Deployment and Operational Aspects of Rural Broadband Wireless Access Networks

Giacomo Bernardi

Doctor of Philosophy

Institute of Computing Systems Architecture
School of Informatics
University of Edinburgh

2012

Abstract

Broadband speeds, Internet literacy and digital technologies have been steadily evolving over the last decade. Broadband infrastructure has become a key asset in today's society, enabling innovation, driving economic efficiency and stimulating cultural inclusion. However, populations living in remote and rural communities are unable to take advantage of these trends. Globally, a significant part of the world population is still deprived of basic access to the Internet.

Broadband Wireless Access (BWA) networks are regarded as a viable solution for providing Internet access to populations living in rural regions. In recent years, Wireless Internet Service Providers (WISPs) and community organizations around the world proved that rural BWA networks can be an effective strategy and a profitable business.

This research began by deploying a BWA network testbed, which also provides Internet access to several remote communities in the harsh environment of the Scottish Highlands and Islands. The experience of deploying and operating this network pointed out three unresolved research challenges that need to be addressed to ease the path towards widespread deployment of rural BWA networks, thereby bridging the rural-urban broadband divide. Below, our research contributions are outlined with respect to these challenges.

Firstly, an effective planning paradigm for deploying BWA networks is proposed: incremental planning. Incremental planning allows to anticipate return of investment and to overcome the limited network infrastructure (e.g., backhaul fibre links) in rural areas. I have developed a software tool called `IncrEase` and underlying network planning algorithms to consider a varied set of operational metrics to guide the operator in identifying the regions that would benefit the most from a network upgrade, automatically suggesting the best long-term strategy to the network administrator.

Second, we recognize that rural and community networks present additional issues for network management. As the Internet uplink is often the most expensive part of the operational expenses for such deployments, it is desirable to minimize overhead for network management. Also, unreliable connectivity between the network operation centre and the network being managed can render traditional centralized management approaches ineffective. Finally, the number of skilled personnel available to maintain such networks is limited. I have developed a distributed network management

platform called `Stix` for BWA networks, to make it easy to manage such networks for rural/community deployments and WISPs alike while keeping the network management infrastructure scalable and flexible. Our approach is based on the notions of goal-oriented and in-network management: administrators graphically specify network management activities as workflows, which are run in the network on a distributed set of agents that cooperate in executing those workflows and storing management information. The `Stix` system was implemented on low-cost and small form-factor embedded boards and shown to have a low memory footprint.

Third, the research focus moves to the problem of assessing broadband coverage and quality in a given geographic region. The outcome is `BSense`, a flexible framework that combines data provided by ISPs with measurements gathered by distributed software agents. The result is a census (presented as maps and tables) of the coverage and quality of broadband connections available in the region of interest. Such information can be exploited by ISPs to drive their growth, and by regulators and policy makers to get the true picture of broadband availability in the region and make informed decisions. In exchange for installing the multi-platform measurement software (that runs in the background) on their computers, users can get statistics about their Internet connection and those in their neighbourhood.

Finally, the lessons learned through this research are summarised. The outcome is a set of suggestions about how the deployment and operation of rural BWA networks, including our own testbed, can be made more efficient by using the proper tools. The software systems presented in this thesis have been evaluated in lab settings and in real networks, and are available as open-source software.

Acknowledgements

I am very thankful to the communities around Loch Hourn, and in particular to some `Tegola` users: Finlay and Anna MacKenzie, Peter Carr, Peter Fletcher, Martin Davies, Ewen Ballantyne, Rick Rohde, Iain Wilson, Mick Simpson. Thanks to Martainn Domhnallach, for the precious assistance at the Sabhal Mor Ostaig campus, our Internet uplink. You all made me experience why the Highlands are so special. After all, Beinn Sgritheall is not that different from Monte Rosa.

Thanks to Dmitry Rykovanov and Francesco Talamona for helping me implement the `IncrEase` system.

For the work on the `Stix` system, I am grateful to Alex Macmillan and Matt Calder, who helped me code a software prototype, and to Robert Macgregor (University of Edinburgh) for his assistance in developing the `StixControl` board.

For the `BSense` project, many thanks to Alessio Botta, Walter de Donato and Antonio Pescapè (University of Naples 'Federico II') for the continuous support on their D-ITG tool.

I am thankful to my supervisor Mahesh K. Marina, who taught me how to do research, guided me along this journey and painstakingly brought me back on track.

I am heavily indebted to Peter and Annie Buneman, for the mixture of academic guidance, algorithmic tips, Highlands hiking and sailing, scaffolding hacking and warm meals they offered me.

My admiration goes to the smart people at NGI SpA. Their example was inspiration for the topics discussed in this thesis. Everyday they demonstrate that the rural ISP business is profitable and socially useful.

A toast with my flatmates, friends and colleagues Sofia Pediaditaki and Damon Fenacci (also co-author of several of my publications), who shared all the ups and downs of student life.

I am grateful to my family and all my friends in Masnago. Thanks to Luca Polinelli and Davide Franzetti for being there when I needed. And to Rita Kosmidou, the best result of my academic research.

This work was funded by the EPSRC, The University of Edinburgh Development Trust, and the School of Informatics IDEAlab.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified. Some of the material used in this thesis has been published in the following papers. For publications with co-authors other than my advisors, their contribution is limited to helping with the implementation of the respective software tools.

- Giacomo Bernardi, Peter Buneman and Mahesh K. Marina. *Tegola Tiered Mesh Network Testbed in Rural Scotland*. In the proceedings of the ACM Mobicom workshop on Wireless Networks and Systems for Developing Regions (WiNS-DR). September 2008.

- Giacomo Bernardi, Francesco Talamona, Dmitry Rykovanov, Mahesh K. Marina. *IncrEase: A Tool for Incremental Planning of Rural Fixed Broadband Wireless Access Networks*. In the proceedings of the IEEE GLOBECOM workshop on Rural Communications (RuralComm). December 2011.

- Giacomo Bernardi, Matt Calder, Damon Fenacci, Alex Macmillan, and M. K. Marina. *Stix: A Goal-Oriented Distributed Management System for Large-Scale Broadband Wireless Access Networks.* In the proceedings of ACM Mobicom. September 2010.

- Giacomo Bernardi, Damon Fenacci, Mahesh K. Marina and Dimitrios P. Pezaros. *BSense: A Flexible and Open-Source Broadband Mapping Framework*. In the proceedings of the International Conference on Networking. May 2012.

(*Giacomo Bernardi*)

In memory of Carlo and Stefano.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Broadband Divide

Over the last decade, broadband markets have been growing in lock step with the increase of Asymmetric Digital Subscriber Line (ADSL) availability. Among the various broadband access technologies, ADSL alone accounts for more than 60% of broadband connections in OECD countries[1]. In everyday language, ADSL has even become a synonym for 'broadband'.

Key to ADSL's success is that it capitalises on the existing telephone network. However, the exact same reason is a prime cause of "digital divide" between metropolitan and rural areas. As the maximum transmission rate of an ADSL connection is a function of the distance between the phone exchange and the user (often referred to as the 'last mile'), the resulting user experience of real world ADSL connections varies significantly. According to (OFCOM, 2009), the theoretical maximum speeds at 3km and 5km are about 7Mb/s and 2Mb/s respectively, but surveys of what people actually get show that the situation is much worse. For example, Figure 1.1 presents the average download speeds achieved by panelists of a Ofcom broadband speeds study in the UK as a function of their distance from the local phone exchange: above 3km of 'last mile' distance, speeds quickly degrade below 2Mbps. Population density and distance between exchanges are also related, with sparsely populated areas having less and scattered exchanges, which results in lower ADSL speeds.

The foregoing discussion suggests that ADSL is unlikely to be the solution for pro-

[1] Source: OECD 2010 statistics, `http://www.oecd.org/sti/ict/broadband`.

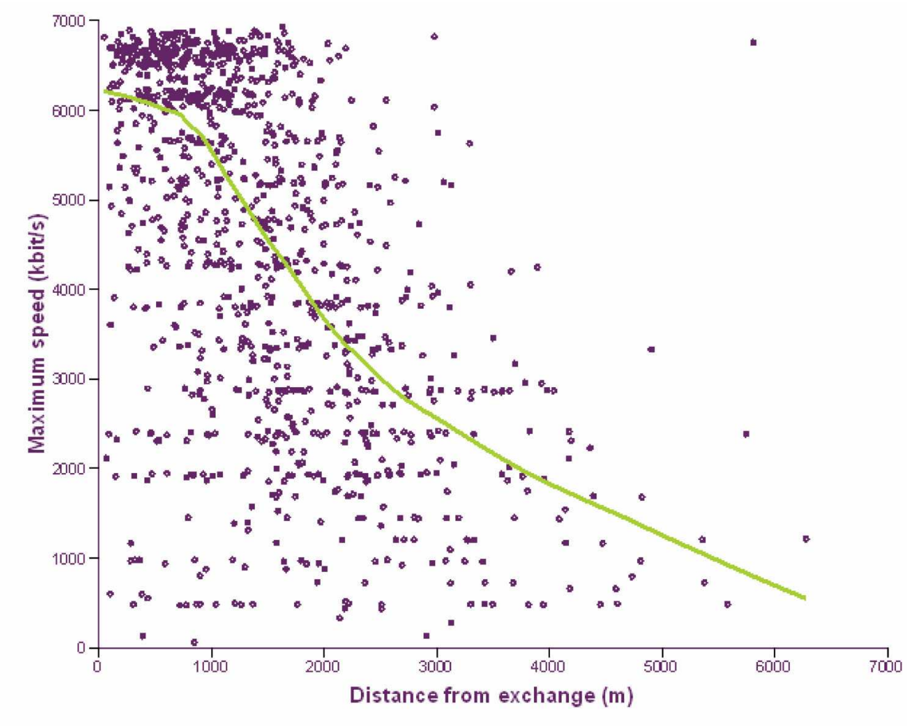


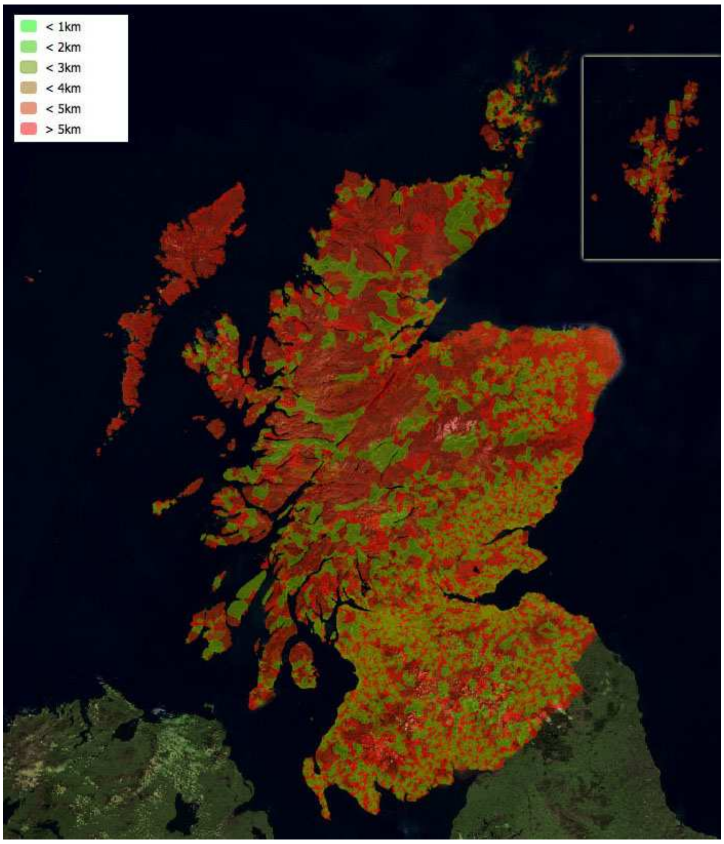
Figure 1.1: *Distance from phone exchange and average download speeds achieved. Source: (OFCOM, 2009).*

viding universal broadband. Scotland is taken as an example for further illustration of this statement. Using a public database of the incumbent phone exchanges, I associate each of the 211,600 postcodes in Scotland[2] with the phone exchange covering it. Since telephone cables are typically laid alongside roads, travel distance is used as an approximation of the local-loop[3] length. Figure 1.2(a) shows the distribution of households in Scotland with respect to their road distance from the closest exchange. I observe that in about 20% of the cases (shown in red) the local-loop is longer than 3km, as detailed in Figure 1.2(b). In these cases, speeds above 2Mbps are unlikely. Similarly, the chances are that people living farther than 5km from their phone exchange will not be able to get any Internet access service at all.

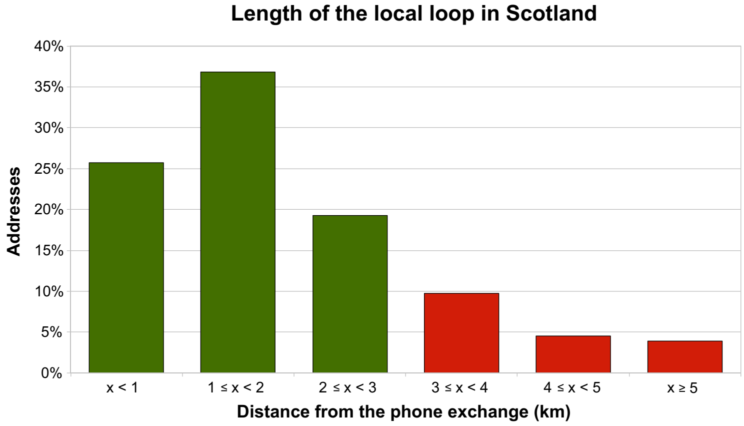
It is worth noting that besides the local loop length, there are several other factors that our calculations did not consider but still have a bearing on ADSL coverage and

[2]Corresponding to 2.65 million households and over 120,000 small business locations.

[3]The *local-loop* is the physical connection between the customer's premises and the edge of the telecommunications provider's network (i.e., the phone exchange).



(a)



(b)

Figure 1.2: *Distance to the phone exchange in rural areas: the example of Scotland.* (a) *Scarcely populated areas are farther away from exchanges.* (b) *Distribution of the local-loop length in Scotland shows that around 20% of the dwellings are farther than 3km from their phone exchange.*

speeds, including: poor quality cables and joints that can dramatically reduce achievable broadband speeds; noise from external and unpredictable factors such as "cross talk" of signals between different lines; the contention between users accessing the same exchange can sometimes saturate the backhaul connection of the Internet Service Provider (ISP). Finally, users in covered areas may still be unable to get ADSL broadband if the phone exchange is "full" – that is if all ports on the communication equipment at the exchange are occupied.

Like ADSL, fibre-based (e.g., FTTB, FTTC, FTTH, etc.) and cable technologies commonly reach only metropolitan areas, as they have an inherently high deployment cost per subscriber. Two-way satellite access, although available virtually anywhere and often subsidised to the most disconnected communities (including in Scotland), comes with a very high round-trip time latency which makes it unsuitable for many Internet applications, including interactive voice and video (e.g., Skype).

As suggested so far, while it is common to assume that digital divide mainly concerns the wide disparity in information access between developed and the developing nations, a similar problem exists even within developed countries between urban and rural areas though the latter problem seems relatively easier to address given the infrastructure and cultural acclimation to technology (Alfonsi, 2006). Without easy and affordable access to information and communications technologies (ICT) like in urban areas, rural communities are severely disadvantaged in several ways (e.g., children's access to educational resources, economic development opportunities), which may potentially lead to their eventual migration to urban areas. According to (Baran, 2000), narrowing the gap to Internet access can enable education equalisation, ultimately leading to a more peaceful world.

Such growing recognition of the detrimental impact of the digital divide is also evident from the Tunis Commitment of the United Nations (UN) sponsored World Summit on the Information Society (WSIS)[4]. The level of broadband penetration is a significant driver for economic growth and high-speed Internet access is rapidly becoming a necessity for the society: broadband networks have gained the same status as streets, water and energy networks. However, despite the increasing public pressure for an accelerated roll-out of broadband, the urban/rural 'gap' is still evident from recent statistics (see Figure 1.3). People living in rural areas suffer the effect of the digital

[4] `http://www.itu.int/wsis/docs2/tunis/off/7.html`

divide even more than urban dwellers, because they are unable to access services such as distance learning, e-health and e-commerce.

Our focus, in particular, is on rural-urban divide in terms of Internet access, which is a small but crucial element of the larger digital divide. By all accounts, rural areas lag behind their urban counterparts in broadband Internet access even in developed countries, with people living in such areas having fewer choices and paying higher prices for slower speeds. This view is also reiterated in an OECD broadband report (OECD, 2008), which analyses differences within the member countries. The root of the problem lies in the fact that rural areas have low user density and large distances between user clusters (Brewer, 2005; Subramanian, 2006), which makes it prohibitively expensive to deploy wired access technologies such as those seen in urban areas (e.g., ADSL, cable, fibre) unless mandated and heavily subsidised by governments. That leaves wireless as the only viable technology approach in the foreseeable future and there seems to be widespread consensus on this. Wireless technology, with its low cost commodity hardware and operation in the unlicensed spectrum lends itself as a natural, readily available, low cost and easily deployable alternative. For example, the IEEE 802.11 standard adapted from its original use as a technology for indoor wireless local area networks to work in outdoor scenarios over large areas, thanks to the addition of mesh networking capabilities and high gain directional antennas for enabling long distance links. The recognition that blanket coverage and mobility support are not needed in rural areas also work in its favour. Therefore, not surprisingly, wireless has become the *de facto* technology choice among researchers and communities seeking broadband Internet access.

1.2 Broadband Wireless Access Networks

Phil Edholm, chief technology officer of Nortel, described in 2004 a long-term trend in the history of wireless technologies (Cherry, 2004), which is illustrated in Figure 1.4: the growth rate of wireless communications is faster than that of wireline technologies. By plotting data rates logarithmically against time, wireline, nomadic and mobile technologies maintain the same (linear) relationship, but the latter two have a higher slope. Looking forward, Edholm's law suggests a convergence between wireline and wireless rates, roughly around year 2030. At that stage, wireline communications will be much

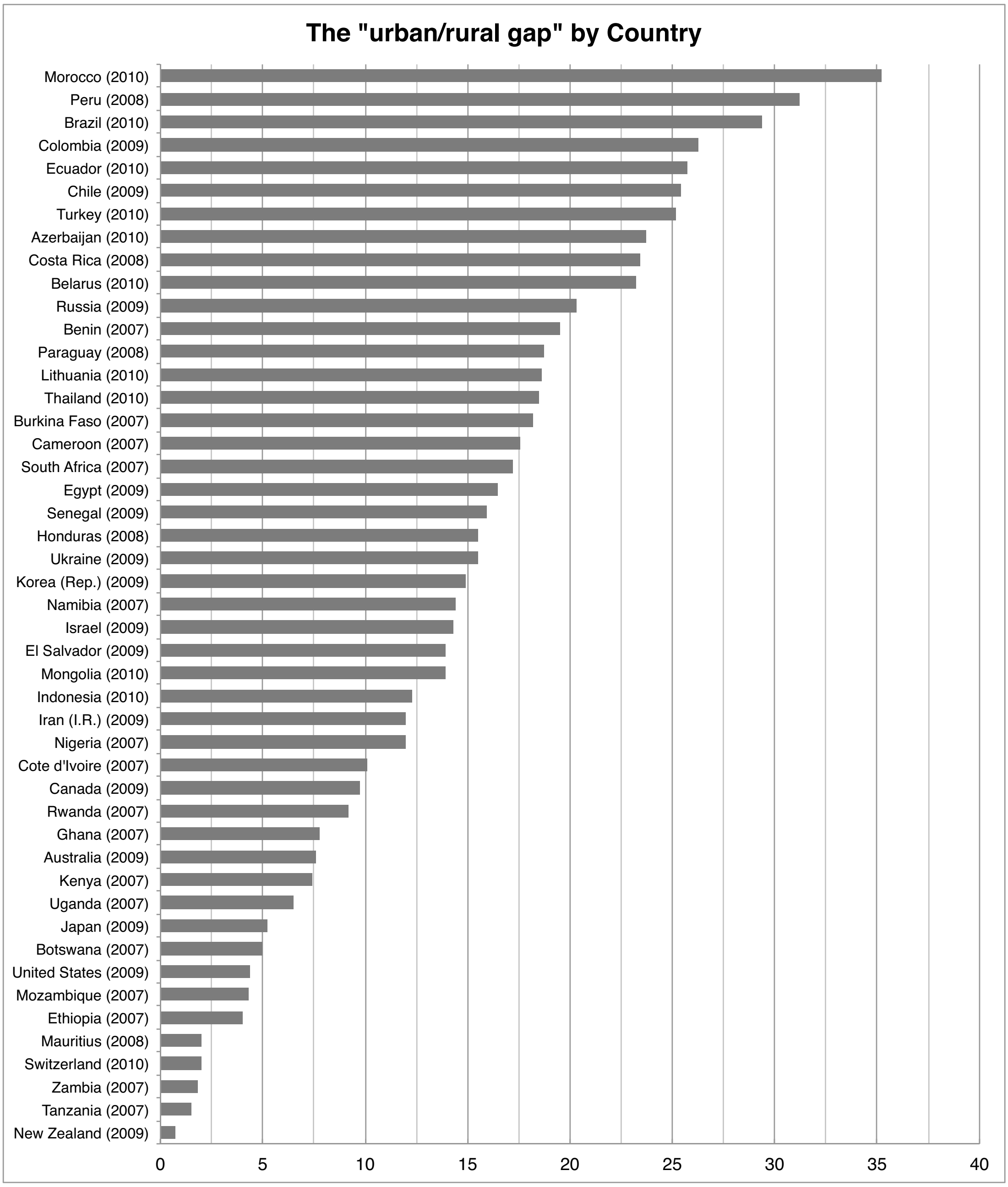


Figure 1.3: *The "urban/rural" gap, measured as difference between Internet penetration levels among individuals living in urban and rural regions of a selected set of countries. Note how this phenomenon is noticeable in both developing and developed markets. In parenthesis, the latest available year. Source of data: ITU-T (2011)*

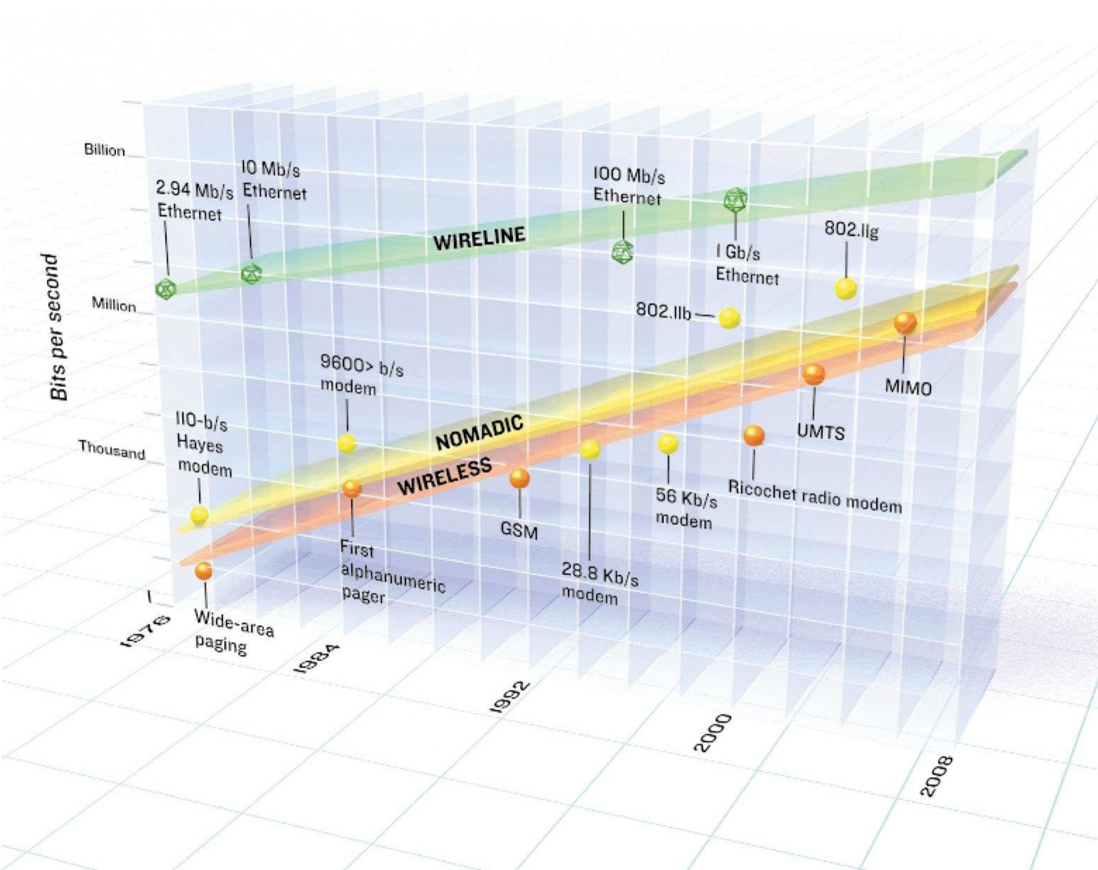


Figure 1.4: *Edholm's law shows the evolutionary trend of wired and wireless technologies, indicating a convergence in speed around year 2030. Source: IEEE Spectrum magazine, July 2004.*

less attractive than today, if the infrastructure cost of wired connectivity continues to remain higher than that of wireless.

Wireless offers financial and operational advantages over wireline access technologies, which are particularly important for rural deployments: it allows for a shorter time-to-market and reduced capital expenses, thanks to the lower infrastructure costs.

Broadband Wireless Access (BWA) is a generic term covering a wide spectrum of wireless technologies with different purposes and characteristics. Operators can choose from a wide range of products, which either implement industry standards (e.g., WiMAX, LTE, IEEE 802.11) or proprietary protocols. While each wireless technology has different capabilities (e.g., low latency, support for asymmetric traffic, point-to-multipoint support, etc.), operators commonly decide to organise their networks in two

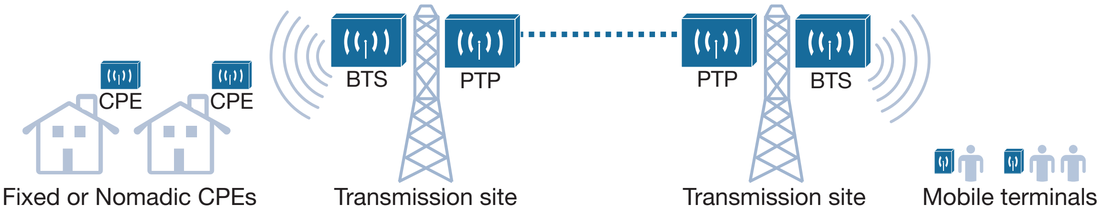


Figure 1.5: *Architecture of a typical two-tier BWA network.*

tiers: an *access tier* connecting the end customers with a transmission site, and a *backhaul tier* interconnecting towers among them to an Internet uplink. This is illustrated in Figure 1.5: in the backhaul tier, transmission sites are inter-connected using long distance point-to-point (PTP) wireless links, while an access tier provides connectivity to customer premises equipment (CPE) via a point-to-multipoint (PMP) link from the base transceiver station (BTS) located at a nearby transmission site. We note that there could be multiple BTSs at a transmission site, each using a sector antenna. It is also possible that some CPEs may act as relays for other CPEs that are outside the coverage area of a BTS.

1.2.1 Access Tier

The most peripheral section of a BWA network is the one that links the end users with a transmission site (e.g., a tower); this part of the wireless network is called the "access-tier". Most commonly, users are given a wireless device, which is either installed indoors or outdoors (e.g., on a rooftop or a window) and connects to the local transmitting station. The user device is often called *Customer Premises Equipment* (CPE), while the central device is a *Base Transceiver Station* (BTS), or simply 'Base Station' (BS).

The resulting topology is a star, where a single base station shares the same medium (i.e., radio frequency) with all the subscriber stations. These point-to-multipoint (PMP) connections allows significant statistical multiplexing and capacity overselling, resulting in increased spectrum efficiency and lower capital costs.

1.2.2 Backhaul Tier

Traffic generated by the end-users is aggregated by the base stations and gathered at each transmission site. The backhaul tier is responsible for transferring such volume of traffic to an Internet uplink and back. Unlike traditional mobile networks, which are dominated by synchronous TDM protocols, the trend with BWA networks is to use all-IP/Ethernet backhaul solutions. The reasons are that IP backhauling matches the type of subscriber traffic, it allows for capacity overselling, and that Wireless ISPs (WISPs) are usually very familiar with IP. Many BWA operators adopt wireless technologies for the backhaul tier, connecting transmission sites over wireless point-to-point (PTP) links.

The key desirable features of backhauling technologies can be summarised as:

- *Capacity.* Backhaul to a single tail site[5] should easily scale to tens or hundreds of Mbps.

- *Low Latency.* Low end-to-end delay is required in order to achieve high capacity and enable real-time application and voice services.

- *Availability.* In order to achieve good availability levels, the operator must plan for network robustness, implemented by using redundant paths and link protection schemes. Also, forecasting the future environmental condition (e.g., rain) and traffic demand is hard, so point-to-point wireless links should be allow for graceful performance degradation, for example by using adaptive modulation techniques.

- *Limited footprint.* Operators need to achieve profitability while dealing with multiple constraints, such as limited cabinet space, power and air conditioning availability, real estate costs, etc. Full-outdoor equipment is particularly attractive for rural WISPs that operate under limited profitability and in smaller or exposed outdoor setups.

Current backhauling products offer capacity in the region of 300Mbps to 800Gbps, with latencies lower than 10ms for distances of several tens of kilometres.

[5] A *tail site* is a transmission tower connected to the rest of the network with a single backhaul link.

1.3 Thesis Statement

Lack of software tools for design, management and evaluation of broadband wireless access networks have hindered their widespread deployment despite their cost and operational advantages over other alternative broadband access technologies. Our aim is to develop a software suite to fill this gap, with particular emphasis on rural and developing regions.

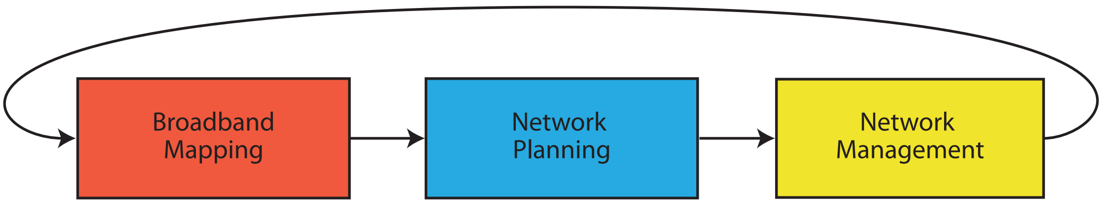


Figure 1.6: *Relationship between the three technical challenges addressed in this thesis in the context of broadband wireless access networks in this thesis.*

I now elaborate on three specific technical challenges tackled in this thesis: *Broadband Coverage and Quality Assessment* (henceforth referred to as *Broadband Mapping*), *Network Planning*, and *Network Management*. Together, these activities constitute the bulk of network operator activities. Moreover, they are very closely related (see Figure 1.6): a typical BWA network life-cycle starts by *assessing* the existing market to gauge the need for broadband connectivity in a given region. A first draft of the network is then *planned* by taking into account the existing infrastructure in the area (e.g., transmission towers, Internet uplinks). As soon as the network roll-out activities start, the devices deployed need to be *managed*. Finally, the operator performs tests and measurements to evaluate network performance (and consequent customer satisfaction), which may lead to further upgrades and expansion. In this section, I provide a high-level introduction to each of these three problems.

1.3.1 Broadband Mapping

Governments across the world are recognising the importance of broadband communications infrastructure in todays economy and society, with rural and underprivileged areas being the focus of policy statements that targets both broadband reach and speeds. Rural regions, which are still an untapped market, are an opportunity for ISPs. However, their limited profitability requires the operator to obtain a clear picture of where the demand is.

Over the last few years, several governments have sponsored large-scale efforts for broadband assessment[6]. These have been done in two ways, either by relying on ISP-provided data (e.g., estimated speeds and coverage) or by providing a sample set of broadband users with some kind of hardware box with customized software that periodically tests their broadband connection. Both of these approaches have limitations: while data provided by ISPs is inherently biased, a hardware based measurement is not only expensive but also only suitable for macroscopic analysis (e.g., at the country level). On the other hand, ISPs still largely rely on market research studies performed by consultancy agencies.

1.3.2 Network Planning

By 'network planning', I mean the process of planning the coverage, topology, and capacity of a BWA network before network equipment is purchased, configured and deployed. It is a key task for any WISP, but even more so for those operating in rural areas, which are faced with the unique challenge of extending their coverage on a tight investment budget in an environment of limited profitability.

The first task for any rural WISP[7] is to identify the most effective deployment strategy for a given capital investment budget. As population density in rural areas is typically scarce and populated areas are scattered (e.g., in towns and villages), the WISP first needs to get a clear picture of where the demand for service is concentrated, how to cover such profitable pockets and how to connect transmission towers back to an Internet uplink.

Beyond the initial deployment stage, the WISP can take two actions to extend its business: to increase the network coverage, or to improve it in areas already covered. In either cases, its moves are likely limited by budget and only a small subset of the potential actions can be addressed.

[6]Such as the $350 million US National Broadband Map initiative (`http://www.broadband.gov`) or the German Broadband Map project (`http://www.breitbandatlas.de`).

[7]Community-owned rural wireless networks also come under this category.

1.3.3 Network Management

Network management encompasses a wide range of activities, such as fault detection, configuration of new equipment, user accounting, performance profiling and the enforcement of security policies. BWA networks are inherently complex to manage, given the wide range of parameters involving both radio-frequency and environmental phenomena (e.g., radio propagation, mobility) in addition to those relevant in wireline network contexts.

Moreover, rural or community networks present additional difficulties, such as: Internet connection is expensive, so it is not desirable to have management traffic interfering with the efficient use of precious bandwidth; unreliable connectivity seen in rural networks means that remote devices may be inaccessible for troubleshooting; and the number of skilled personnel available to maintain such networks is very limited, strengthening the need for easy-to-use management tools.

1.4 Thesis Contributions

In this thesis, I focus on the above three crucial activities in the lifecycle of rural wireless networks: *network planning*, *network management* and *broadband mapping*. I highlight why these are hard problems, especially in the rural wireless context, and argue why current approaches to address these problems are insufficient. Our overall contribution is a suite of three software systems that has the potential to make the rural WISP business simpler and more efficient, ultimately reducing the digital divide.

1.4.1 Rural Wireless Testbed

At the start of my PhD research, I designed and deployed `Tegola`: a wireless testbed that also provides Internet access to some of the most remote communities in Britain. The network has been running since 2008. It operates on license-exempt spectrum and some transmission sites are powered by green sources (i.e., wind and solar). The contribution of this work is two-fold. From a scientific point of view, `Tegola` has been an invaluable tool to inspire and judge the rest of my research, evaluating each of the proposed tools against the actual needs of rural BWA networks. The relationship between our wireless testbed and each of the three software systems developed is illustrated in

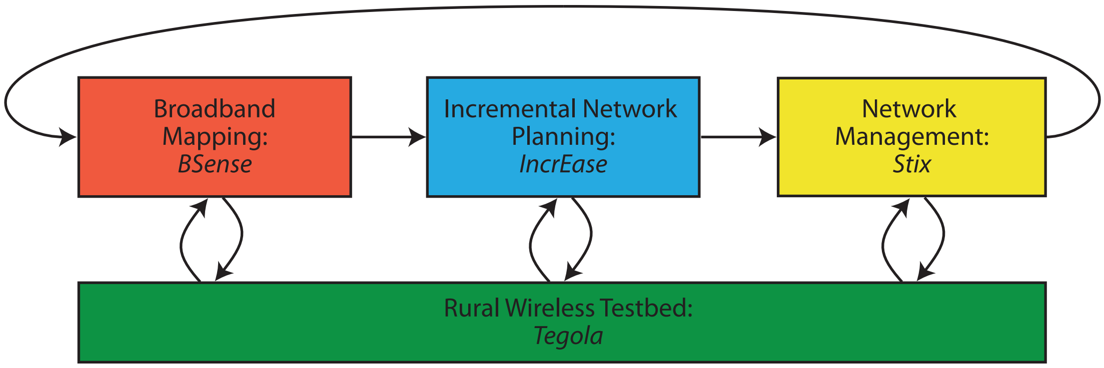


Figure 1.7: *Relationship among the main thesis contributions within the BWA network lifecycle.*

Figure 1.7: key problems encountered from experience of deploying and running the Tegola network led to new research ideas which were realised in the form of software prototypes, which were then brought back for evaluation on the testbed and on another BWA network to which we have access (NGI SpA, see Section 2.1.1). Secondly, from an engineering perspective, the work on Tegola resulted in a hardware and software system suitable to be deployed in very harsh environments.

Key Publication:

Giacomo Bernardi, Peter Buneman and Mahesh K. Marina. *Tegola Tiered Mesh Network Testbed in Rural Scotland*. In the proceedings of the ACM Mobicom workshop on Wireless Networks and Systems for Developing Regions (WiNS-DR). September 2008.

Testbed website:

`http://www.tegola.org.uk`

1.4.2 Incremental Network Planning

I address the network planning problem in a software system that takes an incremental planning approach. It considers a varied set of operational metrics to guide the operator in identifying the regions that would benefit the most from a network upgrade, automatically suggesting the set of "moves" (e.g., new transmission sites to deploy) that leads to the best long term deployment strategy. The key contribution is the design of two algorithms for automated planning, which are implemented in the `IncrEase` software tool. Evaluation has been performed on the network of a real-world WISP, NGI SpA, which is also currently using `IncrEase` in production, allowing me to test the application on a database of over 8,900 existing transmission towers.

Key Publication:

> Giacomo Bernardi, Francesco Talamona, Dmitry Rykovanov, Mahesh K. Marina. *IncrEase: A Tool for Incremental Planning of Rural Fixed Broadband Wireless Access Networks*. In the proceedings of the IEEE GLOBECOM workshop on Rural Communications (RuralComm). December 2011.

Source code and documentation are available at:

> `https://code.google.com/p/wimo-increase`

1.4.3 Network Management

The research focus is on the management aspects of BWA networks, especially those operated by community organisations and rural WISPs. I show why available network management systems (NMSs), often derived from the traditional wired domain and/or based on the centralised paradigm, are inadequate in the context of rural wireless deployments. A novel scheme for 'in-network and goal-oriented' management is then proposed, based on a visual programming language. Finally, I implement the proposed paradigm on the `Stix` platform, which uses distributed agents running on commodity embedded hardware. The evaluation of `Stix` takes place on both the Tegola network, where it has been deployed at each transmission site, and on NGI SpA's acccess network. `Stix` is the major contribution of this thesis.

Key Publication:

> Giacomo Bernardi, Matt Calder, Damon Fenacci, Alex Macmillan, and M. K. Marina. *Stix: A Goal-Oriented Distributed Management System for Large-Scale Broadband Wireless Access Networks.* In the proceedings of ACM Mobicom. September 2010.

Source code, documentation and hardware schematics are available at:

> `https://code.google.com/p/wimo-stix`

1.4.4 Broadband Mapping

I find that existing systems for assessing broadband availability and quality in a given region either rely on models or are based on customised measurement hardware. The former approach is useful from a coverage assessment but can be overly optimistic in terms of actual achievable broadband performance (e.g., speeds). The latter approach, on the other hand, is expensive (requires new hardware to be installed at the user location). There are also software based measurement solutions (e.g., web-based speed test tools) but they lack functionality for providing continuous analysis of a broadband line. Our proposed solution, called `BSense`, is a flexible broadband mapping framework that brings together the positive aspects of these abovementioned approaches to provide a census of broadband availability, choice and quality. `BSense` also incorporates a flexible broadband quality index that summarises the combined effect of various technical attributes (e.g., download/upload speeds and latency) to provide a single measure of user's perceived broadband quality. Both ISPs and governments can make use of such a broadband census system to make informed decisions about public policies and business strategies. `BSense` has been evaluated by recruiting volunteers living in Rural Scotland, around the region where the Tegola network is deployed.

