90	384.00KB	49.20KB
2	1.10	24.80	796.18KB	2.08MB
3	1.20	25.00	7.22MB	6.88MB
4	1.70	25.40	7.89MB	1.80MB
5	3.10	26.20	6.93MB	2.67MB

### 7.3.5 LTSP19+ server results

Table 7.7 shows how the LTSP server performed when serving the five clients. The CPU and RAM usage are promising and Figure 7.7 compares the four Intel NUCs with the Mecer deployed in the Ntsika school (see Chapter 5).

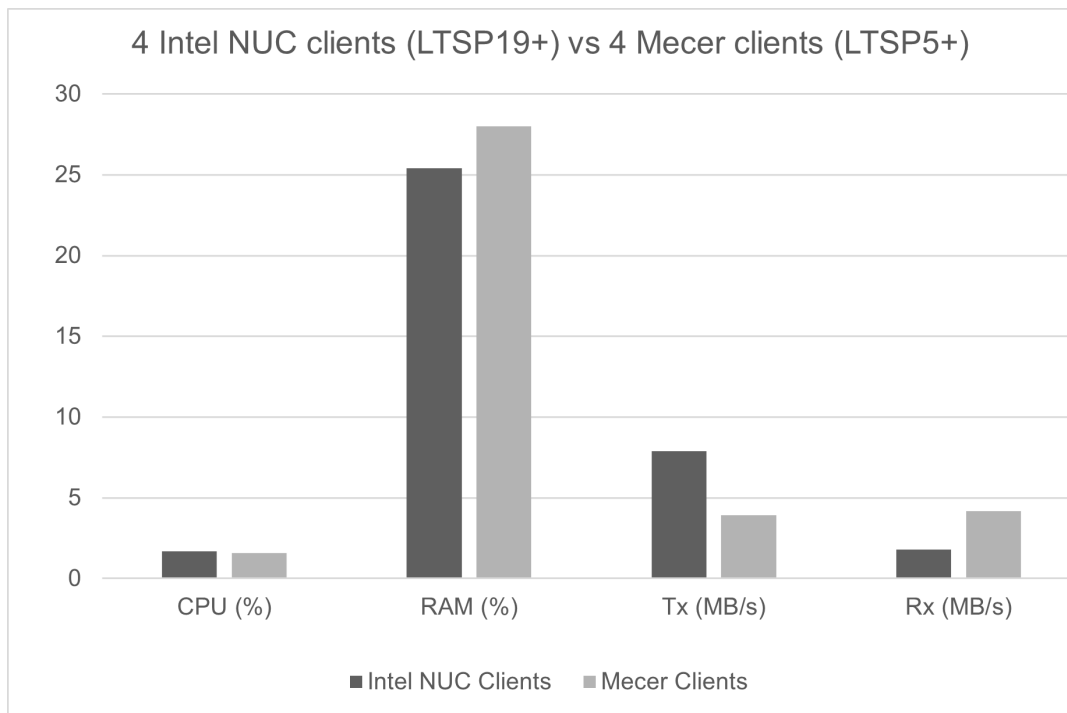


FIGURE 7.7: Intel NUC vs Mecer clients.

Figure 7.7 shows how the Intel NUC clients on LTSP19+ computing infrastructure compares to Mecer clients on LTSP5+ computing infrastructure. The two clients are in a similar class, but the Mecer client has a better processor.

- CPU

Both experiments used an identical Dell server. The LTSP5+ server CPU usage was slightly better than LTSP19+ server. The difference in CPU usage was 0.1%.

Earlier in Experiment 1 (see Section 7.2.6), we observed that the LTSP5+ Supermicro server performed better than the LTSP19+ Dell server. However, in Figure 7.7 the CPU usage was almost the same. The results confirm that the high CPU usage was mainly due to the difference in CPUs. Both LTSP5+ and LTSP19+ have nearly the same impact on the CPU.

- **RAM** Both LTSP5+ and LTSP19+ servers used 8GB of RAM. LTSP19+ server had better RAM usage than LTSP5+ server. As seen earlier in Experiment 1, the LTSP19+ server also had better RAM usage. The better RAM usage is primarily because LTSP19+ is implemented better on the software level.
- **Bandwidth** LTSP19+ server used more bandwidth than LTSP5+ to transmit and less bandwidth to receive packets. Again this depends on the different demands for bandwidth from the clients.

### 7.3.6 Results

The overall LTSP19+ server performance was better than LTSP5+. Notably, LTSP19+ RAM usage is lower than LTSP5+. Both clients performed well, and it seems they have a similar impact on the server. However, the Intel NUC clients was a bit slower compared to the Mecer clients. The Intel NUC clients were slower because of the inferior processor. The Intel NUC results are promising, considering that the Mecer clients gave similar results and were used to deploy a successful computing infrastructure in the Ntsika school (see Chapter 5). Scaling the Intel NUC clients to 27 as in our use case seems feasible.

### 7.3.7 LTSP19+ Adjustments

This section will discuss some of the adjustments required to install LTSP19+ successfully.

#### Missing Kernel on image build

The GNU/Linux kernel is a small program and the main component of the operating system that manages device drivers, system calls, memory and processes [121]. The kernel program is loaded first into the client's RAM before the LTSP server sends the rest of the clients' image files (Section 3.2.1 discussed how LTSP works). If the GNU/Linux OS hosting the LTSP installation is on a separate boot partition, the image building process will miss

some files. The error *'Could not locate vmlinuz and initrd.img'* is reported. The *vmlinuz* is a compressed and bootable GNU/Linux kernel located in (*/boot*) directory, a directory where booting files are located [122]. The *vmlinux* is the kernel state before *vmlinuz*, the *vmlinux* kernel is not bootable, and uncompressed kernel [122]. The initial RAM disk (*initrd*) allows the boot loader to load a RAM disk that is mounted as root filesystem [123]. The solution is to tell the *ltsp image* or *ltsp image x86\_64* CLI commands that there is a separate boot partition [124]. The solution is to run the following CLI commands:

```
ltsp image /,/boot,subdir=boot
```

or

```
ltsp image x86_64,/boot,subdir=boot
```

The CLI *ltsp image x86\_64* depends on the system architecture if the GNU/Linux is on a 32-bit system use *ltsp image x86\_32*.

### Users listed on client login screens

Listing users on the Login screen is not ideal, especially in an LTSP computing environment with multiple users. Even though it is not a huge issue, the issue is solved by editing the LTSP configuration file */etc/ltsp/ltsp.conf*. The adjustments are as follows:

#### [clients]

```
LIGHTDM_CONF="greeter-hide-users=true"
```

```
GDM_CONF="greeter-hide-users=true"
```

For more information about LTSP configuration refer to the Appendix B.

## 7.4 Conclusion

Experiment 1 compared LTSP19+ with LTSP5+ by using similar hardware and procedures as in Chapter 4. The result was that LTSP19+ has a smaller footprint on the server compared to LTSP5+. Even though LTSP19+ reduced the support of thin clients, lower RAM consumption would benefit thin-client deployments. Experiment 2 tested the Intel NUC clients on a smaller scale to prepare for upcoming SLL deployments. The Intel NUC clients had similar hardware specifications as the Mecer clients used in the Ntsika school but a faster CPU. Comparing the two clients showed that Intel NUC clients had almost similar performance with four Mecer clients. In particular, the

Intel NUC on LTSP19+ computing infrastructure showed better RAM usage than Mecer clients on LTSP5+ computing infrastructure. This chapter recommends using LTSP19+ for thick-client deployments, primarily due to its active support. However, LTSP5+ would be recommended has better support for thin clients; therefore, LTSP5+ would be recommended for thin clients deployment. LTSP19+ comes with improvements such as maintaining the client's image as virtual machines, as discussed in the next chapter.

## 8 Making deployment and maintenance easier

Easy deployment and maintenance of LTSP computing infrastructure are key to sustainability. This chapter explores ways to simplify the deployment process and to improve the sustainability of the LTSP computing infrastructure. This chapter also explores ways that may enhance the experience of the LTSP computing infrastructure experience.

### 8.1 Making deployment easier

This section seeks to ease the deployment process and the management of an LTSP computing infrastructure. An LTSP computing infrastructure that is simple to manage or deploy will encourage active involvement from a person with limited technical skills, e.g. educators.

#### 8.1.1 An LTSP ready ISO image

An ISO image contains all files and has a file system of an optical disc, and all files are identical as they would appear on an optical disc [125]. An LTSP ready ISO image is an ISO file with LTSP and OS files remastered into a single ISO file (.iso). Using tools for creating USB bootable devices, e.g. Rufus, the LTSP ready ISO file can be redistributed as bootable USB sticks. Several GNU/Linux OS remastering tools are available, e.g. Distroshare, Cubic, PinguyBuilder. The following section describes the PinguyBuilder remastering tool.

#### **PinguyBuilder Ubuntu-based customiser**

PinguyBuilder comes in either CLI or GUI mode. Figure 8.1 shows the PinguyBuilder menu and the options provided.

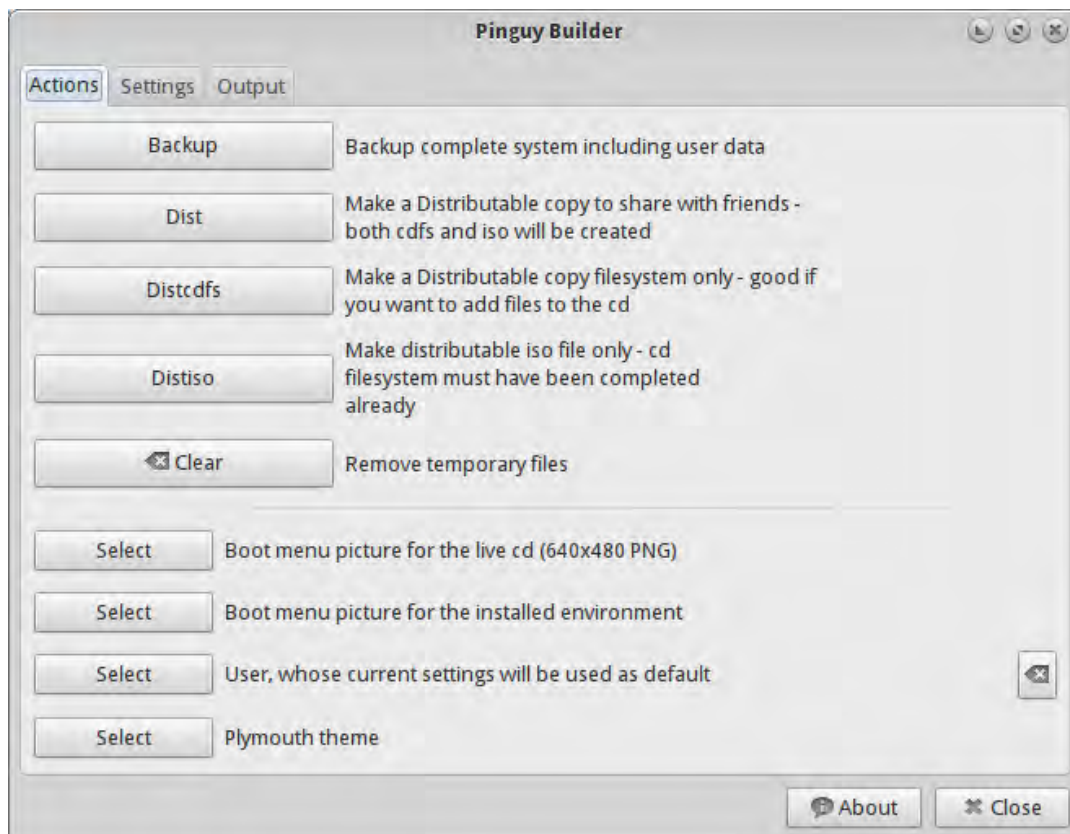


FIGURE 8.1: A GUI-based PinguyBuilder. Source [126]

### Benefits of an LTSP ready ISO image

An LTSP ready ISO image has been naturally tested and generally work as soon as deployed since LTSP and other non-system applications come already installed and configured.

### Drawbacks of an LTSP ready ISO image

Due to architecture incompatibility, the LTSP ready ISO file might fail to install. Most remastering tools use methods (level of interchange) that limits the ISO 9660 filesystem to about 4GiB limit making it impossible to build an ISO image larger than 4.2GB ISO image [127, 128]. An LTSP installation keeps typically two copies of the client image (`/var/ltsp/images/`), i.e. the current and old image. Removing one client image copies will help free up space.

### 8.1.2 Live USB sticks

A live GNU/Linux environment allows a user to install an OS on a USB or disc and run the OS and applications without installing the OS on the

harddrive [129]. Most GNU/Linux OS have an option to run and test the OS without installing it. In our case, the SLL schools can temporarily use USB sticks to boot the LTSP clients in live GNU/Linux OS environments when the LTSP server is unavailable. A single NIC network configuration allows the LTSP clients to connect to the Internet without relying on the LTSP server (see Section 3.10). For this reason, this section recommends deploying the LTSP computing infrastructure using a single NIC network configuration.

Also, the PinguyBuilder may help customise the live GNU/Linux OS environment. LTSP users can continue working using live USB bundled with customised applications, e.g. Tux software packages. However, live USB sticks add to the additional cost of the LTSP computing infrastructure.

### 8.1.3 Offline application installation

Offline application installation allows to load custom applications on a USB stick or disc and install them later without relying on Internet connectivity. Offline applications are ideal when deploying the LTSP computing infrastructure in areas with limited or no connectivity which from experience are far from rare. Offline applications will also reduce connectivity costs and be used for application re-installation or to fix broken packages. CubeGet is a portable package manager that makes offline installation easy [106]. Figure 8.2 shows CubeGet web-server downloading application files for later installation.



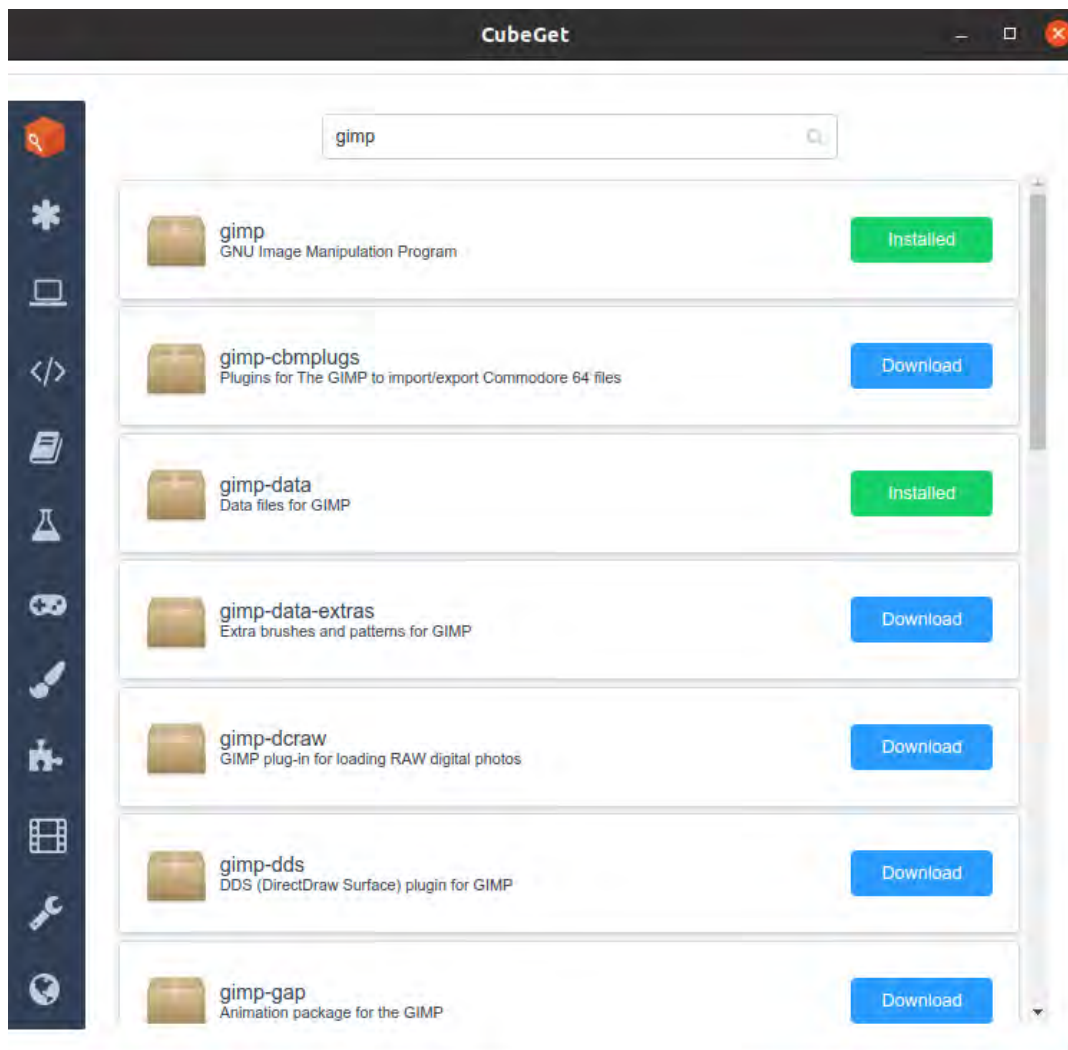


FIGURE 8.2: CubeGet a portable package manager. Source[106]

After installing the CubeGet, a web server launches, and the user can search and download their applications. CubeGet uses an extensive database, and most applications are available [106].

### 8.1.4 Supporting software tools

Supporting software tools assists with the management of the LTSP computing infrastructure.

### 8.1.5 LTSP19+ client image virtual machines

As discussed in Section 3.5, LTSP19+ added the option of preparing and maintaining the client's image as a virtual machines. Virtualisation player

will help a person with limited technical skills manage applications and prepare a new LTSP client's image. In our case, most of the teachers in SLL schools mostly have limited technical skills, and they cannot use a CLI. Teachers in SLL will prefer to maintain their ICT teaching environment using a graphical tool. Maintaining the LTSP' client's image as a VM promotes a more functional computing infrastructure that can transform into revenue-generating platforms.

### **User credentials management**

A GUI environment simplifies LTSP user management. The LTSP-manager application is one of the GUI-based applications specifically designed to manage LTSP users. Using the LTSP-manager, an administrator can create, edit and most importantly, import users from a CSV (comma separated values) file. The LTSP-manager also helps manage user permission by assigning users to groups, e.g. teachers, students, administrators etc. Figure 8.3 shows the LTSP-manager application showing users categorised into different groups, students, teachers and administrators.

FIGURE 8.3: An example of LTSP-manager tool managing user credentials

LTSP Manager						
File	View	Groups	Users	Help		
GID	Group name	UID	Username	Primary group	Real name	Office
1002	students	1000	tichzvidz	tichzvidz	Tichaona Zvidzayi	
1007	teachers	1002	administrator	administrator	ADMINISTRATOR	
1001	administrators	1003	admin2	admin2	ADMIN2	
		1004	tux1	teacher	Tux 1	
		1005	tux2	tux2	Tux 2	
		1006	tux3	tux3	Tux 3	
		1007	tux4	tux4	Tux 4	
		1008	tux5	tux5	Tux 5	
		1009	tux6	tux6	Tux6	
		1010	tux7	tux7	Tux 7	
		1011	tux8	tux8	Tux 8	
		1012	tux9	tux9	Tux 9	
		1013	tux10	tux10	Tux 10	
		1014	tux11	tux11	Tux 11	
		1015	tux12	tux12	Tux 12	
		1016	tux13	tux13	Tux 13	
		1017	tux14	tux14	Tux 14	
		1018	tux15	tux15	Tux 15	
		1019	tux16	tux16	Tux 16	
		1020	tux17	tux17	Tux 17	
		1021	tux18	tux18	Tux 18	
		1022	tux19	tux19	Tux19	
		1023	tux20	tux20	Tux 20	
		1024	tux21	tux21	Tux 21	
		1025	tux22	tux22	Tux 22	
		1026	tux23	tux23	Tux 23	
		1027	tux24	tux24	Tux 24	
Unselect all		1028	tux25	tux25	Tux 25	

Figure 8.3 left column shows the groups whilst the right column shows the user credentials. Note that the primary group is not the group name. Among other features, the LTSP-manager allows new users to register from the LTSP

clients. The drawback of LTSP-manager is that it requires administrative permissions, and for example, a teacher without administrative rights will not be able to add a new student.

### **User session monitoring**

User session monitoring is helpful in learning environments because it gives educators or lab assistants more control of their environment. There are several applications to monitor user sessions, e.g. Veyon (formerly iTalc), Epopetes etc. This section will explore Epopetes, the same application used in previous chapters. Epopetes is a Greek word meaning overseer, is a free and open-source software application used to monitor and manage a computer lab [54]. It allows real-time features such as logging off, shutdown, rebooting, screen broadcasting, sending messages, sound muting etc. The SLL schools can use Epopetes to improve their learning environment by remotely assisting students and broadcasts their sessions for demonstrations. Figure 4.10 shows how an educator uses Epopetes to assist a student (username tux5) to send an email.

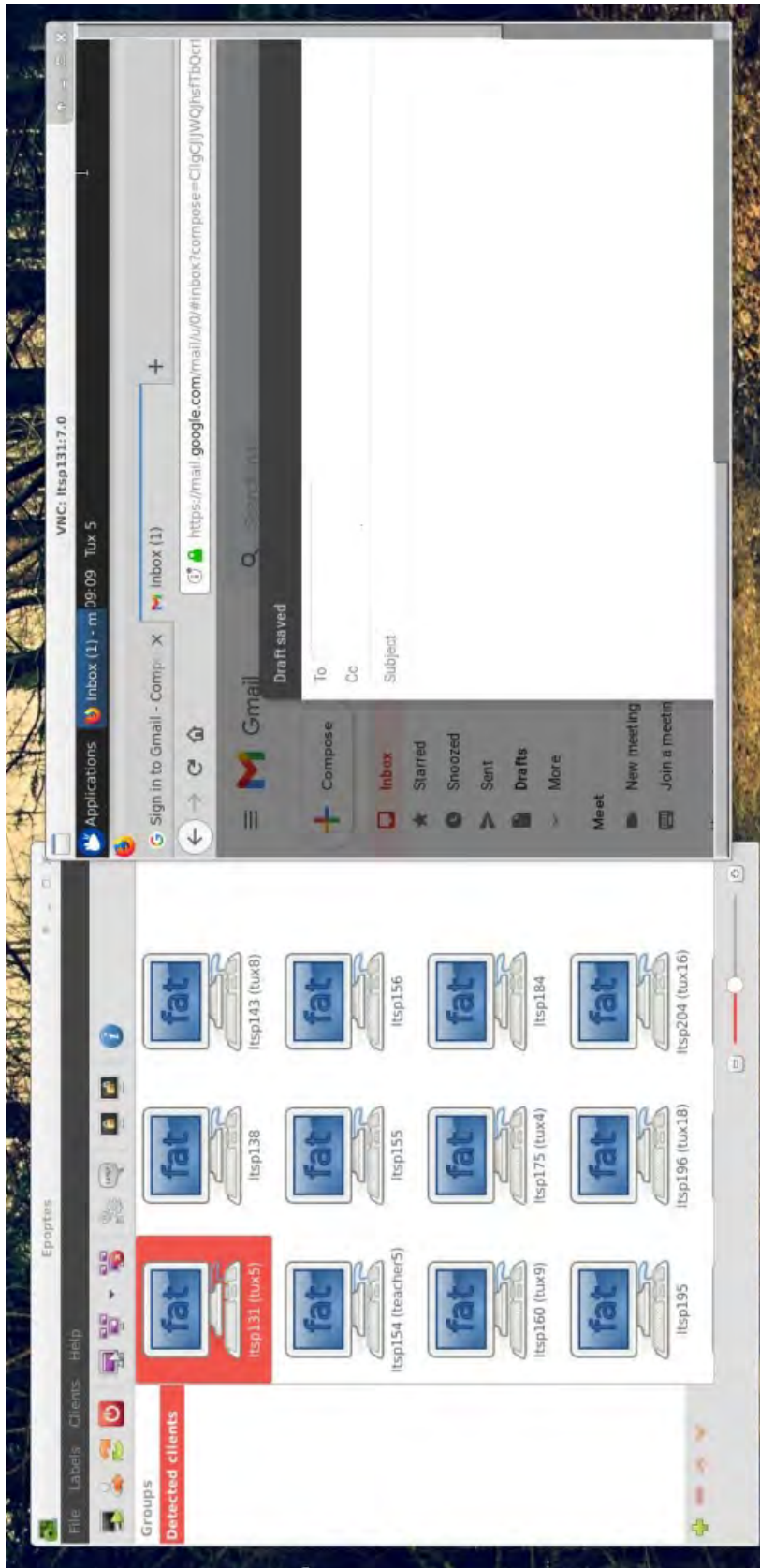


FIGURE 8.4: An example of an educator assisting a student via Epoptes

Epopetes can be operated by a non-administrator person like educators by using the command below.

```
gpasswd -a administrator epoptes
```

As mentioned in Figure 2.4, teachers from the Greek schools used Epoptes to monitor and assist students. Epoptes is ideal for teaching environments because some of the Epoptes developers are IT educators who have experience in learning environments [54]. The drawback of using Epoptes, as seen in previous experiments, is that it requires server resources and might negatively impact an LTSP computing infrastructure, especially an LTSP thin-client computing infrastructure.