Key Publication:

Giacomo Bernardi, Damon Fenacci, Mahesh K. Marina and Dimitrios P. Pezaros. *BSense: A Flexible and Open-Source Broadband Mapping Framework*. In the proceedings of the International Conference on Networking. May 2012.

Source code and documentation are available at:

`https://code.google.com/p/wimo-bsense`

1.5 Thesis Structure

The structure of rest of the thesis is as follows. The body of the thesis begins by describing the Tegola testbed, which helped to identify the key technical problems tackled. Then I discuss our solutions to the network planning and management problems. Finally, the system we designed for broadband mapping is described. While this may seem like a different order from what is shown in Figures 1.6 and 1.7, we present it towards the end because a unique aspect of our solution to broadband mapping is enabling broadband access quality measurement through a continuous stream of measurements.

Background and Related Work *(Chapter 2).* I describe the state of the art in three key research areas within BWA networks that are relevant to this thesis: the literature on wireless network planning and distributed network management is discussed, reporting on active projects aiming at broadband mapping. I also describe existing rural wireless testbed deployments and the network of a commercial WISP covering rural parts of Northern Italy. The latter network is used for some of our evaluations.

Tegola: Wireless Network Testbed in Rural Scotland *(Chapter 3).* The design and deployment of Tegola, our wireless testbed which provides Internet access to rural communities in the north-west of Scotland, is described. Based on the lessons learned, I identify and address three key problems in BWA networks, which are the focus of the rest of the thesis.

IncrEase: Incremental Network Planning *(Chapter 4).* I define incremental planning as the activity of identifying and connecting the geographical regions that would benefit the most from an expansion or upgrade of the existing BWA network. The proposed tool, `IncrEase`, considers a set of metrics (e.g., geographical distribution of users, transmission tower locations, etc) from the current network and visually guides the decision maker through the process of determining the best transmission locations and network topology to expand the current coverage.

Stix: a Goal-Oriented Distributed Management System *(Chapter 5).* I investigate the complexity of managing BWA networks and suggest a comprehensive solution based on distributed software agents running on low cost embedded hardware. Our `Stix` system lets the administrator set out the management goals in term of “workflows” using a visual programming language, which are then interpreted by a distributed set of agents deployed at strategic locations in the network (e.g., at each transmission site). `Stix` features a built-in hardware abstraction layer to run the same workflow on different network devices, and provides a system to generate real-time reports of the network behaviour.

BSense: A Flexible System for Broadband Mapping *(Chapter 6).* This chapter defines “broadband census” as the process of evaluating the availability, quality and choice of broadband connections in a given geographical region. Our contribution is a distributed system that performs configurable network experiments from user-installed agents. The performance metrics gathered are translated into a “broadband quality score” using a tuneable multi-attribute utility function, and presented to the interested party (be it a consumer, an ISP, or a decision maker such as a regulatory authority) as annotated geographical maps. Also, `BSense` enables broadband coverage analysis based on public data from ISPs and/or infrastructure locations (e.g., exchanges) and models of a broadband access technology (e.g., ADSL performance in comparison with distance).

Other Contributions *(Chapter 7)*. I present two other secondary contributions of my research, which lie outside the three main problems on which this thesis is focused.

Conclusions *(Chapter 8)*. I summarise this work and discuss possible directions for future research.

Chapter 2

Background and Related Work

This chapter sets the scene for the following ones, presenting previous work relevant to this thesis. The first section describes the most relevant wireless research testbeds. Section 2.2 focuses on wireless network planning, both for backhaul (also known as core) and access tiers. Section 2.3 presents existing research in the field of network management, focusing on the particular aspects of BWA networks. Section 2.4 gives an outline of the techniques for broadband mapping and of ongoing efforts for broadband mapping.

2.1 Rural research testbeds and deployments

Over the last decade, a large number of outdoor wireless testbeds have been deployed around the world. The availability of wireless testbeds enables researchers to develop and validate new protocols and applications in a real environment, with real users. Often, the usefulness of such wireless deployments goes beyond research, resulting in public access to the Internet to populations living in underserved areas or enabling applications such as tele-medicine or distance learning.

These testbeds can be broadly divided into two categories: *urban* and *rural* outdoor wireless testbeds, respectively targeting high and low user densities.

In the first category, several urban mesh networks in the form of municipal or community wireless networks (Mandviwalla et al., 2008; Kramer et al., 2006) as well as research testbeds (Bicket et al., 2005; Camp et al., 2006; Ormont et al., 2008; Angelakis et al., 2007) have emerged recently as an attractive alternative for broadband Internet access, offering wider coverage at less expense and ease of deployment. For the most part, these deployments are characterised by the use of omnidirectional antennas for communication over short distances and single or multiple radios per node.

More closely related to our research testbed in Scotland, presented in Chapter 3, are rural outdoor wireless networks. There exist a large number and variety of them, especially if we include deployments by communities and non-governmental organisations operating in the field of Information and Communication Technologies for Development (ICTD). The following table presents the prominent testbeds and deployments.

Continent	**Country**	**Project Name**	**Reference**
Africa	Ghana	CUWin Ghana's network	`http://www.cuwin.net`
Africa	Nigeria	Zittnet	`http://zittnet.net`
Africa	South Africa	Meraka Institute's mesh	(Johnson, 2007)
Africa	Zambia	LinkNet	(Matthee et al., 2007)
Americas	USA	Quail Ridge Wireless Mesh Network	(Wu et al., 2007)
Americas	Colombia, Cuba, Peru	EHAS	(Martinez et al., 2004)
Asia	India	Airjaldi's Dharamsala mesh	`http://airjaldi.com`
Asia	India	FRACTEL	(Chebrolu and Raman, 2007; Gokhale et al., 2008)
Asia	India	Ashwini Project	`http://www.ashwini.org`
Asia	India	Digital Gangetic Plains (DGP)	(Bhagwat et al., 2004; Sheth et al., 2007; Raman and Chebrolu, 2007)
Asia	India	Tier's Aravind Network	(Surana et al., 2008a)
Asia	Nepal	Nepal Wireless	`http://nepalwireless.net`
Europe	Denmark	DjurslandS	`http://djurslands.net`
Europe	Sweden	N4C Network	(Farrell et al., 2011)

Europe	UK	Wray Community Network	(Ishmael et al., 2008)

As in our case, some of these deployments employ an overlay of long distance links to cover a larger number of users over a wider area (Gokhale et al., 2008; Matthee et al., 2007; Sheth et al., 2007). Many others, especially in Africa, simply share a single Internet connection (e.g., VSAT satellite connections) among several users much like in a typical mesh network setting, albeit in rural terrains sometimes using planned long distance links with high gain directional antennas to obtain the necessary coverage. Some testbeds span both urban and rural areas (Gokhale et al., 2008; Sheth et al., 2007).

The impact of rural outdoor conditions, especially harsh weather, played a significant role in designing our wireless testbed in Scotland. Another unique characteristic of our deployment is the presence of multiple long-distance links over the sea, which are needed in order to reach costal communities: we are not aware of any wireless rural testbeds involving over-water links. While the presence of foliage has been reported (Gokhale et al., 2008), its impact is different and potentially lower compared to multipath reflections off changing water levels (including due to tidal patterns) and signal attenuation due to water absorption. As a result, the link characteristics observed in our setting are in sharp contrast with characterisations presented in the literature. For instance, Sheth et al. (Sheth et al., 2007) conclude that rural long distance WiFi links exhibit negligible loss rates as they experience little or no external interference from other WiFi/non-WiFi sources, and because of the very low multi-path interference and low delay spreads due to long, line of sight (LOS) links. As another difference, every other existing outdoor rural wireless deployment that requires self-powered masts relies on solar energy as those deployments are mostly in tropical climate areas. Following the same approach is not meaningful in our setting and would result in an expensive and bulky powering solution given the very different climatic conditions.

2.1.1 An example: NGI SpA network

NGI SpA is an Internet Service Provider operating a large BWA network in North and Central Italy. Over the last 5 years, NGI has deployed 1,400 radio sectors at around 600 transmission sites, with a customer base of over 100,000 customers.

NGI's network is structured on two tiers for access and backhaul. Transmission towers are interconnected via wireless point-to-point links ranging up to 60km and operating on a diverse set of unlicensed (i.e., 5 and 17 GHz) and licensed (i.e., 6, 7, 11 and 18 GHz) frequencies. Point-to-point links are arranged to create ring topologies in which towers have multiple backhaul routes to an Internet uplink.

The access network operates in the 5 GHz unlicensed spectrum[1] using Hiperlan/2 and 802.16e technologies. Customers receive an outdoor CPE which is installed by a trained technician, typically on the rooftop, and has to be in line of sight (LOS) with any one of NGI's towers in order for the technician to 'certify' the installation and enable the service.

In running its WISP business, NGI has followed these design principles:

- Incremental network roll-out: deployment started from areas that were not reached by other broadband technologies (e.g., ADSL), expanding then to more competitive urban areas. Such a strategy allows the WISP to fund the network expansion from the early profits, reducing reliance on external financial support.

- The WISP remained technology and vendor agnostic: instead of adopting a single technology vendor throughout the network, NGI decided to partner with brands specialised in either point-to-point or point-to-multipoint technologies. This decisions allowed the company to grow with more flexibility than telcos operating a larger mobile network, enabling the choice of the best technology. The drawback is an increased heterogeneity of the network, especially from the network management perspective.

- Only a few backhaul links are based on fiber technologies, in favour of wireless point-to-point links. The rationale for this decision is to keep operating costs

[1]Under Italian legislation, frequencies between 5.470 and 5.725 GHz are available for BWA connectivity under unlicensed regime, up to a maximum of 1W of Equivalent Isotropically Radiated Power (EIRP)

down, decouple network rollout from the availability of existing fibre and to reduce the dependence on incumbent and wholesale operators.

- All planning, deployment and management activities are operated internally, acquiring personnel and necessary skills, instead of outsourcing such jobs to external agencies. As a result, the company found it easier to develop an integrated software platform rather than adopting third-party software tools for each activity (e.g., planning, provisioning, management, assurance).

NGI's network is likely one of the largest deployments in Europe that operates in the unlicensed spectrum. In this thesis, we refer to it in the evaluation sections of `IncrEase` (Section 4.3) and `Stix` (Section 5.4), to demonstrate that such tools are appropriate for the real world.

2.2 Wireless network planning

The process of selecting the locations for wireless base stations (BS) is often referred to as *cell planning*, and it is likely the most fundamental and challenging part of the wireless network design process. The advent of mobile phones and the inherent complexity of the planning problem attracted a great deal of attention in the research community for many years. As a result, a myriad of tools have been proposed to automate the decision process and heuristic approaches (e.g., direct search, genetic algorithms, simulated annealing, vector quantisation) have been designed for both macro-cellular and indoor environments.

As suggested by St-Hilaire (St-Hilaire, 2009), the planning process can be addressed in two ways: either with a *modular* approach, by breaking the problem into multiple subproblems (cell planning, access network planning and core network planning) to be solved in sequence, or with a *global* approach by tackling more than one subproblem simultaneously in the hope of getting closer to the global optimum. With a coarser level of abstraction and assuming two-tier (i.e. access and backhaul) wireless IP networks such as those commonly deployed by WISPs, we can identify two large problems: *access-network planning* and *core-network planning*. The former requires finding the optimal number of base stations, their type, configuration and best location, while the latter deals with finding suitable multi-hop paths to interconnect transmission towers to one or several back hauling points. Each of these subproblems has been demonstrated to be NP-hard (Amaldi et al., 2001, 2003; Harmatos, 2002). Although attempts have been made to solve both problems at the same time, such as (Zhang et al., 2004), most work either falls into the first or second category. The following are examples addressing one of these two subproblems.

1. **Access-network planning**

   Cell planning is an extension of a well-studied problem: the capacitated facilities location problem with unsplittable demands (Daskin, 1995). Being a hard problem, meta-heuristic techniques have become increasingly popular. For example (Whitaker and Hurley, 2003) provides a complete review, categorising existing work between greedy, exact, genetic, hill-climbing, simulated annealing and tabu-search algorithms and comparing their pros and cons. A further comparison between heuristics is (Raisanen and Whitaker, 2005), which compares coverage and cost achievable by applying four genetic algorithms on a set of candidate locations and test points. Examples of tabu-search algorithms in access-network planning are (Amaldi et al., 2008), applied to 3G networks, and (Gordejuela-Sanchez et al., 2009) which finds the subset of BSes out of a set of candidate locations that optimises a multi-objective function including both economical and technical factors in WiMAX networks. Financial benchmarks and plug-in heuristics are also extensively used in (Hurley et al., 2010), which aims to design profitable networks by bringing capital expenses (CAPEX) and operational expenses (OPEX) into the planning problem. Similar heuristics have been used also for indoor wireless planning such as (Bosio et al., 2007), which formalises the "Minimum Overlap Problem": its idea is to limit the number of BS covering a user in order to reduce interference and increase capacity. On the other hand, a simple, although naive, base solution is the greedy approach presented in (St-Hilaire, 2009): it first tries to find a subset of BSs that maximises coverage, to ensure that the signal can reach the largest number of users, then it allocates different channels to the BSs in order to minimise interference and allow for frequency reuse in other parts of the network.

2. **Core-network planning**

   Once suitable antenna locations have been suggested, the problem is to determine how to connect them (typically by using wireless point-to-point links) so that they all have a path to a network backhaul point. Additionally, redundancy constraints may be imposed to ensure transmission towers have more than one path to the Internet uplink. (Bu et al., 2005) tries to identify a topology of backhaul point-to-point 802.16 links that minimises the number of such radio links while meeting the expected traffic demand. The problem is shown to be NP-hard, and a greedy algorithm is proposed to produce close to the optimal solution both when no path redundancy is needed and in the case of N+1 redundancy.

In the rural domain, (Sen and Raman, 2007) proposes a four-step scheme to determine the network topology, tower heights, antenna types and radio transmit power. The paper provides several interesting design considerations to minimise costs, such as the reduction in tower heights as they often represent the largest investment for a rural WISP. (Panigrahi et al., 2008) tackles a similar problem: constructing a topology for inter-village rural mesh networks with a greedy approximation algorithm.

2.2.1 “Incremental” network planning

In Chapter 4 we introduce our software tool for incremental network planning. The use of the term “incremental” is not new in this context. For example, in the context of capacity planning for wide area networks, (Charzinski and Walter, 2006) compares various strategies for deciding which links to upgrade in order to match the demand. The proposed approach, however, assumes the topology to be fixed, with link capacity being upgraded based on demand growth. This is in contrast with our scenario, where the best topology is to be determined. Similarly, (Reininger et al., 1999) draws an interesting parallel between radio networks (and, in particular, cellular networks) and the life of a living organism: they both go through different ages, such as gestation, birth, growth, and death. In operating radio networks, these phases correspond to planning, deployment, coverage extension and network saturation. From this observation, the authors suggest an “incremental” approach: the first step is to dimension and estimate the cost of a first-cut deployment, which is then refined and expanded to improve the network quality and capacity. The paper proposes a Genetic Algorithm technique for identifying the best strategy. However, in contrast with our approach, the location of CPEs and base stations are fixed and provided as input, and the algorithm does not consider their bandwidth demand.

In the literature, there are two other trends closely related to incremental planning. The first is that of *multi-period planning*, that is the activity of designing network topologies by looking at the budget available to cover the investment in each period (e.g., year) of activity. An example is (Meusburger et al., 2008) which, in the field of optical networks, gives an overview of common multi-period planning approaches and compares the costs of each. The authors emphasise that planning conditions often change significantly over time (e.g., capacity demand can grow, technologies develop, equipment costs decrease) and propose an integer linear optimisation model to identify the best allocation of a restricted investment budget. A further example is (Kubat and Smith, 2001), which addresses topological design for a multi-period cellular system, calculating the interconnection between cells and the core networks in each planning period. However, the algorithm proposed assumes that the location of cells, their demand, and the backbone hubs are given. In general terms, multi-period planning algorithms follow one of five approaches:

- All-period planning, which minimises the network costs over all periods of time. Theoretically, it is an optimal overall solution, but requires the demand forecast for all future periods to be known and correct, which is rare. For this reason, it is used mainly for the initial periods of operation, when accurate forecasts are possible.

- Incremental Planning, which assumes that the network planner only knows the demand for the next time period. In this case, the network topology can only be calculated sequentially year upon year.

- Begin of Life planning, which is a combination of the previous two approaches: often, the network operator can project the demand forecast for the first few years. It is a more practicable model than the all-period approach.

- End of Life planning: it optimises the network for a certain period in the future, reaching the presumed “end of life” of the network, and only requires the demand forecast for this last period. The network is then planned incrementally from the start, in order to converge towards the end-of-life solution.

- Budget-Restricted planning, which acknowledges that networks costs are often restricted to a specific budget per period, which can be used to build and upgrade the network. Algorithms following this approach suggest how to use the remaining budget for equipment purchases strategic for future periods.

A second related trend is *buy-at-bulk network design* (Awerbuch and Azar, 1997; Andrews and Zhang, 1998; Meyerson et al., 2000; Salman et al., 2000), which derives from the design of traditional telephone networks. Buy-at-bulk assumes that different link types (e.g., fiber links, microwave point-to-point, leased lines, etc.) are available, each having a fixed per length cost and some capacity. Provided the demand requirement at each network node, we need to lay a collection of links along pair of nodes in order to install sufficient capacity to route all demands to their respective destinations. Link costs obey economies of scale, in the sense that the cost of routing a unit of demand along a cable with larger capacity is less than using many links of smaller capacity. The goal is to design a minimum cost network that allows all demands to be routed. A special case, more similar to our BWA scenario, is the "Access Network Design" problem formulated in (Andrews and Zhang, 1998), where all demands need to be routed to a central core network (one or more Internet uplinks, in our case). In this field, the closest piece of work to our scenario is (Meyerson et al., 2001). The authors stress that the network provisioning process must be done incrementally, as demand constantly changes and the existing infrastructure needs to be upgraded or moved to cope with the new conditions. Their goal is to plan a network so that each existing node has a path to its sink at all times, with the requirements that already deployed links should not change because of new demand (although it is allowed to add additional links along existing paths to route new demand), demand paths should not be re-routed, and links already laid should not be removed. Despite the similarity in terminology with our formulation, our problem formulation is somewhat orthogonal to that of buy-at-bulk network design algorithms: while algorithms in that category are given the exact location of demand nodes, our aim is first to estimate where 'demand'[2] is and then to strategically plan a two-tier (access and backhaul) network. Also, when designing rural BWA networks in which most of the backhaul links are wireless, it is

[2] In contrast with the meaning traditionally attributed to 'demand' in the networking literature, in this chapter we vaguely intended it as the desire of people in a geographical region to subscribe to the service.

unpractical to assume that any pair of network nodes (i.e., transmission masts) can be connected: orography and distance can substantially limit the link feasibility.

2.3 Distributed network management

Pavlou (Pavlou, 2007) offers a good survey and classification of management approaches that have appeared over the years. Broadly, these fall into two categories: management by remote invocation and management by delegation. These two categories roughly correspond to centralised and distributed management approaches, respectively. Approaches based on remote invocation can be further divided into manager-agent based (e.g., SNMP, NetConf) or distributed object and service interface based (e.g., CORBA, JRMI, web services). The essential idea behind management by delegation is to move the managing entity closer to the managed device. The approach of our management system `Stix` (described in Chapter 5) falls between the two extremes of management by delegation approach (i.e., manager-agent based and full mobile code based), so can be referred to as a distributed cooperative agent based approach (Pras et al., 2007).

Starting from the early 90s, a large number of distributed architectures have been proposed along the directions of agent-based networking. Among the earliest approaches are Active Network technologies (Tennenhouse and Wetherall, 1996; Tennenhouse and Smith, 1997), funded by DARPA with the aim to create more intelligent networks by introducing dynamic network programming and allowing routers to execute binary code received in packets. The main benefit from the viewpoint of an ISP service provider is that Active Networks can reduce the time to develop and deploy new services and to experiment with new applications while preserving network continuity. Such paradigm has also been applied to network management in the 'Smart Packets' project by BBN Technologies (Schwartz et al., 2000): the idea is that modern router architectures tend to have a large amount of idle processing power on the control plane that could be used to dynamically run network applications. It is interesting to note that the advantages identified by the authors are very similar to those of `Stix`: (i) the amount of back traffic to the management center is reduced; (ii) new management activities can be run directly at the nodes; (iii) the 'control loop' becomes shorter as control operation can be taken without intervention of the management station. Another architecture proposed to increase the flexibility of the management agents running on network devices was Netscript (Purakayastha and Mohindra, 1998), developed at the IBM Watson Research Center around year 2000. Its authors see the deployment of agents as *"a fixed contract between the management server and the managed clients"* that severely limits flexibility in administrative procedures. Their solution is instead based on a tiny generic runtime environment that automatically downloads programs written in BASIC-like scripting language on the device. In IBM's original vision, Netscript was intended to become a universal language for managing active networks.

During the same period in which Active Networks were developed, an alternative approach was initiated by the work of Goldszmidt on *Management by Delegation* (MbD) in (Goldszmidt and Yemini, 1995). The underlying principle is that management functions can be effectively delegated to the network elements and executed locally rather than centrally: instead of moving information from the devices to the management platform, MbD moves and runs applications at the devices where the data is. It addresses the scalability limitations of centralised management systems by adopting an architecture whereby delegation agents could be sent from servers to clients, which could then invoke delegation procedures stored on other clients. Finally, to ease the development of practical implementations, Goldszmidt suggested the development of *'Elastic processing'*, a distributed technology that enables applications to move and modify their code around agents. The benefit of MdD were identified in two directions: to reduce the network bandwidth utilization for management purposes (*spatial distribution*) and to give the system the ability to dynamically delegate new management code to remote devices when needed (*temporal distribution*).

Over the years, a number of research projects have found their roots in management by delegation and agent-based management. The research of Mountzia (Mountzia and Dreo-Rodosek., 1996) presents an early classification of MbD schemes according to delegation types, phases, functional areas, trigger modes, and lifetimes of delegated tasks. Steenekamp (Steenekamp and Roos, 1996) proposed a framework that extends the delegated model to provide agents that implement management policies in terms of rules. Suzuki (Suzuki et al., 1996) improves the efficiency of delegated management operations by using dynamic scripts that divides an operation into a series of steps and addressable objects. Primitives for cooperation among management agents, based on the MbD paradigm, are proposed in (Schoenwaelder, 1996), while (Keller, 1996) analyzed the suitability of Common Object Request Broker Architecture (CORBA) to implement MbD systems. The comparative review of Kahani (Kahani and Beadle, 1997) presented several decentralized network management techniques, with more examples of network management applications that can benefit from the MbD paradigm provided in (Kooijman, 1995). The overall conclusion implied by all cited authors is that agent-based approaches reduce the load on the NOC by performing management tasks at the devices. This is expressly one of the design requirements of the `Stix` system that will be discussed in Section 5.2.

The current research on the In-Network Management (INM) paradigm (Dudkowski et al., 2009) is a clean slate design approach that shares similarities with the older MbD and Active Networks attempts. Generally speaking, INM stipulates five fundamental principles: (i) Management is intrinsic to the network; (ii) Management is an inherent part of network elements, protocols and services; (iii) Management is autonomous and does not involve any external technical intervention; (iv) External management operations occur on the highest possible level of abstraction; (v) Any principle is to be implemented in a way that it can be gradually adopted. The authors provide a high-level description of the architecture and an example applied to P2P networks. Ultimately, INM aims to reduce the dependence on a centralised NOC, which is also one of the design principles of `Stix`.

When it comes to practice, most network management platforms are centralised (manager-agent based) and based on SNMP (Simple Network Management Protocol), making it the *de facto* standard. While SNMP allows for distributed monitoring with multiple servers (managers) for load balancing and fault tolerance, it does not as such reduce the communication overhead. `Stix`, on the other hand, deploys managers in the form of `StixAgents` inside the network close to the managed devices (within one hop) with the view of cutting down the communication overhead.

There exists both open source tools (e.g., OpenNMS, Nagios) and commercial tools. In the latter set, many are vendor-specific (e.g., Alvarion Star Management Suite, Motorola Prizm, Ubiquiti AirControl, Meraki's centralised management solution). Though a few multi-vendor tools exist, such as Aruba Networks' AirWave, none of them are for BWA networks. The Customisable Wireless Management System (CWMS) (Ng et al., 2007) proposes to manage heterogeneous BWA networks composed by multivendor devices for both WiFi and WiMax by using metadata expressed in XML to define a glue between different types of devices, but it is a centralised system.

Spotting and automatically identifying problems in 802.11 wireless networks is the aim of MOJO (Sheth et al., 2006), a system that outlines the importance of detailed physical-layer metrics for problem diagnosis, and demonstrates that many higher layer symptoms are manifestations of problems at the physical layer. A large-scale implementation of similar ideas has been proposed by Microsoft Research (Adya et al., 2004), which is also appropriate for troubleshooting disconnected clients by elaborating on the information received by other neighboring devices. Similarly, the Antler project (Raghavendra et al., 2008) focuses on the automated multi-tiered metric collection to assist in fault diagnosis, enabling realtime problem tracking. Many management frameworks are especially tailored on mobile or ad-hoc wireless networks which differ significantly from BWA networks. However, their conclusions can be effectively applied in our context. An example is the DAMON system (Ramachandran et al., 2004), which includes an auto-discovery mechanism to enable the agent to automatically discover the presence of devices to be controlled, thus reducing the effort required in deploy new networks. The idea of implementing hardware agents in commodity hardware is not new to `Stix`. For example, Weaver (Lim and Stadler, 2003) is a framework for code-mobility to control routers based on an external hardware agent.

There is limited work on BWA network management (Schuetz et al., 2007; Baliosian et al., 2008), mostly focusing on autonomic or self management. These do not consider the underlying implementation platform for autonomic management in general. This is evident from the lack of usable software implementations from these efforts. Even those that build demonstration prototypes are too naive. For instance, in (Schuetz et al., 2007), software running on their access points to implement autonomic processes is in fact statically pre-compiled and pre-deployed C programs, which are inflexible and hard to run in heterogeneous environments; moreover, it does not allow code reuse. In contrast, one of the contributions of our work, demonstrated by our case studies in Section 5.4, is to offer a flexible platform to facilitate autonomic management. However, autonomic management by itself (e.g., alarm correlation, fault diagnosis, intrusion detection) is not the focus of our work.

The work of Surana et al. (Surana et al., 2008b) is relevant to our work in the sense that it provides a concrete context where simplifying network management and supporting to heterogeneous devices are essential. The authors provide a detailed description of the challenges they faced in terms of operational sustainability from their experience with two BWA deployments, both having thousands of users. For the network management component, they essentially use a centralised approach via their push-based PhoneHome monitoring mechanism. In contrast to our work, their focus is more on low level issues such as ensuring stable power. There also exists technology specific performance and fault management research for scenarios different to ours, including infrastructure WLANs (Cheng et al., 2007), mesh networks (Qiu et al., 2006) and ad hoc networks (Badonnel et al., 2008).

Our workflow-based modelling language (`StixL`), presented in Section 5.3, is at a high level similar to the approach adopted in the Click modular router architecture (Kohler et al., 2000). However, our focus is on device monitoring and control, whereas Click's focus is on packet processing. Also Click's "workflows" are data-bound in that they are triggered on a per-packet or per-frame basis, whereas ours are event-bound and so are fired after a condition (e.g., timer, value of a variable reaching a threshold, message arrival) becomes true. Also the notion of using visual programming languages in general is not new (e.g., OPNET), but our definition and use of a visual programming language (i.e., `StixL`) for goal-oriented network management is.

2.3.1 Business Process Management

Business Process Modeling Notation (BPMN) is a graphical representation developed by the Object Management Group[3] for specifying business processes as workflows.

It provides a graphical notation for specifying business processes, based on flowcharts, in order to support technical and business users by providing a notation that is intuitive yet able to represent complex process semantics. BPMN is targeted to business users, who can depict the interactions between actors and determine the sequence of tasks to be accomplished.

Models in BPMN are constructed from a small set of graphical elements, organized in four categories (see Figure 2.1):

- **Flow Objects**, which are the core elements of BPMN: *events*, denoting something that happens and triggers the execution of the workflow; *activities*, repre-

[3] `http://www.omg.org/`

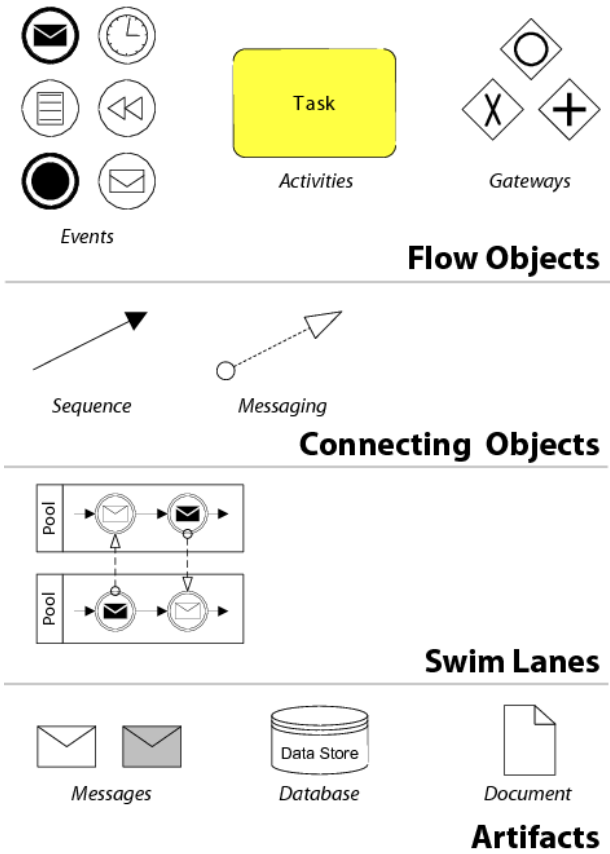


Figure 2.1: *A subset of the symbols available in BPMN.*

senting individual tasks; and *gateways*, that cause forking and merging of paths depending on the conditions expressed.

- **Connecting Objects**, that relate other entities to define relationships such as sequence and messaging.
- **Swim lanes**, that define the responsibilities of each actor.
- **Artifacts**, which allow to bring more information into the diagram, making it more readable, for example by grouping elements or by adding annotations.

In designing our `StixL` visual programming language (described in Chapter 5.3), we took inspiration from BPMN, from which it carries some of the original constructs and principles and borrows the XML schema.

2.4 Broadband assessment and mapping

Market knowledge is precious, and particularly so in the broadband business, which is split between highly competitive urban areas and less profitable rural regions. Stakeholders of such information are the ISPs, eager to know where to invest and how the competitors are performing; policy makers in order to plan regulations, strategies and public intervention; and broadband end users, wanting to know which is the best broadband offering currently available in their area. In order to grow awareness about the market, it is essential to obtain a clear perception of the choice and quality of Internet connections available in a given area, by producing a *broadband census*. Broadband census approaches can be broadly classified into two categories: *model based* and *measurement based*. We discuss each of them below.

2.4.1 Model based Approach

Techniques falling in this category estimate broadband coverage and speeds based on theoretical or empirically derived models of access technologies, knowledge of network infrastructure (e.g., phone exchanges, mobile network base station locations) and configurations (e.g., contention ratio, radio parameters). In the case of DSL, see (Tanenbaum, 2003; Grubesic, 2008) for examples of such models[4] and their use in estimating broadband coverage. Similar approaches can be followed to estimate 3G mobile broadband coverage[5]. Alternatively, estimated broadband data can be provided directly by ISPs, obtained from their public websites or other public sources. Such estimated data could again be based on models and/or field tests (e.g., drive test measurements conducted by mobile telecom operators). As with any model based approach, especially if they are not empirically based, there is the possibility for the estimates to provide overoptimistic results due to underlying model assumptions (e.g., in the case of DSL, perfect line quality and no crosstalk).

Both these approaches have severe limitations. In the absence of a regulatory imposition, it is difficult to obtain coverage data from the ISPs, as they often regard it

[4] The UK's OFCOM has also published the document 'Assessment of the Theoretical Limits of Copper in the Last Mile', available at: `http://www.ofcom.org.uk/research/technology/research/emer_tech/copper`

[5] For example, in the UK: OFCOM 3G Coverage Maps, available at: `http://www.ofcom.org.uk/radiocomms/ifi/licensing/classes/broadband/cellular/3g/maps/3gmaps`

as confidential information sensitive for their business operation. ISPs may claim they are afraid that if they published a map of the services they offered, competitors would know exactly what pitch to send to which customers. Also, network operators inherently have the tendency of overestimating their network coverage, for both marketing reasons and technical factors (i.e. overselling, contention ratio, etc.). Finally, in markets dominated by the network incumbent, a overestimation or error from their side will reflect in a corresponding error to the coverage of all the ISPs to which the incumbent provides wholesale access.

A particular example in this category that is worth mentioning is the approach taken by the US non-profit Connected Nation[6], which relies on broadband data from providers that voluntarily choose to participate in their mapping project. Incentives to encourage providers to participate are unclear. This project depends on consumer feedback and surveys to verify ISP provided data, which potentially introduces a substantial manual element to maintain the mapping effort and thereby making it an expensive operation to sustain. Field measurements conducted by Connected Nation engineers is also mentioned as yet another verification approach, but solely relying on such field tests can also make the project unsustainable. In the UK, the Ofcom has launched a website[7] featuring interactive maps about fixed-broadband, using data provided by ISPs, including: availability of ‘superfast’ (i.e., FTTH and FTTC) broadband; average broadband take-up; average maximum speed for ADSL and cable services; percentage of homes with broadband currently not receiving 2Mbit/s speeds. In the United States, a large-scale broadband mapping effort[8] is underway with $350 million funding from the National Telecommunications and Information Administration (NTIA) as part of the broadband stimulus from the 2009 American Recovery and Reinvestment Act, with the aim of updating the maps every six months. Similarly, the German government is sponsoring the development of a ‘broadband atlas’[9]. However, such data is inherently optimistic as it does not consider various practical impediments (e.g., line quality, contention). It is nevertheless useful in the absence of any other information to estimate the extent of broadband coverage in a region and to identify notspots (locations lacking any Internet access service).

[6]Connected Nation: Broadband Mapping. `http://connectednation.org`
[7]Ofcom fixed-line speed map: `http://maps.ofcom.org.uk/broadband`
[8]US National Broadband Map, `http://www.broadband.gov`
[9]Broadband Map for Germany, `http://www.breitbandatlas.de`

2.4.2 Measurement based Approach

Instead of estimated data, actual measurement data can be used for broadband mapping. Although ISPs can in theory contribute with measurement data that they may collect continuously to monitor and optimise their networks, they may not have any incentive to do so in practice. So we *focus on provider-independent measurement approaches* as practical alternatives to obtain measurement data. They fall under two sub-categories: (1) user-dependent, and (2) user- and provider-independent. We discuss each of these below.

2.4.2.1 User-Dependent Approach

In this approach, users explicitly or implicitly participate in measuring the quality of their broadband connections.

1. *Hardware-based:* An approach that does not involve installing test software on the user's computer is to install a hardware device dedicated to monitoring the user's broadband connection 24/7. This is the approach followed by SamKnows[10] for its UK broadband speeds study with Ofcom. SamKnows was recently selected by the FCC to conduct a similar study in the US and by the European Commission in the EU, deploying thousands of home routers instrumented to periodically record network measurements. This approach requires a large number of diverse volunteer users to agreeing to install such a hardware monitoring device in their homes, which can be expensive. Moreover, the hardware-based measurement approach is unsuitable for studying broadband quality at different granularities (e.g., at the country level, county level and city level) because it relies on the idea of statistical sampling: if a sample is selected for a given sized region (e.g., the whole of UK), that sample will not be appropriate for a smaller region (e.g., the city of Edinburgh) unless the sample itself is very large. Sampling is effective if the sample size is just enough and increasing the sample size is expensive, which makes this approach suitable for macroscopic studies but not microscopic studies. Also, this approach will clearly not work for mobile broadband measurement. Finally, privacy concerns of users may also impede an implementation of this approach. The hardware-based approach was

[10] SamKnows Inc, `http://www.samknows.com`

also recently considered in the academic community: a recent study (Sundaresan et al., 2011) looked into the performance achievable by the aforementioned FCC deployment in the United States and provided insightful findings of how broadband connections should be measured, also giving interesting views of the performance obtained by individual users in the dataset.

2. *Software-based:* There are many broadband tests that the user can run on their computer when connected at home (e.g., NDT, Speedtest.net, BroadbandCensus.com, etc). These tests differ widely in their test methodology (e.g., packet dispersion vs. file transfer time), server location and data collection mechanisms. Traditional approaches of using web-based speed tests, such as `Speedtest.net` from Ookla, can only gather sporadic and geographically non-uniform measurement data and also suffer from measurement biases (e.g., users taking speed tests may have poor broadband connections or not a representative sample). Some of these are open source like the D-ITG test we use in our implementation, whereas others like Speedtest.net are commercial. In fact, these two tests have been chosen for the recently launched Federal Communication Commission (FCC) consumer broadband test[11]. Epitiro/isposure[12] takes a different software-based approach based with a measurement agent running in the background on the users' computers. While this approach has better prospects for improved spatio-temporal measurement data collection, and it similar to the approach we take in our `BSense` tool, Isposure employs proprietary measurement techniques making it hard to audit measurement techniques and results. Moreover, they have an interesting business model in which ISPs pay for the data collected from consumers using Isposure. To be effective, software-based tests have to be easy to setup and configure. Also, when geolocation techniques are used to infer the user's location based on the connection's global IP address, location errors can be introduced, especially in rural areas. Another common problem for user software tests is that they can be used for collecting measurements only when the user's computer is on.

[11] `http://www.broadband.gov/qualitytest/about`
[12] `http://www.isposure.com`

2.4.2.2 User- and Provider-Independent Approach

The above mentioned measurement approaches require user participation either initially, to agree to host a hardware device, or repeatedly in the case of software tests. An alternate approach has emerged recently to enable large-scale active measurement without user or provider involvement by continuous, low-overhead probing of consumer broadband routers from a remote measurement server (Dischinger et al., 2007; Croce et al., 2008). This approach relies on certain specific but standard functionality from routers (e.g., responding with TCP RST packets upon receiving unsolicited ACKs). Such functionality may be disabled due to security concerns. If a particular ISP does not support this functionality on all its broadband routers, then that ISP is effectively ignored by this approach, introducing a measurement bias and thus undesirable from a broadband mapping perspective.

One common limitation of all measurement approaches is that results may be sensitive to cross traffic (other active traffic from the user during the time his/her connection is being tested) and may not be robust across a heterogeneous set of access technologies (e.g., DSL, cable, WiFi, satellite). However, this is a very active research area as evident from several recent papers (Portoles-Comeras et al., 2009; Maier et al., 2009), so it is reasonable to expect improved measurement techniques to be available in near future.

Other related work includes the research on novel measurement techniques along the lines of (Dischinger et al., 2007), such as measurement work looking at the impact of WiFi prevalent inside homes (Portoles-Comeras et al., 2009), focusing on troubleshooting (Kreibich et al., 2010) as well as work discussing the issues underlying broadband speed measurements (Bauer and Lehr, 2010). We note that our focus is not on new measurement techniques but rather on developing a flexible framework for broadband mapping. Though we incorporate a measurement technique in our implementation, our proposed approach is not tied to it and can instead rely on more advanced and comprehensive measurement techniques like (Kreibich et al., 2010).