## 8.2 Maintenance

LTSP servers typically will run into problems at some point during their lifetime. This section proposes methods that might mitigate these problems and so achieve a more sustainable deployment.

### Potential LTSP server challenges

From experience these are some of the typical problems that the LTSP server might encounter during its lifetime.

- *Power surges:*  
Voltage surges are untimely rises in voltage that can damage electrical devices [130]. Some of the hardware components that are damaged by spikes are motherboard, disk and power supplies etc.
- *Power cuts:*  
Power cuts are a loss of power caused by a problem in the infrastructure or service providers. Power cuts risk loss or corruption of data.
- *Dust collection:*  
The LTSP server collects dust as time goes on. Dust collection may cause the server's components to malfunction, e.g. RAM, disk and motherboard.
- *Disk failure*  
When a disk failure happens all LTSP files including user data are lost and a new disk is required. Some of the reasons for disk failures are mechanical failures, power surges and overheating [131].

- *Motherboard failure*

The most common cause of motherboard failure are overheating and power surges. In our case if the motherboard fails there is nothing we can do but to replace the LTSP server.

### 8.2.1 Power surge protector

Voltage surges are caused by many reasons such as bad wiring, lightning, instabilities in the feeding power network and re-establishment of power after power cuts. A surge protector reduces the chances that the LTSP server gets damaged by redirecting extra voltages. One drawback of surge protectors is that they do not last forever and need to be replaced at some point.

### 8.2.2 Uninterruptible Power Supply

A UPS (Uninterruptible Power Supply) provides backup power during power cuts or when voltage drops below a usable level [132]. Note that in our context, LTSP deployment, our interest is not provide backup power but simply to have enough time for a 'clean' shutdown in order to protect the integrity of the filesystem as well as make the battery in the UPS have the longest possible life. A UPS battery uses direct current (DC) and converts it to alternating current (AC) to allow the connected devices to use it. UPSes fall into the following categories standby, line-interactive and double-conversion.

- A standby (offline) UPS switches to battery when power fails or when the voltage dips (surges) [133, 132]. The offline UPS device functions by switching to battery power and redirecting the power to the connected devices. A standby UPS has a simple and least expensive implementation than other UPSes. A standby UPS is ideal for SLL schools.
- A line-interactive UPS includes the capability to correct minor voltage fluctuations without switching to battery [132]. The connected devices never receive power from the AC outlets but from the UPS itself. A line-interactive UPS can also be used as a backup battery when the power is cut off.
- A double-conversion (online) UPS provides near perfect power by converting incoming AC power to DC and back to AC [134, 132]. The online UPS is always running off the battery whilst the battery is charging by the power line [134]. Even though online UPS are inefficient because

of heat lost during conversions, there are ideal for protecting mission-critical equipment. a double-conversion UPS is most expensive UPS type.

In the LTSP context the first two options are reasonable, the third is too expensive. If a surge protector is already available then a standby UPS is the correct choice.

### **Network UPS tools**

The Network UPS Tools (NUT) are software tools that provide support for power devices, UPS, power distribution units, and solar controllers [135]. NUT is a cross-platform software supporting GNU/Linux, Windows, BSD, Apple OS, Solaris etc. NUT supports serial, USB and network as communication channels. NUT has a client-server model whereby the server monitors the UPS status and gives feedback to the client on request [136]. NUT tools help the UPS to perform automatic tasks such as safe shutdown in case of power cuts [137]. As shown in Figure 8.5 NUT exists in three layers which are drivers, a server and clients.



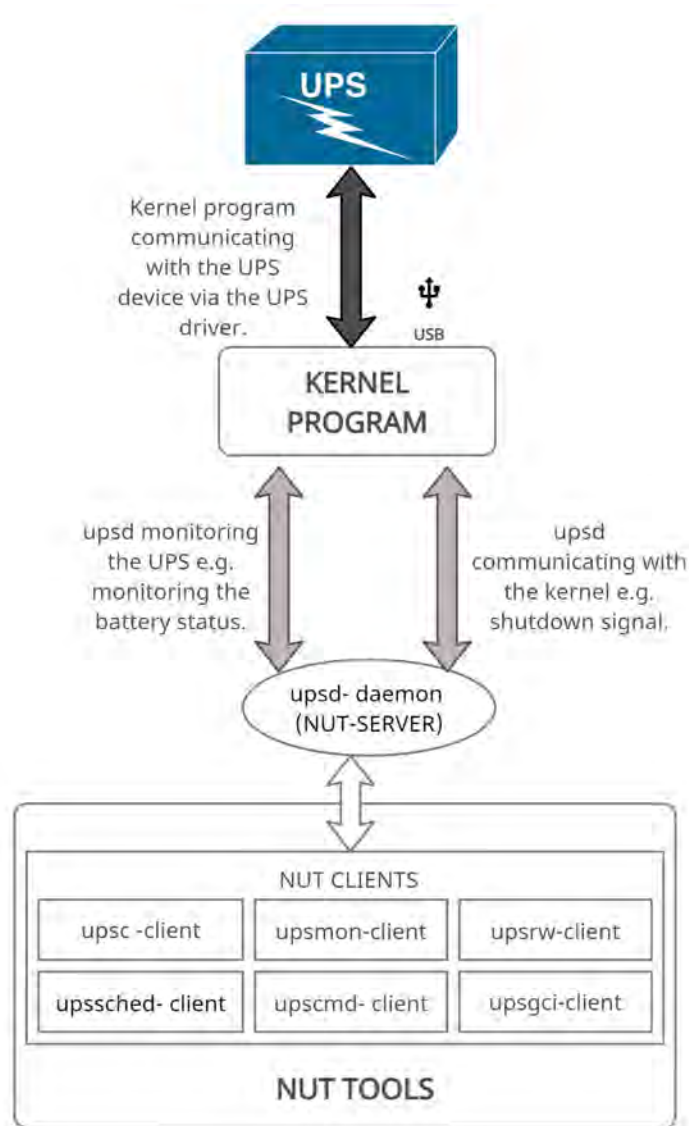


FIGURE 8.5: An architectural view of how the NUT tools communicate with the UPS.

Drivers enable the kernel component of the OS to communicate with the UPS. Since a UPS does not support all drivers, the user should choose the correct one in the NUT configuration files. The server layer implemented by the `upsd` (UPS-daemon) shares the UPS' information with NUT clients on the host or over a network. A daemon is a background process that often implements servers [138]. NUT clients can monitor and manage the UPS and for example, shutdown a host. The NUT clients include `upsc`, `upsw`, `upscmd`, `upsmon`, `upssched`, `upsgci` etc:

- **upsc** (UPS-client) provides a quick and easy way to see the status of the server[139]. Figure 8.6 shows an example of the output after running

the `upsc` command.

```
tichzvidz@ltspserver2:~$ upsc fry
Init SSL without certificate database
battery.charge: 85
battery.voltage: 12.60
battery.voltage.high: 13.00
battery.voltage.low: 10.40
battery.voltage.nominal: 12.0
device.type: ups
driver.name: blazer_usb
driver.parameter.pollinterval: 2
driver.parameter.port: /dev/ttyS0
driver.parameter.synchronous: no
driver.version: 2.7.4
driver.version.internal: 0.12
input.current.nominal: 3.0
input.frequency: 0.0
input.frequency.nominal: 50
input.voltage: 8.2
input.voltage.fault: 228.5
input.voltage.nominal: 230
output.voltage: 232.9
ups.beeper.status: enabled
ups.delay.shutdown: 30
ups.delay.start: 180
ups.load: 1
ups.productid: 0000
ups.status: 0B
ups.type: offline / line interactive
ups.vendorid: 0001
tichzvidz@ltspserver2:~$
```

FIGURE 8.6: A diagram showing the UPS metrics

Figure 8.6 shows the UPS information displayed by the `upsc`, including that the UPS is not charging (offline) and not connected to the power supply and the current charge is 85%.

- **upsrw** allows the read and write parameters within the UPS driver depending on the capability of the UPS [140].
- **upscmd** allows to call instant commands depending on the UPS [141]. For example starting and battery tests.
- **upsmon** is a daemon responsible for UPS monitoring [142]. For example powering down a computer when battery is low

- **upssched** allows the scheduling of events being monitored by the upsmon [143].

## NUT installation

In Ubuntu-based distributions, the NUT tools can be installed via command line as below.

```
apt install nut
```

Alternatively, the NUT binaries can be downloaded as a Debian package from the Debian website on [144]. The NUT has the following configuration files under 'etc/nut' folder to help identify the UPS, monitor and communicate feedback.

### nut.conf

The nut.conf file setups the startup mode for the NUT. Examples of modes are none, standalone, netserver and netclient. None means that NUT tools are not configured and it comes as the default setting. Standalone means a local configuration with a single UPS protecting the host computer. Standalone starts the upsmon, upsd and the UPS driver. The netserver is similar to the standalone mode but adds network access control tools, e.g. firewall. Netclient requires the upsmon component of NUT.

### ups.conf

The ups.conf file configures all the UPS devices that the NUT will be directly monitoring. It is recommended to give an alias that is easier to remember. Below is an example of a fry UPS that interfaces with the NUT tools via a USB.

```
[fry]
driver = blazer_usb
port = /dev/ttyS0
desc = "My LTSP server UPS."
```

FIGURE 8.7: An example of the UPS configuration file.

**upsd.conf**

The upsd.conf is a daemon that monitors and collects the data from the UPS devices and forwards it to the upsmon daemon.

**upsd.users**

Sets the user credentials including the password for accessing the NUT Tools such as upsmon. The file is useful when configuring user privileges and the network services.

**upsmon.conf**

The upsmon is a monitoring daemon. A new user can be created under NUT for security reasons, or NUT can be run with root privileges. The upsmon acts as the server, and it monitors the UPS and responds to the clients' messages from the UPS. An example will be responding to low battery messages by initiating a system shutdown process.

**upssched.conf**

The upssched.conf file is used to schedule tasks to be performed by the NUT tools.

**A Customised shell script**

Most of the cheap UPSes do not send the low battery signal, so they never shut down in practice. A lead acid battery degrades with time and will not last if continuously discharged. The lead-acid battery should never discharge below 80% of its capacity [145]. In our case, during power cuts, the LTSP clients will have already lost their sessions, and there is no need to keep the server running. Since NUT tools do not customise the low battery level, the solution is to create a customised shell script. A shell script is useful when a user intends to customise tasks when a certain event occurs. An example is shutting down the computer when the UPS battery reaches a critical level. A shell script that immediately shuts down the server will protect the filesystem and extend the UPS battery life. Figure 8.8 is an example of a shell script that automatically shuts down the server as soon as power is cut:

```
#!/bin/bash
echo "started shutdown script"
notify-send "problem with power"
notify-send "shutting down in 10 seconds"
sleep 10
# start the UPS driver shutdown sequence,
# which initiates the count down to switch off the
UPS
upsdrvctl shutdown SLL-UPS > /dev/null
# and shutdown the server
upsmon -c fsd > /dev/null
```

FIGURE 8.8: An example of a script to shutdown the server

### 8.2.3 Dust collection

Since the LTSP server is deployed in schools, the server is most likely to collect dust from chalk. If possible, the use of chalk should be avoided. The server machine should be periodically cleaned with a cloth to remove accumulating dust. Section 6.3 demonstrates how to re-sit and clean the RAM module in case the RAM module collect dust.

### 8.2.4 Redundant Array of Inexpensive Disks

A Redundant Array of Inexpensive Disks (RAID) is a data storage technology that uses two or more disks (drives) for better performances and or data safety [146, 147]. Note that at times the 'I' in the acronym can be interpreted as independent. RAID uses at least two drives that work in parallel. In our case, the focus is on the uptime (fault tolerance) of the LTSP server. Therefore this section only explores RAID 1.

#### RAID 1 mirroring

Figure 8.9 shows a RAID 1 implementation with drive 1 mirrored (duplicated) to drive 2.

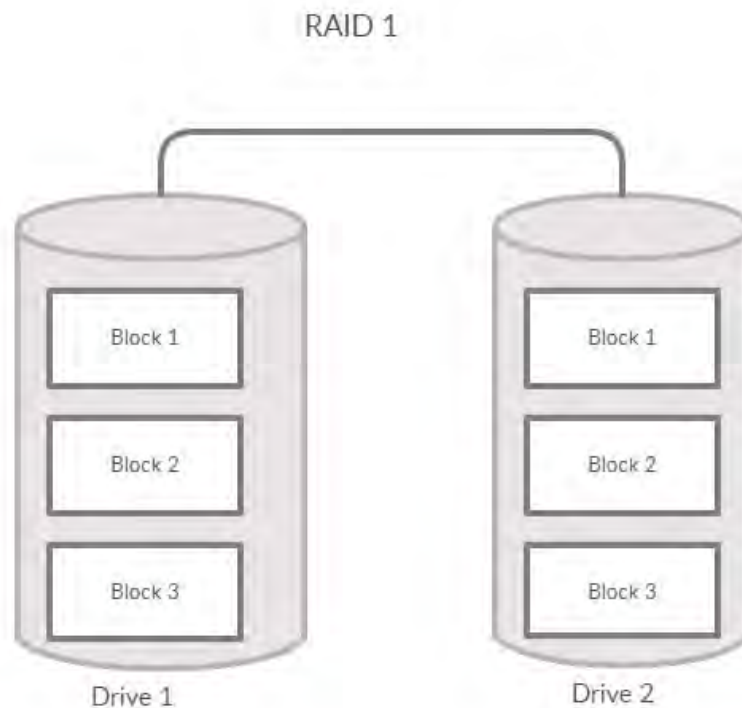


FIGURE 8.9: RAID 1 mirror of drive 1 and drive 2

RAID 1 stores data as duplicates or mirrored across at least two drives. Only one drive is considered active or online at a time. The other drive functions as a backup in case the online drive fails. RAID 1 has better fault tolerance as there is a backup drive if the LTSP server's online drive fails. The drawbacks of RAID 1 are the additional costs of an extra drive(s) and the periodical duplication of data negatively impacts the LTSP computing infrastructure, especially an LTSP computing infrastructure with thin clients.

### 8.2.5 Extra disk deployment

Since data safety is not so critical in the SLL's LTSP infrastructure deployments, an extra disk is simpler and therefore a better alternative to RAID 1. Each deployment comes with two disks that have the same copy of the LTSP installation. That way, if the disk in use fails, it is replaced by the second one. Unlike the RAID 1 implementation, an extra disk is a more straightforward solution. Using remote assistance tools (as discussed in the next section), the administrators can help a person on-site in the SLL school to replace the disk.

## 8.2.6 Motherboard failures

In our case, if the LTSP server motherboard fails it is replaced by another server. However, if there is enough technical ability one can deploy the LTSP server with an extra motherboard in storage.

## 8.2.7 Remote system logging

Remote system logging is whereby files that records system or program events are sent to a remote destination such as a website and email [148, 149]. Remote logging tools might help prevent a problem before it occurs and also helps troubleshoot a problem after it has occurred. The GNU/Linux system, by default, uses the 'var/log' directory to store all system log files from the system and services. In our case, the administrators are not always on-site, and administrators need to access the log files for troubleshooting purposes. Figure 8.10 shows how Logwatch is set-up to manage system logs.

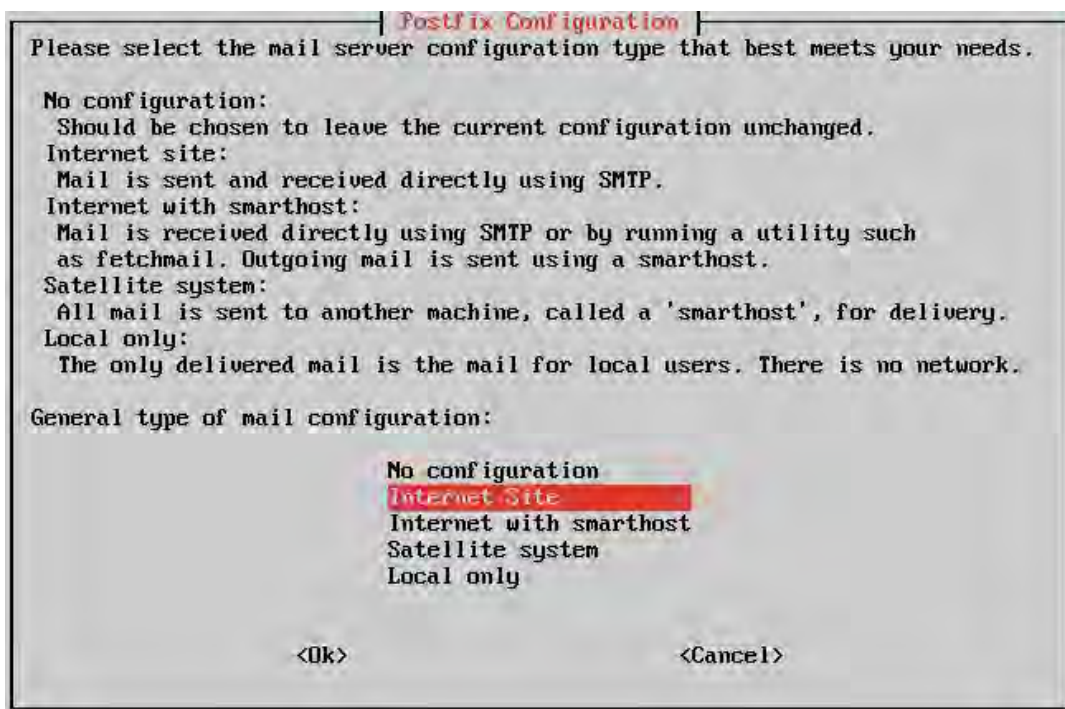


FIGURE 8.10: Logwatch postfix configuration menu

Figure 8.10 shows the Logwatch menu with options available. The LTSP administrator for the SLL schools can configure the Logwatch to send log files to a remote email periodically. The logs will help the LTSP administrators monitor the server's health and troubleshoot potential problems before they occur.

### 8.2.8 Remote access

In remote assistance a computer controls another computer located somewhere with or without the assistance of a person on-site [150]. Remote assistance tools are mostly used to troubleshoot or modify the server, but in some cases, they are used to prevent a problem before it occurs. Since the LTSP administrators and SLL schools are usually in a different network, we need an application that can remote assist another computer over the Internet. Several GNU/Linux remote assistance Internet-based applications are available and are free or at least free for non-commercial use, e.g. TeamViewer, Anydesk etc. This section recommends including remote assistance applications as part of LTSP installation. Figure 8.11 shows an example of a TeamViewer application remotely accessing an LTSP server in the Ntsika school from a different network (Rhodes University network).



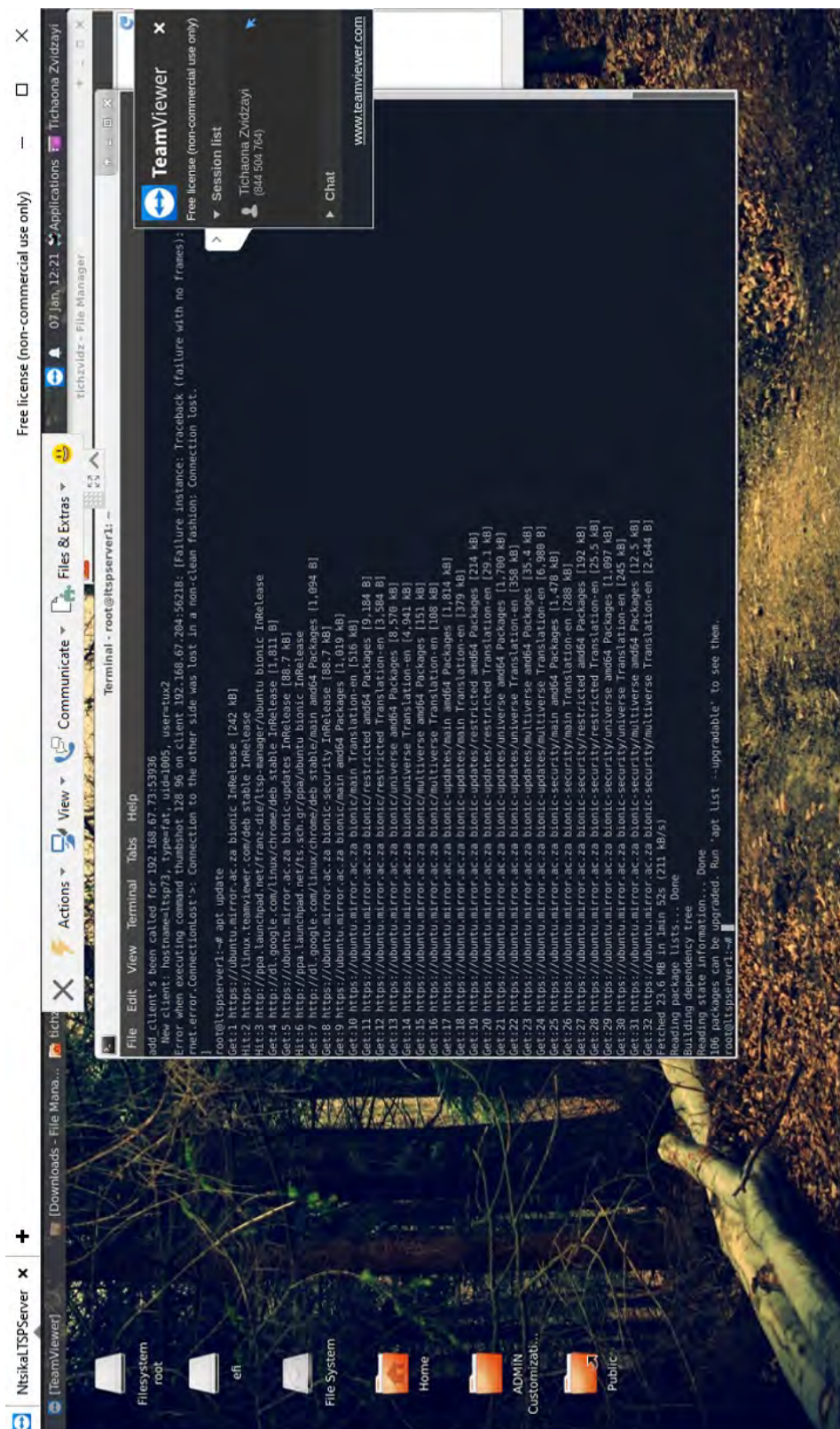


FIGURE 8.11: An LTSP administrator using TeamViewer to update the LTSP server in Ntsika school remotely

TeamViewer also offers relay services that allow a user in a private network to control another computer in a private network. TeamViewer uses unique login credentials for the host computer, and a remote computer uses these credentials to access and control the host computer. The TeamViewer application was successful in the Ntsika school. The task was to add the VLC media player application and update the LTSP client's image in the Ntsika school remotely from Cape Town. The experience showed that the TeamViewer application requires reasonable bandwidth for its use.

## 8.3 Other ways to improve the LTSP experience

This section explores the other ways that can improve the LTSP computing infrastructure experience.

### 8.3.1 Solid State Drive

A Solid State Drive (SSD), sometimes called a Solid State Device, uses flash memory to store data persistently and are significantly faster than the traditional harddrives [151]. An SSD has fewer I/O overheads and will improve the LTSP computing infrastructure experience for both thin and thick client computing. The drawback of an SSD is that they cost more than traditional harddrives and have normally shorter lives (this is actually by now no longer true for many of them).