Chapter 3

Tegola: Wireless Network Testbed in Rural Scotland

3.1 Introduction

This chapter gives an account of the process of designing and operating our testbed network, and provides a backdrop for the following three chapters, which elaborate on three crucial areas of running a BWA network.

We deployed a wireless testbed network called Tegola[1] in collaboration with the University of the Highlands and Islands (UHI)[2] as an inspiration to guide our research, pointing us to the most critical problems in deploying rural BWA networks. It also serves as a realistic experimentation platform for characterisation and evaluation studies. Finally, the testbed accomplishes a useful social purpose, as it connects remote communities that were relegated to slow dialup Internet access and unreliable landlines, with virtually no GSM and terrestrial TV coverage. The northwest of Scotland is known for its mountainous terrain and for its harsh and unpredictable weather, even when compared to the rest of the British Isles, making it a challenging environment for deploying and operating wireless networks.

Our network has important similarities with other community and research deployments: it is organised into two tiers (Gokhale et al., 2008; Matthee et al., 2007) and

[1] 'Tegola' means 'roof tile' in Italian.

[2] UHI is a federation of 13 colleges and research institutions in the Highlands and Islands of Scotland delivering higher education courses. In particular, Tegola is based at the Sabhal Mòr Ostaig campus on the Isle of Skye: `http://www.smo.uhi.ac.uk`

Figure 3.1: *The Tegola testbed on a map.*

the underlying wireless technology adopted (i.e., 802.11) is the same as most existing research testbeds.

The remainder of this chapter is structured as follows. In Section 3.2 we give an overview of the testbed along with details of the hardware and software used. We then look at the three main issues in deploying a wireless network: network planning (Section 3.3), network management (Section 3.4) and the assessment of broadband quality (Section 3.5). A summary of this chapter is given in Section 3.6.

3.2 The Testbed

In this section, we give an overview of the testbed that we deployed.

Area. Figure 3.1 provides an illustration of the testbed on a map. It is situated in the northwest of Scotland, connecting rural (mostly coastal) communities in the Glenelg and Knoydart peninsulas on the British mainland to the Sleat peninsula on the Isle of Skye, via long distance wireless links over water across the Sound of Sleat. The population in each of these communities is around 50-100, more or less evenly spread across all age groups. These communities, though inhabited, are quite remote[3]; and

[3] In fact, Knoydart peninsula is regarded as *'mainland Britain's most dramatic and unspoilt wilderness area'*. (From: "The Rough Guide to Scotland", Apr 2006, Seventh Edition)

there is no road access to Knoydart, which is only reachable by boat or a 2-day hike. Before our deployment began, residents in these communities had no broadband via traditional means from ISPs, and even the existing telephone and dial-up Internet connections tended to be unreliable in some locations. Climate in this region is maritime temperate; frequent, unpredictable changes are common as is the case in the rest of the British Isles. Weather can be quite harsh as it can often be wet and cloudy with strong winds, and things can change quite rapidly to severe conditions, especially on the mountains. From a network deployment and operation standpoint, this region is quite challenging given its inhospitable weather conditions and rugged mountainous terrain.

Architecture. The testbed is organised as a two-tier network, with a backhaul tier consisting of a ring of long-distance wireless links connecting different target communities with a site linked to the wired Internet. Backhaul nodes in turn distribute the available bandwidth to rooftop nodes in the individual communities via point-to-multipoint links, and each rooftop node further down to the end-user client computers. Figure 3.1 shows the deployed backhaul wireless network. Nodes marked *S* and *I* are on the Sleat peninsula in the Isle of Skye, nodes *B* and *C* on the Glenelg peninsula, and node *K* on the Knoydart peninsula. Node *S*, located on the campus of Sabhal Mor Ostaig (Gaelic college on Skye) is linked to the wired Internet via the UK academic network. We have used standard WiFi (i.e., 802.11a/b) as the underlying wireless technology given its low cost and ready availability. Currently, our testbed connects around 40 houses from widely dispersed coastal and remote communities living in the villages of Arnisdale and Corran (connected via nodes *B* and *C* respectively), Inverguseran and the west coast of Knoydart (connected via *K*), Duisdalemore (via *I*), and Doune (via *S*). The houses in Arnisdale and Corran are linearly distributed along the coast, which is quite unlike that in typical urban mesh deployments. The network also provides connectivity to a community centre and several small businesses.

Hardware. We use Avila GW2348-4 router boards from Gateworks[4] as nodes in the backhaul tier. These boards, based on the Intel IXP425 processor, come with 64MB RAM and 16MB of Flash memory, and provide 4 mini-PCI slots and two Ethernet ports. In our configuration, one of the mini-PCI slots is typically used for local point-to-multipoint distribution via 802.11b/g in the 2.4 GHz band, and the remain-

[4] Gateworks Corporation: `http://www.gateworks.com`

ing 3 slots are used for backhaul connectivity over 802.11a in the 5 GHz band. We have used mini-PCI radio modules from Ubiquiti Networks[5] as they support higher transmit powers and exhibit better receiver sensitivities — XtremeRange5 (XR5) for 5 GHz operation and XtremeRange2 (XR2) for 2.4 GHz. For the backhaul, we have used high performance dish antennas operating in the 5 GHz band with 29dBi gain and dual-polarity support from Laird Technologies[6] as they are quite rugged and also exhibit high cross-polarization isolation. Each backhaul link is in fact comprised of two links using orthogonal polarizations (i.e., horizontal and vertical). This is useful for three reasons: first, it enables us to run wireless experiments on one polarization while keeping the network operational for our end user; on the other, it allows for increasing the link capacity; lastly, it helps counter multi-path fading effects due to over-water propagation. We equipped node *C*, the site connected to the wired Internet, with two Gateworks boards for added fault tolerance. For local access at each backhaul node, we are using 19dBi panel antennas from Laird Technologies that operate in the 2.4 GHz band. Because of the high power requirements (around 6W at 3.3V) of each mini-PCI radio module, the router boards often work close to the maximum supported power limit. Also, in order to reduce the power loss in cabling, none of our base stations are powered using Power over Ethernet. On the CPE side, we are providing testbed users with a custom device we assembled in a rugged outdoor box (see Figure 3.2). They are based on alix3c2 boards from PC Engines[7], which is equipped with a 500 MHz x86 CPU, 256MB of RAM, a CompactFlash slot for persistent storage, an ethernet plug and two mini-PCI interfaces. Our CPE node simultaneously acts as an 802.11 client to the local base station, and as an AP to devices within the house.

Nodes *B* and *I* in our testbed (see Figure 3.1) are self-powered using a wind generator as well as a solar panel. Node *B* is in fact located on a mountain (called Beinn Sgritheall, see Figure 3.3 for a picture of this mast) at a height of about 1,000 feet. We use a Rutland Furlmatic FM910-3 windcharger[8] that can generate around 24W at wind speeds of around 5m/s. We use a Kyocera KC130GH T-2 solar panel[9] that has a maximum power output of 130W. To serve as a buffer during periods of low power

[5]Ubiquiti Networks wireless products: `http://www.ubnt.com`
[6]Laird Technologies antennas: `http://www.lairdtech.com`
[7]PC Engines embedded computers: `http://pcengines.ch`
[8]Rutland windchargers from Marlec Engineering Co Ltd: `http://www.marlec.co.uk`
[9]Kyocera solar panels: `http://www.kyocerasolar.com`

Figure 3.2: *The custom Tegola CPE (left picture) is typically installed on users rooftops (on the right).*

generation, we use two identical Elecsol 125amp/hr 12v deep cycle batteries[10] connected in parallel and in turn connected to solar/wind power generators via a charge regulator. The load (Gateworks board) is directly connected to the battery bank.

Masts. Our masts need to be strong and rigid enough to withstand substantial wind forces without deflection. They also need to be light enough that the individual components can be carried in by a small – three or four men – construction team. However they need not be tall: the terrain provides the height; the land is heavily grazed, so there is little chance of obstruction by vegetation; and the wind is strong enough that there is no need for height in order to obtain added wind velocity at the turbine. The main need for height is to keep the turbine clear of people and animals. Our initial design, shown in Figure 3.3, was a single, guyed, vertical pole supporting an "H" frame with approximately 2m between the verticals to provide adequate separation of the dishes. However we found that we needed to install extra guy wires to stop the frame twisting.

Our current design uses aluminium scaffold poles and galvanized connectors throughout. As shown in Figure 3.4, it consists of a horizontal bar at head height sup-

[10]Elecsol batteries: `http://www.elecsolbatteries.com`

Figure 3.3: *Self-powered mast on Beinn Sgritheall (node B in Figure 3.1).*

ported by two verticals about 2m apart. The verticals support the larger (long-distance) antennae and the power generating equipment when present. The structure is diagonally braced with the same material. We have found this to be much better than using guy wires, which tend to become loose with time. The basic structure can be erected in a matter of minutes and is easy to adjust for uneven ground. A further improvement would be to incorporate some kind of shelter into the design — perhaps a tent using the horizontal bar as a ridge-pole. The Scottish Highlands seldom provides weather in which it is possible to do basic wiring, let alone electronics, outdoors.

It is also important to note that the materials required for building the masts were largely provided by the local communities. People from the communities also actively helped with the mast installations, which made it a significantly easier task.

Software. We have used OpenWrt[11] with the 2.6 Linux kernel as the operating system at all nodes, both in the backhauling and in the CPEs. We used a slightly modified version of the MadWifi radio driver. Radios for the backhaul wireless links are

[11] OpenWrt Linux distribution for embedded devices: `https://openwrt.org`

Figure 3.4: *Mast at node K in Figure 3.1, illustrating our current approach to mast construction.*

configured to work in "adhoc demo mode" (pseudo-IBSS) so that neither management frames nor beacons are ever sent in order to minimise the protocol overhead. RTS/CTS protection modes are disabled, and the ACK timeout and slot time values are configured using the MadWifi `athcrtl` utility based on the distance between the endpoints. We have used the default MadWifi rate adaptation algorithm, unless otherwise specified.

The entire network uses a private IP address space, with OSPF routing in order to deal with route changes due to failure of point-to-point links and for load balancing over multiple links (when using both polarizations simultaneously over different channels at the backhaul nodes). Connection to the Internet is routed via a Linux server located at node S, which provides NAT and firewalling facilities. We also records statistics such as anonymized traffic flow and bandwidth transferred in each direction by each testbed user.

Frequencies. In the UK, as per Ofcom regulations, high power transmissions for

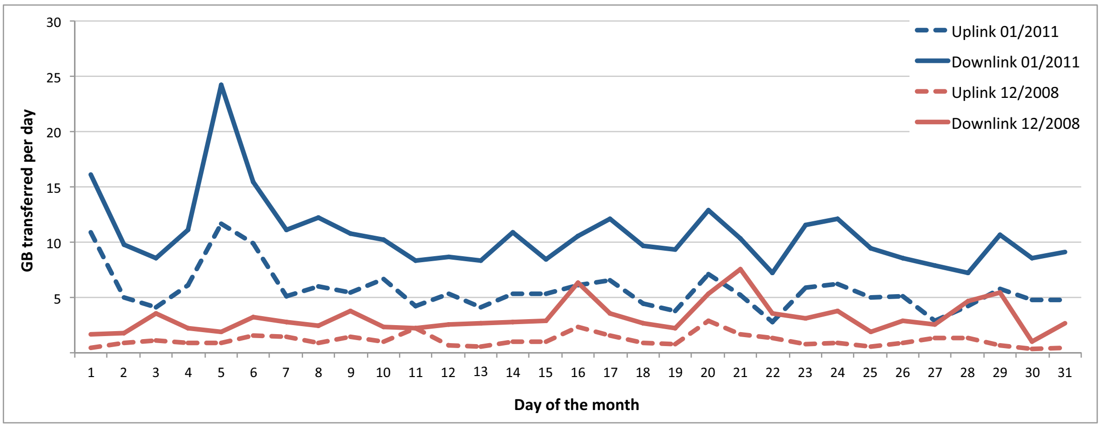


Figure 3.5: *Total daily traffic exchanged between the Internet and the Tegola network during December 2008 and January 2011.*

fixed wireless access requires a license and is only allowed in the 5725-5850 MHz band with a maximum transmission power limit stipulated at 4W EIRP. Therefore, we obtained a 5.8 GHz fixed wireless access license and use only channels in that band in our testbed for backhaul communication.

Current usage. The take-up and usage of the Tegola network has been steadily growing over the years. Figure 3.5 clearly demonstrates this trend: we look at the total daily traffic exchanged between the Internet and our users (in either directions) for two distinct months, December 2008 and January 2011. Figures are consistently higher in the second case, because of both increased usage from old users and due to new households and communities being reached.

A recent and more detailed statistical analysis[12] performed on three months worth of network traces from Tegola showed some expected facts, including: asymmetric network usage dominated by downstream traffic, majority traffic from TCP (and within TCP, HTTP), and different usage patterns between weekdays and weekends.

3.3 Network Planning

Our goal in the initial phase of deployment was to connect the residents in the two villages of Arnisdale and Corran in the Glenelg peninsula and the communities in the

[12] Included in the M.Sc. thesis of Daniel Tyrode, see Appendix B for details.

west coast of Knoydart peninsula to the Internet (see Figure 3.1). Since connection to the Internet was only feasible via the Gaelic college on the Sleat peninsula in the Isle of Skye at location marked *S* on the map, it was obvious that we needed long distance over-water radio links across the Sound of Sleat, but the exact mast locations were not clear. We took a pragmatic stance in wanting to have a working system first before concentrating on the optimal mast placement problem, yet we wanted the topology to be redundant enough to survive a mast failure.

We adopted an ad-hoc approach to identifying the suitable mast locations. We used topographic maps of the area in combination with the several GPS locations gathered via site surveying and also taking into account accessibility and closeness to a grid-connected power supply. Once we identified a pair of locations for a link, we tested its feasibility using the Radio Mobile tool[13]. Two key observations came out of the above topology planning exercise: (i) the only straightforward way to reach the Glenelg peninsula (the part of the map in Figure 3.1 with nodes *B* and *C*) required a self-powered mast close to the top of a nearby mountain (Beinn Sgritheall); otherwise, we would have needed several – and still self-powered – relays incurring a significantly higher cost. (ii) It was not possible to directly complete the “ring” from the Glenelg peninsula to Knoydart peninsula given that residents in Knoydart were along its west coast. This meant we had to bounce the signal off of a site on the Sleat peninsula in the Isle of Skye (the part of the map in Figure 3.1 with nodes *S* and *I*).

The resulting set of locations and links are the ones shown in Figure 3.1. Sites are interconnected in a “ring” topology, so that each one has two independent paths to the Internet backhaul, resulting in improved reliability. In fact, the decision to have redundancy in the backhaul has proven useful at various times. For example, when one of the two boards at node *C* failed, the network automatically switched to the alternate path without any noticeable disruption.

Even though our network is composed of only five transmission sites, we spent considerable effort on the network planning task. Our ad-hoc approach proved to be naive and ineffective, as it involved numerous inspections at potential transmission sites and field surveys of remote areas before each link could be established. Adopting such a trial-and-error process to larger networks would be unacceptably time-consuming. In Chapter 4 we explicitly focus on planning of rural BWA networks and present the

[13]Radio Mobile tool: `http://www.cplus.org/rmw/english1.html`

`IncrEase` tool, which aims to simplify the design process.

3.4 Network Management

Establishing a solid support infrastructure for a wireless network is as important as deploying the network in the first place.

The troubleshooting of wireless networks is more complex than that of similarly sized wired deployments, because of the larger set of performance metrics and configuration parameters involved, which requires operators to have knowledge in both the radio and IP networking fields. Sometimes, problems on wireless network are 'invisible': they are asymptomatic, hard to detect, and are harder to diagnose and take longer to fix. For example, when operating on unlicensed spectrum, interference can be a recurring problem which can affect the network at unpredictable times. Also, weather factors, such as wind, rain and snow, can damage outdoor equipment in subtle ways. For example, over one winter we experienced power outages at one of our self-powered masts, because ice had clogged its wind turbine and cold temperatures lowered the efficiency of the backup batteries.

Rural networks, such as Tegola, are even harder to manage because of two reasons. First, most of the equipment is installed at extremely remote locations. If an 'on-site' intervention becomes necessary, for example to replace faulty electronics, the journey has to be planned days in advance, especially during the winter season. Such remoteness taught us the importance of automated redundancy (e.g., redundant power sources and batteries, backup radio links automatically managed via IP routing, etc.), and to be able to remotely control and monitor every aspect of our network (e.g., power consumption, powering on and off the backhaul boards, upgrade software) to avoid local intervention as much as possible. The second reason is that Tegola is operated as a community network, administered by the users themselves: we taught local residents to configure new CPEs and connect them to the network, to replace backup batteries and to perform some basic troubleshooting. Creating such strong links with the community is helpful and rewarding, but requires clear communication and procedures to be set: network management is harder and more complex when operations are delegated and ownership is distributed.

Despite the myriad of software projects for network management, we felt the lack

Figure 3.6: *Pictures from the Tegola network deployment process: (a) Preparing the hardware; (b) Control visit at site* B *during a storm; (c) Two dual-pol dishes at site* C*; (d) The self-powered mast at site* I*, surrounded by sheep fencing; (e-f) Local residents helping to move the equipment on site; (g) Connecting to the Internet from a fishing boat at sea; (h) Maintenance operations at remote locations provided no shelter from harsh weather (here: using a laptop in the rain).*

of tools addressing the specific issues of rural and community-operated networks. In Chapter 5, we dig deeper into the problems of network management for rural and community-run BWA networks. The result of our research is `Stix`, a goal-oriented distributed network management system. In `Stix`, management tasks are defined in a simple visual-programming language, which makes our system well-suited for rural networks, where the number of skilled personnel available is limited.

3.5 Assessing the User Experience

In late 2007, while we were looking for a suitable region to deploy a research testbed, we were recommended the two small villages of Arnisdale and Corran in the western Highlands. Residents told us they could not access the Internet (often not even via slow dial-up access), landline phone quality was poor, TV was available only via satellite, and there was virtually no mobile coverage. Despite their enthusiasm in the project, we did not know exactly how widespread and serious such digital 'notspots' were. With hindsight, we were facing two problems:

- Our knowledge of the local broadband market was based solely on word of mouth.

- To provide broadband quality details, residents used online speed measurement tools (such as the `http://speedtest.net` website) which provide a picture of the broadband line in a given moment in time, rather than continuously over a period of time.

We encountered similar issues every time we expanded our network to reach new communities: we did not have a clear idea of how many people we would be covering by the access tier and how many would be interested in joining. Similarly, when we had to plan upgrades to our backhauling links, we could only estimate the bandwidth usage without knowing the actual network quality perceived by the users. While it is very important for a network owner to understand the ongoing status and 'health' of the network, distilling effective statistics from piles of network traces can be a daunting task. Even in our small testbed the ad-hoc process of collecting, processing and deciphering network data took a significant effort.

Often, network operators perform automated measurements, such as ‘speed tests’ for throughput and `ping` for latency estimation, to ensure the level of quality of their services. However, it is hard to translate these crude figures into a measure of the quality *perceived* by the user, which varies according to the applications in use. In Chapter 6 we present `BSense`, a system for assessing the quality of broadband connection, boiling large data sets down to a customisable ‘Broadband quality score’, and presenting the results on a map.

Such geographical information can be intended as a ‘census’ of broadband availability and quality, invaluable information for ISP entrepreneurs and public policy-makers.

3.6 Summary

In this chapter we have described the Tegola testbed, which has been deployed in rural Scotland to enable research into rural BWA networks and to provide robust broadband Internet connectivity to remote communities. This testbed differs from other existing rural outdoor network deployments in two key aspects (see Section 7): (i) the presence of long distance over-water links; (ii) the need for relying on energy sources beyond solar to keep cost and size of self-powered masts low. Moreover, our design uses judicious amount of redundancy at various levels for uninterrupted and robust operation (e.g., backhaul ring topology, multiple router boards at the Internet gateway node, dual battery bank at self-powered masts). This design choice has proven to be quite useful with the network functioning virtually uninterrupted[14] over a 4 year period by adapting to hardware failure at backhaul nodes and by being immune to variations in output from power sources at self-powered masts.

More significantly, deploying and operating the Tegola network helped us identify three crucial technical issues for BWA networks as motivated in this chapter: network planning, network management, and broadband mapping. In the next three chapters we present our contribution with respect to each of these problems.

Other important issues related to our wireless testbed that are not the focus of this thesis but where I have been part of the effort to develop an initial understanding — wireless propagation over sea water and power planning for self-powered sites — are discussed in Chapter 7. Additionally, in Appendix B we note how the Tegola network is being used by other researchers and is encouraging other communities in Scotland to deploy similar access networks.

[14]This was demonstrated in October 2011, when many telephone lines were knocked out by lightning strikes. The community became entirely dependent on our network. Even emergency health services to a broader area were diverted through Tegola. Two community members, Andrew (a NHS doctor) and his wife Emma, wrote to us: *"We personally have been reliant on the Tegola Internet for our sole means of telecommunication [...] for nearly 2 weeks now and our land line has still not been repaired by BT. Contact links for emergency medical cover to Glenelg and Arnisdale were maintained with the Scottish Ambulance Service and also the NHS 24's Highland HUB Out of Hours Service for a 3 day period via the Internet telephone. [...] Emma has been able to continue her work as Local Development Officer using only the Tegola network for all comms with the Trust, Highlands and Islands Enterprise and international renewable energy companies. Tegola has allowed and is still allowing us both to operate socially and business as usual. [...] We are total fans and supporters of the work that you have achieved here and on other parts of the west coast and many of our friends and colleagues are fascinated to hear that your Tegola method of providing reliable high speed broadband to remote areas even exists!"*. This achievement was later acknowledged by the Scottish Parliament with motion S4M-01260 on 7 November 2011, which was undersigned by 10 MSPs. Also, Tegola was awarded the first prize in the UK NextGen challenge as *"the best in next generation digital access infrastructure innovation"* (`http://www.nextgenevents.co.uk/ngchallenge`).

Chapter 4

IncrEase: Incremental Network Planning

4.1 Introduction

In the previous chapter, we shared our experience in deploying a rural BWA network. We concluded that, even for a small testbed like Tegola, ad-hoc planning is inefficient and time-consuming. We now elaborate on the concept of ‘incremental’ network planning and propose a software tool we refer to as `IncrEase` for guiding through such a process.

Academia and industry offer a myriad of software tools for wireless network planning. Broadly speaking, their aim is to identify the best locations for transmission towers and/or to plan their interconnection to the rest of the network. However, these tools are often unavailable or unsuitable to communities and small WISPs, which often resort to an ad-hoc approach towards network planning. Our focus is on the needs of WISPs operating in rural areas, which are faced with the unique challenge of extending their coverage on a tight investment budget in an environment of limited profitability. The key, for such organisations, is to identify the most cost-effective deployment strategy for planning their core network while taking user coverage into consideration.

The design of rural fixed Broadband Wireless Access (BWA) networks is significantly different from mobile broadband networks (e.g., 3G), and its planning process can benefit from the following two observations:

- *Only outdoor propagation is relevant*, as client devices (Customer Premises

Equipment, or CPE) are typically installed on rooftops. Taking this notion into account helps in driving down costs (or increases coverage for the same cost) because of the lower path loss as signal does not have to penetrate walls. Several recent market studies, such as (Mason, 2010), conclude that the use of outdoor CPEs is the most cost effective choice to reach the “final third” of the population residing in rural areas.

- CPEs are fixed and there is rarely any need to support nomadic or mobile services. There is *no need to provide blanket coverage* or overlap in coverage between cells, since no handover is required. The planning process can concentrate only on residential locations where outdoor CPEs will be placed, thereby simplifying the coverage planning aspect of the problem. In comparison, mobile network planning software typically calculates coverage over a grid of equally spaced points, requiring each of them to be above a threshold signal level.

The network model of rural fixed BWA networks almost invariably follows a two-tier model as discussed at the outset of this thesis: an access tier composed of point-to-multipoint (PMP) between transmission towers and customers; and a backhaul tier formed of point-to-point (PTP) links typically or exclusively wireless, as dedicated wired links are seldom available in rural areas.

While wireless network planning is traditionally a very active area of the research community, the focus of research literature is mainly on mobile broadband networks and wireless local area networks, such as (Amaldi et al., 2003; Bosio et al., 2007; Amaldi et al., 2008; St-Hilaire, 2009). More importantly, the network planning formulation is aimed at an *“all-at-once”* deployment, largely based on mathematical optimisation methods and meta-heuristics like (Whitaker and Hurley, 2003; Bu et al., 2005; Raisanen and Whitaker, 2005; Gordejuela-Sanchez et al., 2009; Hurley et al., 2010). However, as we argue below, this is an impractical approach to deployment for rural WISPs and community organisations. Limited research has so far focused on the rural domain, which also follows the all-at-once deployment approach. Examples include (Sen and Raman, 2007; Panigrahi et al., 2008).

We instead advocate the importance of *incremental planning*, a design methodology that guides WISPs – especially those operating in rural scenarios – in planning their growth by extending their coverage. Our approach is based on the following observations:

- Rural deployments are typically *coverage-driven*, rather than capacity-driven[1]. A reason is that low population density of rural areas plays a positive factor in keeping the required capacity of a cell low. Another reason is that WISPs often operate on a tight budget, so they need to anticipate return on investment from the early stage of roll-out. Even more, in an environment of limited profitability such as rural regions, their priority is to reach the largest number of potential customers as early as possible in the deployment phase by focusing on areas where users are clustered (e.g., larger towns and villages lacking broadband access). A similar reasoning is also applicable to community or council networks, where part or all the population of a region needs to be covered.

- Network infrastructure (e.g., fibre links for backhauling) is often limited or unavailable in rural areas, which means that the operator has to roll-out its own (wireless) backhaul tier as it expands, entailing an additional deployment cost.

- Limiting the geographical extent in which the WISP initially operates is effectively a way to keep ancillary operational costs (e.g., engineers for on-site customer support, marketing) low.

Beyond the initial deployment stage, the network operator can take two actions to extend its business: to increase the network coverage, or to improve it in areas already covered. In either cases, its moves are likely limited by budget and only a small subset of the potential actions can be addressed. Our aim is instead to systematically identify, and suggest to the network operator, a sequence of actions that results in the best long term deployment strategy. Towards this end, we develop an open-source tool called `IncrEase` that enables the incremental planning paradigm in practice.

The remaining of the chapter is organised as follows. Section 4.2 presents the two operational modes supported by the `IncrEase` tool: (1) Targeted Increase, where the operator selects a specific region to be covered as part of network expansion; and (2) Strategy Search, where the tool guides the operator in deciding the deployment order of transmission sites in the near to long-term horizon based on expected profitability.

[1]The number of base stations is determined either by the size of the region to be covered ("coverage-driven") or by the total traffic that needs to be carried ("capacity-driven")

In Section 4.3, we evaluate the tool on the NGI SpA Network (see Section 2.1.1 for further details about this WISP): a real-world scenario of over 8,000 available towers and benchmark its computation time. To validate the quality of its output, we posed a sample set of planning scenarios to experienced wireless engineers working at NGI SpA. We then compared their answers with the deployment strategies proposed by `IncrEase`, obtaining similar or better results in most cases.

4.2 The Increase Tool

`IncrEase` is open-source software, implemented as a cross-platform desktop application in Java. It is based on the NASA World Wind Java[2] open-source GIS library and the Neo4J[3] graph database. It allows the network operator to import, model and elaborate customer-related statistics from the BWA network in order to systematically identify the strategy that best improves and extends the network coverage.

An example of such information flow is presented in Figure 4.1. A set of XML files containing network statistics is read and parsed. In the current implementation, we consider three sources of data. The first is *coverage demand*: the list of requests for coverage received (e.g., on the WISP website) by potential users living in unserved areas. The second is the set of details regarding those new users that *failed installation* stage due to insufficient coverage. Finally, we also import the *log of support calls* to the WISP helpdesk and the location of existing users. Extra data sets can be imported to capture other influencing factors (e.g., availability of DSL, 3G coverage, demographics, etc.). `IncrEase` elaborates each source of data to form a bi-dimensional array covering the geographical region of interest, with each cell value representing how many "items" (e.g., current users) are contained in the cells region. Cell values are then normalised as a fraction of the cell containing the most items. `IncrEase` visually presents the three 2D arrays on the map as *heatmaps*, and combines them into one as a weighted average, where weights are configured according to each metric's relative importance to the operator. Such a combined heatmap provides an at-a-glance view of the areas that would benefit the most by a network upgrade. In this sense, *heat* (i.e. high cells values in the 2D array) is an indication of inadequate wireless coverage,

[2] `http://worldwind.arc.nasa.gov/java/`
[3] `http://neo4j.org/`

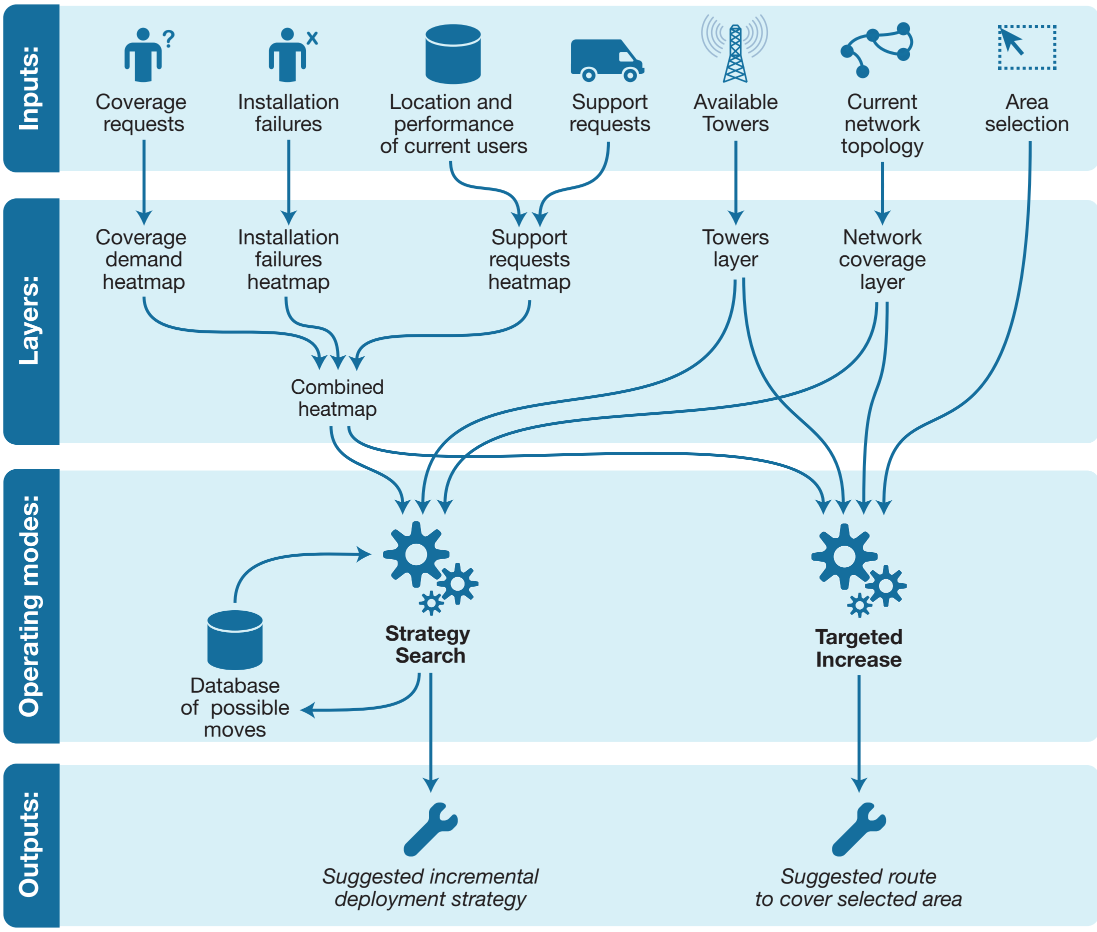


Figure 4.1: *Example of information flow in the* `IncrEase` *tool.*

which can be removed by a new transmission tower or directive sector. Heatmaps are stored in memory, and can be zoomed, shown or hidden by selecting the appropriate GUI elements.

`IncrEase` can import additional list of additional towers available to be deployed. Such an inventory could include towers that exist already (e.g., available for rent from a tower operator) or suitable locations where new ones could be built. An XML description of the current network topology, including the location and height of each tower and configuration and number of the sector antennas can also be imported into `IncrEase`. Such information is used to generate a "network coverage" layer by performing, with a configurable granularity, line-of-sight calculation from each existing tower and considering azimuth and tilt of existing sectors and a "maximum distance" parameter specifying the admissible range for access-tier wireless links.

We define the following notation for the remainder of the chapter:

- T: is the set of all the towers (deployed and available)
- N: a subset of T containing only the towers currently in use in the network topology.
- $h(t)$: is the “total amount of heat” for tower $t \in T$, defined as the sum of the heatmap cell values covered by the tower.
- $c(t)$: is the deployment cost of the tower t.

At launch, the software performs a pre-calculation step to store the memory structures needed at later stages. The most important is G, an “inter-visibility” undirected graph, in which vertices are elements of T. Two towers $t_1, t_2 \in T$ are connected by an edge if they are in line of sight and not farther apart than the maximum allowed distance for point-to-point links. As G is far from a complete graph, we save G to an internal graph database which allows for efficient storage of sparse graphs, and because the LOS calculation is computationally intensive, the graph database is persistently stored on disk, saving time at subsequent launches.

Based on the data infrastructure described, the tool offers two operational modes: (A) Targeted Increase; and (B) Strategy Search. We describe these two modes in the following subsections.

4.2.1 Targeted Increase

Heatmaps are a visual aid for the network operator to see the areas that would benefit the most by an improvement in coverage. The *targeted increase* operation mode provides the lightest level of automation available in `IncrEase`, keeping the “human in the loop” by asking the operator to visually select on a map the geographical region where coverage should be improved. `IncrEase` then automatically identifies which is the hottest cell in the region, defined as the one with the highest value in the combined heatmap layer. The application determines the set of closest towers (i.e., 20 in the current default settings, as discussed later in Section 4.3.1) from set $T - N$ that are in line-of-sight of the hottest cell to form the set of candidate locations that will cover the hotspot. Considering multiple source towers allows selection from among a larger number of potential backhaul paths, compared to focusing only on the single tower closest to the hotspot.

The software finds the best way to connect each of those towers to the existing network topology (i.e., the set N) by traversing links in the G graph. The "best" solution is the route that provides the lowest value for the $c(t) - h(t)$ difference for each tower t traversed. In this calculation, we carefully avoid accounting multiple times the "heat" associated with a cell that may be in line-of-sight with different towers, as it would bias the results. We consider heat for such cells only once.

For pathfinding over the G graph, `IncrEase` uses the A* (A-star) algorithm. A* uses a best-first search, based on a distance-plus-cost heuristic function, to find the least-cost path from an initial node to a goal node. Our implementation has two slight changes with the original A* algorithm. First, it takes as input a set of start nodes (closest towers to the hottest cell in the selected region) and a set of goal nodes instead of a single start/end nodes, as the backhaul path could start from any of the candidate locations and terminate at any of the existing towers. Second, in the G graph, costs are associated with the vertices (i.e., towers) rather than edges, so we consider the cost of an edge (i, j) to be that of the departing node i (Skiena, 1998).

A* requires a heuristic function that is the minimal lower bound of possible path cost (e.g., for traveling between two cities, it is distance by straight line), so in our case we need to design an estimate of the best $c(t) - h(t)$ achievable for the rest of the path from a given tower to the goal towers. We adopt $(l/d) \cdot c_{min}$ as such a heuristic, where l is the straight-line distance between the current tower being analysed and any one of goal towers, d the maximum distance allowed for point-to-point links (both in km), and c_{min} the minimal $c(t) - h(t)$ of all towers.

Finally, we introduced two modifications to the cost functions presented above. As A* requires non-negative edge costs, we sum an arbitrary constant large positive value to all $c(t) - h(t)$ values. Lastly, to let the user balance the importance of saving money and extending coverage, we allow two variable coefficients c_0 and h_0 and define cost as $c_0 \cdot c(t) - h_0 \cdot h(t)$.

The search result for the best path is presented as a path on the map together with a text indication of which towers to be deployed and their order, as shown in Figure 4.2(d-f).

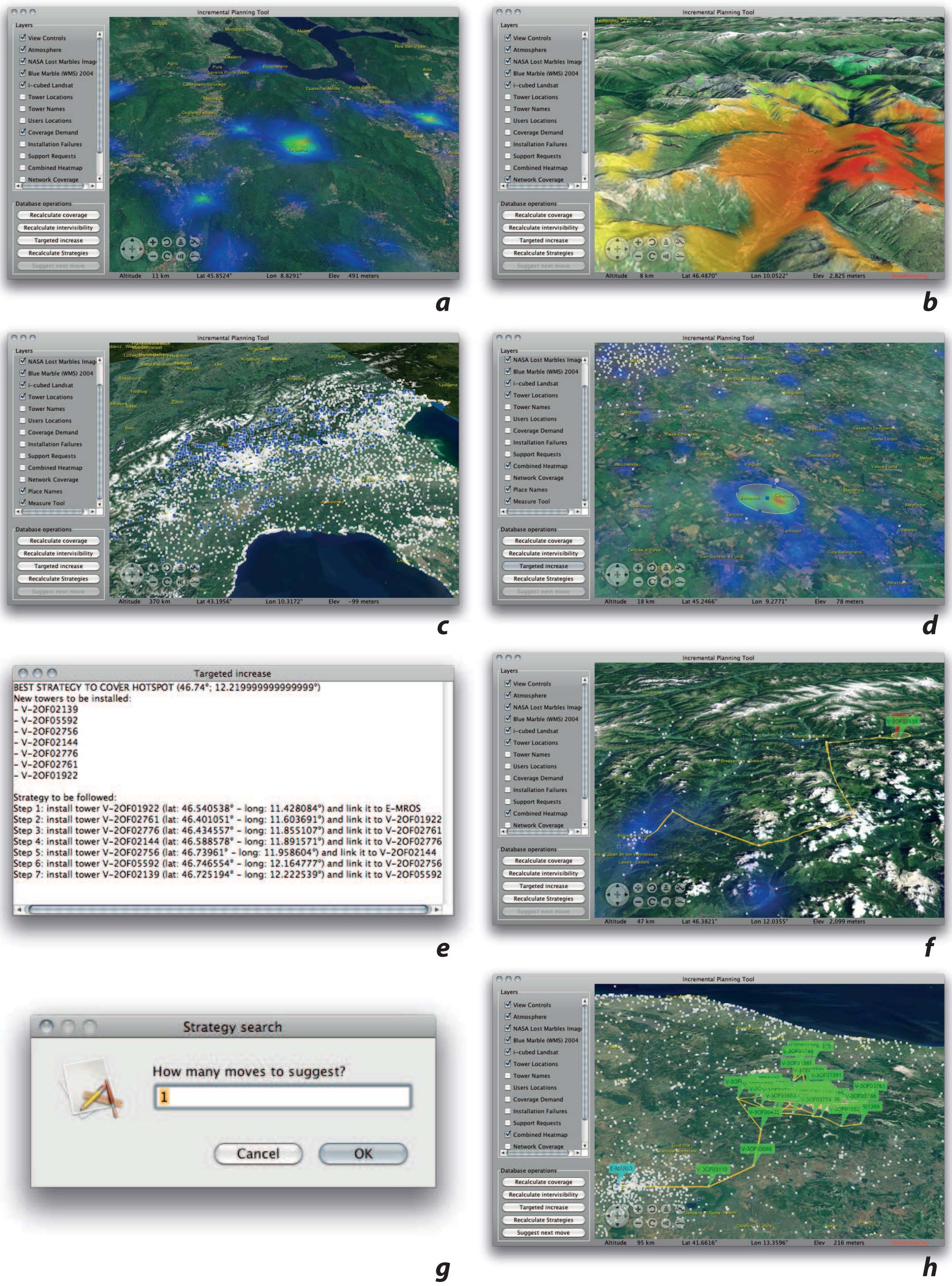


Figure 4.2: *Screenshots of the* `IncrEase` *application: (a) Coverage Demand heatmap; (b) Coverage layer, showing the locations in LOS with any of the deployed towers; (c) Location of the over 8,000 towers in set T; (d) Selecting an area from the GUI for the* Targeted Increase *operating mode; (e-f) Results of the* Targeted Increase *algorithm are displayed as text and on the map; (g) Selecting the number of steps to calculate for the* Strategy Search *mode; (h) Results of a* Strategy Search *are displayed on a map.*

4.2.2 Strategy Search

While targeted increase is a semi-automatic mode that requires the operator to select a region, the *strategy search* operational mode identifies and suggests the best network expansion strategy. We assume the network topology to evolve over arbitrary discrete intervals of time (e.g., weeks, months) and the capital investment budget of the WISP to be limited by a discrete parameter that determines how many moves (i.e., tower installations) can be performed in each time interval. The overall aim of the strategy search is then to suggest the "best" actions that the WISP should take during the next interval of time. An obvious practical limitation is the so-called *horizon effect*: as in many artificial-intelligence games, the number of possible states is so large that it is only feasible to search a small portion of all the potential moves within the time horizon. The search algorithm needs to be able to cut down the number of possible strategies to analyse, while limiting the risk of excluding potentially good ones.