### 8.3.2 LTSP configuration file

The LTSP server can be optimised to function in a specific way using the configuration file. A user can configure display resolution, sessions, sound etc. Note that for LTSP5.2 the file is `lts.conf` and for LTSP19+ the file is `ltsp.conf`. The `*.conf` file can be edited using the editor of choice e.g. `vim`, `nano` etc. For LTSP5, if the `ltsp.conf` does not exist the following command will install it:

```
ltsp-config lts.conf
```

The `lts.conf` file is found in the variable directory `/var/lib/tftpboot/ltsp/*/lts.conf`. The asterisks `'*'` is a placeholder for the processor e.g. `amd64`. Note that LTSP5+ called the configuration file `lts.conf` and LTSP19+ called it `ltsp.conf`.

For the LTSP19+, if the file does not exist it can be installed using the following command:

```
install -m 0660 -g sudo /usr/share/ltsp/common/ltsp/ltsp.conf /etc/ltsp/ltsp.conf  
[152].
```

### Avoiding user session collisions

By default, LTSP does not restrict users from sharing one session. However, errors may occur when users attempt to use and save data from the same application. An example is when two users share the same session and one attempts to logoff. These errors can be prevented by editing the `lts*.conf` file as follows.

```
LDM_LIMIT_ONE_SESSION=True
```

### Guest sessions

There are cases where the LTSP clients are used as once-off sessions. An example is when groups of people from the community are using computers occasionally. It is not necessary to create credentials for non-frequent users. The guest login feature will allow the user to login as a guest and the LTSP handles the guest accounts as separate accounts. Below is an example of enabling guest sessions.

```
LDM_GUESTLOGIN=True
```

For more detailed versions of both LTSP5+ and LTSP19+ configuration files see Appendix B.

### Single NIC over a dual NIC configuration

In our case, if the LTSP server fails during a lab session, the lab can use live-USB sticks to restart the lab (as discussed in Section 8.1.2). On a single NIC, unlike a dual NIC configuration, the thick clients can use a live OS and access the Internet. Accessing the Internet is impossible on a dual NIC configuration since the LTSP server provides a private network and is responsible for NAT routing the LTSP clients to the Internet. Compared to a dual NIC configuration, a single NIC network configuration makes the installation of LTSP easier to configure as no additional network settings are required. The drawback of a single NIC is that the LTSP clients will rely on an external DHCP, and if the external DHCP is misconfigured or fails, the LTSP client will not boot.

## **8.4 Conclusion**

This chapter explored ways to make deployment and maintenance easier. Easy deployment and maintenance of the LTSP computing infrastructure are vital for sustainability. An LTSP computing infrastructure that is easy to manage or deploy will encourage active participation from people with some technical skills, e.g. educators and lab assistants. The maintenance tools will help the infrastructure to last longer. The next chapter will conclude this research, outlines contributions and proposes future work.

## 9 Conclusion and future work

This chapter concludes by revisiting the aims and objectives outlined in Chapter 1. It also lists areas that were not covered in this research and proposes future work.

### 9.1 Research summary

The core task of this thesis was to review and assess the Siyakhula Living Lab (SLL) blueprint. Two questions were asked; to determine whether LTSP was still a viable component for the SLL deployments and how it could support a more sustainable deployment. The introduction chapter was followed by reviewing related work that supports the deployment of ICT4D computing infrastructure. The chapter also explored alternative technologies that implement server-client models similar to LTSP, e.g. BeTwin ES and OpenThin-Client. LTSP was the preferred solution for the computing infrastructure. A more technical presentation of the LTSP computing infrastructure followed, to help the reader better understand the technology. This involved explaining core technologies such as DHCP, PXE, TFTP, X11 etc. The chapter also explored LTSP19+ improvements and detailed how virtual machines work an important novelty in LTSP19+.

Chapters 4 to 7 reported on specific work packages, organised in chronological order. As a matter of course, lab experiments supported all real-life deployments to reduce time and complications during the deployment.

Experiments from chapter 4 supported the deployment of a thick-client based LTSP computing infrastructure in the Ntsika school in Makhanda. The Ntsika installation has been successfully running for almost two years and has helped the school and the surrounding community. In late 2020 and early 2021, the Ntsika infrastructure was used to teach digital literacy and certify members of the community.

Experiments from chapter 4 also supported the re-deployment of thin-client based LTSP computing infrastructure in the Ngwane school in the Mbhashe Municipality. The Mpume school infrastructure did not need much attention, besides the clean up of the filesystem.

Experiments reported in Chapter 7 focused on the new version, LTSP19+. The aim was to compare the two versions of LTSP, in preparation of the upcoming Siyakhula Living Lab (SLL) deployments. The upcoming deployments involve thick-client based LTSP computing infrastructure, similar to the Ntsika school. The LTSP19+ has a smaller memory footprint and has similar CPU usage to LTSP5+. Most importantly, LTSP19+ includes the ability to prepare and maintain LTSP clients' image as virtual machines. Virtual machines will help the SLL schools by giving teachers or lab assistants a graphical way to manage classroom applications.

Chapter 8 focused on making the LTSP computing infrastructure easier to deploy and maintain. An easy deployment will encourage people with limited technical skills to actively participate in the deployment of LTSP computing infrastructure. For example, a teacher can easily use a USB stick to install offline applications without connectivity. The maintenance devices such as power surge protectors, Uninterruptible Power Supply (UPS) and an extra disk help prevent problems or solve them quickly if they happen. In some cases where a problem can be remotely solved, tools such as TeamViewer are helpful, as they do not require an expert to be on-site. Making the deployment and maintenance of the LTSP computing infrastructure easier is critical to sustainability and will extend the lifetime of the deployed infrastructure.

## 9.2 Revisiting the research questions and goals

### Research questions

1. *Is LTSP still a good tool for the SLL blueprint?*

The researcher was requested to review and assess whether LTSP was still a good component for the SLL blueprint. Currently, LTSP is well supported and a cost-effective deployment compared to deployments based on tablets and laptops. Tablets and laptops have the advantage of mobility, and educational institutions can allow students to take tablets or laptops home. However, it is a challenge to allocate every student a

laptop or a tablet because of affordability and, in some cases, accountability. One example is the One Laptop Per Child (OLPC) (as discussed in Section 2.2.2) failed to achieve its expectations due to costs. In 2021, South Africa's education department failed to account for a total of 55000 laptops distributed to matric students [153]. Laptops and tablets are much easier target of theft.

As seen in the Ntsika school when thieves broke in and stole all the laptops.

LTSP deployment will also provide a naturally centralised filesystem, whereby only one installation (LTSP client's image) is maintained and served to clients. All user files will reside on the LTSP server, and unlike tablets and laptops, a modification on the client's image reflects on all clients immediately. LTSP users can login to any of the LTSP clients and continue working without having to use the same client. Also if one client fails, it is replaced without losing user data (on the other hand the server represent a single point of failure see next subsection). An LTSP installation will provide a cost-effective and sustainable deployment compared to tablets and laptops.

## 2. *What improvements should be made to have a more sustainable deployments?*

The preparation and maintenance of LTSP19+ clients' image as virtual machines will encourage participation from non-technical people. In our case in the SLL, most of the time, the administrators are not on-site, and tasks such as installing an application demand unnecessary administrative work. Virtual machines will empower teachers and lab assistants by allowing them to manage applications and prepare the client's image graphically.

The SLL will use inexpensive Uninterruptible Power Supply (UPS)es, and in practice, most cheap UPSes never sends the shutdown signal, so a customised shell script was created. The UPS use lead batteries that will not last if discharged continuously. Typically the lead batteries should never be discharged beyond 80% [145]. UPSes should be configured to shutdown immediately after power cuts. Naturally only the LTSP server is connected to the UPS because the entire files and filesystem reside on the server.

Wherever possible, LTSP deployments should use thick clients that offloads most computing to client computers, resulting in a less expensive

server than thin-client computing. Thick-client computing on a single NIC configuration will allow a lab to quickly recover from a server sudden downtime using live USB sticks.

### 9.3 Research work contributions

First of all, this research contributed to the SLL blueprint upgrade, by confirming that LTSP is still a good tool for computing infrastructure deployment. Tools such as virtual machines, power surge protectors and UPSes contribute to a longer lifetime of an LTSP deployment. This research also established that as a quick fix, thick-client computing on a single NIC would help reduce lab sessions downtime when the LTSP server is unavailable.

Secondly, this research contributed one real-life deployment in the Ntsika school and redeployment in the Ngwane school. For about two years, this research also supported continuous maintenance for the Ntsika, Ngwane and Mpume schools. The three labs now help the schools with learning and the surrounding communities with Internet access. The work in this research will still contribute to the deployments and maintenance of the upcoming SLL projects.

Finally, part of this research was presented in the Innovation, Science and Technology (IST) Africa 2021 Conference paper. The paper was entitled '*LTSP Client Image Maintenance: Utilising a Virtualisation Player to Support Educators to Directly Manage Classroom Applications*' reported the ability to help educators in the SLL manage their classroom applications using virtual machines [154]. Another paper reporting on a different angle of this research is under review for Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2021 Conference.

### 9.4 Future work

There should be an easier way for LTSP19+ to support thin clients. Even though the technology is now moving towards thick clients, schools in poor communities might only afford thin clients for more years to come. Besides, even new thin clients might become a more effective technology in the future.

LTSP can be deployed in Docker containers. Docker uses OS-level virtualisation to package applications and their dependencies, making it easier to



create, deploy, and run applications as containers [155, 156]. A Docker container would allow a more straightforward deployment as the LTSP comes already installed and configured. A containerised LTSP might be easier to deploy for people low technical abilities.

The introduction of the Wayland technology, an improvement on the X11 that is easier to maintain and develop [157], can reduce LAN bandwidth consumption. A reduction in LAN bandwidth consumption, together with improvements in WiFi technology, will open the way to implement LTSP over wireless mediums such as WiFi. This may reduce the cost and the maintenance of the network infrastructure to support LTSP.

## 9.5 Parting thoughts

The LTSP-based computing lowers the costs of deploying infrastructure, especially in schools. Thick clients are becoming the current favourites, and as seen in Section 3.2.2, the creators of LTSP have limited the support of thin clients. Some of the reasons are the availability of better computing hardware and the possibility of reducing LTSP server costs.

Governments should play a vital role in bringing ICTs to poor communities and therefore take full advantage of blueprints such as the one developed through this research work reported in this thesis.

# Bibliography

- [1] *Effects of poor infrastructure in education & transport* | DBSA. URL: <https://www.dbsa.org/article/effects-poor-infrastructure-education-transport-and-communities>.
- [2] Ifzal Ali and Ernesto M. Pernia. "Infrastructure and Poverty Reduction - What is the Connection?" In: (Jan. 2003). ISSN: 1655-5260. URL: <https://think-asia.org/handle/11540/613>.
- [3] T P Ogun. "Infrastructure and Poverty Reduction: Implications for Urban Development in Nigeria". In: (2010). DOI: 10.1007/s12132-010-9091-8.
- [4] World Bank. "World Development Report 1998/1999". In: *communications technologies* (Sept. 1998). DOI: 10.1596/978-0-1952-1118-4. URL: <https://openknowledge.worldbank.org/handle/10986/5981>.
- [5] *Bridging the digital divide: Discovery Service for Rhodes University Library*. URL: <http://0-eds.a.ebscohost.com.wam.seals.ac.za/eds/detail/detail?vid=0&sid=f2f5c855-486d-4154-ba9e-c9290c69bbe4%40sdc-v-sessmgr01&bdata=JnNpdGU9ZWRzLWxpdmU%3d#AN=146576742&db=buh>.
- [6] *Top 7 Reasons Why Most ICT4D Projects Fail - ICTworks*. URL: <https://www.ictworks.org/top-7-reasons-why-most-ict4d-projects-fail/>.
- [7] Ingrid Sieborger and Alfredo Terzoli. "The network society: A model for computing infrastructure in South African schools". In: *2016 IST-Africa Conference, IST-Africa 2016*. Institute of Electrical and Electronics Engineers Inc., Aug. 2016. ISBN: 9781905824557. DOI: 10.1109/ISTAFRICA.2016.7530673.
- [8] Ed. "Linux Terminal Server Project Administrator's Reference Linux Terminal Server Project Administrator's Reference A Guide to LTSP Networks". In: (2008).
- [9] *HP Thin Clients Unraveling The Mystery* | HP® Tech Takes. URL: <https://www.hp.com/us-en/shop/tech-takes/hp-thin-clients>.

- [10] *Fat Client | IT Pro*. URL: <https://www.itprotoday.com/compute-engines/fat-client>.
- [11] *The Evolution of Thin Client Computing: From Mainframes to Zero Clients | Hacker Noon*. URL: <https://hackernoon.com/the-evolution-of-thin-client-computing-from-mainframes-to-zero-clients-cf1431bu>.
- [12] Ethan Mollick. “Establishing Moore’s law”. In: *IEEE Annals of the History of Computing* 28.3 (July 2006), pp. 62–75. ISSN: 10586180. DOI: 10.1109/MAHC.2006.45.
- [13] *Over 50 Years of Moore’s Law*. URL: <https://www.intel.com/content/www/us/en/silicon-innovations/moores-law-technology.html>.
- [14] *The Possibilities and Pitfalls of Harnessing ICTs to Accelerate Social Development*. URL: [http://colfinder.net/materials/Supporting\\_Distance\\_Education\\_Through\\_Policy\\_Development/resources/butcher1/unrisd.htm](http://colfinder.net/materials/Supporting_Distance_Education_Through_Policy_Development/resources/butcher1/unrisd.htm).
- [15] *The Practice and Need for Rural ICT for Development Evaluation: An Experience of the Siyakhula Living Lab Baseline Study*. URL: <https://dora.dmu.ac.uk/handle/2086/19865>.
- [16] Alfredo Terzoli, Ingrid Siebörger, and Sibukele Gumbo. “Community ‘Broadband Islands’ for digital government access in rural South Africa”. In: *Proceedings of the European Conference on e-Government, ECEG Part F1294* (2017), pp. 204–211. ISSN: 20491034.
- [17] *Technology solutions to strengthen the integration of marginalized communities into the global knowledge society - IEEE Conference Publication*. URL: <https://ieeexplore.ieee.org/document/5753005>.
- [18] *About · GitHub*. URL: <https://github.com/about>.
- [19] *What Is GitHub? A Beginner’s Introduction to GitHub*. URL: <https://kinsta.com/knowledgebase/what-is-github/>.
- [20] *Telkom talks up mobile growth*. URL: <https://businesstech.co.za/news/mobile/434575/telkom-talks-up-mobile-growth/>.
- [21] Jan A G M Van Dijk. “Digital divide research, achievements and shortcomings”. In: (). DOI: 10.1016/j.poetic.2006.05.004. URL: <http://www.gw.utwente.nl/vandijk..>
- [22] Masuma Rahaman and Arifur Rahaman. “The influence and effect of digital divide on students purchase: Comparison of urban and rural cases”. In: *Revista de Cercetare si Interventie Sociala* 69 (June 2020), pp. 206–226. ISSN: 15845397. DOI: 10.33788/rcis.69.13.

- [23] (No Title). URL: <https://www.statssa.gov.za/publications/P0318/P03182016.pdf>.
- [24] *Science and Innovation on South Africa's expenditure on research and development* | South African Government. URL: <https://www.gov.za/speeches/science-and-innovation-south-africas-expenditure-research-and-development-25-oct-2019-0000>.
- [25] *Peri-Urban Landscapes; Water, Food and Environmental Security*. URL: <https://en.unesco.org/events/peri-urban-landscapes-water-food-and-environmental-security>.
- [26] (PDF) *Defining the peri-urban: Rural-urban linkages and institutional connections*. URL: [https://www.researchgate.net/publication/287613842\\_Defining\\_the\\_peri-urban\\_Rural-urban\\_linkages\\_and\\_institutional\\_connections](https://www.researchgate.net/publication/287613842_Defining_the_peri-urban_Rural-urban_linkages_and_institutional_connections).
- [27] *Marginalized populations* | National Collaborating Centre for Determinants of Health. URL: <https://nccdh.ca/fr/glossary/entry/marginalized-populations>.
- [28] *Rural population (% of total population) - South Africa* | Data. URL: <https://data.worldbank.org/indicator/SP.RUR.TOTL.ZS?end=2018&locations=ZA&start=2018&view=bar>.
- [29] *ICT4D* | CRS. URL: <https://www.crs.org/our-work-overseas/ict4d>.
- [30] *Strategic Partnerships*. URL: <https://www.imd.org/research-knowledge/articles/strategic-partnerships/>.
- [31] *About UNCTAD* | UNCTAD. URL: <https://unctad.org/about>.
- [32] (No Title). URL: [https://www.siyakhulall.org/sites/default/files/SiyakhulaLL\\_Booklet\\_2012.pdf](https://www.siyakhulall.org/sites/default/files/SiyakhulaLL_Booklet_2012.pdf).
- [33] *Global Information Technology Report 2016 - Reports - World Economic Forum*. URL: <https://reports.weforum.org/global-information-technology-report-2016/economies/#economy=ZAF>.
- [34] *SA Connect: Another year, another moving target* | ITWeb. URL: <https://www.itweb.co.za/content/lwrKxv3JRQGqmg1o>.
- [35] (PDF) *Role of ICT on Enhancing Quality of Education*. URL: [https://www.researchgate.net/publication/330778182\\_Role\\_of\\_ICT\\_on\\_Enhancing\\_Quality\\_of\\_Education](https://www.researchgate.net/publication/330778182_Role_of_ICT_on_Enhancing_Quality_of_Education).
- [36] *The spectacular failure of One Laptop Per Child -Philanthropy Daily*. URL: <https://www.philanthropydaily.com/the-spectacular-failure-of-one-laptop-per-child/>.

- [37] *Our Story – One Laptop Per Child*. URL: <https://www.onelaptopperchild.org/about/>.
- [38] *OLPC's \$100 laptop was going to change the world — then it all went wrong - The Verge*. URL: <https://www.theverge.com/2018/4/16/17233946/olpcs-100-laptop-education-where-is-it-now>.
- [39] *Why did One Laptop Per Child fail? – Foreign Policy*. URL: <https://foreignpolicy.com/2009/09/09/why-did-one-laptop-per-child-fail/>.
- [40] *Low-Cost OLPC Laptop Hits Assembly Line | LinuxInsider*. URL: <https://linuxinsider.com/story/low-cost-olpc-laptop-hits-assembly-line-58453.html>.
- [41] *Hardware | One Laptop per Child*. URL: <http://one.laptop.org/about/hardware>.
- [42] *Our software - OLPC*. URL: [http://wiki.laptop.org/go/Our\\_software#What\\_software\\_will\\_be\\_used\\_with\\_the\\_.24100\\_laptop.3F](http://wiki.laptop.org/go/Our_software#What_software_will_be_used_with_the_.24100_laptop.3F).
- [43] *What is a Mesh Network? | Soracom*. URL: <https://www.soracom.io/iot-definitions/what-is-a-mesh-network/>.
- [44] *Why OLPC failed | TechSangam*. URL: <https://www.techsangam.com/2020/10/09/why-olpc-failed/>.
- [45] *BBC News - OLPC software to power ageing PCs*. URL: <http://news.bbc.co.uk/2/hi/technology/8117064.stm>.
- [46] *How Google Took Over the Classroom - The New York Times*. URL: <https://www.nytimes.com/2017/05/13/technology/google-education-chromebooks-schools.html>.
- [47] *Introduction to Client Server Networks*. URL: <https://www.lifewire.com/introduction-to-client-server-networks-817420>.
- [48] *Client-Server networks*. URL: <https://www.omnisci.com/technical-glossary/client-server>.
- [49] *Thinsoft BeTwinVS, virtual desktop, vdi, multipoint desktop, remote access, pc sharing, virtual desktop software, remote access software, pc sharing software, virtual desktop infrastructure, multipoint desktop software*. URL: [http://thinsoftinc.com/product\\_pc\\_sharing\\_betwin\\_ES.aspx](http://thinsoftinc.com/product_pc_sharing_betwin_ES.aspx).
- [50] *openthinclient | openthinclient*. URL: <http://openthinclient.org/en/open-thinclient/>.
- [51] *DRBL - About*. URL: <http://drbl.sourceforge.net/>.
- [52] *29.4. Network Information System (NIS)*. URL: <https://www.freebsd.org/doc/handbook/network-nis.html>.

- [53] *LTSPManager: how 1000+ Greek schools switched to Debian-based distributions - DebConf17*. URL: <https://debconf17.debconf.org/talks/76/>.
- [54] *About - Epopetes*. URL: <https://epoptes.org/>.
- [55] Birgitta Bergvall Kareborn and Anna Stahlbrost. "Living Lab: an open and citizen-centric approach for innovation". In: *International Journal of Innovation and Regional Development* 1.4 (2009), p. 356. ISSN: 1753-0660. DOI: 10.1504/ijird.2009.022727.
- [56] *Living labs for regional innovation ecosystems WHY INVEST IN PROMOTING LIVING LABS?* Tech. rep. URL: <http://www.citysdk.eu>.
- [57] Alfredo Terzoli et al. "Digital inclusion: A model for e-Infrastructure and e-Services in developing countries". In: *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST 250* (2018), pp. 85–98. ISSN: 18678211. DOI: 10.1007/978-3-319-98827-6{\\_}7.
- [58] *Project Overview | Siyakhula Living Lab*. URL: <https://siyakhulall.org/project-overview>.
- [59] J Fielder and P W Foltz. *Benefits and Challenges of Multidisciplinary Project Teams: "Lessons Learned" for Researchers and Practitioners*. Tech. rep. 1. 2012, p. 33. URL: <https://commons.erau.edu/publication/108>.
- [60] (PDF) *Information and Communications Technology for Development (ICT4D) - A Design Challenge?* URL: [https://www.researchgate.net/publication/251829011\\_Information\\_and\\_Communications\\_Technology\\_for\\_Development\\_ICT4D\\_-\\_A\\_Design\\_Challenge](https://www.researchgate.net/publication/251829011_Information_and_Communications_Technology_for_Development_ICT4D_-_A_Design_Challenge).
- [61] Hossana Twinomurizi. "The role of ICT in sustainable and responsible development: E-skilling". In: *IFIP Advances in Information and Communication Technology*. Vol. 386 AICT. Springer, Berlin, Heidelberg, 2012, pp. 90–99. ISBN: 9783642333316. DOI: 10.1007/978-3-642-33332-3{\\_}9. URL: [https://link.springer.com/chapter/10.1007/978-3-642-33332-3\\_9](https://link.springer.com/chapter/10.1007/978-3-642-33332-3_9).
- [62] *Network - DHCP | Server documentation | Ubuntu*. URL: <https://ubuntu.com/server/docs/network-dhcp#download>.
- [63] *Dynamic Host Configuration Protocol (DHCP) | Microsoft Docs*. URL: <https://docs.microsoft.com/en-us/windows-server/networking/technologies/dhcp/dhcp-top>.
- [64] *What Is DNS? | How DNS Works | Cloudflare UK*. URL: <https://www.cloudflare.com/en-gb/learning/dns/what-is-dns/>.

- [65] *How DNS Works In Six Steps - Verisign*. URL: [https://www.verisign.com/en\\_US/website-presence/online/how-dns-works/index.xhtml](https://www.verisign.com/en_US/website-presence/online/how-dns-works/index.xhtml).
- [66] *Advantages of Using DHCP - Oracle Solaris Administration: IP Services*. URL: [https://docs.oracle.com/cd/E23824\\_01/html/821-1453/dhcp-overview-12a.html](https://docs.oracle.com/cd/E23824_01/html/821-1453/dhcp-overview-12a.html).
- [67] *11.4. NFS File Server*. URL: <https://debian-handbook.info/browse/el-GR/stable/sect.nfs-file-server.html>.
- [68] *SettingUpNFSHowTo - Community Help Wiki*. URL: <https://help.ubuntu.com/community/SettingUpNFSHowTo>.
- [69] *Remote Procedure Call*. URL: [https://www.ibm.com/support/knowledgecenter/ssw\\_aix\\_72/commprogramming/ch8\\_rpc.html](https://www.ibm.com/support/knowledgecenter/ssw_aix_72/commprogramming/ch8_rpc.html).
- [70] *What Is RPC? (Introduction to RPC)*. URL: [https://www.w3.org/History/1992/nfs\\_dxcern\\_mirror/rpc/doc/Introduction/WhatIs.html](https://www.w3.org/History/1992/nfs_dxcern_mirror/rpc/doc/Introduction/WhatIs.html).
- [71] *The Trivial File Transfer Protocol (tftp)*. URL: <https://www.erg.abdn.ac.uk/users/gorry/course/inet-pages/tftp.html>.
- [72] *Setting up DHCP and TFTP servers*. URL: <https://tldp.org/HOWTO/Clone-HOWTO/setting-up.html>.
- [73] *What is PXE? | PXE Boot Server - ManageEngine OS Deployer*. URL: <https://www.manageengine.com/products/os-deployer/pxe-preboot-execution-environment.html>.
- [74] *Understanding PXE Booting and Kickstart Technology*. URL: [https://docs.oracle.com/cd/E24628\\_01/em.121/e27046/appdx\\_pxeboot.htm#EMLCM12198](https://docs.oracle.com/cd/E24628_01/em.121/e27046/appdx_pxeboot.htm#EMLCM12198).
- [75] *iPXE - open source boot firmware [start]*. URL: <https://ipxe.org/>.
- [76] *(No Title)*. URL: <http://www.pix.net/software/pxeboot/archive/pxespec.pdf>.
- [77] *X Window System Internals*. URL: <http://xwindow.angelfire.com/page0.html>.
- [78] *X.Org*. URL: <https://www.x.org/wiki/>.
- [79] *X11 | Article about X11 by The Free Dictionary*. URL: <https://encyclopedia2.thefreedictionary.com/X11>.
- [80] *File:X client server example.png - Wikimedia Commons*. URL: [https://commons.wikimedia.org/wiki/File:X\\_client\\_server\\_example.png](https://commons.wikimedia.org/wiki/File:X_client_server_example.png).
- [81] *How to use XDMCP for Remote Desktop Access on Linux*. URL: <https://linuxide.com/how-tos/how-to-identify-your-xdmcp-server/>.