Below we describe the strategy search algorithm, which is triggered via the "recalculate strategies" button in the IncrEase user interface.

Step 1. A "multiple-source lowest-cost" path search algorithm is run on the inter-visibility graph G to identify the lowest-cost paths, with costs being $c(t) - h(t)$ as before, from each of the nodes in the set $T - N$ (e.g., the available towers) to any of the nodes in N. The output is a tree R, which intuitively provides the best path from the existing network to each available tower. To generate R, `IncrEase` adds a fictional zero-cost 'root' tower connected to every tower in N and runs Dijkstra's algorithm from such a root to each node of the graph G. An example is provided in Figure 4.3, where (a) shows the inter-visibility graph G with shaded nodes being those already deployed and (b) the resulting R paths after Dijkstra is run.

Step 2. Graph R is traversed depth-first starting from the 'root' node and, while doing so, towers are tagged with the score:

$$\frac{h(r) - c(r)}{(1+C)^{distance(r)}}$$

where $distance(r)$ is the count of newly-deployed towers that have to be traversed to reach r from the 'root' on R (e.g., towers that can be directly connected to the network have $distance = 0$). As not all towers can be immediately connected to the network, to gain the coverage benefits associated with any one of them, others may have to be deployed first to serve as back-hauling relays. To weight future coverage benefits to

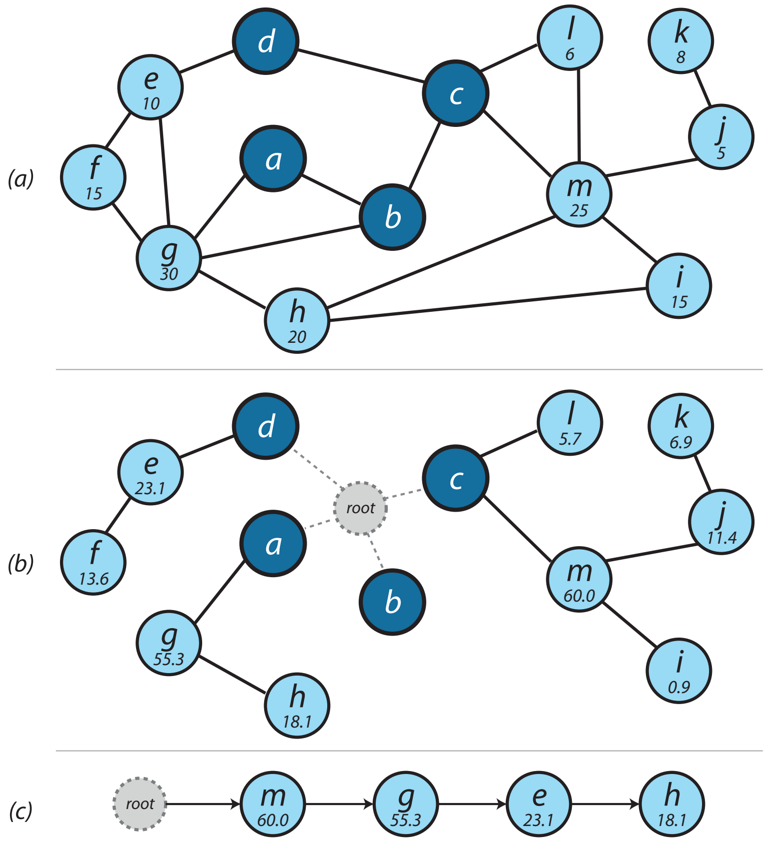


Figure 4.3: *Data structures used in the* Strategy Search *operation mode. (a) G graph. (b) R graph. (c) L list.*

the present day we bring in the financial concept of Net Present Value (NPV), which applies a constant discount rate C (e.g., 5% being 0.05) to revenues that will happen in the future. An example is provided in Figure 4.4(a). Here, two towers a and b could be installed and directly connected to the existing mast n. Nodes b and f each bring a $h(t) - c(t)$ benefit of 100, while all other towers provide significantly lower benefits. Parameter C is a measure of the greediness of the selection: if we were to pick between a and b based on the total benefit that they and their descendants could bring, we would decide to install tower a as in Figure 4.4(b). However, if we increase the value of C to 5%[4], b becomes more attractive as in Figure 4.4(c). NPV controls how far it is worth going for installing profitable towers, allowing the network owner to tune the duration

[4]The $C = 0.05$ value is provided here as an example of what may be considered reasonable by an ISP. In reality, the operator may want to tune C to express the intention to invest for future profits in a given region, in a way that could be supported by its cash flow.

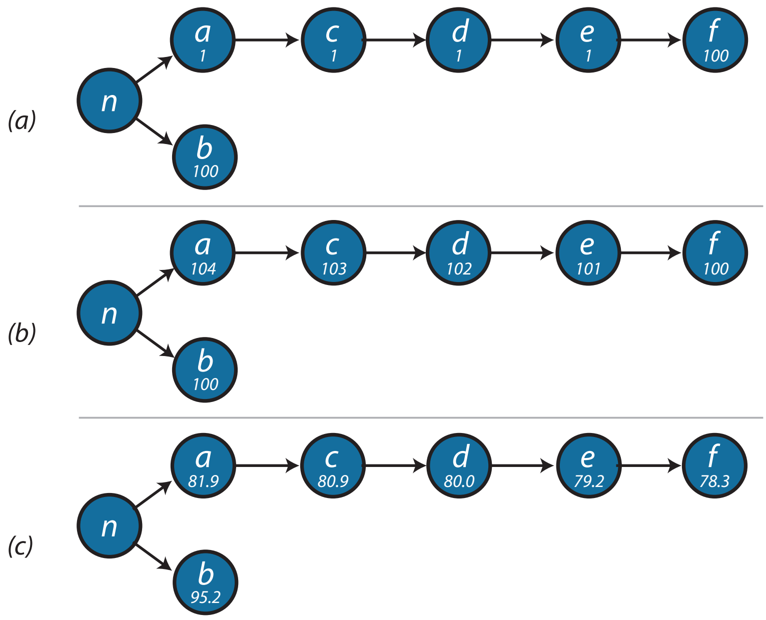


Figure 4.4: *Net Present Value based score adjustment on an example network (a) in Strategy Search mode: increasing the discount value, C, from 0% (b) to 5% (c) changes the greediness of the selection.*

of the benefit delay.

Step 3. R is traversed again, this time from leaves to the root. While doing so, we update the score of each node r to the sum of its own score and that of its descendants. Figure 4.4(b) and 4.4(c) actually show the scores obtained after step 3.

Finally, at each click on the "suggest next moves" button of the UI, `IncrEase` asks for the number of moves (towers to be deployed). It then generates a sorted list L (Figure 4.3(c)) which includes the towers that could be immediately deployed, ordered by decreasing benefit score as calculated at the end of step 3; subsequently, it extracts the top nodes from L; and, finally, presents them on the map as results.

4.3 Evaluation

We have implemented a prototype of `IncrEase` that implements both operation modes described in the previous section. It is being used at NGI SpA, a large Italian WISP operating a fixed 802.11 and 802.16e wireless access network. The service covers Northern Italy, including both metropolitan cities, towns and small rural villages (further

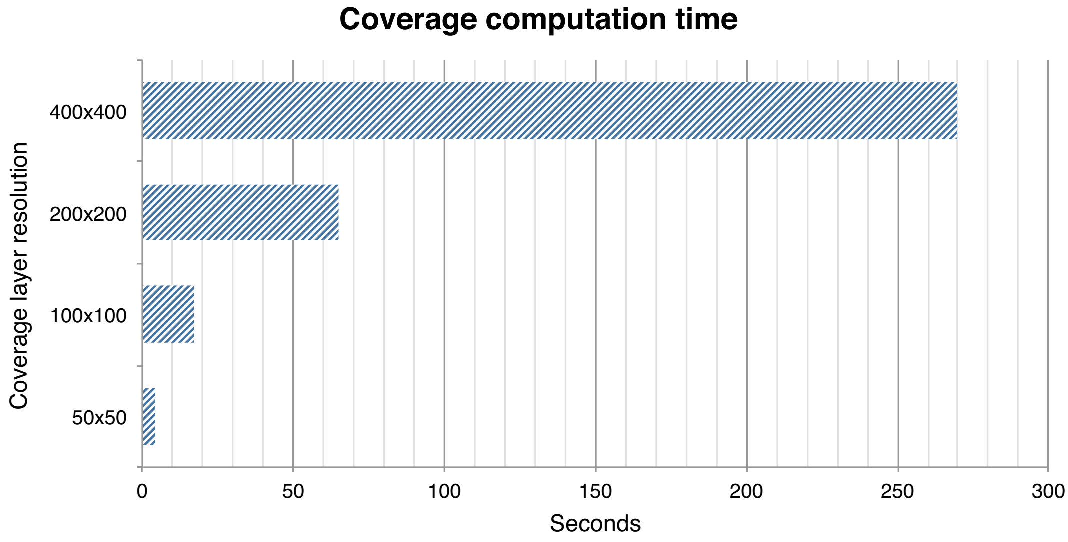


Figure 4.5: *Benchmarking the computation time of the coverage layer on a network of 513 towers on an area covering the whole of Northern Italy.*

details are provided in Section 2.1.1). As WISPs commonly do, NGI has agreements with mobile operators and TV/radio broadcasters in order to acquire space on existing towers, resulting in over 8,900 towers available for immediate installation. The existing network spans over 513 transmission sites, mostly connected over wireless point-to-point (PTP) links of up to 7km in length. For determining coverage, NGI uses a simple line-of-sight (LOS) criteria with a 20km maximum allowed distance, since the access tier operates on the 5.4-5.7 GHz spectrum with outdoor CPEs typically on customers' rooftops. NGI's network is a typical BWA network, so we use it as a representative scenario to benchmark `IncrEase` and demonstrate its usefulness.

A large set of performance metrics has been obtained including coverage requests from prospective clients, details of customers that could not be connected to the network because of insufficient coverage, the log of support request received at NGI's helpdesk, and the geographic location of all current users. We imported this data set in `IncrEase` to generate heatmaps and drive the incremental planning process, as outlined in Figure 4.1.

The first step to visually assess areas that would benefit the most by an improvement in coverage is to calculate the current coverage extent. Figure 4.5 shows the time

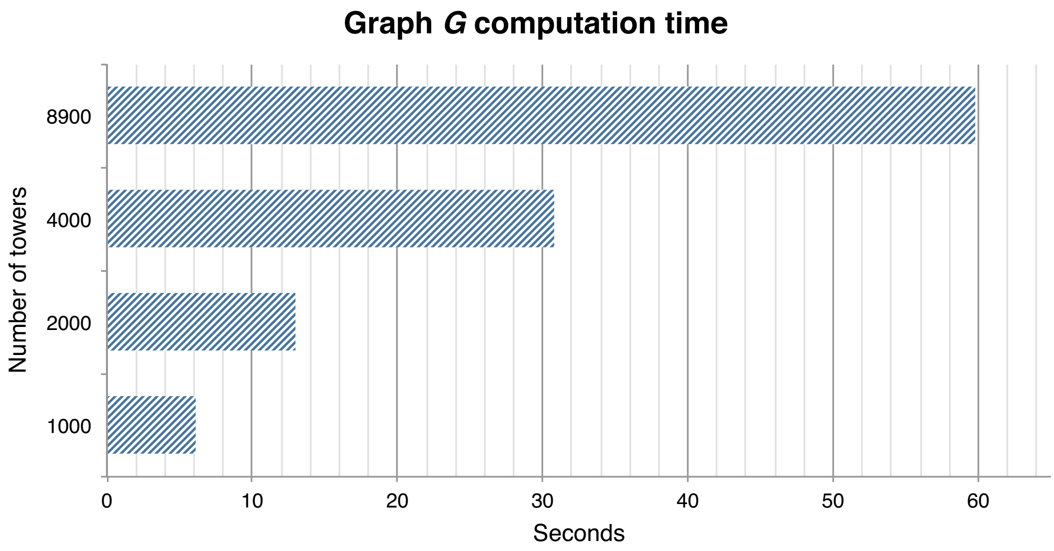


Figure 4.6: *Time taken to calculate the inter-visibility graph* G*, depending on the number of available towers.*

taken to calculate and display on a map the area in LOS with any of the towers in the existing network over an area of 273,000 square kilometres. The coverage calculation considers the type, orientation and number of sector antennas installed on each tower. In NGI's network, there are normally 3 sectors per tower, except small towers in mountain areas composed by a single omnidirectional cell, and a few critical sites where up to 20 sectors have been installed to add capacity. Benchmarking results, taken on a 2.7 GHz dual-core CPU, validate our implementation by showing strictly quadratic time complexity on the map resolution (in points per latitude/longitude degree), which is expected as the number of points in the region has a power of two relationship with the resolution.

Computation time for building the inter-visibility graph G (Figure 4.6) shows linear complexity with the number of available towers in the network, taking a minute to compute G with 8,900 available towers: we get this result by comparing the LOS path of only those tower pairs that are within the maximum allowed distance configured for PTP links (7km in our experiments). It is important to note that both the coverage layer and G are persistently saved in the internal graph database and are recomputed only if needed.

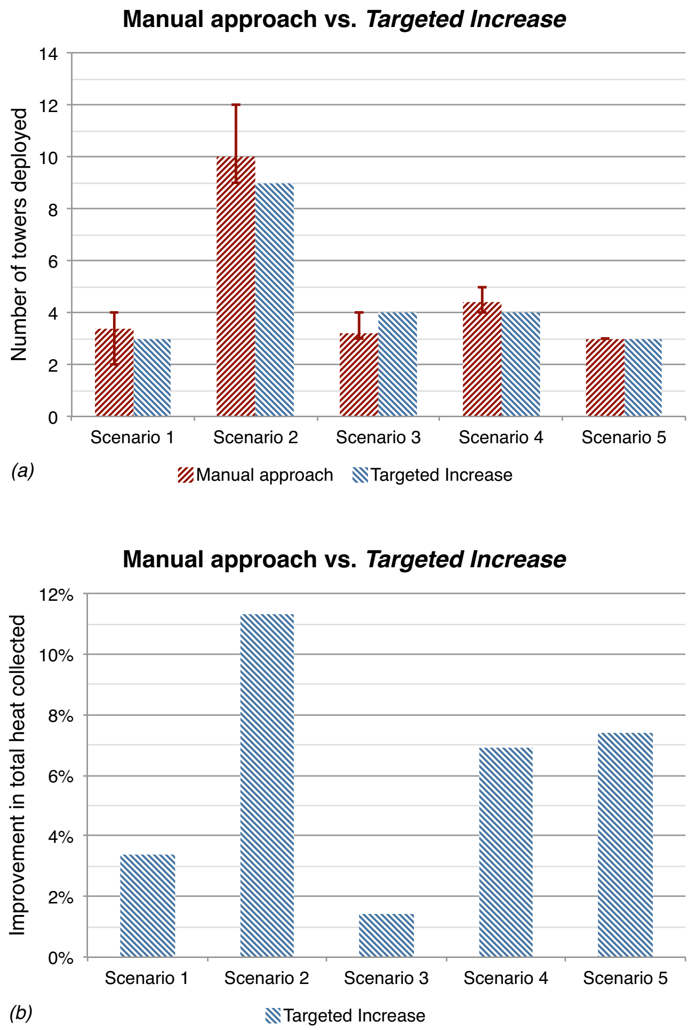


Figure 4.7: *Comparison, in terms of* (a) *average number of new towers deployed (bars shows best and worst case) and* (b) *total "heat" collected, between manually planning five deployment scenarios and by using the Targeted Increase mode in* `IncrEase`.

4.3.1 Targeted Increase

Targeted Increase assists the network operator in finding the "best" strategy to cover a given geographical region and to build a set of suitable back-hauling connections to link it back to the existing network. `IncrEase` provides a suggestion in real time, by exploiting A* heuristics to navigate the inter-visibility graph G. We assess the quality of the proposed solutions with the following comparative study. The locations of available and existing towers were given to five different wireless engineers at NGI, with the request to cover five given small geographical areas and to propose valid backhauling strategies. Engineers could use any tool or technique to work their solution. We then ran the same scenarios on `IncrEase`, and compared the results in terms of number of new towers deployed and total heat removed from the map. Results are presented in Figure 4.7: (a) in 4 cases out of 5, the strategy suggested by `IncrEase` requires fewer or equal number of towers to be deployed than solutions provided by wireless engineers; and (b) in all 5 cases better in terms of "quality" (total heat collected). Results demonstrate that Targeted Increase is able to find routes between towers that engineers may overlook. Also, in Scenario 3, `IncrEase` suggests to deploy one tower more than the engineers proposed in order to collect more heat. Based on this data, we believe that without the aid of a tool like ours and heatmaps, planning the best backhaul route may be non-obvious, even for a skilled technician, especially when a large number of towers are available. The algorithm complexity and behaviour can be controlled by several user-configurable parameters. One of them is the "initial towers set size": `IncrEase` determines the set of closest towers in sight with the hottest spot of the selected region, and tries to find a backhauling path from each of them to any one of the towers in the existing network. Clearly, this parameters is a trade off between computation time (the higher the set size, the larger is the number of paths that `IncrEase` will analyse) and quality of the results. In Figure 4.8 we analyze the total amount of heat removed ($h(t) - c(t)$) of the best solution that `IncrEase` could find for 3 selected scenarios on NGI's network, for different sizes of the initial tower set. The average time taken to compute each solution is also included. While computation time increases steadily with the number of initial towers, the net amount of 'heat' removed tends to growth marginally after the most obvious solutions (i.e., those starting from the closest towers to the hottest point of the given region) have been analysed. Based on this graph, we default this parameter to 20 towers in our current implementation, which is around the inflexion point of the heat removed curve.

Finally, the solutions generated with `IncrEase` were given back to the wireless engineers, in order to get feedback about their quality and suitability. We learnt from the engineers that, while all the solutions proposed by the tool are considered valid, in some cases they may not have been the most straightforward to build for a number of

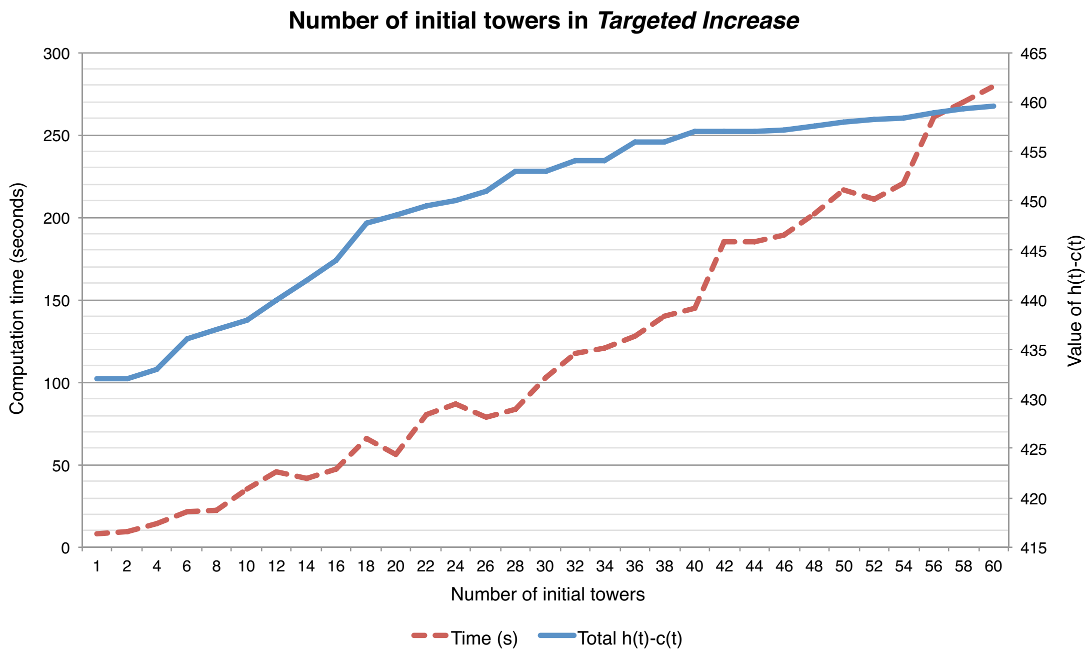


Figure 4.8: *The "initial tower set size" parameters of Targeted Increase is a trade off between computation time and quality of the solutions identified.*

reasons, among which:

- some towers are known to be almost completely full, with the remaining space in unattractive locations of the mast (e.g., low altitude);
- obtaining permit to install on some towers may take longer or to attach additional required documents to the application (e.g., towers located in natural reserves);
- some towers may be more attractive than others because of ease of access (e.g., availability of roads), structural robustness or type of the tower itself;

In the existing implementation, `IncrEase` does not include any of these details in the tower model (e.g., available space, ease and cost of obtaining permit to install, structural quality, etc.). A simple workaround could be to add an optional 'ignore' flag that can be set to any towers that the engineer wants to exclude from the calculation: `IncrEase` would then try to suggest a solution that does not include any of the 'ignored' towers, keeping them for last-resort.

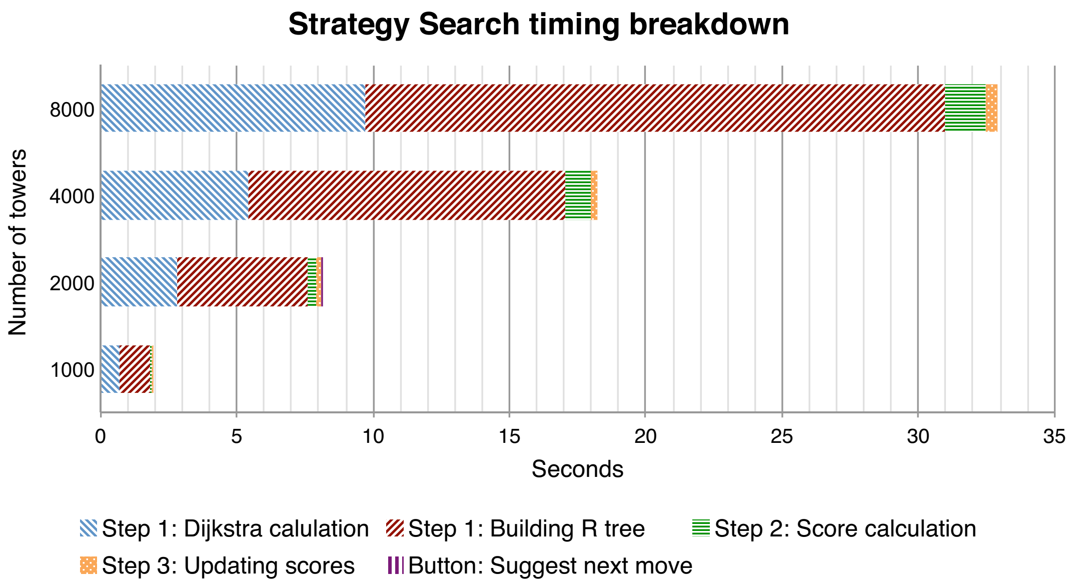


Figure 4.9: *Profiling of the Strategy Search operation mode, showing the computation time for different steps of the algorithm.*

4.3.2 Strategy Search

The Strategy Search operation mode analyses the current network topology and the set of available towers to identify the long-term strategy that leads to the best long-term results. When the "Recalculate Strategies" button is pressed, the three steps presented in Section 4.2.2 are executed. Figure 4.9 shows the time taken for their completion, which is clearly linear with the number of available towers. Note that timing of Step 1 is split between our modified Dijkstra calculation and the construction of the *R* tree in memory: the latter is a relatively more expensive operation as it requires the creation of a new Neo4J graph structure, which is then populated using Dijkstra's output. Finally, at each request to retrieve the next *n* best actions, the algorithm performs an $O(1)$ operation by selecting the top *n* elements from the ordered list *L*.

4.4 Summary

In this chapter we have presented `IncrEase`, an open-source tool for *incremental planning* of rural BWA networks. Although the `IncrEase` implementation is currently only a prototype, it efficiently handles large data sets and it is being trialed by the network engineers of NGI SpA, a large BWA operator which provided the real-world data we used for our evaluation, including a database of over 8,900 real towers. We are gathering results and suggestions from their daily experience with the tool, from which we expect to learn how to tune the algorithm parameters (e.g., the discount rate C, the A* weights, the maximum allowed distance for PTP and PMP links, etc) and which new feature to implement. In Section 8.2 we discuss the enhancements that could be done to the `IncrEase` tool in the future.

Chapter 5

Stix: a Goal-Oriented Distributed Management System

5.1 Introduction

In the previous chapter, we proposed a scheme for automated network planning of BWA networks. As soon as the network deployment starts, equipment needs to be configured, monitored and controlled. In this chapter, we introduce a network management framework designed for BWA networks, especially those located in rural areas and developing regions.

Network management encompasses a wide range of activities as captured by the ISO FCAPS model (ITU-T, 1996), which defines five areas of network management: fault, configuration, accounting, performance and security. We focus in particular on performance, fault and configuration management. These include activities such as monitoring for performance bottlenecks and faults via device level statistics collection, and upgrading the software on managed devices (e.g., PTP devices, BTSs, CPEs).

BWA networks (and more generally, wireless networks) are inherently complex to manage (Raman and Chebrolu, 2007), given the wide range of parameters and environmental phenomena (e.g., radio propagation, mobility) that can affect network operations. BWA networks can also be quite diverse from rural/community networks to large-scale WISP deployments, so network management platforms should be adequately flexible to suit a wide range of deployment scenarios. However, *in-band management* (i.e., management traffic going over the production network being managed) is one aspect that is common to all BWA network scenarios as a dedicated management network infrastructure maybe infeasible or impractical due to the additional cost and deployment burdens. Consequently, management traffic contending with user traffic is an issue and as such it is an overhead. Today is a time of concern with network management systems in use, as they employ a centralized paradigm that puts most of the management related intelligence at one location, typically called a network operations center (NOC). The information about all the managed devices needs to be continually available at the NOC for network monitoring and control decisions. Clearly this becomes an even bigger concern as the network size increases. We use the term "scalability" to refer to this issue.

Rural or community networks present an additional set of issues: (a) Internet connection is often the most expensive part of the operational expenses for such deployments, so it is not desirable to have communication with the NOC interfering with the efficient use of that precious bandwidth; (b) unreliable connectivity between the NOC and the network being managed seen in rural networks can render the centralized approach ineffective; (c) the number of skilled personnel available to maintain such networks is limited (often, just one person from the community).

Simplifying network management is therefore crucial, more so in developing regions as shown by the experiences of Surana et al. (Surana et al., 2008b). Simplifying the specification of network management goals is also key for effective distributed management (as outlined under the management by objectives paradigm in Clemm (2007)). Finally, as network size increases, the potential for using heterogeneous devices increases too (Surana et al., 2008b), so it is important to be able to seamlessly support management of devices from different vendors.

The need for network management platform and tools becomes more important as the size and complexity of the network to be managed increases. Both of these are true for BWA networks — wireless network management is more complicated than in wired networks. However, most network management systems in use today are infrastructure heavy in the sense that network operators rely on expensive network operations centers (NOCs) to centrally manage their networks. While centralized management system may be easier to implement, it has the obvious scalability limitation of having high communication overhead due to network management traffic between the NOC and devices being managed. More crucially, as small- and medium-sized enterprises

(SMEs) and community organisations may form the bulk of the BWA network operators, the cost associated with maintaining a NOC may force them to do away with network management altogether or at best employ low-cost, ad hoc approaches (e.g., ping checks). Also, unreliable connectivity between the NOC and the network being managed seen in rural deployments can render the centralized approach ineffective.

We developed a novel network management framework and system called `Stix`, motivated by the above considerations. The `Stix` distributed management architecture facilitates in-network execution of monitoring and control operations as well as in-network storage of management data. Thus it reduces the reliance on the central NOC for management operations and storage of management data. This, however, does not mean completely eliminating the NOC and the network administrator — they are still needed for specifying network management activities, network visualisation/analysis, software updates, billing and accounting operations.

A key contribution of our work is the implementation of the `Stix` system, including the `StixAgent` on embedded boards (Section 5.3.2). We demonstrate the ability of `Stix` to support management of heterogeneous devices. Using the network topology and logging data from our partner WISP NGI SpA (see Section 2.1.1), we evaluate the scalability and efficiency of the `Stix` distributed monitoring and control approach relative to the traditional, centralized SNMP-based management approach and show that `Stix` approach is more scalable and efficient. We also present two case studies using the implemented `Stix` system to demonstrate the ease with which it can be used for realistic network management activities such as seamless device reconfiguration and adaptive spectrum management.

In contrast to the commonly used centralized and/or vendor-specific network management tools (e.g., OpenNMS, Nagios, Alvarion Star Management Suite, Motorola Prizm), `Stix` enables distributed and scalable management of large-scale multi-vendor BWA networks. Simplifying the specification of network management activities using a graphical workflow-based modelling language is a feature unique to `Stix`. Existing work on BWA network management tends to focus on decision making processes in the context of self-management (Schuetz et al., 2007; Baliosian et al., 2008) without regard to the underlying implementation platform. No such platform exists currently. We provide a flexible framework and platform that fills this void, thus enabling self-management of BWA networks. Although self-management *per se* is not the focus of our work, the case studies as part of our evaluation do demonstrate the feasibility of using `Stix` for self-management processes. While being focused on commercial and non-profit deployments, our tool can also support research on wireless network. For example, its pragmatic approach is useful to instrument experimental devices, collect results on large-scale testbeds or provide control on prototype hardware.

The remainder of this chapter is structured as follows. An overview of the `Stix` framework and system is given in Section 5.2, and its design and implementation are described in Section 5.3. Its efficiency and scalability using data from a real large-scale WISP as well as realistic case studies are presented in Section 5.4. In Section 5.5 we demonstrate the usefulness of `Stix` on our Tegola network. Possible directions for future research on `Stix` are given later in Chapter 8.

5.2 Stix overview

We consider the problem of developing a network management platform for broadband wireless access (BWA) networks. As already described, our design requirements for such a platform are as follows:

(a) Simplified network management to allow us enable the network administrator to focus on management goals rather than burden him with the tedium of low level details. This is especially important in community deployments and developing region settings;

(b) Given that management in BWA networks is likely done in-band, the management platform has to keep the overhead low, thus it scales better to large deployments;

(c) Reduced dependence on a centralised Network Operations Center (NOC) can potentially enable the management system to function smoothly even during periods when the NOC is unreachable;

(d) It has to seamlessly support multi-vendor devices, which are likely to be the rule rather than the exception, especially in large-scale deployments;

(e) Lastly, the platform should be flexible enough to facilitate self management.

We present a novel network management platform for BWA networks called `Stix` that meets the aforementioned requirements. `Stix` design is based on two key principles:

1. goal-oriented management;

2. the network is the NOC.

Regarding the first principle, *goal-oriented* means allowing the administrator to focus on the network management objectives and the specification of associated activities rather on the low level details. To realise this principle, `Stix` incorporates a high-level, visual, workflow-based modelling language (referred to as `StixL`) to easily express network management activities, enabling the administrator to describe the goals of the network by modelling processes as workflows, thus meeting requirement (a) above. A workflow is defined as a sequence of tasks that needs to be performed in order to achieve a high level network management goal (e.g., upgrade firmware on all CPEs). A workflow can be applied to a specific device or to a set of them with the aid of associated qualifying expressions in a purpose-designed, query language; it is formed by combining pre-defined elements such as decision gateways, event triggers and purpose-written code called tasks that takes the form of pluggable boxes to facilitate code reuse. `StixL` also helps realise a distributed management architecture by providing a flexible way for specification of network management activities that are actually executed at management entities inside the network.

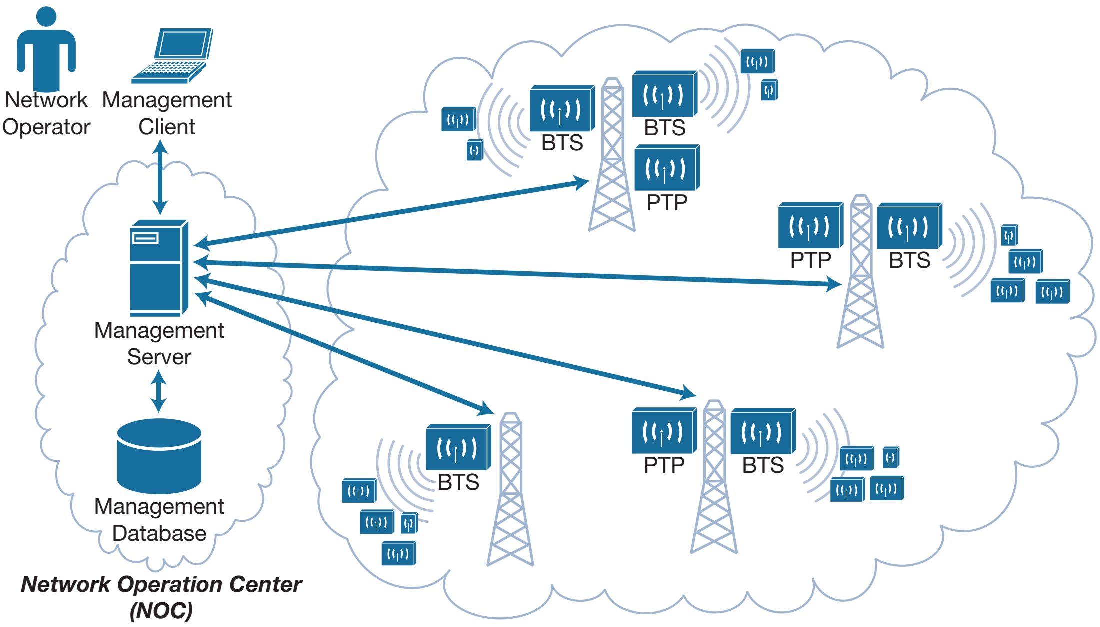
The second principle is realised by adopting a distributed cooperative agent management architecture for monitoring and control that pushes the manager (called `StixAgent`) of a device being managed closer to that device. A workflow that is designed using `StixL` is converted into an XML file and disseminated to relevant `StixAgents` in the network, which generate executable code on the fly. This principle is essentially aimed at improving scalability by dividing and distributing network management activities within the management infrastructure. Making the management hierarchical and introducing layers of abstraction are well known strategies to tackle network management related scalability and complexity (e.g., arising from heterogeneity) issues (Clemm, 2007). By employing a distributed cooperative agent-based architecture, `Stix` meets the requirements (b) and (c) above.

Another unique aspect of `Stix` is that logs generated as a result of executing network management activities (e.g., monitoring statistics) reside *in* the network at `StixAgents` and are replicated locally around the source for high availability using a mechanism called *Sprinkle* that implements a "*log overlay*" within the network. Management data stored on the log overlay is available to the network administrator for on-demand and asynchronous retrieval via a web based graphical user interface with wiki-like syntax and SQL-like querying called `StixView`, thereby decoupling the execution of a network management activity from the retrieval of its results.

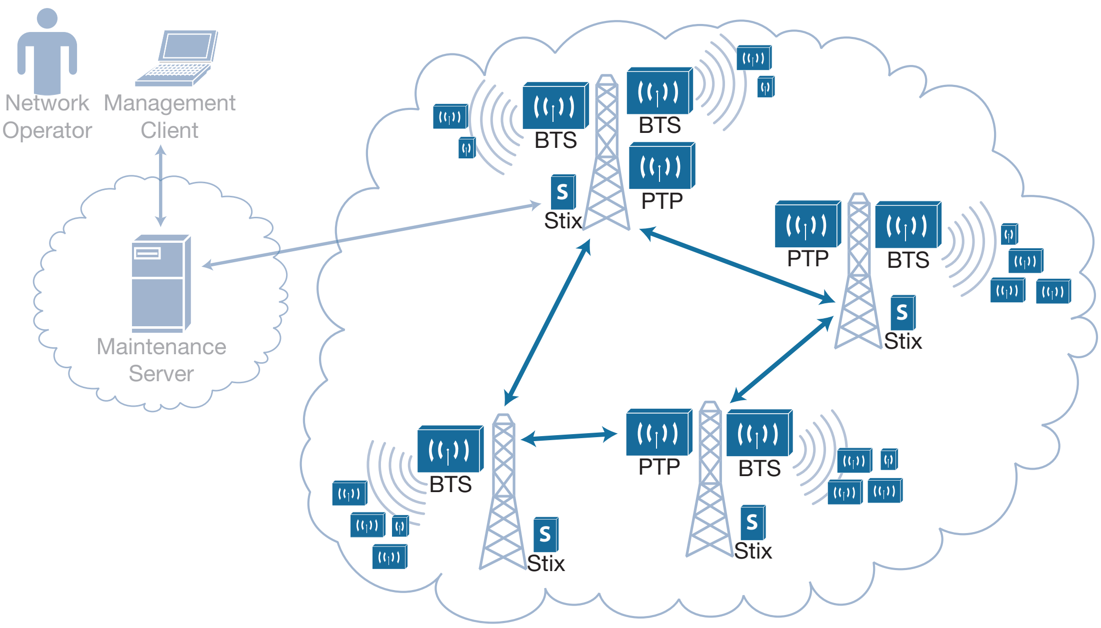
`Stix` also provides a *hardware abstraction layer* in the form of a device manager within each `StixAgent` to support devices from multiple vendors. Finally, the `Stix` framework is technology independent, so it can be employed for managing IP-based BWA networks using any underlying wireless communications technology (e.g., outdoor WiFi, WiMax/LTE, etc).

Figure 5.1 compares the `Stix` distributed management architecture with the traditional approach. Essentially, to satisfy a network management goal, `Stix` deploys a corresponding workflow to the appropriate set of management entities (referred to as `StixAgents`) situated at transmission sites, which execute the workflow locally and usually upload the results to the *log overlay* using a replication mechanism called *Sprinkle*. The log overlay is, in other words, an in-network overlay storage system for keeping the logs (e.g., monitoring statistics). The log overlay is asynchronously queried via the `StixView` web interface *as needed* by the administrator to fetch the monitoring results, network health status updates and so forth. Thus the `Stix` system shifts the burden of monitoring, control and storage from the NOC to `StixAgents` via workflows and the log overlay store, thus reducing the dependence on the NOC; the NOC is only used for a limited set of operations such as software updates, network visualisation, billing and accounting. The hardware abstraction layer within each `StixAgent` helps meet requirement (d), and all the above components in `Stix` collectively satisfy the last requirement (e).