- [82] *About - LTSP*. URL: <https://ltsp.org/>.
- [83] *NBD - The Wireshark Wiki*. URL: <https://wiki.wireshark.org/NBD>.
- [84] *The Linux Terminal Server Project: Thin clients and Linux | USENIX*. URL: <https://www.usenix.org/techsessionssummary/linux-terminal-server-project-thin-clients-and-linux>.
- [85] *Linux Today - After two years of development, LTSP 5.2 is now out*. URL: <https://www.linuxtoday.com/developer/2010022201135NWCYRL>.
- [86] *Thin Client: Linux Terminal Server Project - LTSP 5.2 Released*. URL: [https://thinclient.org/archives/2010/02/linux\\_terminal\\_1.html](https://thinclient.org/archives/2010/02/linux_terminal_1.html).
- [87] *Announcing LTSP 19.08 · Discussion #268 · ltsp/ltsp · GitHub*. URL: <https://github.com/ltsp/ltsp/discussions/268>.
- [88] *Preparation - LTSP*. URL: <https://ltsp.org/docs/preparation/>.
- [89] *Network Address Translation (NAT) FAQ - Cisco*. URL: <https://www.cisco.com/c/en/us/support/docs/ip/network-address-translation-nat/26704-nat-faq-00.html>.
- [90] *Discussions · ltsp/ltsp*. URL: <https://github.com/ltsp/ltsp/discussions>.
- [91] *What is Desktop Virtualization? | IBM*. URL: <https://www.ibm.com/cloud/learn/desktop-virtualization#toc-what-is-de-3GqWuys8>.
- [92] *What is VDI? – Virtual Desktop Infrastructure Definition - Citrix*. URL: <https://www.citrix.com/glossary/vdi.html>.
- [93] Jason Nieh, S. Jae Yang, and Naomi Novik. “Measuring thin-client performance using slow-motion benchmarking”. In: *ACM Transactions on Computer Systems* 21.1 (Feb. 2003), pp. 87–115. ISSN: 07342071. DOI: 10.1145/592637.592640. URL: <https://dl.acm.org/doi/10.1145/592637.592640>.
- [94] *LTSP/Ltsp Hardware Requirements - Debian Wiki*. URL: <https://wiki.debian.org/LTSP/Ltsp%20Hardware%20Requirements>.
- [95] *LTSP – Concept and Implementation | Techidnayan*. URL: <http://www.neeleshgurjar.co.in/ltsp-concept-and-implementation/>.
- [96] *What Is LTSP? | The Official Ubuntu Book, 2e: Using Edubuntu | InformIT*. URL: <https://www.informit.com/articles/article.aspx?p=1186281&seqNum=2>.
- [97] *Installation - LTSP*. URL: <https://ltsp.org/docs/installation/>.
- [98] *ltsp-image - LTSP*. URL: <https://ltsp.org/man/ltsp-image/>.
- [99] *chroot - ArchWiki*. URL: <https://wiki.archlinux.org/index.php/chroot>.
- [100] *VMware: An Essential Guide | IBM*. URL: <https://www.ibm.com/cloud/learn/vmware>.



- [101] VMware Workstation Player | VirtualizationWorks.com. URL: <https://www.virtualizationworks.com/VMware-Workstation-Player.asp>.
- [102] Keith Shaw. *What is a hypervisor*. 2017. URL: <https://www.networkworld.com/article/3243262/what-is-a-hypervisor.html>.
- [103] *Hypervisor: what is it and how it works for virtualization*. URL: <https://itglobal.com/company/glossary/hypervisor/>.
- [104] *Type-1 Hypervisor In figure 5, the most common Type-1 hypervisors are... | Download Scientific Diagram*. URL: [https://www.researchgate.net/figure/Type-1-Hypervisor-In-figure-5-the-most-common-Type-1-hypervisors-are-ESXi-of-VMware\\_fig5\\_344553205](https://www.researchgate.net/figure/Type-1-Hypervisor-In-figure-5-the-most-common-Type-1-hypervisors-are-ESXi-of-VMware_fig5_344553205).
- [105] *Explaining Soft Link And Hard Link In Linux With Examples - OSTechNix*. URL: <https://ostechnix.com/explaining-soft-link-and-hard-link-in-linux-with-examples/>.
- [106] *CubeGet*. URL: <https://cube.camicri.com/#/>.
- [107] *Do I Need a Gigabit Switch or 10/100Mbps Switch? | by Kelly Zeng | Medium*. URL: <https://medium.com/@kellyzs123/do-i-need-a-gigabit-switch-or-10-100mbps-switch-f2d286b410d9>.
- [108] *LTS - Ubuntu Wiki*. URL: <https://wiki.ubuntu.com/LTS>.
- [109] *What is a Long Term Support Release? What is Ubuntu LTS?* URL: <https://itsfoss.com/long-term-support-lts/>.
- [110] *Monitoring Linux and Windows hosts with Glances | Opensource.com*. URL: <https://opensource.com/article/19/11/monitoring-linux-glances>.
- [111] *Google Summer of Code - Eoptes*. URL: <https://eoptes.org/documentation/gsoc/>.
- [112] *What is and how do I enable IP forwarding on Linux? | OpenVPN*. URL: <https://openvpn.net/faq/what-is-and-how-do-i-enable-ip-forwarding-on-linux/>.
- [113] *RFC 2993 - Architectural Implications of NAT*. URL: <https://tools.ietf.org/html/rfc2993>.
- [114] *Intel Xeon E3-1220 @ 3.10GHz vs Intel Xeon E3-1225 @ 3.10GHz [cpubenchmark.net] by PassMark Software*. URL: <https://www.cpubenchmark.net/compare/Intel-Xeon-E3-1220-vs-Intel-Xeon-E3-1225/1196vs1198>.
- [115] Anuj Verma and Pawan Kumar Dahiya. "PCIe BUS: A State-of-the-Art-Review". In: *IOSR Journal of VLSI and Signal Processing* 07.04 (July 2017), pp. 24–28. ISSN: 23194197. DOI: 10.9790/4200-0704012428.

- [116] *Troubleshooting PCIe Bus Error severity Corrected on Ubuntu and Linux Mint - It's FOSS*. URL: <https://itsfoss.com/pcie-bus-error-severity-corrected/>.
- [117] *PCIe Bus Error Status 00001100 - Thomas-Krenn-Wiki*. URL: [https://www.thomas-krenn.com/en/wiki/PCIe\\_Bus\\_Error\\_Status\\_00001100](https://www.thomas-krenn.com/en/wiki/PCIe_Bus_Error_Status_00001100).
- [118] *About SGT Solutions*. URL: <https://sgtsolutions.co.za/about.php>.
- [119] *Color-coded RAM slots - CompTIA Server+ Certification Guide [Book]*. URL: <https://www.oreilly.com/library/view/comptia-server-certification/9781789534818/35707f57-5b29-4502-84cf-b511189e3170.xhtml>.
- [120] *fsck(8): check/repair file system - Linux man page*. URL: <https://linux.die.net/man/8/fsck>.
- [121] *What is the Linux kernel?* URL: <https://www.redhat.com/en/topics/linux/what-is-the-linux-kernel>.
- [122] *vmlinuz Definition*. URL: <http://www.linfo.org/vmlinuz.html>.
- [123] *Using the initial RAM disk (initrd) — The Linux Kernel documentation*. URL: <https://www.kernel.org/doc/html/latest/admin-guide/initrd.html>.
- [124] *Boot Partition · Issue #43 · ltsp/ltsp*. URL: <https://github.com/ltsp/ltsp/issues/43>.
- [125] *What Is an ISO File? (And How to Open or Use One)*. URL: <https://www.lifewire.com/iso-file-2625923>.
- [126] *Pinguy Builder – An App to Backup/Remix \*buntu | Pinguy OS*. URL: <https://pinguyos.com/2015/09/pinguy-builder-an-app-to-backupremix-buntu/>.
- [127] *Pinguy OS - Browse /ISO\_Builder at SourceForge.net*. URL: [https://sourceforge.net/projects/pinguy-os/files/ISO\\_Builder/](https://sourceforge.net/projects/pinguy-os/files/ISO_Builder/).
- [128] *ISO 9660 - OSDev Wiki*. URL: [https://wiki.osdev.org/ISO\\_9660](https://wiki.osdev.org/ISO_9660).
- [129] *LiveCD - Community Help Wiki*. URL: <https://help.ubuntu.com/community/LiveCD>.
- [130] *What is Voltage Surge? Definition & Types of Voltage Surge - Circuit Globe*. URL: <https://circuitglobe.com/voltage-surge.html>.
- [131] *6 Worst Reasons of hard disk failure and Data Recovery Solutions*. URL: <https://www.stellarinfo.com/blog/6-worst-reasons-of-hard-disk-failure/>.
- [132] *How does an Uninterruptible Power Supply (UPS) work? | CyberPower Power Blog*. URL: <https://www.cyberpowersystems.com/blog/how-does-a-ups-work/>.

- [133] *Standby UPS / Standby Power Supply*. URL: <http://209.68.14.80/ref/power/ext/ups/typesStandby-c.html>.
- [134] *Online ("True") UPS*. URL: <http://209.68.14.80/ref/power/ext/ups/typesOnLine-c.html>.
- [135] *Network UPS Tools User Manual Network UPS Tools User Manual Network UPS Tools User Manual ii REVISION HISTORY NUMBER DATE DESCRIPTION NAME Network UPS Tools User Manual*. Tech. rep. 2016.
- [136] *wiki.ipfire.org - Detailed NUT Configuration*. URL: <https://wiki.ipfire.org/addons/nut/detailed>.
- [137] *Linux UPS Without Tears*. URL: <https://www.crn.com/recipes/blogs/199000818/linux-ups-without-tears.htm>.
- [138] *13. Daemon Processes - Advanced Programming in the UNIX® Environment, Third Edition [Book]*. URL: <https://www.oreilly.com/library/view/advanced-programming-in/9780321638014/ch13.html>.
- [139] *UPSC(8)*. URL: <https://networkupstools.org/docs/man/upsc.html>.
- [140] *UPSRW(8)*. URL: <https://networkupstools.org/docs/man/upsrw.html>.
- [141] *UPSCMD(8)*. URL: <https://networkupstools.org/docs/man/upscmd.html>.
- [142] *UPSMON(8)*. URL: <https://networkupstools.org/docs/man/upsmon.html>.
- [143] *UPSSCHED(8)*. URL: <https://networkupstools.org/docs/man/upssched.html>.
- [144] *Debian - Package - nut*. URL: [https://packages.debian.org/cgi-bin/search\\_packages.pl?searchon=names&version=all&exact=1&keywords=nut](https://packages.debian.org/cgi-bin/search_packages.pl?searchon=names&version=all&exact=1&keywords=nut).
- [145] *Battery Basics - Progressive Dynamics*. URL: <https://www.progressivedyn.com/service/battery-basics/>.
- [146] *What is RAID and how does it work with my ReadyDATA storage system? | Answer | NETGEAR Support*. URL: <https://kb.netgear.com/23529/What-is-RAID-and-how-does-it-work-with-my-ReadyDATA-storage-system>.
- [147] *RAID level 0, 1, 5, 6 and 10 | Advantage, disadvantage, use*. URL: <https://www.prepressure.com/library/technology/raid>.
- [148] *Logwatch - ArchWiki*. URL: <https://wiki.archlinux.org/index.php/Logwatch>.
- [149] *The Benefits of Logging Tools | Loggly*. URL: <https://www.loggly.com/use-cases/the-benefits-of-logging-tools/>.