We now discuss the types of network management activities that can lead to reduction in management traffic using `Stix`. It is worth noting that event-based management is more scalable and responsive compared to the repeated polling based approach commonly used by network management systems (Clemm, 2007). `Stix` naturally supports event based management as `StixL` workflows are event driven.



(a) Traditional Centralized Management



(b) `Stix` Distributed Management

Figure 5.1: *`Stix` management architecture compared to a traditional centralized architecture.*

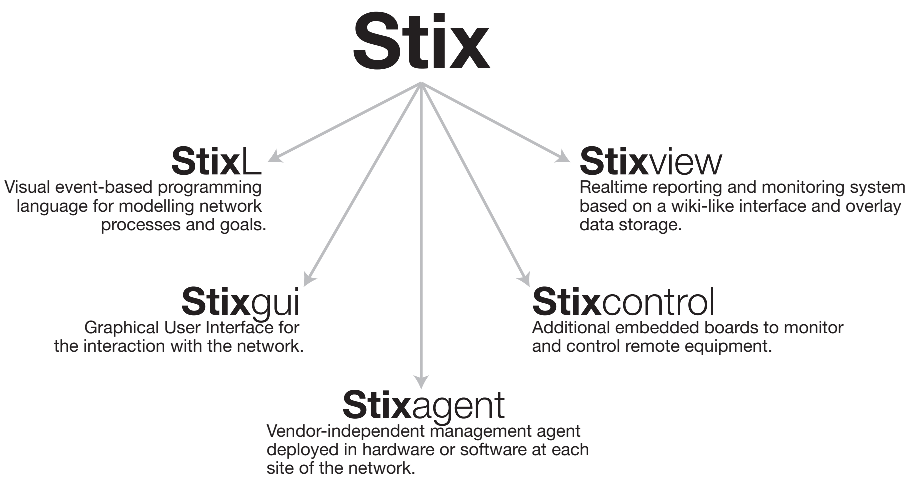


Figure 5.2: *`Stix` system components.*

`Stix` also realises management by delegation. There are a wide range of network management activities that can be delegated to other management entities or agents in the network such as logging, de-duplication and correlation of events, polling of devices for statistics, preprocessing of statistical information and software upgrades; these span all areas of network management from fault and performance management to configuration and accounting management (Clemm, 2007). We consider the specific case of firmware upgrades in our evaluation (Section 5.4.1.1). A different way to identify cases benefiting from `Stix` is to look at network management activities as a combination of monitoring and control (especially true for performance and fault management).

All such activities that can be done in a self-managing manner as a collection of workflows executing on a distributed set of agents can gain from using `Stix`. Even activities that are inherently centralized and require polling may benefit from in-network processing and aggregation (e.g., usage data for billing purposes). In-network storage and on-demand retrieval of management information further contributes towards system scalability as typically a network administrator is interested in querying a small portion of a network (e.g., a transmission site or a subnet). Execution of monitoring and control operations at `StixAgents` naturally provides opportunities for delegating management tasks and in-network aggregation of management data, thus improving scalability.

In the following, we describe the design and implementation of each component of `Stix` (see Figure 5.2).

5.3 Stix system design and implementation

5.3.1 StixL: a visual language for describing network processes

From our perspective, a *'network process'* is a set of activities performed during the network lifecycle. Examples of processes are routine software upgrades, periodic reporting of link status, and emergency routing reconfigurations. We believe that for a network management scheme to be successful, the administrator must be able to focus only on the operational goals and to express them in a natural language. Our proposal, `StixL`, is a visual programming language that enables network processes to be described graphically, enables software reuse and hides the complexity of dealing with heterogenous hardware.

`StixL` is loosely derived from the Business Process Modelling Notation (BPMN) [1], a graphical representation developed by the Object Management Group for specifying business processes as workflow. We have adapted the concepts from the business community to the specific wireless networking domain. The language is composed of only 17 elements, shown in Figure 5.3, which are used in flowchart-like representations called *workflows*.

Round symbols are *events*, and can be classified into three categories: 'start', 'intermediate' and 'end'. Start events trigger the execution of a workflow using timers, incoming messages or conditions. Intermediate events can be used to send and receive messages, to log activities or control the execution flow via error handling. Finally, 'end' and 'terminate' events determine the end of a workflow execution.

A core language concept is represented by the *'task'* element. These are elementary units of work (e.g. a network `ping`, a reboot command, routing control functions) and are implemented as hot-pluggable Java classes. In `Stix`, the administrator is presented a library of common tasks that can be (re-)used in workflows. This library can be continually expanded by writing new tasks or by importing them from 3rd parties.

A workflow is contained in a rectangular *pool* element (Figure 5.3). This has an

[1] Business Process Management Initiative, `http://www.bpmn.org/`

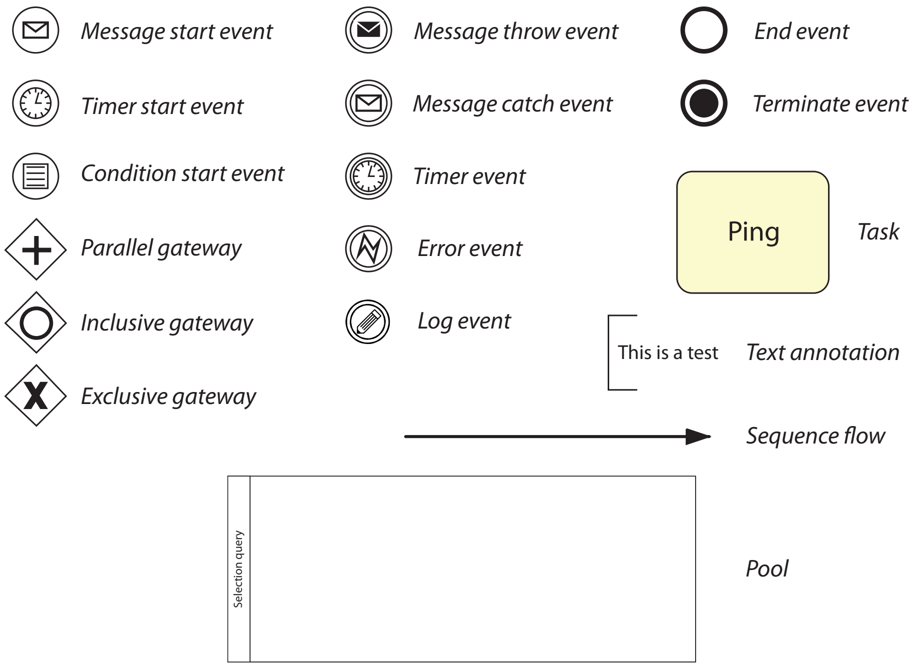


Figure 5.3: *The* `StixL` *language elements.*

associated query string, which lets the administrator specify the set of devices that fall within the scope of the workflow. The query is expressed in a well-defined syntax, which allows queries to be evaluated against the properties of a particular device. As an example, the query `ON devicemodel='VendorA' AND uptimedays>'10' DO` will cause the associated workflow to apply to all devices of model 'VendorA' with uptime larger than 10 days. We note that while a workflow is executed separately for each device that matches the query executed at the corresponding `StixAgent`[2], a single execution can affect more than one device by using the "message send/receive" events. Additionally, each workflow has associated metadata information which includes a globally unique identifier, a revision counter, and optional author names and notes. Metadata also includes an option to limit the temporal validity of a workflow (e.g., "don't run before..." and "don't run after...").

`StixL` programs are coded visually using a graphical user interface, which generates an equivalent text serialisation in XML format. An XML Schema is used to parse

[2]In `Stix`, each managed device comes under the purview of a unique `StixAgent`. For example, a CPE is typically managed via the `StixAgent` attached to the transmission site with a BTS with which the CPE is associated.

and validate workflows that are exported from the GUI and exchanged between agents.

`StixGUI` is a web-based application that runs on a centralized server and allows the administrator to design and deploy new workflows and to edit existing ones. It is divided into three interfaces: a topology view, a repository with the list of existing workflows and the actual workflow editor.

The topology view (Figure 5.4(a)) shows the `StixAgents` deployed on the network, with links showing the overall network topology. It is also possible to see the various devices managed at each site (PTP devices, BTSs, CPEs, etc.).

The repository in Figure 5.4(b) illustrates all the workflows existing in the network, their name, a brief description, their revision number and other optional details. From this screen it is possible to open the actual process edit window, Figure 5.4(c), which we built on the existing Oryx editor project[3] (Decker et al., 2008). The editor presents the administrator with the following: a list of `StixL` elements along with the task library; a 'flow mapping' pane to compose the active workflow and wire elements and tasks in sequence; and a 'data mapping' pane to map workflow variables to input and output parameters. These two panes are necessary to define both the order in which events happen, also called "execution flow", and to wire the input/output assignment between successive tasks in the workflow (i.e., the "data flow").

In Appendix C we give further details about `StixL` and its grammar, and in Appendix D we show prototypes and schematics of `StixGUI`.

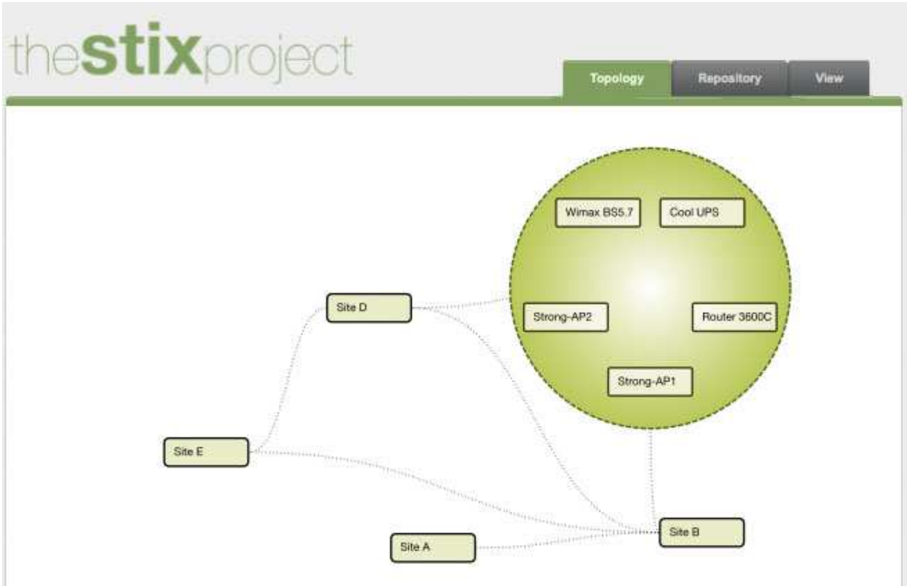
5.3.2 StixAgent: distributed monitoring and control of heterogeneous hardware

`StixAgent` is the core component of the system, as it enables distributed monitoring and control of remote devices. It is internally composed of six parts (Figure 5.5):

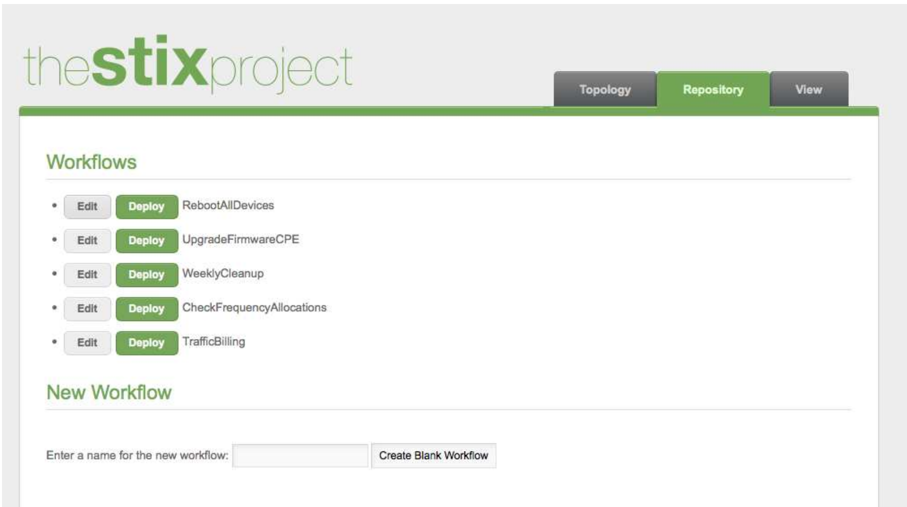
- a "communication manager" that connects with other agents via SOAP[4] messages. It listens on a network socket for incoming messages and subsequently dispatches them to the appropriate internal part.

[3] `http://www.oryx-project.org`

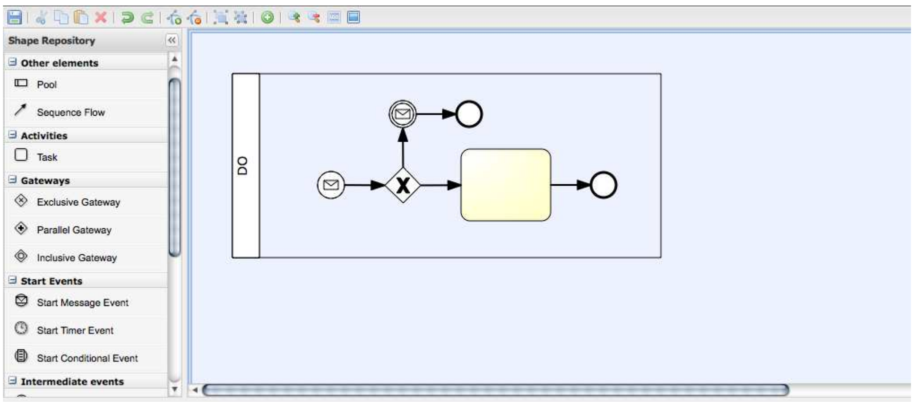
[4] Throughout we use SOAP as the messaging protocol in `Stix` because of its extended support in Java and because it is efficiently verifiable against an XML schema, which enabled us to implement a straightforward validation of all incoming messages.



(a) Topology view



(b) Repository view



(c) Editor view

Figure 5.4: *Three sample screenshots of the `StixGUI` in action.*

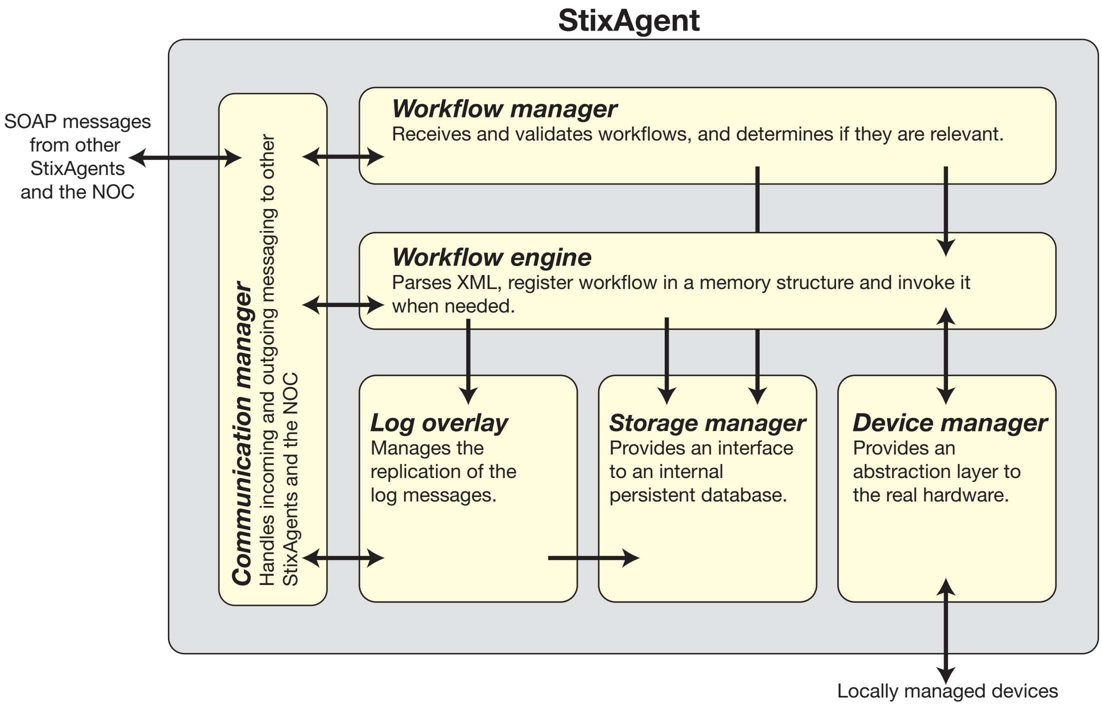


Figure 5.5: *The `StixAgent` architecture.*

- a "workflow manager", which receives new workflows via the communication manager and determines if they are 'relevant' for the locally managed devices. If so, it stores them to disk, transforms the XML representation to appropriate memory structures and passes them to the workflow engine.
- a "workflow engine", which registers and schedules the execution of a workflow by interpreting its XML representation.
- a "log overlay", that keeps track of the most recent log messages originated locally or at neighbouring sites.
- a "storage manager" that provides persistent storage with an appropriate database interface.
- a "device manager", which is responsible for communication with locally managed devices (see Section 5.3.4).

5.3.2.1 Communication Manager

`StixAgent` consumes and distributes information by using SOAP messages sent over TCP. Each `StixAgent` runs a server, controlled by the Communication Manager module, which accepts four kinds of XML messages: Log, Event, Workflow and LogQuery. When the Communication Manager receives a message, it determines the message type and finally passes it to the appropriate component within the `StixAgent`.

When workflows are designed and deployed to the network, the `StixAgent` forwards them using a purpose-designed flavour of directed flooding, which we call *Pick and Forward* and is implemented between the Communication Manager and the Workflow Manager (described in the next subsection). In this technique, a `StixAgent` forwards a workflow to each of the neighbouring `StixAgents` except the one from which the workflow was received ("Split-horizon"). In case any of the outgoing links is temporarily unavailable, the workflow is locally cached.

5.3.2.2 Workflow Manager

The Workflow Manager component first performs XML and logical validation on the incoming workflow program. For example, the number of starting and ending events are checked from the workflow metadata to assert that there are at least one of each. If validation succeds, then the Workflow Manager distributes the workflow message to the neighbouring agents and also forwards it to the Workflow Engine. However, in order to preserve memory, not all workflows are registered in memory by the agent. Instead, the query string contained in the Pool is evaluated in order to determine whether the particular workflow will ever be executed on the local devices. Since queries can involve fields that change over time, the pick and forward technique uses simple heuristics to determine whether an agent should pick up and locally store a workflow passing by. For example, a workflow for which the Pool query specifies a minimum value on a monotonically varying field (e.g., the packet counter on an interface) is considered true if the given value is higher than the current field value. By adopting the *Pick and Forward* technique, we are able to save on the memory usage within each `StixAgent`.

5.3.2.3 Workflow Engine

The Workflow Engine registers the workflow XML in memory, creates a set of supporting data structures and manages the actual workflow execution by interpreting its XML representation. Start events can be triggered by a timer, a condition or an incoming message. In all cases, the engine looks for the corresponding start event, allocates a structure for local variables (which can be optionally declared "persistent"), creates a new thread for the workflow and starts it. The thread interprets the workflow by following the paths through the events and by dynamically loading tasks.

When the execution flows reaches a task, the engine tries to dynamically load the Java class defined for the task. If the binary format of the task is not available locally, the Agent tries to download it directly from the Task Library at the NOC. Each task offers a *runTask* method as an entry point for execution. Once an end or terminate event is reached, the engine deallocates the local variable structure and ends the workflow thread thus freeing its resources. Depending on the execution flow of a workflow (e.g., if there is a loop in the workflow specification), the corresponding thread can be active for an arbitrarily long period of time in which case the associated local variable structure remains allocated.

5.3.2.4 Log Overlay and Storage Manager

Using the 'Log event' element, shown in Figure 5.3, a workflow can persistently save any piece of management data. We use the term "log entry" to refer to such data; log entries are represented as tuples containing a timestamp, a reference to the agent that originated it and the data payload. `StixAgents` coordinate to create 'log overlay' storage in which the most recent or important logs generated at an agent are automatically replicated at a few other agents in the network[5]. By doing so, such log data remains available even when the originating site is temporarily down or unreachable and is particularly useful for troubleshooting activities.

For replication, we designed a new mechanism called *Sprinkle* that is simple and localised to reduce replication-related network traffic. With Sprinkle, a log entry to be replicated is sent by the originating agent to `jmax` of its `j`-hop neighbours. Each receiving agent will store up to `jnum` messages from the originating agent and will itself pass on the data to `kmax` of its `k`-hop neighbours (with $k > j$). Each recipient of such relayed data will store up to `knum` messages from the originating agent (typically `jnum > knum`). The behaviour of the replication mechanism is thus controlled by six configurable parameters. We use `j=1`, `k=2`, `jnum=1000`, `knum=100`, `jmax=2`, `kmax=1`

[5]Note that all logs generated at an agent are always stored locally regardless of whether or not they are replicated. Locally generated logs in the agent are managed separately from remote logs replicated at the agent.

as default values in our implementation[6]. To avoid storing data in poorly connected network regions, each `StixAgent` avoids to select those neighbours for replication that are solely dependent on itself as a bridge to connect to other transmission sites. Replicating agents determine the log entries to drop based on a dropping policy whenever the number of logs for an originating agent exceeds the limit (`jnum` or `knum`). In the current implementation, we use a simple policy to maintain only the most recent logs from an agent. It is also possible to extend the log management at replicating nodes to apply an aggregation operation to dynamically reduce the number of remote logs maintained. The mechanism we use for retrieving management information from the log overlay is described in Section 5.3.3.

The Storage Manager component is a unified database abstraction, which hides the connection and transaction details of reading and writing to the actual database.

5.3.2.5 Implementation

For implementation we used Gumstix Overo Earth [7] embedded boards. These feature a System-on-Chip (SoC) based on a 600MHz ARM Cortex-A8 CPU, 256MB RAM and a microSD slot. We added a 4GB memory card and a daughter board, which provides a FastEthernet interface and an I^2C bus. The bus is used for `StixControl`, a small low-cost embedded board which we designed to allow power monitoring and control, described below.

The whole setup runs with a single 5V supply and consumes less than 2.5W. The overall cost of a single unit so configured is around 200 USD, but we expect the price to drop significantly for large-scale deployments in a way that it could be considered a commodity. Note that one such unit is deployed *per transmission site* to implement the `StixAgent` functionality for all devices managed through that site; this is acceptable when compared to the cost of communication equipment involved in the deployment of a transmission site. We run a modified Linux 2.6.29 distribution and, as Java Virtual Machine, the Sun “J2SE for Embedded 6” [8], which has the advantage of a reduced

[6] These parameters mean that, for a given originating node in the network, up to two of its 1-hop neighbours will store the last 1000 entries, and up to one of its 2-hop neighbour will store the last 100 entries. Such values are provided as a simple suggestion in our evaluation, their sensitivity analysis is provided in Figure 5.13.

[7] Gumstix embedded boards, `http://gumstix.com/`

[8] SUN J2SE for Embedded, `http://java.sun.com/javase/embedded/`

Figure 5.6: *Artist depiction of the* `Stix` *and* `StixControl` *boards.*

memory footprint while being largely compatible with the "desktop" J2SE.

StixControl. In several operational scenarios, including those addressing critical situations, is it important to have complete control over the remote hardware. In order to give the administrator the possibility of controlling remote devices, we developed the `StixControl` board. It is an embedded hardware adapter that sits between a device and its power supply and enables reading the realtime current consumption (in Amps), read realtime voltage (in Volts); it also performs power off, power on and power cycle operations. The board thus provides 'eyes-and-hands' access to remote devices. It is equipped with current/volt sensors, a 12-bit ADC, a relay and an opto-isolated bus which can currently host up to 8 daisy-chained `StixControl` boards. Current and voltage sensors can be read and the relay can be operated using a specific task object. We have designed the printed circuit-boards (PCB) and built a few units with an estimated cost of around 15 USD each. Further details about `StixControl` are given in Appendix E. We envision the `Stix` and `StixControl` boards to look like those pictured in Figure 5.6.

Our implementation of the Storage Manager uses the HSQL database. In practical terms, assuming the average size of a log overlay tuple is 200 bytes, with 3GB of storage space available in each `StixAgent`, one can store in the local part of the log overlay more than twelve million records, which may be sufficient even for mid to long-term storage. For example, in NGI's network, around 20 counters are saved every five minutes for each point-to-multipoint device. Assuming an agent is deployed at a typical '3-sector site', the local database can store data for roughly two years of operations.

5.3.3 StixView: visualizing the distributed network knowledge

`StixView` can be used to generate and view reports, which we call "network perspectives", from the data stored in the log overlay in real time. To enable multiple users to operate concurrently, we modelled perspectives as pages in a wiki engine. A network administrator can create new ones by using a simple syntax that, besides allowing basic text formatting, also includes SQL-like primitives to retrieve and select data from the log overlay. Query results can be rendered in realtime as simple text, as tables or as line, bar or pie charts.

We implemented a two-step rendering engine: in the first step, the perspective page source is compiled into HTML and sent to the web client. Then, Javascript methods scan the document for queries and interrogate the webserver via AJAX calls. A module of the webserver is in charge of interpreting such queries and in turn interrogate the log overlay. Following this approach, more complex and lengthy queries can be displayed as soon as they are resolved without blocking the rendering of the rest of the page while also allowing multiple queries to be resolved in parallel.

The mechanism for retrieval of management information from the log overlay works as follows: the server always tries to first query the `StixAgent` to which the device in question is mapped to. If that agent is unreachable, then the server queries each of the agent's `j`-hop neighbours, stopping when it receives the requested data. If they are also unreachable, then the same procedure is repeated with each of the `k`-hop neighbours of the original target agent. This on-demand retrieval mechanism results in just one query message in the normal case and O(*max. node degree*) query messages in the worst case, where *max. node degree* corresponds to the maximum number of neighbours of a transmission site in the network. As a further optimisation, the data received in response to prior queries can be cached at the server for fast retrieval of the same data later on. We discuss the impact of different failure patterns in Section 5.3.5.

5.3.4 Hardware Abstraction: the Device Manager

Tasks and events communicate with managed devices through the Device Manager (see Figure 5.5), which acts as a hardware abstraction layer. Device drivers implement the interface for each device type and are dynamically loaded as needed depending on the set of devices under consideration. This allows us to flexibly define new drivers to support new devices and to offer a common set of system calls or an API (across all devices) to the workflow engine, while throwing an exception when a system call is not supported by a given device. If the `StixAgent` is running as a software agent on networking equipment, then the Device Manager could just rely on Operating System tools, otherwise via a management protocol (e.g., SNMP).

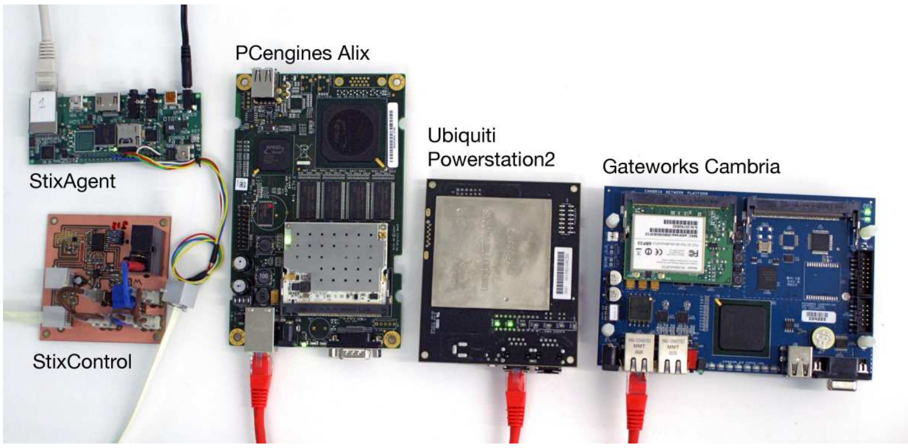
Implementation and Demonstration. Currently, for a hardware based agent, we have implemented drivers for several different devices making use of a handful of protocols (e.g., HTTP, SNMP, SSH).

To demonstrate that `Stix` can support multi-vendor devices, we deployed a small laboratory testbed composed of hardware from different vendors, each equipped with a `StixAgent` and a `StixControl` board. Specifically, we consider different types of wireless devices: PcEngines Alix[9], Ubiquiti Powerstation2[10], and Gateworks Cambria[11] (left to right in Figure 5.7(a)). Each of these products has a different architecture, and we deliberately configured their software so that they necessitate a diverse spectrum of protocols: the Alix board is managed over SNMP mostly using objects in a private MIB tree, the Cambria device is controlled by issuing commands in a shell over an SSH connection and the Powerstation2 board is managed via its web interface.

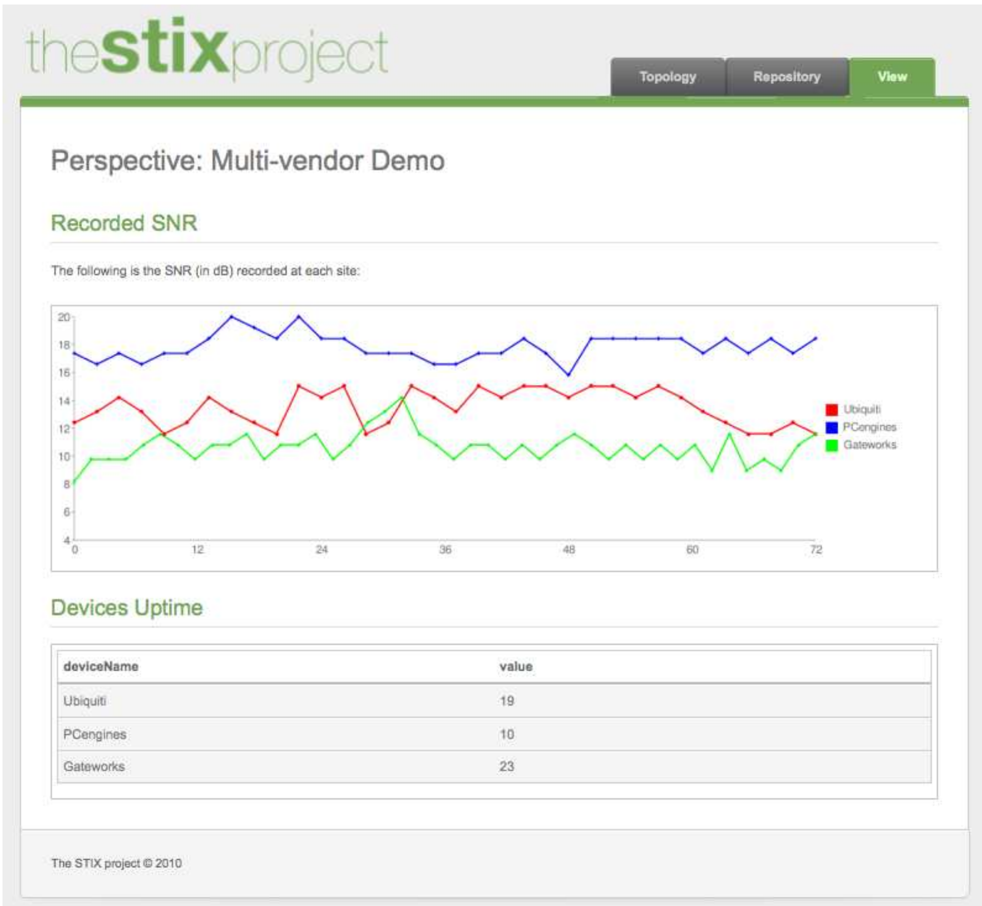
The Device Manager in `Stix` hides this heterogeneity via three different device drivers operating over different management protocols, all presenting a common API. The result is that the administrator can design workflows without worrying about the actual type of devices that will be executing them. For example, a ‘Reboot’ task can be used from the task library and deployed in a workflow; it is up to the device manager to translate the internal `reboot()` method call to the appropriate management procedure. `StixView` is also capable of handling device heterogeneity; for example, in Figure 5.7(b), we can observe a screenshot of values collected in realtime from the

---

[9] `http://www.pcengines.ch`
[10] `http://www.ubnt.com`
[11] `http://www.gateworks.com`



(a) Sample of devices from our hardware testbed



(b) Screenshot of `StixView` showing data acquired from devices of different vendors

Figure 5.7: *Monitoring of heterogenous hardware using* `Stix`*.*

three devices mentioned above.

5.3.5 Discussion

Robustness to Failures. For `Stix`, we are mainly concerned about transmission site failures as they are the locations where `StixAgents` are deployed. We identify three patterns of site failures: *random* (uncorrelated) failures, where a site becomes unreachable but does not affect connectivity to any other site; *spatially-correlated* failures, in which a set of sites in a specific geographical area is disconnected because of an event in that area (e.g., due to a storm); and *cascade* failures that is a variant of the spatially correlated failures resulting from connectivity failure to a set of sites when a site effectively acting as a "bridge" to the rest of the network fails randomly. The likelihood of such failures is dependent on the redundancy in the network topology. The Sprinkle replication mechanism in `Stix` is inherently robust to random failures. It can be configured to be robust against spatially correlated failures (albeit at a higher replication related communication overhead) when j or k are larger than the typical scope of such failures. Careful selection of replication sites in Sprinkle helps improve robustness to cascade failures. In well-provisioned WISP networks such as the one considered in our evaluation (Section 5.4), historical data suggests that site failures are usually rare (as evident from the bottom graph in Figure 5.10), and when they do happen they are uncorrelated (mostly due to grid power outage).

Security. While securing access to management schemes is of primary importance, the current `Stix` system does not provide any specific application-level security mechanism, but it relies on an underlying network-level separation (e.g., different VLANs) between user data traffic and management traffic. Traffic segregation is done routinely on carrier networks to prevent unauthorised access to management interfaces, and enables further advantages such as prioritisation against data traffic. Additionally, we suggest a Public Key Infrastructure (PKI) to be included in the `StixAgent` code base, although it is not part of the current implementation. PKI mechanisms would allow developers of `StixL` "tasks" to sign their code, thus enabling the agents to recognise them as trusted. Mandatory access control schemes could also be included to limit users of the GUI to access specific regions of the network, classes of devices, etc.

To prevent new workflows from causing damage to network operation, a possible solution is to have a simple runtime simulation scheme: a workflow can be simulated and evaluated by adding a "dry run" option in the engine of the `StixAgent` that, while allowing read access to all device configurations, will "trap" any write access and keep modified variable state in Copy-on-Write (CoW) registers. Our initial investigation shows that implementation of this functionality in the current codebase is possible.

5.4 Evaluation

The previous section described our implementation of the `Stix` system and used it to demonstrate the capability for seamless monitoring of multi-vendor devices. In this section, our focus is on evaluating the scalability and efficiency of the `Stix` system, and to demonstrate its utility for BWA network monitoring and control through realistic case studies.

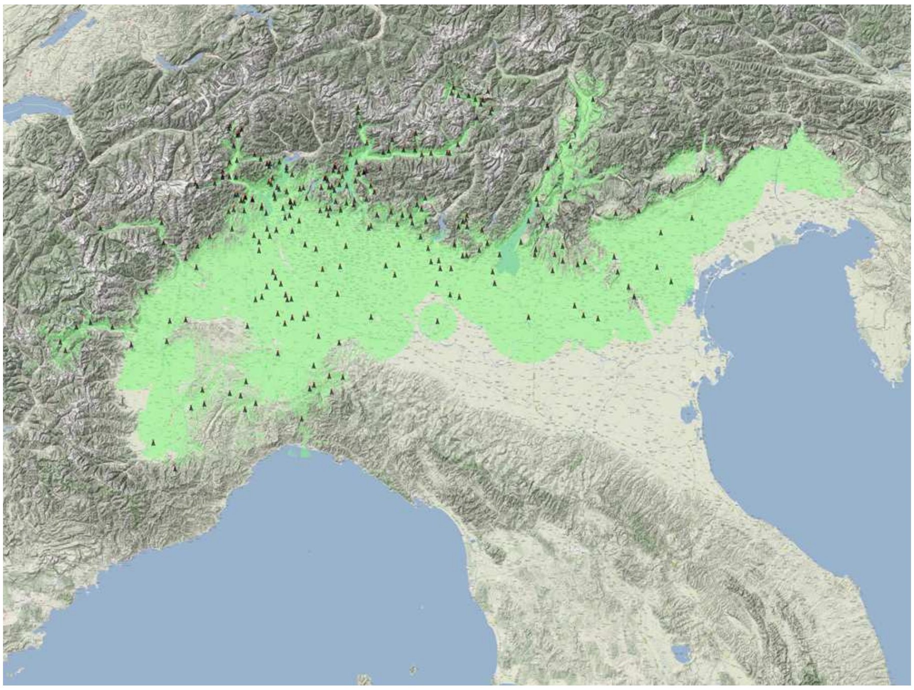
During the initial design process, we had interviewed BWA operators in order to gather a list of the most common network management challenges they face, and drive the design of `Stix` accordingly. Such use cases are gathered in Appendix F.

5.4.1 Scalability

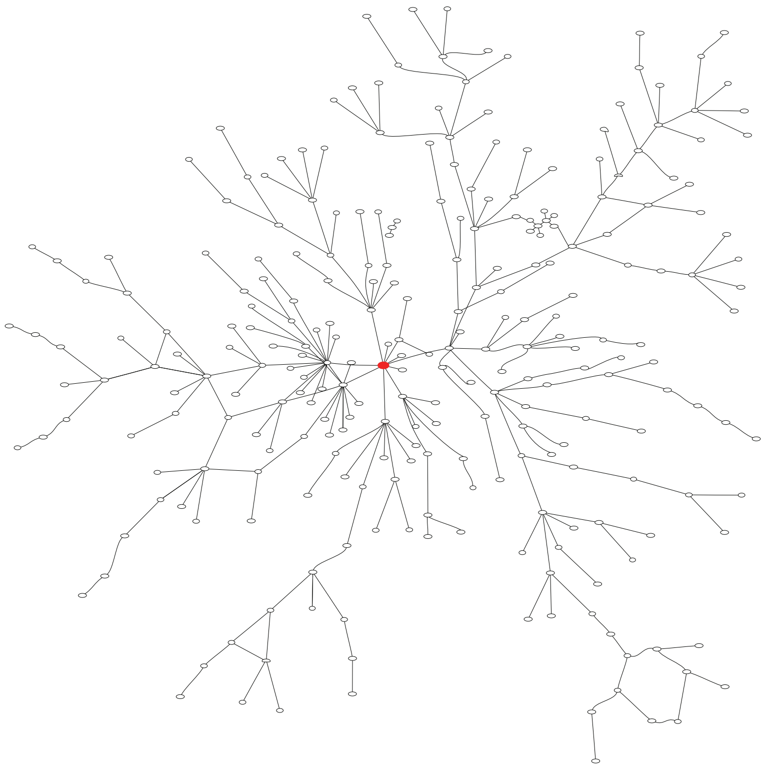
Here our aim is to quantify the communication and storage overhead of the `Stix` system relative to the traditional centralized management approach for BWA network monitoring and control applications. To do this realistically, we leverage real topology and logging data of our partner NGI SpA, which operates one of the largest BWA network deployments in Europe as described in Section 2.1.1. As the time of this evaluation in February 2010, NGI SpA's network had 259 transmission sites; 1,112 BTSs and 51,200 CPEs. Figure 5.8 shows the coverage on the map and the topology showing point-to-point links between transmission sites.

5.4.1.1 Distributed Control: A Firmware Upgrade Scenario

Currently, many features of wireless devices are implemented in software, and service providers rely on firmware upgrades to expand the available services and to distribute bug fixes. This makes firmware upgrade a good candidate network control application to quantify communication overhead with `Stix` relative to the commonly used



(a) Coverage on Map



(b) Topology of Point-to-Point Links

Figure 5.8: *Coverage and topology of our partner NGI SpA's BWA network in northern Italy. Each black dot in (a) corresponds to a transmission site. The red dot in (b) around the middle of the graph is the management server node at the NGI SpA's NOC.*

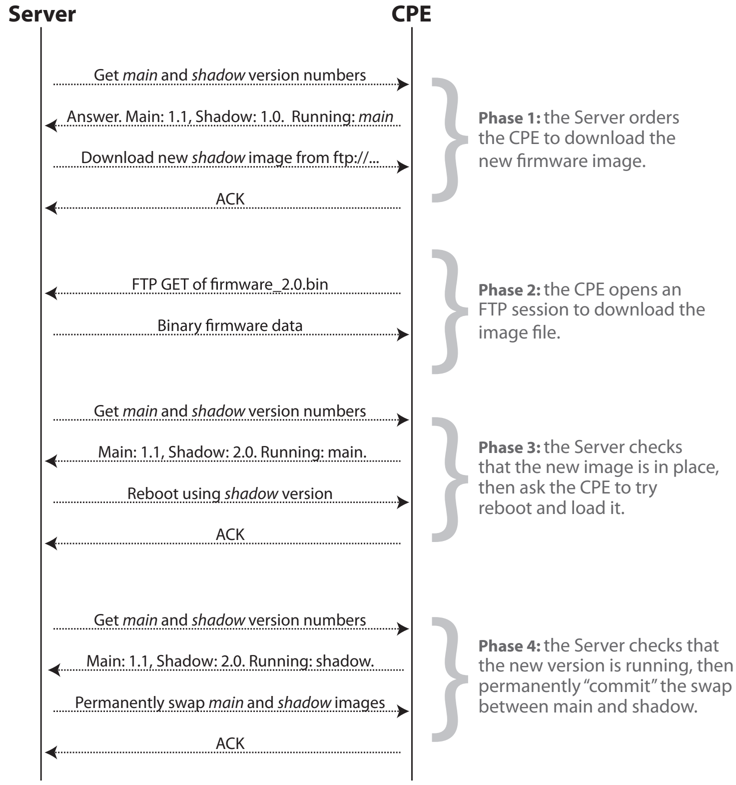


Figure 5.9: *The firmware upgrade dialogue between the centralized management server and a CPE in the NGI SpA network.*

approach.

As CPEs are typically larger in number compared to BTS and PTP devices (also true for the NGI SpA network), we will focus on the case of upgrading the firmware on all CPEs in the network for this scalability assessment. In the NGI SpA network, configuration and management happens over SNMP, and remote firmware upgrade is supported via a commonly used "dual firmware" technique. This technique requires equipping the device with enough persistent memory (i.e., onboard flash) to concurrently store two firmware images. The BIOS then controls which of the two binaries starts at boot based on a configuration parameter. Figure 5.9 presents a sequence graph illustrating the communication between the centralized management server and a CPE: the dialogue is composed of four phases that begin with the server asking the CPE to download a firmware image, which is then downloaded by the device being upgraded over a separate session. Then the server asks the CPE to try rebooting using the new

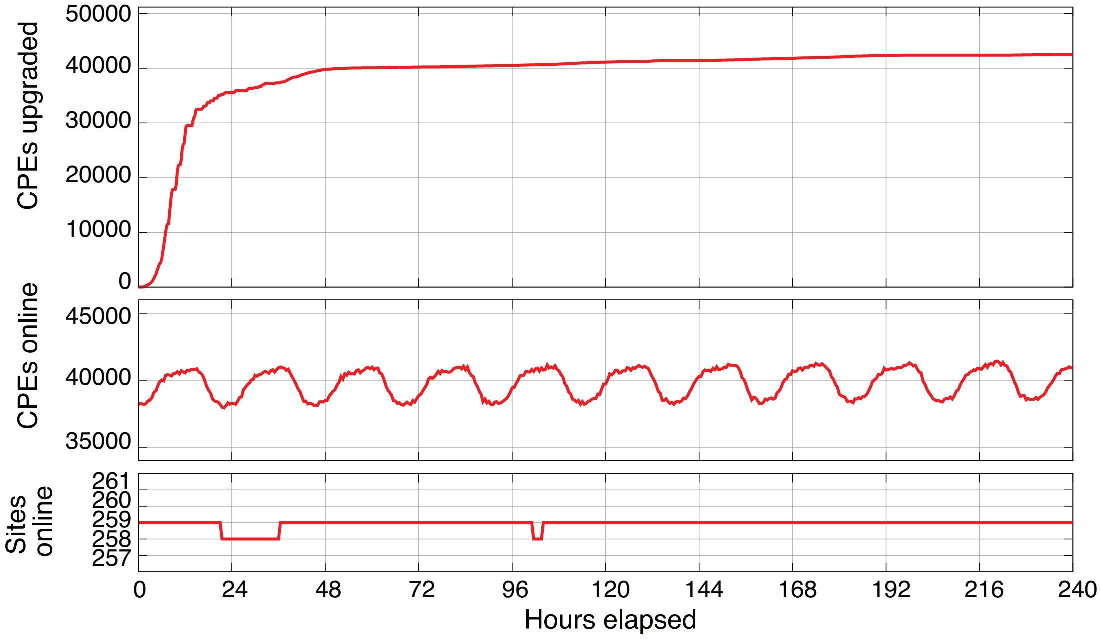


Figure 5.10: *Number of CPEs successfully upgraded in the first ten days (from the time the upgrade was initiated by the server) compared with the number of CPEs and sites online during the same period. Overall, there are 259 transmission sites and 51,200 CPEs in the network. Number of CPEs online shows a diurnal pattern, suggesting that some customers tend to switch off their CPEs at night. Most of the time, almost all transmission sites are reachable.*

image. If the process succeeds, the change is made permanent, i.e., committed.

Clearly, the success of such a centralized approach depends on the existence of an end-to-end path between the server and the remote device (CPE in this case). If the CPE is turned off or if there is some kind of communication failure, the process may need to be repeated multiple times. It should be noted that CPEs in the NGI SpA network are outdoor devices installed by qualified personnel on the customer's rooftop, so they are more likely to be turned on at all times compared to indoor or mobile CPEs. Nevertheless, it may still take time to upgrade each CPE in the network.

For a recent firmware upgrade performed by NGI SpA in Jan 2010, Figure 5.10 shows, from top to bottom, the cumulative number of devices upgraded for the first ten days of the process, the volume of CPEs online in the network and unreachable

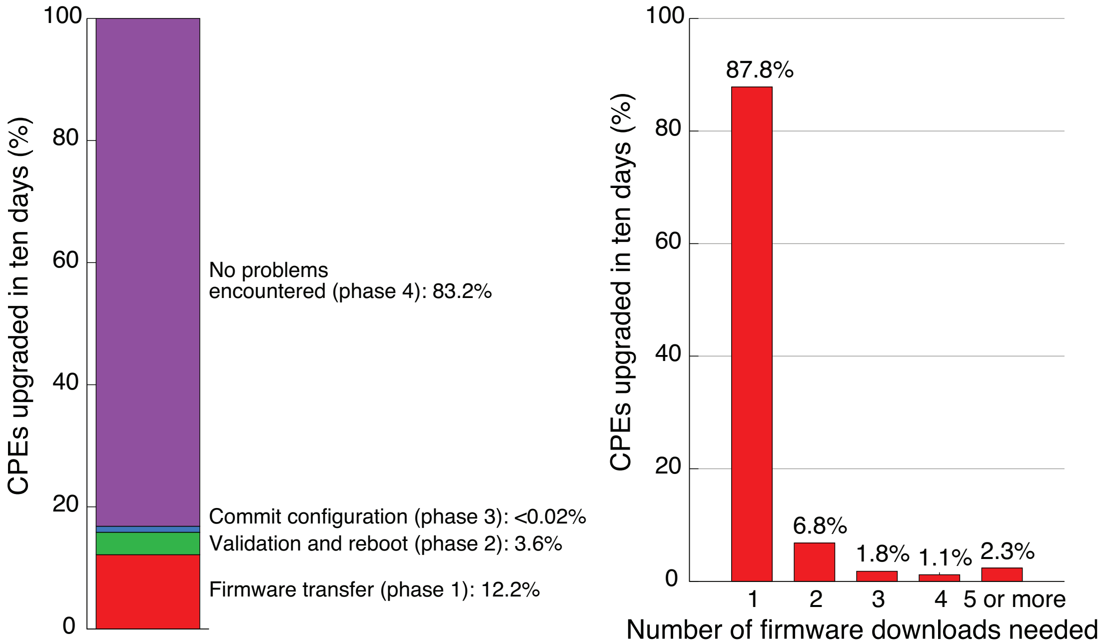


Figure 5.11: *Breakdown of causes behind the failure of firmware upgrade for online CPEs. The graph on the left shows that 16.8% of the online CPEs experience firmware upgrade failure. For 12.2% of the online CPEs, the problem is due to the failure of the firmware transfer; the graph on the right indicates the number of transfer attempts required to resolve this problem for all CPEs.*

transmission sites at a given time. After upgrades are swiftly performed during day one, the process slows for two main reasons. The first is that the number of CPEs online and still to be upgraded runs out, which causes the centralized server to continuously "hunt" for them once they appear on the network. The second is that a remote firmware upgrade can fail even when a CPE is online for several reasons, for which we give a break down in Figure 5.11, slowing the process even further.

Using `Stix`, the network administrator can design a simple workflow that runs on the `StixAgent` at each transmission site *per BTS* such that it is triggered by the association of a new CPE; at that point, it checks whether an upgrade is needed and, accordingly, performs it. Such distributed control is beneficial in three ways. The first is that the BTSs have knowledge about the CPEs associated to them, thus they know exactly when to trigger the upgrade operation without needing the central remote

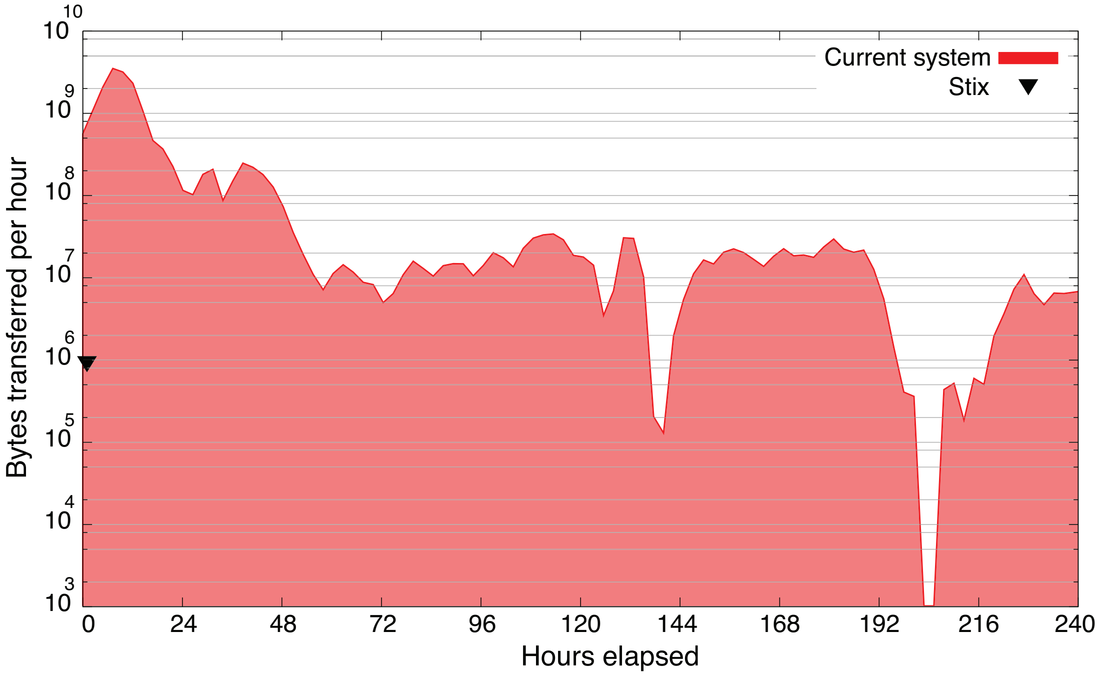


Figure 5.12: *Comparison of estimated network traffic generated for the CPE firmware upgrade application between the current approach used in the NGI SpA network and the* `Stix` *approach. The shaded area corresponds to a total of 40GB.*

server to continuously track for new CPE associations. The second is that the firmware upgrade operation becomes entirely local, making all data-intensive communication (firmware transfer in this case) mostly between the CPE and its BTS; this may greatly reduce the most significant cause of failure we recorded for online CPEs in Figure 5.11, i.e., "Firmware transfer: 12.2%". The third is a remarkable reduction in network traffic caused by the firmware upgrade. We estimate this by calculating the total network traffic outgoing from the central management server based on the fact that the firmware image for the CPEs is 944KB in size. This is shown by the shaded area in Figure 5.12, which amounts to a total of 40GB in traffic volume. To obtain this plot, we have assumed that firmware transfer for online CPEs succeeds in the first attempt as we do not have access to the detailed time series of failures, so in reality it is a conservative estimate. Based on the observed failure numbers we obtained (Figure 5.11), we can estimate that more than 96,000 firmware transfers had been done in the first 10 days of the process. This roughly corresponds to 94GB of traffic volume, which is more than twice compared to the case with the simplifying assumption in Figure 5.12. Using `Stix` instead, only a single transfer of the image to each `StixAgent` is sufficient to delegate the upgrade operation to the individual `StixAgents` at different transmission sites. This results in the total volume of data exiting the central server with `Stix` to be around 1MB, significantly lower than the currently used approach.

5.4.1.2 Distributed Monitoring using the Stix Log Overlay

Here we study Sprinkle's behaviour by simulating it on NGI SpA's network topology, shown in Figure 5.8(b), to see the impact of the parameters `j`, `k`, `jnum`, `knum`, `jmax` and `kmax`. We use (`h`, `hnum`, `hmax`) as generic variables that could correspond to either (`j`, `jnum`, `jmax`) or (`k`, `knum`, `kmax`). Figure 5.13 shows the average number of log entries stored at a replicating node on the log overlay as a consequence of a *continuous* stream of logs from an originating agent, calculated on the network of NGI SpA for different settings of `h`, `hnum` and `hmax`. As seen from Figure 5.13, the `hmax` parameter can be used to bound the replication overhead in terms of storage for a given set of `h` and `hnum` values. Note that the `h` and `hnum` are parameters used for controlling how far and how much to spread the data on the log overlay around the source agent.

Let us consider an example with the following parameter settings to show how the result in Figure 5.13 can be used to estimate the storage overhead on the log overlay due to the Sprinkle mechanism for the NGI SpA network:

$$\begin{aligned} \mathtt{j} = 1; \mathtt{jnum} = 1000; \mathtt{jmax} = 2 \\ \mathtt{k} = 2; \mathtt{knum} = 100; \mathtt{kmax} = 1 \end{aligned}$$

This means that, given an originating agent, up to two of its 1-hop neighbours will keep its last 1,000 logs and that up to one of its 2-hop neighbours will keep its last 100 logs. The average number of logs replicated on the log overlay for the agent can be read from the graph as the sum of the $\mathtt{hnum} = 1000$ curve at coordinates $(\mathtt{h} = 1; \mathtt{hmax} = 2)$ with the value of the $\mathtt{hnum} = 100$ curve at $(\mathtt{h} = 2; \mathtt{hmax} = 1)$, which equals 1545. This essentially means that 1000 (`jnum`[12]) logs from the originating agent result in 1545

[12]Note that `jnum` ($>$ `knum`) is the maximum number of unique logs from the originating agent that can be replicated on the log overlay at a given time.

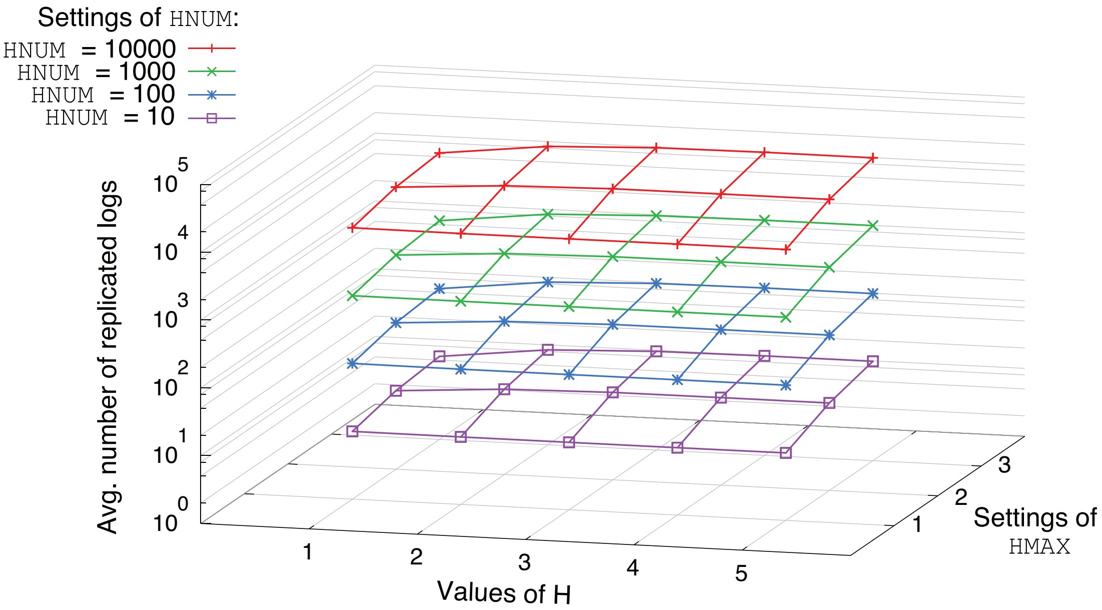


Figure 5.13: *Average number of log entries stored at a replicating node on the `Stix` log overlay using the Sprinkle mechanism for the NGI SpA network.*

logs (including duplicates) replicated on the log overlay. In general, `jnum` logs from an agent can produce up to $(\texttt{jnum} * \texttt{jmax} + \texttt{knum} * \texttt{kmax})$ logs on the overlay using Sprinkle. The overhead matches the upper bound for a network topology with node degree greater than or equal to `jmax`. Multiplying by the average size of a log gives the storage overhead due to replication.

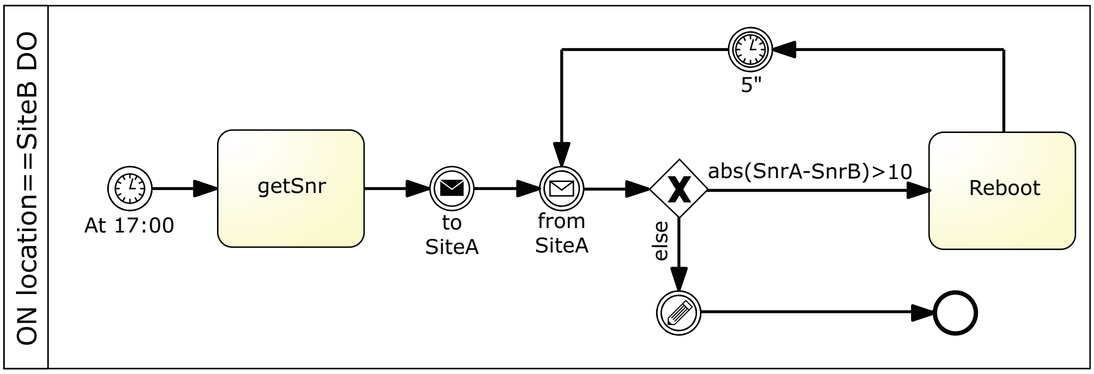
Now turning our attention to the replication overhead in terms of network bytes: to replicate a log entry, the `Stix` log overlay mechanism has a communication overhead which is upper bounded by the product of $(\texttt{j} * \texttt{jmax} + \texttt{k} * \texttt{kmax})$ and the size of the log entry. This is however reasonable given that `j`, `k`, `jmax` and `kmax` are small constant numbers (1 or 2 in our implementation). In contrast to the above, the communication overhead of the traditional centralized management approach is a function of the path length between the managed device and management server, which can be long (as high as 11 hops in the NGI SpA network).

Retrieval of a log from the log overlay in general depends on the failure probability of transmission sites, Sprinkle parameters and the dropping policy used. Note that replicating nodes are queried only when the original source of the log is down or unreachable. Also, prioritising certain types of data during replication (e.g., billing related data) would improve its availability in the event of failures. In Section 5.3.5, we have discussed Sprinkle's robustness to various failure patterns in general terms. A detailed analysis of the probability of retrieving a log in the presence of failures and limited available storage at replicating nodes requires further investigation based on the specific environment.

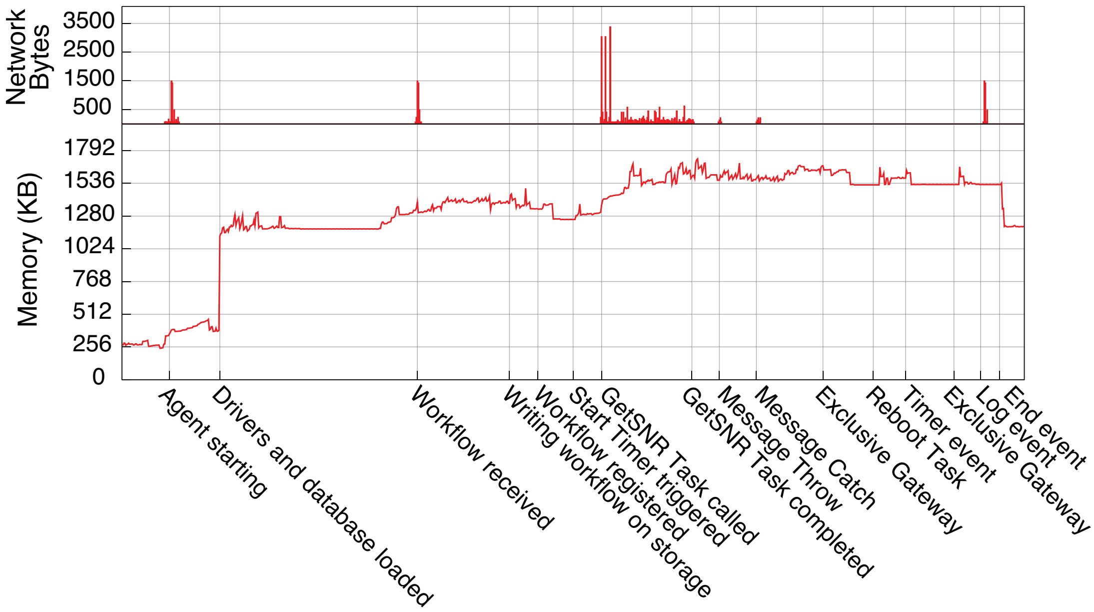
5.4.2 Efficiency: StixAgent Resource Consumption Profiling

We now evaluate the efficiency of the `StixAgent` implementation in terms of memory and CPU utilisation by stress testing it. For this, we created a special synthetic workflow shown in Figure 5.14(a) that includes most of the `StixL` language elements: it is started by a timer, then it invokes a Task object that gets the Signal to Noise Ratio (SNR) value from a particular wireless interface on the device. That SNR value is sent to a workflow running on another agent, which then answers back with the remote SNR value. An Exclusive Gateway is used to determine whether the difference between the two SNR values is above a threshold. If so, the "Reboot" Task is called, resulting in the local device getting restarted, then a timer waits for the device to come back online. Otherwise, a message is saved in the log overlay and the workflow terminates.

In Figure 5.14(b), we show a record of the traffic generated on the network and the memory allocated on the system heap inside the Java Virtual Machine (JVM) at intervals of 10ms. We can observe that the baseline memory footprint of the `StixAgent` implementation is around 1200KB without any workflows registered or executing. When the workflow is received from the server and registered, the memory footprint goes up by a modest amount of about 100KB. When it starts executing, this workflow consumes less than 500KB on top of the memory footprint at the time of registration. Although our Workflow Engine implementation within the `StixAgent` has not yet been optimised for memory savings, we believe that the memory consumption of our implementation is quite acceptable (compare with the total RAM size of 256MB). This experiment also allows us to estimate the number of concurrent workflows (assuming the one used is a typical workflow) that can run on a `StixAgent` before the OS has to resort to memory paging. The CPU utilisation throughout the above experiment remained around 5%, which is again a positive result.



(a)



(b)

Figure 5.14: *(a) Synthetic workflow used in the resource consumption profiling of the* `Stix` *implementation; and (b) the resulting network and memory overhead trace over time.*

5.4.3 Case Studies

We now present two realistic case studies to demonstrate the usefulness of the `Stix` system for BWA network management. For these case studies, we created a small indoor wireless testbed network as depicted in Figure 5.15. We use commodity WiFi hardware for the testbed network. Specifically, each of the PTP links are realised using

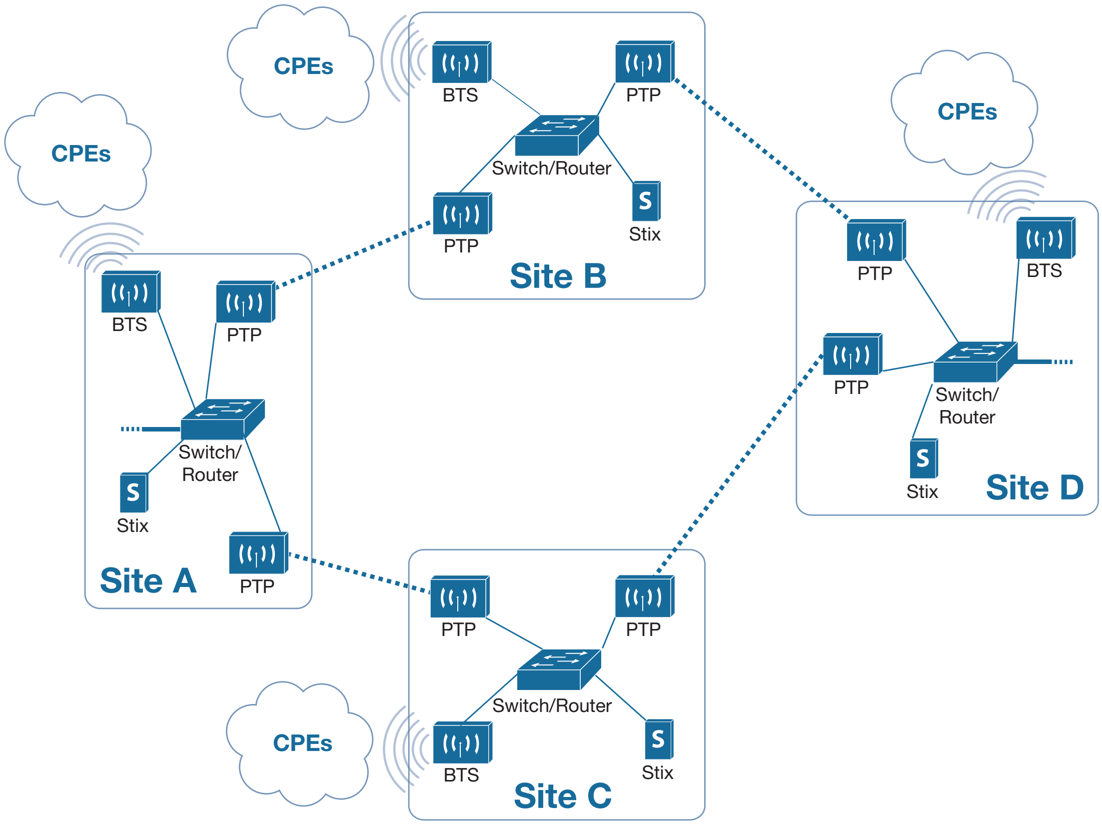


Figure 5.15: *Testbed network used for the case studies.*

a different (unused) channel in the 5GHz band, whereas BTS-CPE communication is over 2.4GHz band as in a typical infrastructure wireless LAN. These configurations together allow us to create a BWA network in an indoor setting. A `StixAgent` based on our implementation described in Section 5.3 is deployed at each "Site", representing a transmission site in the real world. Routing in this testbed network is performed using OSPF, which also handles link failures.

5.4.3.1 Seamless Device Reconfiguration

It is increasingly common for WISPs to have "all-wireless" networks in which both the access tier and the backhauling tier are wireless. In these cases, network maintenance operations such as upgrades and reconfigurations are disruptive operation for customer traffic. As an example, suppose that a major upgrade is to be performed at Site B on our small testbed. Such an activity would have impact on ongoing data traffic

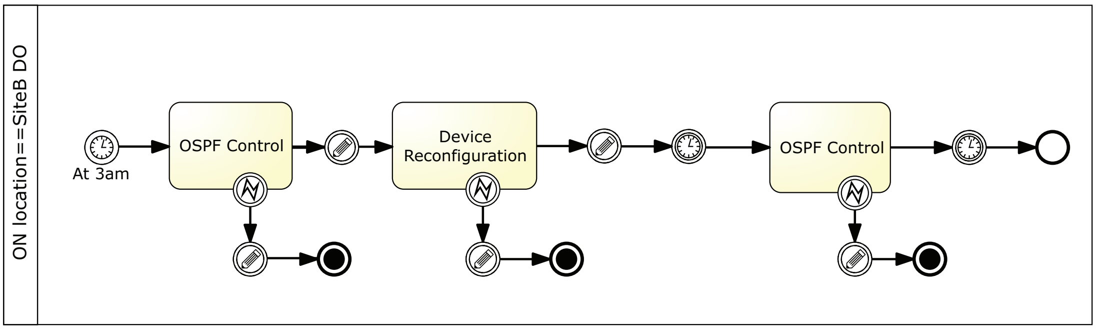


Figure 5.16: *The device reconfiguration workflow.*

for customers connected to the local base station, and potentially for other areas of the network. However, the administrator can design a simple workflow like the one in Figure 5.16 to trigger a series of tasks that reconfigures the network routing prior to the upgrade and finally restores the previous routing state. We implemented this by providing an "OSPF Control" object in the `Stix` Task library; it can modify the administrative cost of any local network link based on an input parameter.

The graph in Figure 5.17 is a plot of the RTT measured from Site A to Site D of our testbed. Prior to the upgrade, all the traffic is routed via Site B, which offers the best path. When the workflow is triggered, the `StixAgent` running at Site B automatically announces the unavailability of the local links, then waits for the routing modification to be complete, then performs the device reconfiguration and finally announces the availability of the routes. During the time when Site B is not available, traffic is routed via Site C which, despite being a sub-optimal path initially, ensures data packets continue to get routed between Site A and Site D. The step-wise increase in the RTT is because of changes to forward and reverse paths at different times.

5.4.3.2 Adaptive Spectrum Management

The amount of spectrum available for BWA networks is limited, especially in unlicensed bands or those involving nominal license cost (e.g., 5.8GHz band in the UK). This makes adaptive spectrum management a crucial network management activity from a performance standpoint. A simple approach for balancing the number of users (CPEs) associated with a set of base stations (BTSs) has been proposed in (Moscibroda et al., 2008) for 802.11 networks. In this technique, a central allocation server (which

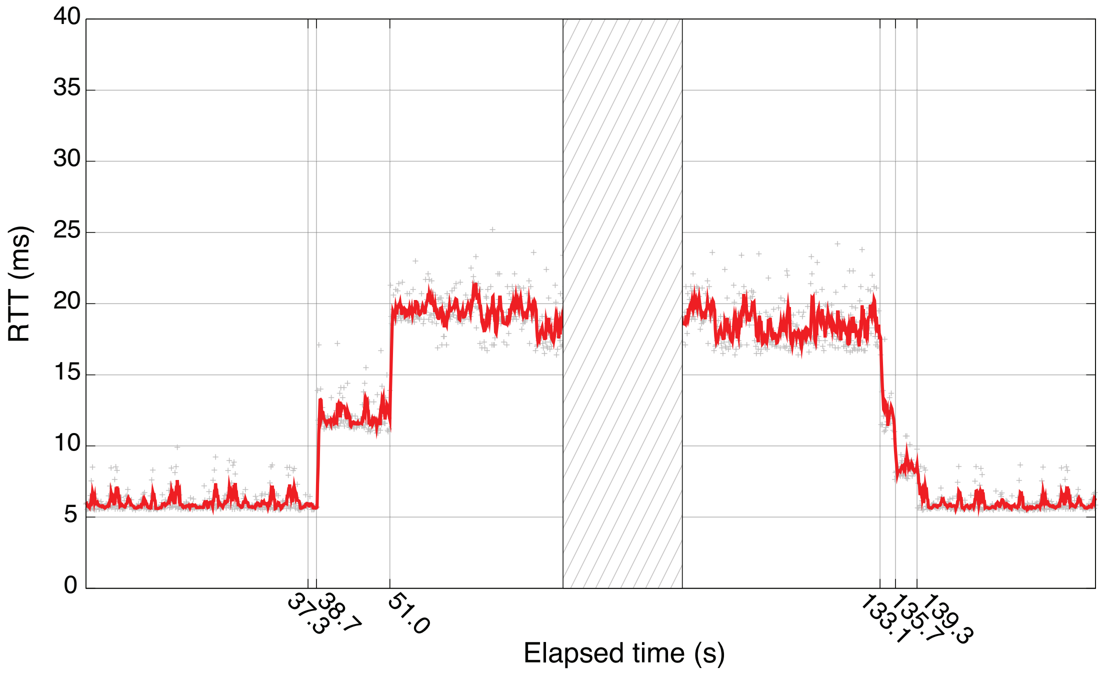


Figure 5.17: *The measured RTT over the network during the execution of the device reconfiguration workflow.*

we call *CAS*) determines the spectrum allocation between BTSs based on the number of CPEs at each site, and communicates with each BTS to dynamically adjust the channel width and center frequency. The advantages over static channel width allocation are an increased spectrum utilisation and better per-CPE and per-site fairness.

We implemented this mechanism in `Stix` by designing the two workflows shown in Figure 5.18, which run on each `StixAgent` for the co-located BTS. The first workflow on the left is triggered by a Condition Start Event every time a new CPE associates or disconnects and causes a message to be sent to the CAS. On the other hand, the second workflow on the right is triggered by a spectrum allocation message sent by the CAS, modifies the center frequency and channel width using the specified Task. In a real world scenario, CAS could be implemented in a server that is external to the network. In our case, it is instead implemented at one of the agents in the network using a workflow that is triggered by messages from the workflow shown on the left in Figure 5.18.

We deployed the workflows on our testbed network, where a CPE moves from

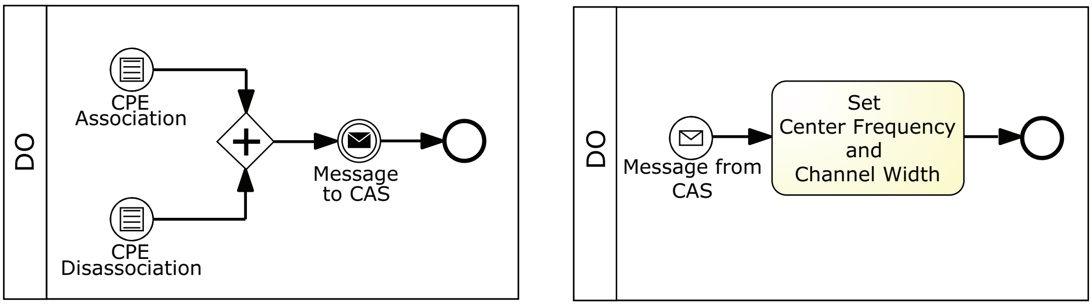


Figure 5.18: *Workflows used for adaptive spectrum management case study.*

being associated with one BTS to another BTS while receiving a 5Mbps constant bitrate UDP stream from an external server. In this case, all the BTSs operate at a fixed 802.11g datarate of 18Mbps and, initially, the BTS at Site B is allocated a 10MHz wide channel and has only one CPE connected, which is able to receive the stream with 0% packet loss. However, when a second CPE associates to the same BTS, saturation occurs and each of the two streams drops to around 3.5Mbps. As `Stix` reconfigures the network, allocating 20MHz to the BTS at Site B, it is able to deliver a full 5Mbps again to each of the two associated CPEs (see Figure 5.19).

5.5 Demonstration of Stix on the Tegola Testbed

The `Stix` system was demonstrated on several occasions. The latest, which featured the most complete software implementation, was presented at the WiNTECH workshop of ACM Mobicom[13] in September 2010. Our Tegola network, located in the Western Highlands of Scotland and described in detail in Chapter 3, was used to show the usefulness of `Stix` in rural networks, demonstrating a real-world use case that involves device self-configuration to adapt to link quality fluctuations.

As shown in Figure 3.1, the coastal communities reached by the Tegola network connect to the Internet via long-distance links over tidal sea water at low altitude. Tidal fading is the main source of link quality degradation for these links, and it can severely affect network connectivity. Such correlation is clearly shown by the graph

[13] `Stix` was awarded the first prize of the WINCOOL (aka the Next Big Thing in Wireless) competition.