- [150] *What is Remote Access? Connect to your Computer from Anywhere*. URL: <https://www.splashtop.com/what-is-remote-access>.
- [151] *What is an SSD? | Solid-State Drive Definition | Avast*. URL: <https://www.avast.com/c-what-is-ssd>.
- [152] *ltsp.conf - LTSP*. URL: <https://ltsp.org/man/ltsp.conf/>.
- [153] *Eastern Cape matric pupils fail to return 55 000 Samsung tablets to education dept | News24*. URL: <https://www.news24.com/news24/southafrica/news/eastern-cape-matric-pupils-fail-to-return-55-000-samsung-tablets-to-education-dept-20210422>.
- [154] "IST-Africa 2021 Final Programme SSIT". In: (). URL: [www.IST-Africa.org/Conference2021](http://www.IST-Africa.org/Conference2021).
- [155] *What is Docker? | Opensource.com*. URL: <https://opensource.com/resources/what-docker>.
- [156] *What is Docker—Container Registry | Oracle South Africa*. URL: <https://www.oracle.com/za/cloud-native/container-registry/what-is-docker/>.
- [157] *Wayland*. URL: <https://wayland.freedesktop.org/>.

# A LTSP Documentation

## A.1 LTSP Background

LTSP is free and open source project released in early 2000s that helps terminal computers boot from another computer. This document outlines the installation procedures for both the older version of LTSP, version 5 and the latest version 19+. For a more detailed overview of how LTSP works revisit Chapter 3 of the attached thesis.

## A.2 LTSP 5+ Installation procedures

Since LTSP clients require a graphical interface to do computing one can choose any GNU/Linux desktop e.g. Xubuntu, Ubuntu mate, Linux Mint etc. To save RAM for older hardware when running Ubuntu mate run the following commands.

```
apt purge --yes --auto-remove indicator-application mate-hud snapd  
apt install --yes synaptic
```

### A.2.1 Adding the LTSP5+ Personal Package Archive

```
add-apt-repository --yes ppa:ts.sch.gr  
apt update
```

### A.2.2 Installing the LTSP5+

The LTSP can either be installed in chrootless mode or chroot mode. The chrootless mode is recommended if both the server machine and LTSP clients share the same architecture.

### A.2.3 Installing in chrootless mode

Here the server machine image is used as the template for the clients and all changes on the server machine reflects on the LTSP clients without using chroot commands.

```
apt install --yes --install-recommends ltsp-server-standalone ltsp-client epoptes
      gpasswd -a ${SUDO_USER:-$USER} epoptes
      ltsp-update-image --cleanup /
```

### A.2.4 Installing in chroot mode

```
apt install --yes --install-recommends ltsp-server-standalone epoptes
      gpasswd -a $SUDO_USER:-$USER epoptes

ltsp-build-client --purge-chroot --mount-package-cache --extra-mirror
      'http://ppa.launchpad.net/ts.sch.gr/ppa/ubuntu bionic main'
--apt-keys '/etc/apt/trusted.gpg.d/ts_sch_gr_ubuntu_ppa.gpg' --late-packages
      epoptes-client
```

## A.3 Network configuration

### A.3.1 Single NIC configuration

```
ltsp-config dnsmasq
```

### A.3.2 Dual NIC configuration

```
ltsp-config dnsmasq --no-proxy-dhcp wget
https://git.launchpad.net/sch-scripts/plain/debian/sch-scripts.if-up -O
/etc/network/if-up.d/sch-scripts chmod +x /etc/network/if-up.d/sch-scripts
systemctl restart network-manager
```

### A.3.3 LTSP configuration file

```
ltsp-config lts.conf
```

## A.4 Thick client support

```
ltsp-chroot -m apt install xubuntu-desktop ltsp-update-image
```

### A.4.1 Configuring thick client settings

Ubuntu 18 and the other versions moved from the traditional network manager and it is vital to learn to Netplan. Below is a sample template of how to configure network settings in Netplan so that the LTSP server can communicate with the thin and thick clients.

```
network:
  ethernets:
    enp0s316f:      #Connects to the public network
      dhcp4: true
      dhcp6: true
    enp4s0f1:      #Connects to the LTSP network
      dhcp4: false
      dhcp6: false
      addresses: [192.168.67.1/26]
      routes:
        - to : 0.0.0.0/0
          via: 192.168.67.1
          metric: 150
  version: 2
```

FIGURE A.1: An example of a dual NIC configuration.

## A.5 LTSP 19+ Installation procedures

### A.5.1 Adding the LTSP19+ Personal Package Archive

```
add-apt-repository ppa:ltsp  
apt update
```

For Debian-based distributions use the following commands

```
wget https://ltsp.org/misc/ltsp-ubuntu-ppa-focal.list -O  
/etc/apt/sources.list.d/ltsp-ubuntu-ppa-focal.list  
wget https://ltsp.org/misc/ltsp_ubuntu_ppa.gpg -O  
/etc/apt/trusted.gpg.d/ltsp_ubuntu_ppa.gpg
```

### A.5.2 Installing LTSP19+ packages

```
apt install --install-recommends ltsp ltsp-binaries dnsmasq nfs-kernel-server  
openssh-server squashfs-tools ethtool net-tools epoptes gpasswd -a administrator  
epoptes
```

### A.5.3 Single NIC

```
ltsp dnsmasq
```

### A.5.4 Dual NIC

```
ltsp dnsmasq --proxy-dhcp=0
```

## A.6 Maintaining the client image

There are three methods of maintaining the LTSP client's image namely chrootless, chroot and virtual machine.

### A.6.1 Chrootless mode

The chrootless maintains the LTSP client image by using the server machine operating system as the template.

```
ltsp image
```



## A.6.2 Chroot mode

The chroot maintains the LTSP client image within a separate chroot jail environment. This is done using chroot a CLI commands.

```
ltsp image x86_64
```

## A.6.3 Virtual machine

The raw virtual machine allows the image to be modified via a virtual machine.

```
ln -s "/home/user/VirtualBox VMs/xubuntu/xubuntu-flat.vmdk"  
/srv/ltsp/xubuntu.img ltsp image xubuntu
```

## A.7 Configuring iPXE

This generates the iPXE menu and copy the binaries in TFTP. The CLI needs to be run every time the LTSP client image is modified.

```
ltsp ipxe
```

## A.8 Configuring NFS

The NFS needs to be configured to allow the LTSP to serve LTSP client's images or run chroots.

```
ltsp nfs
```

## A.9 Generating ltsp.img

```
ltsp initrd
```

This is a new procedure that was not there in LTSP5 and it compresses the LTSP files in share and configurations directory and the public SSH keys to 'ltsp.img' this is transferred as an additional initrd. The CLI needs to be run after every LTSP update, new users and when the 'ltsp.conf' file is modified.

# B Appendix

## B.1 LTSP5+ Configuration file

# This file is served via TFTP from /var/lib/tftpboot/ltsp/<arch>/lts.conf when using the default NBD setup.

# If using NFS, it goes in /opt/ltsp/<arch>/etc/lts.conf. A list of directives is available at <http://manpages.ubuntu.com/lts.conf>

# Or your installed lts.conf man page. # Lines starting with '#' are comments.  
# Avoid empty [Sections] as they're considered syntax errors.

# This section applies to all clients and is overridden by the other sections.  
Default

# For troubleshooting, the following open a local console with Alt+Ctrl+F2.  
SCREEN\_02=shell  
SCREEN\_07=ldm

# Don't encrypt X traffic. Gives much faster graphics at a security cost.  
LDM\_DIRECTX=True

# Force all clients to a specific resolution.  
XRANDR\_MODE\_0=1024x768  
XRANDR\_RATE\_0=85

# If the default 16bit color mode is giving you problems on thin clients:  
X\_SMART\_COLOR\_DEPTH=False

# Separate fat clients from thin clients based on their RAM.  
FAT\_RAM\_THRESHOLD=400

# Use any local swap partitions on the clients.

USE\_LOCAL\_SWAP=True

# Provide a "Login as Guest" button, which uses LDM\_USERNAME/LDM\_PASSWORD.

LDM\_GUESTLOGIN=True

# Automatically login clients with the specified LDM\_USERNAME/LDM\_PASSWORD.

LDM\_AUTOLOGIN=True

# Map HOSTNAMEs to LDM\_USERNAMEs. E.g. for HOSTNAME=pc01,

LDM\_USERNAME=guest01.

HOST\_TO\_USERNAME="pc/guest"

# Some other frequently used directives, described in the lts.conf manpage.

# For really low RAM clients, disable SOUND, LOCAL\_APPS and LOCALDEV.

SOUND=False

VOLUME=50

LOCAL\_APPS=False

LOCAL\_APPS\_EXTRAMOUNTS=/srv

LOCALDEV=False

LOCALDEV\_DENY\_INTERNAL\_DISKS=False

FSTAB\_1="server:/home /home nfs defaults,nolock 0 0"

XSERVER=vesa

X\_BLANKING=0#

#X\_NUMLOCK=True

#XKBLAYOUT="us,gr"

#XKBOPTIONS="grp:alt\_shift\_toggle,grp\_led:scroll"

#LDM\_LIMIT\_ONE\_SESSION=True

#LDM\_LIMIT\_ONE\_SESSION\_PROMPT=True

#LDM\_PRINTER\_DEFAULT="psc-1310-series"

# You can define sections to be included later on with "LIKE" directives.

OLD\_MONITOR

# The following directives are useful in cases where the driver can't detect

# the (possibly old CRT) monitor timings and only offers low resolutions. #

Consult your monitor manual for the correct values, those are for 1024x768.  
X\_HORZSYNC=30.0-88.0 X\_VERTREFRESH=50.0-88.0

```
# You can specify directives per client with MAC address (or IP or hostname)
# based sections. # One way to get their MAC addresses is to run 'arp -n'
when they're at LDM. #[A1:B1:C2:D4:E5:F6]
#HOSTNAME=pc01
#LDM_USERNAME=user01
#LDM_PASSWORD=pass01
#LTSP_FATCLIENT=False
# Include another section.
LIKE=OLD_MONITOR
```

## B.2 LTSP19+ Configuration file

### [server]

```
# Enable NAT on dual NIC servers
```

```
NAT=1
```

```
# Provide a full menu name for x86_32.img when 'ltsp ipxe' runs
```

```
IPXE_X86_64_IMG="Xubuntu"
```

```
# The special common section is evaluated by both the server and ltsp clients.
```

### [common]

```
# Specify an alternative TFTP_DIR
```

```
# TFTP_DIR=/var/lib/tftpboot
```

```
# In the special [clients] section, parameters for all clients can be defined.
```

```
# Most ltsp.conf parameters should be placed here.
```

### [clients]

```
LIGHTDM_CONF="greeter-hide-users=true"
```

```
GDM_CONF="greeter-hide-users=true"
```

```
# Specify an /etc/fstab line for NFS home; note this is insecure
```

```
# FSTAB_HOME="server:/home /home nfs defaults,nolock 0 0"
```

```
# MAC address, IP address, or hostname sections can be used to apply settings
```

# to specific clients.

**[61:6c:6b:69:73:67]**

# HOSTNAME=pc01

# Include parameters from another section, defined below.

- # INCLUDE=crt\_monitor

# Shell "case" expressions can be used in MAC, IP, or hostname sections. #  
This matches all Raspberry Pi MAC addresses.

**[b8:27:eb:\* | dc:a6:32:\***

# FSTAB\_BOOT="/dev/mmcbk0p1 /boot vfat defaults 0 2"

# If you have proper DNS, you can use hostname sections

**[administrator-pc]**

# Only allow the administrator to log in to this client # PWMERGE\_SUR="administrator"

# You can also group parameters into named sections and INCLUDE= them  
elsewhere [crt\_monitor] # Force EDID and resolution to 1024x768 for clients  
with old CRT monitors X\_HORIZSYNC="28.0-87.0"