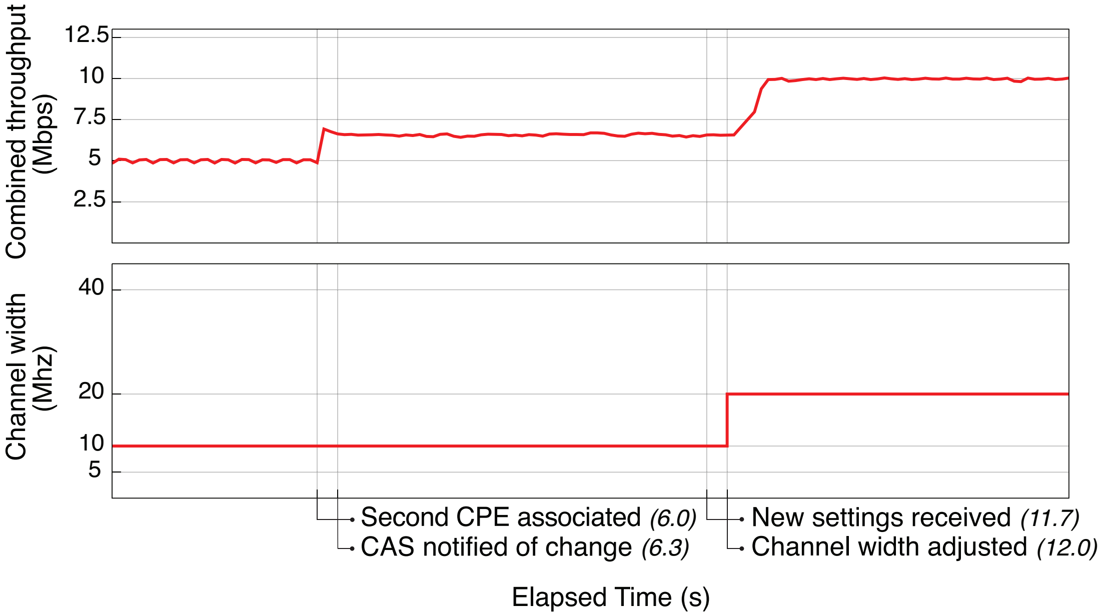
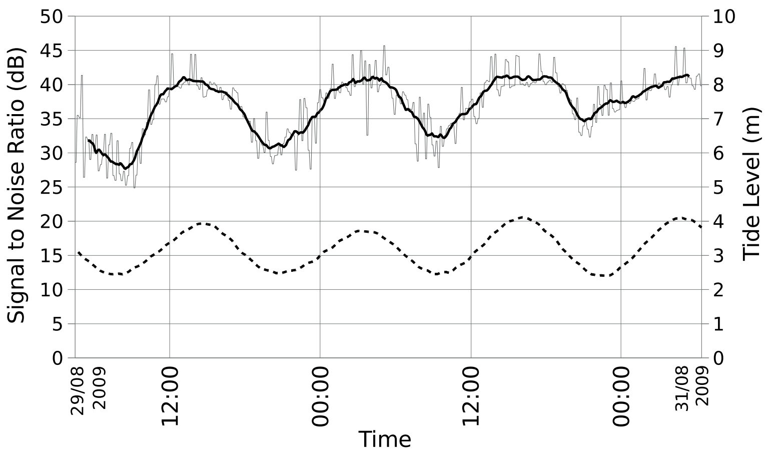


Figure 5.19: *Channel width and aggregate throughput at Site B over time in the adaptive spectrum management case study.*

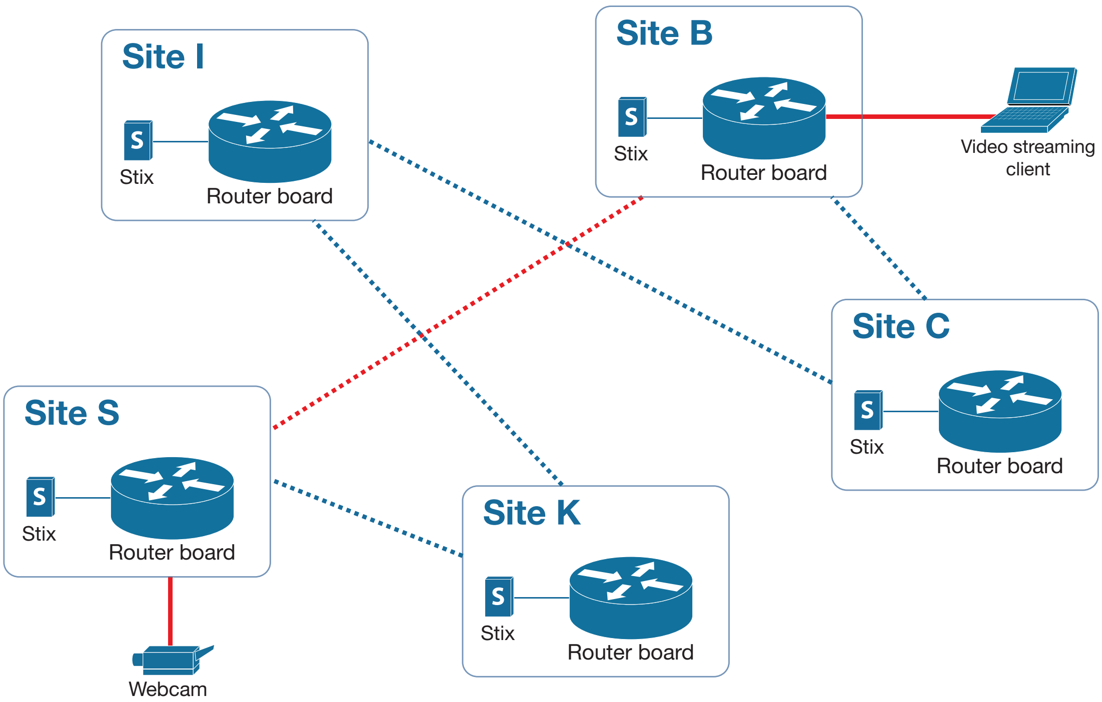
in Figure 5.20(a), from (Macmillan et al., 2010), which shows the signal strength data collected on the link between Site I and Site C: the above curve corresponds to the measured SNR and its two-hour average, and the dashed line shows the tide height data recorded at a nearby location for the same period.

Each transmission mast of the Tegola network has been equipped with a `Stix` dongle (see Figure 5.21). By designing a simple `StixL` workflow, we show how `Stix` can react to cyclic variation in the signal strength: when the signal strength of a long-distance link drops below a given threshold, a workflow is triggered to adjust the OSPF link cost, which in turns causes the the routing layer to recalculate the topology and re-route traffic via unaffected links.

In our demonstration, we configured various Gateworks Avila router boards (the same used on the Tegola network) to setup a tabletop testbed that resembles the topology of our Highlands network, as in Figure 5.20(b). We then streamed a video from a source connected to Site S to a client connected to Site B: when all links have equal OSPF cost the stream is sent over the shortest path, displayed in red in the picture. We then simulated the tidal path loss by reducing the antenna gain on the S-B link: when we do so, the link scales down to less efficient 802.11 modulations, which pro-



(a)



(b)

Figure 5.20: *Using* `Stix` *to mitigate tidal fading on long-distance links over sea water: (a) the link quality of a long distance links in the Tegola network varies periodically as an effect of tides fluctuation. Source: Macmillan et al. (2010). (b) Simplified layout of the Tegola network (compare with Figure 3.1), the red path is the default route for the video stream from Site S to Site B.*

Figure 5.21: *Four of the `Stix` network management dongles deployed at the Tegola network backhaul masts.*

vide reduced bitrate causing the video quality to degrade or drop completely. However, if `Stix` is enabled, a `StixL` workflow is automatically triggered when the SNR level of a link goes below some threshold. The workflow varies the OSPF cost of the link to force the routing subsystem to recalculate the new shortest path between S and B, which becomes S-K-I-C-B. In this simple example, `Stix` is used to pro-actively handle variations in the network environment: as the adjustments are made before the links become seriously degraded, the video stream quality remain good at all times[14].

In our demonstration, we have also shown how the `StixView` interface can be used to generate realtime graphs of links signal strength which reveal the tidal fluctuation patterns.

5.5.1 Community use cases

In Section 5.4.1, the scalability of `Stix` was demonstrated on NGI's network by implementing a management task which is routinely performed on that network: CPE firmware upgrades. However, scalability is only one of the benefit of `Stix`, and much smaller networks (such as rural and community deployments) can also take advantage of the distributed approach of our system.

In Appendix F, we present seven management use cases which we took into con-

[14]An alternative approach of using `Stix` to mitigate tidal fading is to adopt the slow frequency hopping technique presented in Section 7.1 and described in detail in (Macmillan et al., 2010). In this case, a workflow running on `StixAgent` builds a two-ray propagation model of each link and slowly varies the transmission frequency to reduce interference.

sideration when designing the Stix system. We discussed the implementation of two of such scenarios in Section 5.4.3 ("Seamless Device Reconfiguration" and "Adaptive Spectrum Management") and demonstrated the use of Stix to address an actual issue of our Tegola network (dynamic routing to address environmental changes). While operating the Tegola network, we indeed had to face some of these scenarios, such as:

- *Power monitoring*, to troubleshoot problems with the two self-powered towers (sites *I* and *B* in Figure 3.1).
- *Remote hardware control*, to perform emergency reboots of remote devices that crashed and stop responding to management commands.
- *Automatic channel monitoring*, to avoid radio interference on radio sectors, which may happen at any time as Tegola operates on license-free spectrum.
- *Historical performances query*, to understand network usage over time, identify bottlenecks and enforce Tegola's acceptable usage policies (i.e., avoid excessive filesharing).

Despite a Stix agent has been deployed at each transmission site of the Tegola network, so far I have been the only author of the management workflows inserted in the system. I suggest, as future work in the evaluation of the Stix system, to train the local network users to use the StixGUI and to design management workflows by themselves. Such activity would allow to measure human-computer interaction (HCI) metrics and, ulimately, to validate Stix as a network management system for community deployments.

5.6 Summary

Given the size and complexity of emerging and future BWA deployments, the design of *usable* management systems is difficult. We have taken a pragmatic approach to address this challenge.

We wanted our system to benefit three categories of users: commercial operators of large-scale BWA networks, non-technical personnel in community deployments and researchers working on self-management techniques. Our contributions can be summarised as follows: (a) Stix introduces a visual paradigm to ease the definition of management processes as workflows; (b) it provides a distributed cooperative agent architecture to run such workflows and store results in the network, thereby reducing management traffic and improving scalability, a fact confirmed by our evaluations; (c) it abstracts hardware heterogeneity and enables “eyes and hands” control to devices being managed. Through our case studies, we have also shown that `Stix` can be useful in easing the implementation of self-management mechanisms. Possible directions for future enhancement to the `Stix` software are given in Chapter 8.

Chapter 6

BSense: A Flexible System for Broadband Mapping

6.1 Introduction

The previous two chapters proposed new approaches for the planning and management of rural BWA networks. This chapter now turns to the complementary problem of evaluating the quality of finding broadband availability in a given geographic region and, where available, evaluating the quality of broadband connections in use.

There exists a digital divide in terms of broadband Internet access in most countries across the world. The underlying reasons can vary from location (remoteness, terrain) and population density to lack of infrastructure, deployment costs and socio-economic factors. The 2008 global financial crisis has been a boon for broadband related public policy and investment as evident from the attention it drew in the European Economic Recovery Plan published in Nov 2008 and the American Recovery and Reinvestment Act of 2009.

Recognising the importance of broadband communications infrastructure as a critical element of the economy in today's information society, governments across the world have made it a key priority, issuing policy statements and setting targets both in terms of broadband reach and speeds. For example, the US national broadband plan aims to provide at least 100Mbps download and 50Mbps upload to at least 100 million homes by 2020[1]; other regions and countries have come out with equally ambitious

[1] See: `http://www.broadband.gov/plan/`

targets[2]. Issues of public funding and incentives for private investment are also being debated in relation to these plans. Some countries like Finland have even gone so far as to make broadband access a legal right for every citizen.

Against this backdrop, to track progress towards achieving the set targets and accountability for the use of taxpayers' money, quantifying the state of broadband in a region over time is vital. Such quantification, referred to as *broadband mapping* or *broadband census*, fundamentally concerns *broadband coverage* to understand the extent to which universal access is not met; this goal can be restated as identifying *"notspots"*, i.e., areas not serviced by even one broadband access technology. For those areas that *are* covered, assessing *broadband quality* is the main concern. Quality is measured using a set of metrics such as download/upload speeds, latency, jitter and packet loss rate. Several technology-specific and network provisioning factors affect quality in practice (e.g., length of the local loop, number of concurrent users, contention ratio, backhaul capacity). Choice and cost for subscribing to a broadband connection are additional aspects that are of interest to regulators for assessing the broadband market in a region: in order to determine the amount of choice that a consumer has, one needs to find out the number of access technologies and ISPs available at the consumer's location. Greater choice usually also implies lower cost (per Mbps) for the consumer. Moreover, choice and cost both tend to depend on the coverage and quality aspects — poor broadband coverage or quality in a region correlates well with lack of choice and/or higher costs for consumers in that region. Finally, we note that all of these aspects (coverage, quality, choice and cost) vary with time.

Not surprisingly, several broadband mapping efforts are currently underway, the largest among them being the $350 million US National Broadband Map (NBM)[3] initiative from the National Telecommunications and Information Administration (NTIA) in collaboration with the Federal Communications Commission (FCC). NBM, first published in Feb 2011, is primarily based on maximum advertised speed data collected from Internet Service Providers (ISPs) and aggregated at the state level. Germany also has a similar mapping initiative[4], and the European Commission is in the process of working out a Europe-wide broadband mapping program. Other mapping projects are

[2] Such as the European Union's plan to boost broadband by 2020 to provide 30Mbps to each household: `http://www.bbc.co.uk/news/10128190`

[3] US National Broadband Map, `http://www.broadband.gov`

[4] Broadband Map for Germany, `http://www.breitbandatlas.de`

measurement based (OFCOM, 2011; Dischinger et al., 2007) and are usually carried out in cooperation with consumers: they take different approaches as discussed in the next section.

Existing mapping approaches have certain limitations: the approaches based on ISP-supplied data fail to capture the discrepancy between expected and actual broadband quality experienced by consumers, whereas measurement-based approaches are either expensive, based on proprietary measurement techniques or subject to measurement bias.

In this chapter, we present a flexible mapping framework called `BSense` for fixed wired/wireless broadband networks. Our approach combines the best aspects of existing approaches, using data from ISPs for coverage analysis and relying on consumer side measurements for quality assessment. Our framework also incorporates a flexible broadband quality index based on multi-attribute utility theory (Dyer, 2005) to ease assessment of broadband quality in a region that is dependent on a myriad of underlying (technical and non-technical) characteristics. We implement the `BSense` framework using open-source software components. We extensively evaluate the `BSense` system, especially focusing on demonstrating its use for assessing broadband coverage and quality in a given geographic region.

The remainder of this chapter is structured as follows. In the next section, we describe in detail the proposed `BSense` system and measurement methodology, including the proposed broadband quality index (Section 6.2.3) In Section 6.3 we evaluate it together with two real-world usage examples. In Section 6.5, we discuss issues for future work and other relevant issues. We summarise this chapter in Section 6.6.

6.2 The BSense broadband mapping framework

6.2.1 The Broadband Census Lifecycle

Broadband mapping is needed by all stakeholders — consumers, ISPs, policy makers and regulators. As such, an effective broadband mapping framework should engage and involve all of them. Moreover, broadband coverage and quality varies over time with newer deployments, network upgrades and the emergence of new access technologies, so we believe that cooperation among different stakeholders is needed for a

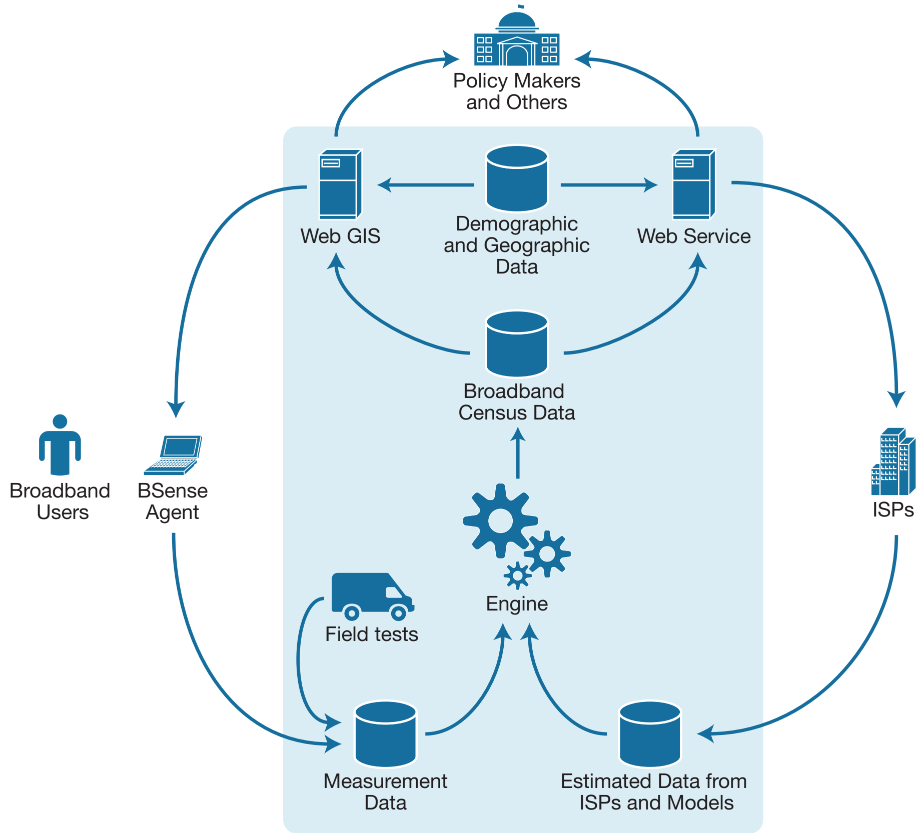


Figure 6.1: *BSense broadband mapping framework.*

complete, reliable and an evolving broadband map. With this in mind, our proposed BSense framework, illustrated in Figure 6.1, includes incentives for everyone to continually contribute to the broadband mapping exercise, as follows.

Consumers would benefit from comparing their observed broadband quality over time with the advertised service of their ISP. Equally, they would be interested in knowing the broadband coverage and quality in their neighborhood from different ISPs. With BSense, a consumer can obtain such information from the mapping system in exchange for installing and running a broadband connection quality measurement application on a home computer[5]. Results of periodic tests continually feed into the measurement database.

ISPs normally pay for market research information to determine the areas in which

[5]If a user has more than one computer behind the same Internet connection, only a single BSense client is to be installed, to prevent tests from different computers interfering with each other and to limit test related overhead on the user's broadband connection.

to upgrade their networks, improving their service quality with the view of maintaining and expanding their customer base. They could obtain such information from the mapping system in return for proactively updating it with data about their "estimated" coverage and speeds at different locations along with associated information on their various service offerings (packages). This is facilitated in BSense via a set of webservice APIs, which will be described in the next section. Estimated data can alternatively be computed based on network infrastructure information (e.g., phone exchanges and mobile phone base stations along with their locations and configurations) and theoretical or empirically derived models of different access technologies (e.g., DSL, 3G).

Finally, *policy makers and regulators* can query the broadband mapping system using web service APIs to obtain a true picture of broadband coverage and quality in a given region, and make informed decisions accordingly (e.g., public sector intervention, provide incentives for investments in underserved areas, telecom market regulation). Policy makers could potentially fund the mapping initiative, as is already the case in US, Germany and UK.

6.2.2 Design and Implementation

BSense brings together different types of data for the purpose of broadband mapping. Data supplied by the ISPs is fed into the BSense database via webservice API calls and broadband users are the key source of continuous measurement data for the mapping system. This is enabled by a lightweight software agent termed *BSense Agent* that runs in the background on a user computer and periodically communicates with BSense *Test Servers* to measure technical attributes of a user's broadband characteristics such as download speed, upload speed and latency. Digital geographic data from country-specific sources and demographic data from population census records are additionally used as layers supporting estimated or measured broadband statistics to generate broadband coverage or quality maps, respectively.

Figure 6.2 shows the BSense software architecture, which is described in detail in the rest of this section. Although parts of the system follow a centralized paradigm, it does not present any single points of failure. Indeed, we foresee that large BSense setups will be, in practice, deployed using a collection of geographically distributed

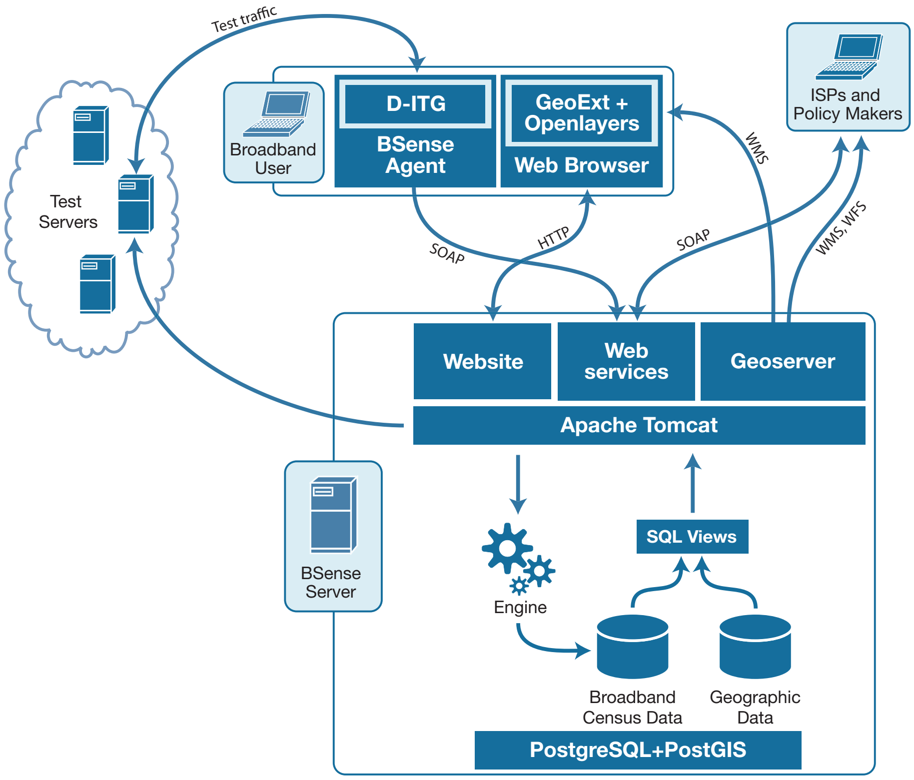


Figure 6.2: *BSense software architecture.*

servers, or by exploiting a global Content Delivery Networks (CDN)[6] as is commonly done today for large websites.

6.2.2.1 Data warehousing

We start by looking at data management in the mapping system. All data is stored in a relational database, the schema for which are shown in Figure 6.3.

As broadband mapping is usually carried out at some geographic granularity (e.g., national-level, state-level), geographic units play a key role in the broadband mapping data. We assume that the geographic region of interest is organised into distinct `GeoUnits`, each with an associated unique ID, name and boundaries. Fine-grained `GeoUnits` may in turn belong to several coarser-grained `GeoUnits`. We incorpo-

[6] A CDN is a network of identically-configured web servers installed at distributed locations, which is used to maximise the uptime of a website and to spread the load over multiple servers. Popular examples are Akamai (`http://www.akamai.com`), Amazon EC2 (`http://aws.amazon.com/ec2`) and Rackspace (`http://www.rackspacecloud.com`).

GeoUnits		
GeoUnitID	varchar	PK
GeoUnitName	varchar	
Boundaries	polygon	
GeounitParentID	varchar	
TierID	int	

GeoUnitTiers		
TierID	int	PK
TierName	varchar	

GeoUnitStatistics		
GeoUnitID	varchar	PK
Population	int	
AddressCount	int	
SmallBusinessCount	int	
…		

Operators		
OperatorID	int	PK
OperatorName	varchar	

Technologies		
TechnologyID	int	PK
TechnologyName	varchar	

Packages		
PackagtID	int	PK
OperatorID	int	
TechnologyID	int	
PackageName	varchar	
Description	varchar	
AdvertisedSpeedDown	float	
AdvertisedSpeedUp	float	
Details	xml	

BroadbandUsers		
UUID	uuid	PK
Postcode	varchar	
EmailAddress	varchar	
RegistrationTimestamp	datetime	
OperatorID	int	
TechnologyID	int	
OSVersion	varchar	
AdvertisedSpeedDown	double	
AdvertisedSpeedUp	double	
SuccessfulTestsNumber	int	
SuccessfulTestsLast	datetime	
FailuresNumber	int	
FailuresLast	datetime	
DownloadSpeed	double	
UploadSpeed	double	
Latency	double	
Jitter	double	
PacketLoss	double	
BSenseScore	double	

RawMeasurementData		
MeasurementID	int	PK
UUID	uuid	
ClientIp	varchar	
Timestamp	timestamp	
RawResultsUp	text	
RawResultsDown	text	
UpstreamExperimentID	int	
DownstreamExperimentID	int	
BSenseVersion	varchar	

ParsedMeasurementData		
MeasurementID	int	PK
UploadSpeed	double	
DownloadSpeed	double	
Latency	double	
Jitter	double	
PacketLoss	double	
Valid	boolean	

EstimatedData		
GeoUnitID	varchar	PK
PackageID	int	PK
ExpectedSpeedDown	float	
ExpectedSpeedUp	float	
Details	xml	

Figure 6.3: *Database schema in the current* BSense *implementation.*

rate such hierarchical geographic subdivisions via the `GeoUnitTiers` tables. Taking Scotland as an example, postcodes would be a suitable candidate for fine-grained geographic units: there are 152,000 postcodes in total. Postcodes in turn are aggregated into 1222 Scottish Census Area Statistics (CAS) wards with a minimum size of 50 residents and 20 households. CAS wards are further aggregated into 32 council areas of varying size. Demographic statistics for a geographic unit are stored in `GeoUnitStatistics`. In our current implementation, we have successfully imported spatial data for postcodes, wards and councils for Scotland from governmental sources into `BSense`, and population data from the 2001 census of Scotland.

Details about participating ISPs are listed in the `Operators` table, whereas their service packages are stored in the `Packages` table, and the available access technologies (e.g., ADSL, Cable, Wireless) are in the `Technologies` table. The `EstimatedData` table contains the estimated broadband data and service availability obtained from ISPs (or equivalent means) for each GeoUnit.

When broadband users register with the mapping system by downloading and installing the agent software (described in Section 6.2.2.3), their details are stored in the `BroadbandUsers` table. Every time the software agent completes an experiment, the "raw" results (i.e., download/upload speeds, delay, packet loss rate and jitter for each sampling interval[7]) are uploaded and stored in the `RawMeasurementData` table along with a reference to the particular type of experiment. A server-side component periodically analyzes the recent results and computes average values across all the raw samples for each experiment, storing them in the `ParsedMeasurementData` table. Summary statistics across all experiments/measurements for a user's connection are updated in the tuple corresponding to that user in the `BroadbandUsers` table as well as the user's broadband quality index value (see Section 6.2.3).

Note that broadband speed appears in the database in three different places with different meanings:

1. The attribute *AdvertisedSpeed* in the `Packages` table refers to the service characteristic advertised by an ISP, typically an *"up to"* value and tied to the subscription cost.

2. An ISP can also provide an *ExpectedSpeed* for each GeoUnit, which is stored in table `EstimatedData`. This matches the maximum advertised speed across all packages from the ISP at the location corresponding to the GeoUnit with PackageID in the table indicating the corresponding service package. It can be empirically derived or based on a theoretical model (as is typically the case with DSL or mobile broadband access).

3. Finally, speeds obtained from continuous measurements are included in `BroadbandUsers` table and related tables.

In our implementation, we used the open source PostgreSQL[8] database management system augmented with the PostGIS[9] extensions to handle spatial data.

[7] Set to 100ms in the current implementation. We adopted this default value for the reasons discussed later on in Section 6.2.2.3.

[8] `http://www.postgresql.org`

[9] `http://postgis.refractions.net`

6.2.2.2 BSense server

Besides the database just described, the `BSense` server incorporates a web server for hosting a public website that users can access for registering and downloading the `BSense` agent software and subsequently to retrieve their broadband connection statistics. The web server also supports a set of web service API calls over SOAP for interaction between the `BSense` system and various stakeholders. The current API consists of the following calls:

- `BroadbandTestRecord()`, called by the `BSense` client every time a broadband test is completed. It records the results into `RawMeasurementData` table in the database.

- `AddPackage()`, `EditPackage()`, `DeletePackage()` are used by participating ISPs to manage their broadband service packages stored in the database.

- `AddEstimatedData()`, `EditEstimatedData()`, `DeleteEstimatedData()`, are called by ISPs to update the estimated broadband data for each geographical region covered by any of their service packages.

- `LookUpCensusData()`, invoked by consumers via the public website and by ISPs and policy makers using SOAP calls to query the `BSense` system.

These API calls are handled by a server-side component that enforces security and access control, validating the input and checking whether an API call is made by a party with the required permissions.

In addition to SOAP based web services, `BSense` provides external access via the Open Geospatial Consortium's standard WMS (Web Map Service) and WFS (Web Feature Service) to obtain raster and vector geo-referenced images, respectively, of a geographical area of interest. Most open-source and commercial GIS software products can directly use WMS and WFS services.

`BSense` also provides a built-in web application based on WMS, developed using the open-source GeoExt[10] and OpenLayers[11] frameworks, to further ease access and visualisation of broadband maps.

[10] `http://www.geoext.org`
[11] `http://openlayers.org`

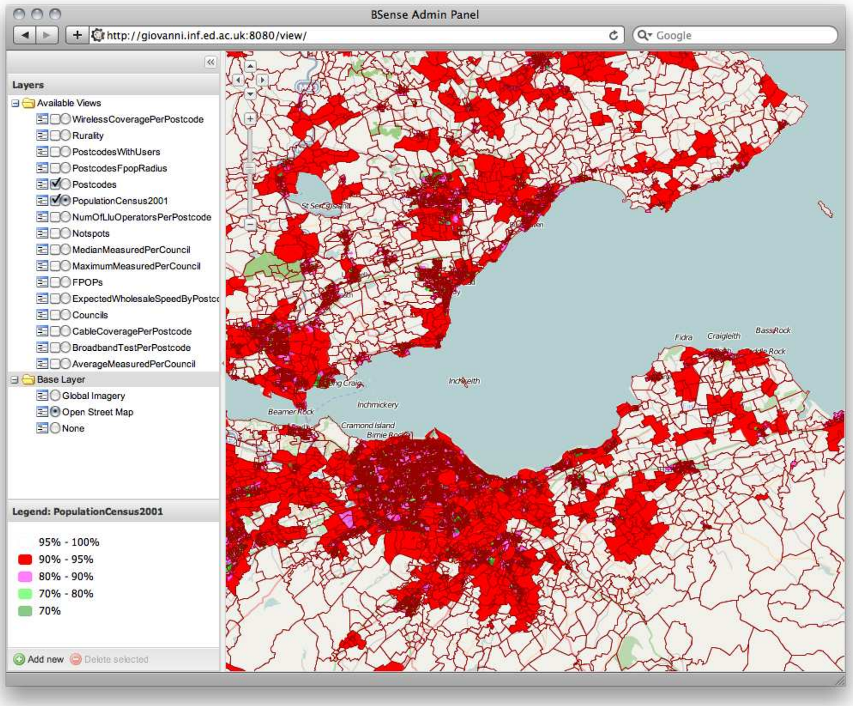


Figure 6.4: *The geographical web interface of `BSense` allows third parties (e.g., regulatory bodies and policy makers) to study the data gathered by the system, by visualising it on a map.*

Access to the `BSense` database via WMS and WFS is enabled by the well known open-source GeoServer[12], a Java application that allows users to view and edit geospatial data. Our implementation's reliance upon GeoServer reduces the complexity of projecting data in maps and rendering them as images. In our implementation, spatial data is stored in the PostGIS database and is accessed by Geoserver via SQL `VIEW` constructs to dynamically retrieve required broadband data in real time (see Figure 6.4). Besides broadband mapping data, `BSense` allows the storage of additional geospatial layers with geographic and demographic data that are useful when generating maps.

These geographical web services and APIs can also be used to access the measurement data and the estimated data provided by the ISPs.

[12] `http://www.geoserver.org`

6.2.2.3 BSense agent

Each broadband consumer participating in the `BSense` based mapping exercise runs a software agent that facilitates continuous and cost-effective measurement of the consumer's broadband connection. As such, the agent is a key element of the `BSense` framework in gleaning the quality of broadband provisioning in a given region. Given the diversity of operating system platforms used by consumers in the real world, the agent should function on different, commonly used platforms to avoid measurement bias. The `BSense` agent was designed explicitly keeping in mind this requirement for multi-platform support. Specifically, it has been developed using Qt[13], an open-source and cross-platform application/UI framework. Qt is available for Windows, Linux and Mac OS X. D-ITG (Botta et al., 2007), the measurement tool chosen in our implementation, was also modified to work on the same platforms.

A participating broadband user downloads the agent from a public website, such as the one we developed[14]. Once installed on the user's home computer, the agent presents the user with a short questionnaire (Figure 6.5(a)) to collect relevant details, such as user location[15], the current ISP name, the access technology, subscribed broadband package and so forth. From that point on, the agent keeps running in the background, and an icon in the OS notification area (e.g., the Windows `systray`, as in Figure 6.5(b)) enables the user to see when a measurement test is in progress as well as to access the statistics of the broadband connection under test (Figure 6.6). We tried to make `BSense` as simple as possible to set up and operate. For example, measurements are taken automatically without the need for user intervention, as long as the user's computer is connected to the Internet. Also, to identify the user, we decided not to provide authentication credentials (e.g., username and password) to remember, adopting instead a unique hash generated at setup-time and stored in the application.

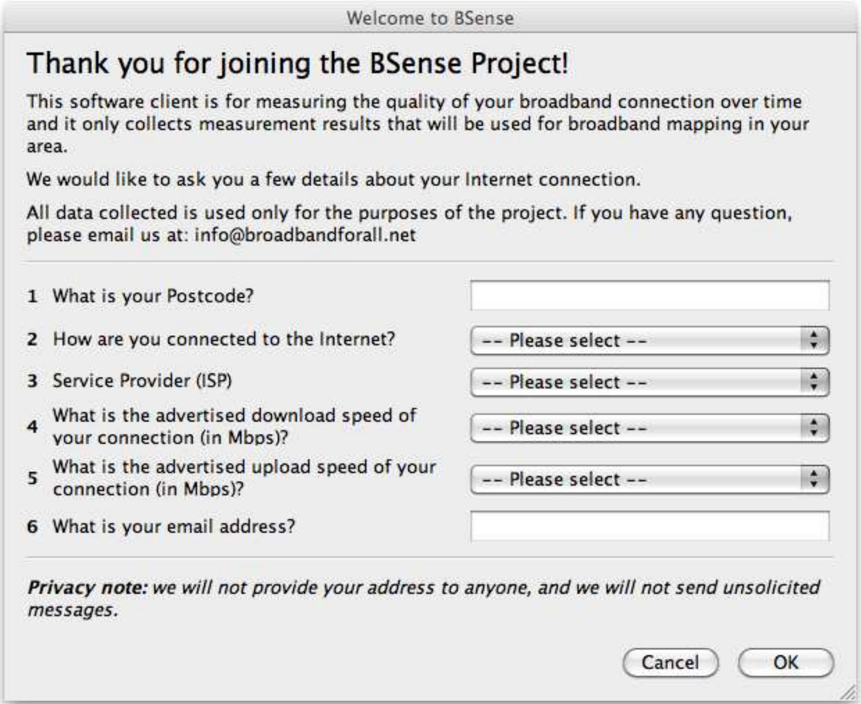
We first give a high-level overview of the measurement process before going into the details. The agent running on a user's computer periodically wakes up to perform a measurement test of the user's broadband connection[16]. Each measurement test con-

[13] `http://qt.nokia.com`

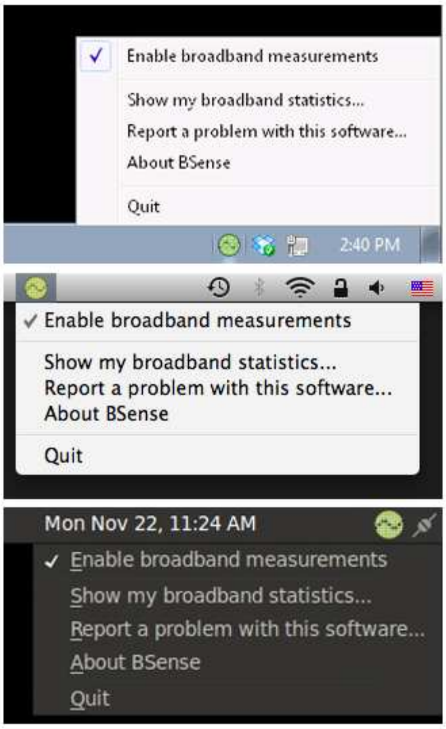
[14] `http://broadbandforall.net`

[15] We prefer to ask the user his postcode instead of guessing his location with a geocoding service, as these tend to be unreliable in rural areas.

[16] The time interval between measurements (equivalently, measurement frequency) is a customisable parameter whose setting is a tradeoff between gathering fine-grained measurement samples over time and measurement overhead. The value is set to 15 minutes in the current implementation.



(a)



(b)

Figure 6.5: *BSense agent: (a) the initial form used to gather user's details, (b) the agent icon and menu as it appears in Windows, Apple OSX and Linux.*

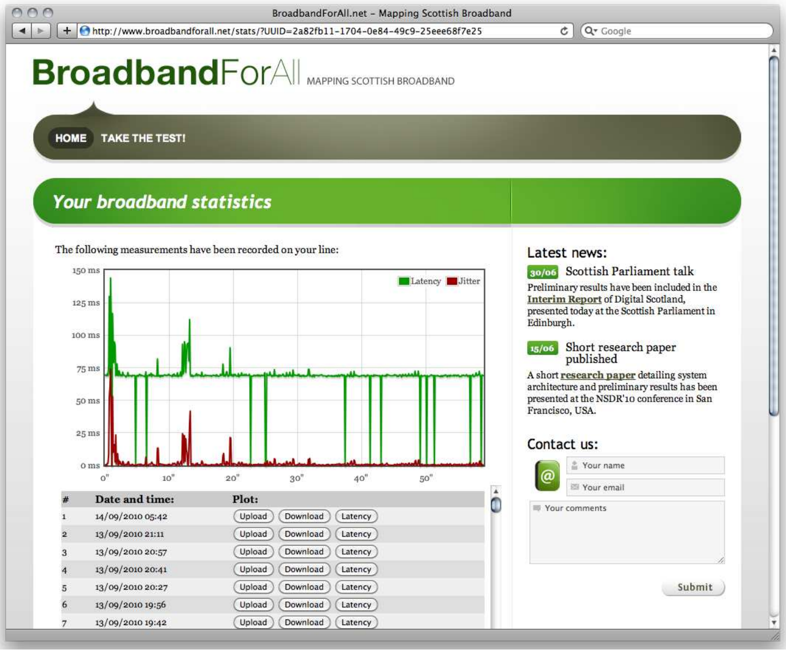


Figure 6.6: *Sample webpage showing the measurement results and statistics collected by* `BSense` *for a user's broadband connection when the user clicks on the agent icon and selects the "Show my broadband statistics..." option from the menu.*

sists of the following sequence of steps (Figure 6.7):

1. The agent queries the `BSense` server[17] to obtain details of the next measurement to be performed.

2. The `BSense` server replies with an "experiment definition" (elaborated below) as well as details for a test server to be used (e.g., IP address, port number).

3. The `BSense` server also simultaneously notifies the test server about the impending measurement test from the user's agent.

4. The agent interprets and follows the experiment definition received, generating the traffic flow requested and/or receiving the incoming traffic to/from the specified test server.

[17] As previously suggested, large deployments could serve the `BSense` web pages and webservices via a Content Delivery Network to increase availability.

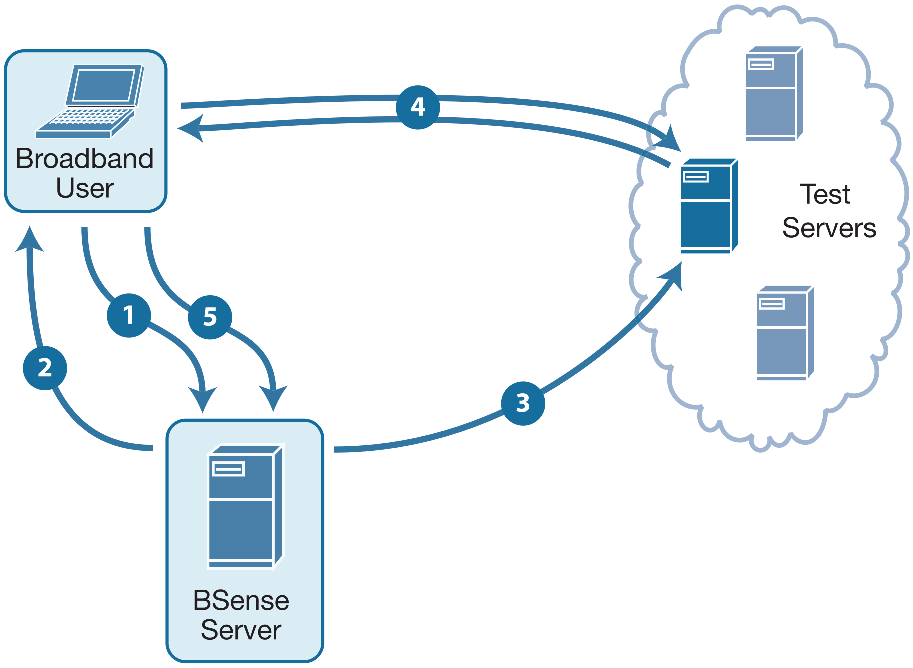


Figure 6.7: *The sequence of steps making up a measurement test initiated by an instance of the `BSense` agent.*

5. Upon test completion, the agent summarises the traffic traces from the test and uploads it to the `BSense` server.

Since the effectiveness of the mapping framework for regional broadband quality assessment improves with a larger number of participating users, the mapping system should be scalable and robust to server failures. In the case of the `BSense` server, this can be achieved through the use of a server farm (the current approach) or by installing the `BSense` server on a cloud-based hosting. The use of multiple test servers also contributes towards scalability and fault tolerance. As regards to the location of test servers, we advocate their deployment at neutral Internet exchange points (IXPs) (e.g., the ones listed at `http://www.euro-ix.net/`) to avoid introducing bias against users of any ISP.

The `BSense` agent could in principle use any multi-platform network performance measurement tool. In our implementation, we chose a widely used traffic generator called D-ITG (Botta et al., 2007) with the `BSense` Agent acting as a wrapper application. While D-ITG can be seen just as a placeholder in the current implementation, it also has several attractive features that have influenced our choice.

- D-ITG is open-source and works on Linux and Windows platforms out of the box; we easily ported it to work Apple MacOS as well.

- It provides a high degree of flexibility for generating a wide range of traffic types from synthetic traffic to ones reflecting real applications; it supports both TCP and UDP flows as well as multi-flow traffic sessions.

- We could make it work behind most common types of NATs without much effort and in collaboration with the D-ITG team. This is important because most residential broadband users have NAT on their home Internet router. Specifically, we have implemented a “Passive Mode” which allows downstream tests to operate in both UDP and TCP modes as follows. Before the start of a measurement test, the test server spawns a D-ITG Receive daemon that listens on a port made known to the `BSense` agent by the BSense server initially. The BSense agent connects to the port and expects traffic to come in the reverse direction (server to agent), which works naturally when in TCP mode. Downstream traffic in UDP mode exploits the “hole punching” effect[18].

- It can be easily interfaced with our `BSense` agent by passing measurement test parameters from the command line. We have also been able to easily create a “patched” version of D-ITG that outputs a file with fine-grain traces of the results. In the current implementation, sampling is performed at 10Hz, which we believe to be a ‘safe’ default value: although higher sampling frequency settings would provide finer-grained resolution over the measurements, they come at the expense of increased storage requirements and longer time to parse and consolidate the result of each measurement in the database. The 10Hz sampling used in our implementation is also motivated by the fact that we are not interested in micro-fluctuations (i.e., shorter than 100ms) to speed and latency. A table is exported, with an entry per time slice containing the measured speeds, latency, jitter and packet loss. We have also taken special care to limit the overhead of the measurement related operations of the agent on the host platform (e.g., by using internal OS variables to record the time skew between the actual inter-departure time of packets and the configured value).

[18] `http://en.wikipedia.org/wiki/UDP_hole_punching`

Every time the `BSense` agent awakes to performs a measurement, it executes an *experiment*: a set of traffic session specifications, with each session potentially consisting of multiple concurrent or partially overlapping flows. Specifically, each experiment is a sequence of three traffic sessions: an initial ping-like UDP session with short packets to measure latency, jitter and packet loss rate, followed by an upstream traffic session and then a downstream traffic session[19]. Depending on the round trip-time between the `BSense` server and agent, we may not be able to simply run experiments for a fixed amount of time. Our solution is to measure the latency first, so that we can adjust the length of the downlink and uplink tests as a function of (estimated) bandwidth delay product between client and `BSense` server. Note that it is straightforward in our framework to define new experiment types. Also, rules can be defined to have the system determine which experiment types to assign to each user, for example based on the access tier technology to which they are connected (e.g., shorter tests for satellite users), the operating system, etc.

When parsing results of a measurement, the `BSense` server checks for its validity by performing a set of sanity checks as well as making sure that the test was done when the user was associated with the registered broadband connection. It does this by querying the Regional Internet Registry[20] using the `whois` protocol to find the Autonomous System (AS) to which the current IP address of the user computer belongs. For a measurement to be considered valid, the AS number must be among those associated to the ISP registered against the user in the `BSense` database. Result of this validity check is included as a boolean flag in the `ParsedMeasurementData` table (see Figure 6.3).

[19]Unless otherwise mentioned, in our evaluations, we used the following parameters. The initial UDP session is a bidirectional flow with 56 byte packets and 10 packets/second for 60 seconds. The upstream traffic session is composed of 8 concurrent UDP/TCP flows with 1024 byte packets at 400 packets/second for 15 seconds. The downstream traffic session has similar parameters as the upstream session. In these initial tests, we picked small (1KB) packets in order to fill the MTU of most connections without incurring in the additional complexity of fragmentation. We consider two experiment types corresponding to using TCP or UDP and alternate between them, i.e., experiment type with UDP upstream/downstream traffic sessions is carried out every 30 minutes with the 15 minute default time between measurement tests and likewise for the other experiment type with TCP upstream/downstream traffic sessions.

[20]In Europe, the RIPE NCC: `http://www.ripe.net`

6.2.3 Broadband Quality Index

As noted earlier, coverage and quality are the two fundamental aspects of interest for broadband mapping. While broadband coverage in a particular location can be easily quantified as a binary variable, the same is not the case for broadband quality, as the latter is dependent on several underlying technical attributes such as download speed, upload speed and round-trip latency. Due to the lack of standard ways to summarise the collective impact of those attributes, the focus is often solely on download speeds even though it is widely recognised that upload speeds and latency have to be also considered, at the very least.

Defining an *index* is a common approach to deal with problems of the above nature. A simple example is the 'h-index' used to measure the scientific impact and productivity of a researcher (Hirsch, 2005) by combining attributes such as number of publications and citation counts of each paper (i.e, "A scientist has index h if h of [his/her] N_p papers have at least h citations each, and the other (N_p - h) papers have at most h citations each"). Corporate governance index is another somewhat more complex example that is used for comparing corporate governance regulation across countries and its impact on economic activities (Martynova and Renneboog, 2010); it involves 3 high-level attributes which are in turn based on 8 lower level attributes. The case of corporate governance index also illustrates that there is no single best way to define an index; in practice, the first reasonable proposal that is widely adopted becomes the standard, much like the widely successful communications technology related standards (e.g., IEEE 802.11). The only work of which we are aware that tries to address the issue of developing a broadband quality index is surprisingly a sociological study (Vicente and de Bernabè, 2010), relying on expert surveys to determine the relative importance of various technical attributes. More importantly, it only provides a very specific approach to defining the broadband quality index, while our interest is in a more flexible and general framework.

Our main idea in addressing this issue is to model each attribute impacting broadband quality as a *utility function* and then draw upon multi-attribute utility theory (MAUT) to define the *broadband quality index (BQI)* as a composite function of utility function values for the individual attributes.

Utility functions derive from the economical concept of relative satisfaction generated from the consumption of certain goods or services. Their use in the networking context is not new. For example, the authors of (Fiedler et al., 2006, 2005) suggest two utility functions that model the influence of network performances on application behaviour and, consequently, on user satisfaction.

Multi-attribute utility theory is an evaluation scheme used to obtain a conjoint measure of the utility, or attractiveness, of a set of alternatives when competing objectives into accounts. MAUT explicitly compute the utility of the different options available, which are then compared on the basis of their utility values. The MAUT function produces a single score, or *index*, representing the utility of each alternative, enabling a decision maker to select the one that gives the highest score.

Each item is described in terms of n attributes $A_1, A_2, ..., A_n$ which can assume values from the domains $V_1, V_2, ..., V_n$. An option o could then be represented as a tuple $o = \{x_1, x_2, ..., x_n\}$. A set of functions $u_i(\cdot)$ is defined on each attribute domain, providing the perceived utility of each.

The combined utility for o is then defined by some function f as:

$$U(o) = f(u_1(x_1), u_2(x_2), ..., u_n(x_n)) \tag{6.1}$$

When the attributes x_i are mutually independent, the most commonly used technique for combining the utility functions $u_i(\cdot)$ is the additive method, based on the weighted sum:

$$U(o) = \sum_{i=1}^{n} w_i u_i \tag{6.2}$$

with:

$$\sum_{i=1}^{n} w_i = 1 \tag{6.3}$$

where x_i is the actual value for the ith attribute of o and w_i is a relative weight of the importance of parameter A_i as compared to the other attributes.

A key step before applying MAUT is to determine the '*utility*' provided to decision-makers by each attribute value. This is commonly done by first determining each single-attribute utility functions $u_i(\cdot)$ for a sample set of attribute values x_i. The single-attribute utility functions are then obtained either by interpolation or by using a curve-fitting method to obtain a continuous function. The weight coefficients w_i of the utility function are then estimated based on the relative importance of the attribute domains to the decision maker.

MAUT applies a class of psychological measurement models and scaling procedures to the evaluation of alternatives which have multiple relevant attributes. In our case, it allows us to decompose a complex evaluation task into a set of simpler subtasks. MAUT has been extensively used in the networking domain, for example in (Bari and Leung, 2007) for the purposes of selecting a wireless network among multiple alternatives; in (Lamparter et al., 2005) to enable intelligent agents to choose the most suitable service from a large number of providers; and in (Musolesi et al., 2005), where MAUT was used to evaluate and fuse information about the state of a node (e.g., battery energy level and the rate of change of connectivity) in a routing protocol for asynchronous communication in partially-connected mobile ad hoc networks.

In our model, we denote $F = [f_1, ..., f_n]$ for the set of network attributes (features) to be included in the BQI, covering important attributes characterising the performance of a broadband connection. For the sake of concreteness, in the current implementation we focus on three key performance attributes: download speed (f_d, in Mbps), upload speed (f_u, in Mbps) and round trip-time latency (f_l, in milliseconds). The relation $\succeq$ provides a complete, transitive, and reflexive preference structure that expresses "preference" between two any broadband connections $b \in B$. For example, a broadband connection b_1 is preferred to b_2 if $b_1 \succeq b_2$. Preference can be derived from the multi-attribute utility function $u(\cdot)$ as follows:

$$\forall b_1, b_2 \in B : b_1 \succeq b_2 \Leftrightarrow u(b_1) \geq u(b_2) \tag{6.4}$$

The function $u(\cdot)$ translate a set of individual attributes into a single numerical representation of their combined utility.

We will now proceed to assessing a suitable family of single-attribute utility (SAU) functions for modelling each of three attributes considered, and their composition into a multi-attribute function.

The choice of utility functions in other contexts, especially in economics and finance, is often governed by analytical tractability even though the chosen functions (e.g., exponential, logarithmic, negative exponential) may not always be realistic. Our concern is the other way around. Besides, for specific applications like multimedia applications, there has been much work on subjective quality metrics such as mean opinion score (MOS) and quality of experience (QoE) that are used for capturing user satisfaction. These are not suitable for our purpose as we are not interested in a specific

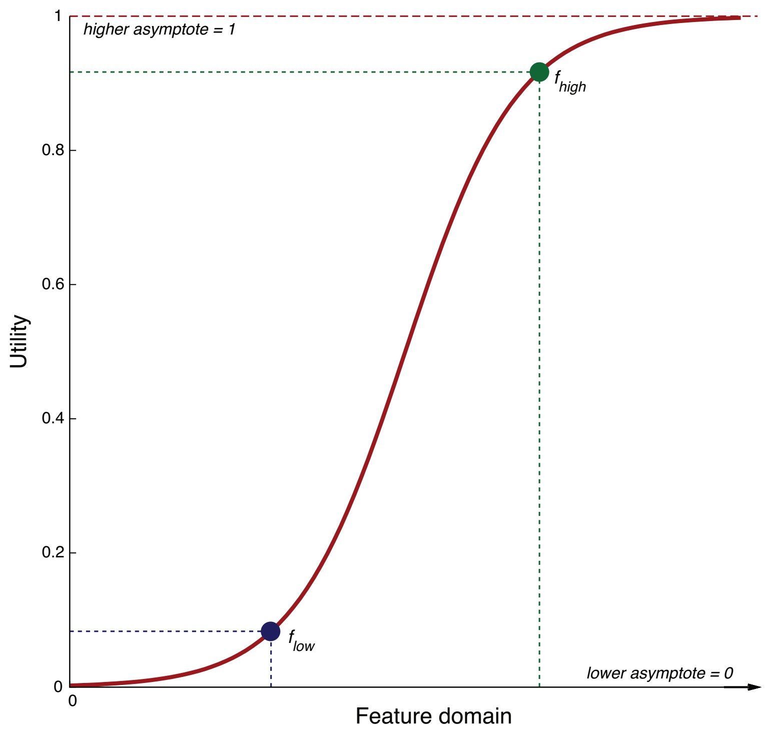


Figure 6.8: *The sigmoid utility function can be seen as a transfer function between a given attribute f and the perceived utility associated with specific values of f. In fact, the function shown is a modified sigmoid function as in equation 6.5 to realise zero utility when the value of f is zero.*

application.

For the attributes under consideration (i.e., download speed, upload speed and latency), we observe that sigmoid functions, whose graphs are “S-shaped” curves (see Figure 6.8), better reflect user satisfaction. This is because improvement in utility from improving any of these attributes beyond a point is marginal. Equally, when these attributes are below a certain threshold (for speeds) and above a certain threshold (for latency), the change in utility beyond those thresholds is again marginal. In between these extremes, the improvement in utility with improvement for any of these attributes is noticeable and substantial.

Thus, we define a set of utility functions $u_f(f)$, each defined on a given attribute $f \in F$ as:

$$u_f(f) = \frac{e^{a+bf} - e^a}{1 + e^{a+bf}} \tag{6.5}$$

where the parameters a and b denote the behaviour of the curve. These functions satisfy a set of interesting properties: for example, in the case of the download speed utility function[21] $u_d(d)$:

- the function $u_d(d) : D \rightarrow U$ is continuous, defined on the given domain D (i.e., speed in Mbps) and with codomain $U = [0,1]$, where zero implies that a user is completely unsatisfied, while one implies that the user is completely satisfied.
- $\lim_{d \rightarrow +\infty} u_d(d) = 1$. This property reflects maximum user satisfaction when the networks offers very high speed.
- $\lim_{d \rightarrow 0^+} u_d(d) = 0$. The user satisfaction is null when the speed is zero.
- $if\ d_i > d_j \Rightarrow u(d_i) \geq u(d_j)$. The utility function is monotonically increasing with speed.
- the function is convex on the left of the higher inflection point and concave on the right of it. In utility theory, this can be interpreted as the user being risk prone (i.e., willing to take risks to improve his condition) for low utility values and becomes proportionally risk averse as the utility increases.

Having adopted Equation (6.5) as parametric family of utility functions, we proceed with quantitative assessment to find which specific member of that family is most appropriate. We take a pragmatic approach by having the `BSense` administrator pick two strategic values of each feature f and to specify their corresponding utility u. Two data points are thus provided in the form:

$$(f^o, u^o) \quad \text{and} \quad (f^*, u^*) \tag{6.6}$$

with:

$$u^o = u_f(f^o) < u^* = u_f(f^*) \tag{6.7}$$

By carefully picking these two points (e.g., with $u^o = 0.2$ and $u^* = 0.8$ or $u^o = 0.1$ and $u^* = 0.9$), they represent the utility of low-end and high-end broadband connections. Intuitively, the lower knee in the curve represents the domain values which are deemed as insufficient, and the upper knee describe the domain values which are good

[21] Similar properties are valid for the other features to be considered, such as upload speed and latency. In the case of latency, the utility function is actually 1 minus the value in equation 6.5.

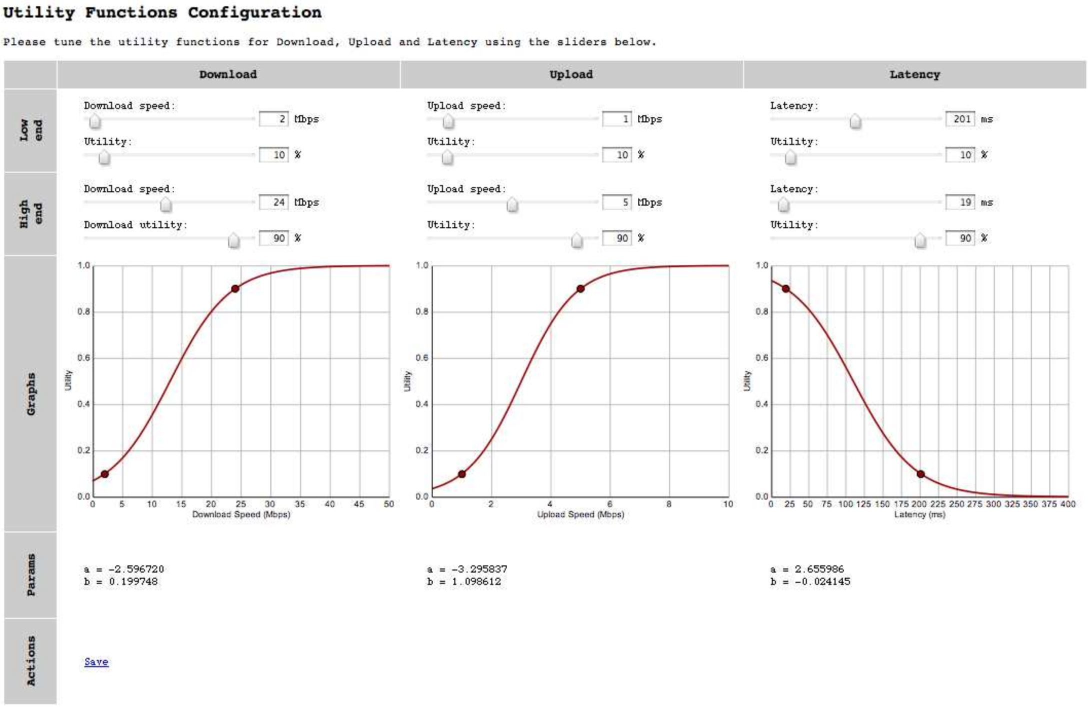


Figure 6.9: *The web-based graphical tool in* `BSense` *to assist the administrator in specifying the parameters for each of the SAUs.*

enough for the service. As a consequence, "poor" broadband lines that are only able to offer features (e.g., speed) below the lower point offer only marginal utility to the users. Similarly, the incremental utility above the upper threshold is only marginal. The low and high values can be based on current policies and regulations (e.g., the "Universal Service Obligation" sets the bottom bar that ISPs have to provide and the policy maker must enforce) or current state-of-the-art (e.g., the fastest fiber-based commercially available sets the top bar). We developed a web-based graphical tool (see Figure 6.9) as part of `BSense` to let the administrator define the two 'knobs' for each attribute and visually see the obtained sigmoid function. Storing a utility function for a domain parameter only requires storing the two knee values, as the upper and lower limits are known to be one and zero, respectively.

The parameters *a* and *b* of each of our SAU function can then be derived from the two specified 'knobs' by solving the following set of equations:

$$\begin{cases} u_f^o = \frac{e^{a+bf^o}-e^a}{1+e^{a+bf^o}} \\ u_f^* = \frac{e^{a+bf^*}-e^a}{1+e^{a+bf^*}} \end{cases} \tag{6.8}$$

To provide a quality index for each broadband connection, we first need to generate an aggregate value from all the measurements that have been gathered by `BSense` for the given connection. In the current implementation, the median values of download speed, upload speed and latency are considered. We then feed these values, together with the a and b parameters, as input to Equation (6.5), to determine the utility of a broadband connection with respect to each of the three attributes.

Multi-attribute utility theory assists us in combining the various SAU functions into a single equation, whose form depends upon the particular independence conditions fulfilled by the different SAU functions. For simplicity, we assume mutual additive independence. Then the resulting multi-attribute utility function can be represented as:

$$u(f) = \sum_{f \in F} k_f u_f \tag{6.9}$$

where the scaling constants (weights) k_f satisfy the conditions:

$$\sum_{f \in F} k_f = 1 \quad \text{and} \quad k_f > 0, \quad \forall f \in F \tag{6.10}$$

It is important to note that by talking about "independence", we are not referring to the numerical values of the technical attributes (e.g., the download speed in Mbps) but of their associated utility in the range $[0,1]$. While it is clear that the former are not independent[22], their associated utilities can be empirically checked against the additive independence constraint by selecting pairs of attributes. Additive value functions like that in Equation (6.9) are valid in many real world scenarios, and can be considered a good approximation even when the required independency assumptions do not precisely hold over the domains of all the attributes. Examples of this pragmatic simplification are presented, for example, in the work of (Winterfeldt and Edwards, 1973; Stewart and Scott, 1995; Belton and Stewart, 2001; Russell et al., 1996). (Keeney and

[22] Consider the following two examples: (1) on shared-media (e.g., wireless or "cable" access connections) network technologies download and upload speeds are mutually dependent; (2) network throughput and delay are not independent because of the bandwidth-delay product and its consequences on transport protocols.

Raiffa, 1976) also presents techniques to allow some violation of the additive independence condition and to assess the consequences of such approximation.

The scaling constants k_f enables the modelling of the relative importance given to each utility function. Interview-based techniques (see for example Farquhar (1980) and Johnson and Huber (1977)), try to assess the values of the scaling constants by asking the decision maker a set of qualitative questions. For example, we can rephrase the technique proposed in (Keeney and Raiffa, 1976) and ask the `BSense` administrator: *"Imagine that each of the broadband features are at the low state (f^o). Would you prefer the download speed (attribute f_d) to be pushed to the high end rather than both attributes upload speed (f_u) and latency (f_l) pushed to their respective high-end (f^*) values?"*. A 'yes' would imply that $k_d > k_u + k_l$, which means $k_d > .5$. The following question would be: *"Would you rather have attribute upload speed (f_u) pushed from low end to high end, than latency (f_l) pushed from low to high?"*. An affirmative answer we would lead to $k_u > k_l$. By successive iterations, the values of the k_i's can be estimated.

In our experiments, we made an additional simplifying assumption that all weights k_f take identical values. This is reasonable given that our main purpose is to demonstrate the value of multi-attribute utility theory and utility functions in providing a flexible framework for defining broadband quality index.

Also, weights can be useful to temporarily 'bypass' the influence of one or more network metrics. For example, a policy maker may want to study the availability of broadband quality in a geographical region solely based upon download speed, to ensure compliance to Universal Service Obligation (USO) regulations[23]. Similarly, a Service Provider may want to study the distribution of broadband with adequate levels of upload speed and latency to evaluate the market for cloud-based services, such as remote desktop virtualization and voice services. For simplicity, in our case study we consider all weights to be of equal value, but these parameters are easily tunable from the `BSense` administration web interface.

The quality score is defined in the interval $[0, 1]$ which, for simplicity, we present to the broadband user as an integer percentage value (i.e., from 0% to 100%). To ensure consistency, we currently release quality scores only for those broadband lines

[23]Such as the 2Mbps minimum speed set by the UK Government, or similar regulations for other European countries.

for which a minimum number of valid samples has been gathered.

6.3 Evaluation

In this section, our goal is two-fold: (1) to evaluate the measurement methodology adopted in BSense under different conditions; (2) to present two case studies using BSense to demonstrate its value for broadband mapping based investigations aimed at coverage analysis and quality assessment, respectively.

6.3.1 BSense Measurement Methodology: OS Impact

BSense relies on various OS calls to manage timers, access the real-time clock, the network and the local filesystem. It is also affected by OS process scheduling. Since the BSense agent software can be deployed on different OS platforms, it is useful to characterise the OS impact on performance measurement results.

To meet this goal, we perform the controlled lab experiment described below. We set up a small testbed as shown in Figure 6.10. A BSense test server was installed on a Linux computer whose configuration is similar to those used in our real-world evaluations described in later sections. The server is connected over a 1Gbps full-duplex Ethernet link to a second computer, which is configured to run any of the three OS platforms (Microsoft Windows, Apple Mac OS X and Ubuntu Linux) as well as the BSense agent on those platforms. The measurement test between agent and server follows the experiment definitions and default settings described in Section 6.2.2.3. On the network segment between the server and agent, we inserted a third Linux machine that acts as a router, performing NAT for the traffic going to/from the agent. The router also runs the dummynet tool (Carbone and Rizzo, 2010) to allow us to emulate network characteristics of various typical broadband connections. We selected the following five scenarios[24]:

[24]For simplicity, in all scenarios we configured zero drop rate.

Scenario	**Name**	**Characteristics**
(a)	Unlimited	Unlimited 1Gbps full-duplex link (dummynet not active)
(b)	ADSL 2+	24Mbps downstream 3Mbps upstream 70ms RTT latency
(c)	ADSL 2 ("ADSL Max")	8Mbps downstream 1.3Mbps upstream 70ms RTT latency
(d)	Satellite connection	512Kbps downstream 256Kbps upstream 800ms RTT latency
(e)	"Exchange Activate"	512Kbps downstream 256Kbps upstream 70ms RTT latency

Configurations for scenarios (b)–(e) are influenced by common broadband connection types available in the UK. Here, traditional 8Mbps ADSL lines are marketed under the name of "ADSL Max", whereas "Exchange Activate" ADSL lines are available only in rural areas; configuration for the satellite connection reflects the Government-subsidised service for residents in remote and rural parts of Scotland.

The `BSense` client was configured to perform a round-trip latency measurement, followed by two sets of eight parallel streams of traffic (first UDP, then TCP) to measure application-level throughput in both directions[25].

Figure 6.10 shows the result of running measurement tests using the three different OS platforms for the computer where the agent software resides. The Linux platform is arbitrarily chosen as the reference and relative differences with the other two platforms are shown. Differences for upstream/downstream speeds for different scenarios is mostly around 1% and never exceeds 2%. Relative error for the delay is within 4%, higher for the Windows platform as it slightly overestimates the round-trip delay.

[25] The detailed configuration is as follows. For the delay test, we send 56 byte UDP packets at 10 packets/s for 15 seconds (resulting goodput: 4.48 Kbps). For the upstream and downstream tests we use 8 parallel flows, each sending 1024 byte UDP or TCP packets at 400 packets/s for 15 seconds (resulting goodput: 26.21 Mbps). For the upstream tests we use 8 parallel flows, each sending 1024 byte UDP or TCP packets at 400 packets/s for 15 seconds (resulting goodput: 26.21 Mbps).

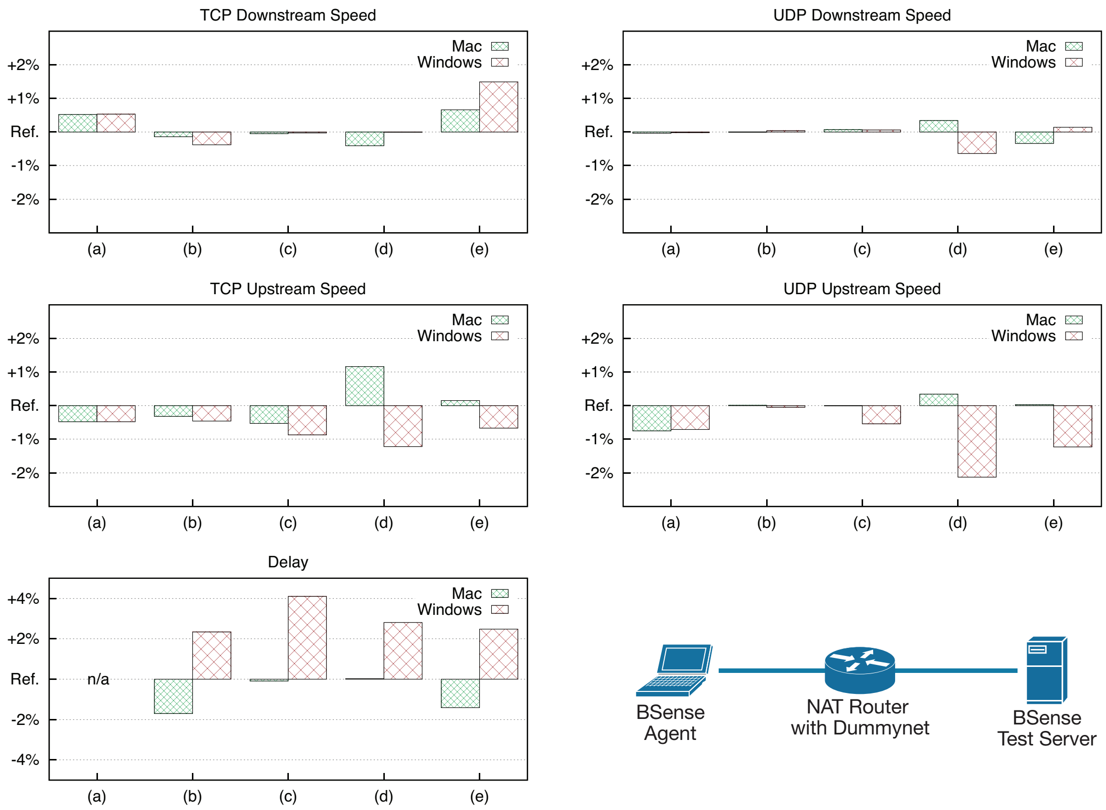


Figure 6.10: *Performance comparison across different OS platforms (with respect to the Linux platform) in controlled lab experiments.*

From these results, we conclude that the OS impact on our measurement methodology is minimal under ideal conditions.

6.3.2 BSense Measurement Methodology: Real World Validation

The real world is much less ideal with several factors differing between different broadband connections even when using the identical host platform and hardware, for example broadband router hardware and configuration, access technologies and other differences in the end-to-end path. We characterise the result of using the `BSense` measurement methodology under real world conditions in comparison with several alternative measurement tools, even though some of them are not readily suitable for continuous measurement on different OS platforms (e.g., NDT).

As in earlier controlled lab experiments, we consider several common broadband scenarios including cable and wireless technologies. Specifically, we consider 6 representative users as follows:

- 3 broadband users on "unlimited 24Mbps" contracts with different ISPs connected over ADSL2+ connections to three different phone exchanges;
- a "Cable" user on a 20Mbps downstream, 768Kbps upstream contract;
- a user of our Tegola (see Chapter 3) testbed in rural Scotland, which provides Internet access to remote communities via long-distance point-to-point wireless links;
- a remote user connected via earlier mentioned Government-subsidised satellite connection with speeds limited to 512Kbps downstream and 256Kbps upstream.

Each of these users was given a Linux laptop pre-configured to run back-to-back measurements using a suite of software measurement tools when connected over Ethernet to the broadband router at the user premises. Measurements are between laptops and a server machine on the Internet that we configured.

Upstream and Downstream speeds were measured using the following on laptops handed to users:

- `BSense` agent (based on the D-ITG tool);
- ShaperProbe bandwidth estimation tool (Kanuparthy and Dovrolis, 2010);
- Network Diagnostic Tool (NDT)[26], which is one of the tools used for FCC consumer broadband measurement tests in the US[27].

The techniques underlying each of these tools are markedly different. Our `BSense` agent uses multiple parallel streams of TCP or UDP traffic to saturate the access path under the assumption that the access tier is the bottleneck. We use multiple connections to prevent the agent or server from becoming a bottleneck instead of the access path because of the known limitations of the TCP receive window mechanism[28], a single

[26] `http://www.internet2.edu/performance/ndt/`

[27] `http://www.broadband.gov/`

[28] The TCP receive window is used by the receiver to tell the sender the buffer size available to store incoming data. The TCP window scale option, described in RFC 1323, is needed when the bandwidth-delay product is greater than 64K. If not supported or enabled, the achievable throughput of a single TCP connection may be limited (e.g., in case of a 80ms link, it cannot exceed 6.55 Mb/s).

connection may be unable to exploit the full speed available. Multiple simultaneous traffic flows are representative of many popular Internet applications (e.g., web browsing) and do not penalise the overall speed result. ShaperProbe estimates the upstream and downstream capacities using packet trains of back-to-back packets over UDP: K packet trains, each composed of L packets of size S are sent over the network. The receiver measures their dispersion Δ and calculates the path capacity as $C_a = \frac{(L-1)S}{\Delta}$. The median value of C_a is given in output as a result. NDT uses packet dispersion techniques, measuring the inter-packet arrival times for all data and ACK packets sent or received. By also taking packet size into account, it can calculate the speed for each pair of packets sent or received.

For RTT latency measurements, we compared the `BSense` agent with the NDT tool and `ping` command line utility (invoked with the default parameters).

All measurements were carried out several times, each time back-to-back for the compared alternatives, for at least 6 hours at night-time without any other local network activity to keep measurement related noise low and allow fair comparison.

Figures 6.11 and 6.12 show the results as Cumulative Distribution Function (CDF) for downstream and upstream speeds, respectively, obtained using different tools and across different types of user connections. We observe that UDP speeds with the `BSense` agent are slightly higher than those based on TCP, which is expected because of the inherent TCP overheads (e.g., due to the slow-start mechanism, the recover time from packet losses and the additional header size). We also observe that `BSense` results are fairly similar to those obtained using ShaperProbe. NDT seems to consistently underestimate speeds, an observation also made in (Bauer and Lehr, 2010) and which may be due to the use of a single TCP connection for speed measurement.

The latency measurement comparisons, shown in Figure 6.13, seem to suggest that `BSense` provides very similar results to those obtained using the `ping` command: in three scenarios the differences are within a few milliseconds. This result is reasonable given that `BSense` measures latency using a bidirectional UDP flow with similar sized packets (56 bytes) as that of ICMP ping $echo-reply$ packets. NDT, on the other hand, tends to consistently overestimate the latency, possibly because of its use of the average of all TCP round trip delay samples to estimate round trip latency for the connection.

Comparison with hardware-based measurement approach. We have also compared the software-based measurement approach used in `BSense` against the hardware-

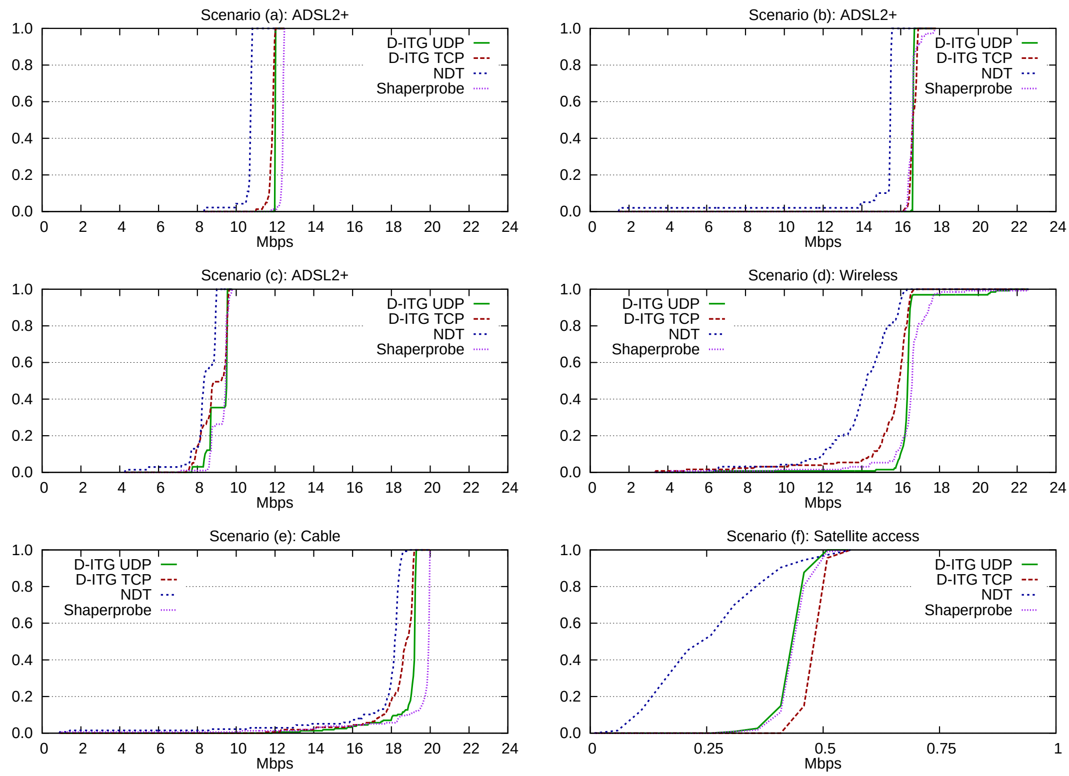


Figure 6.11: *Comparison of downstream speed measurements with different tools and across different user connections.*

based approach employed in the Ofcom/SamKnows broadband speeds study (OFCOM, 2011). For this comparison, we deployed a laptop running the `BSense` agent at a user's home which also had the SamKnows measurement box connected directly to the user's broadband router; the agent on the laptop measured the broadband connection characteristics periodically by communicating with the `BSense` test server on the Internet. The summary of daily measurements collected by the SamKnows box was retrieved from the web based dashboard accessible to the user. Figure 6.14 compares the CDFs of the daily maximum speeds and minimum latencies measured using both the methodologies over a two week period. We find that speed measurements in both cases are fairly similar with less variation whereas latency with the SamKnows case is a bit lower, potentially because its test server is located at a different location from that of `BSense`.

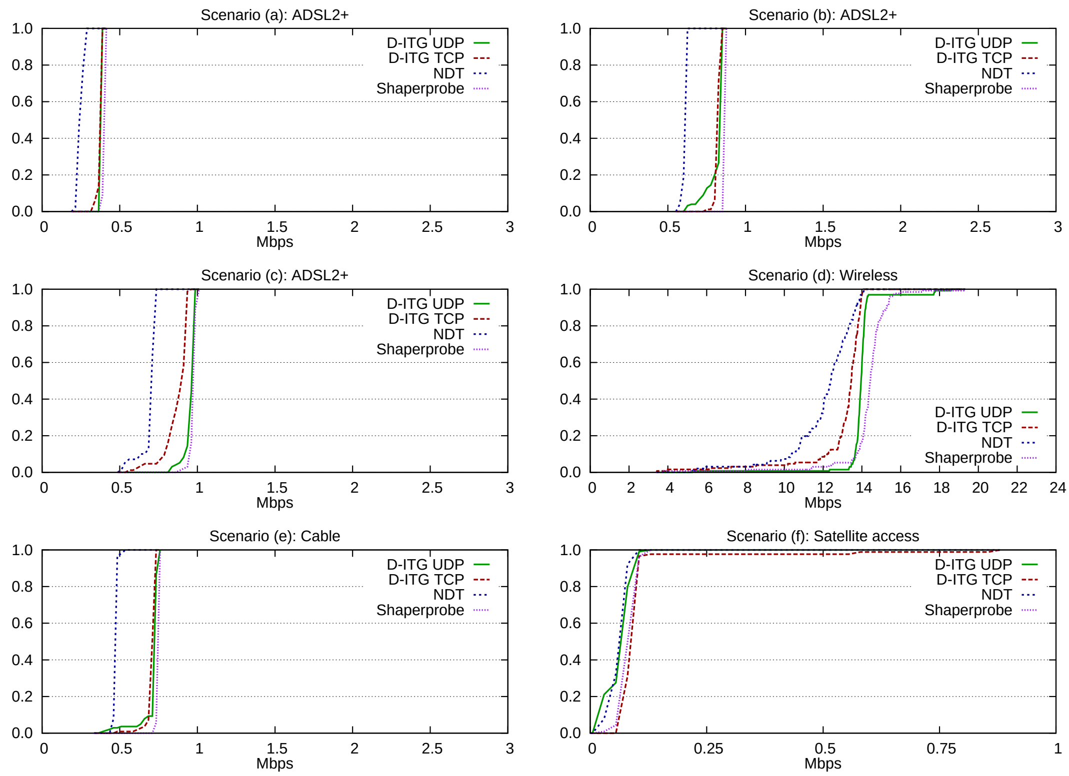


Figure 6.12: *Comparison of upstream speed measurements with different tools and across different user connections.*

6.4 Case Studies

6.4.1 Broadband Coverage Analysis for Scotland

Besides collecting and aggregating broadband measurements, `BSense` also provides an infrastructure that can be used to organise, fuse and analyse broadband coverage and quality data coming from various sources. In this section, we present an example of such activities, which have been carried out using the current version of the software.

In 2010, the Royal Society of Edinburgh initiated the development of the 'Digital Scotland' report[29], a multi-disciplinary study focused on defining a long-term strategy to improve the quality of the Internet 'ecosystem' in Scotland. The document analyses the available broadband infrastructure, combined with demographic and geographical data, to provide key recommendations to the regulatory authority and the ISP industry.

[29] Available at: `http://www.royalsoced.org.uk/enquiries/Digital_Scotland`

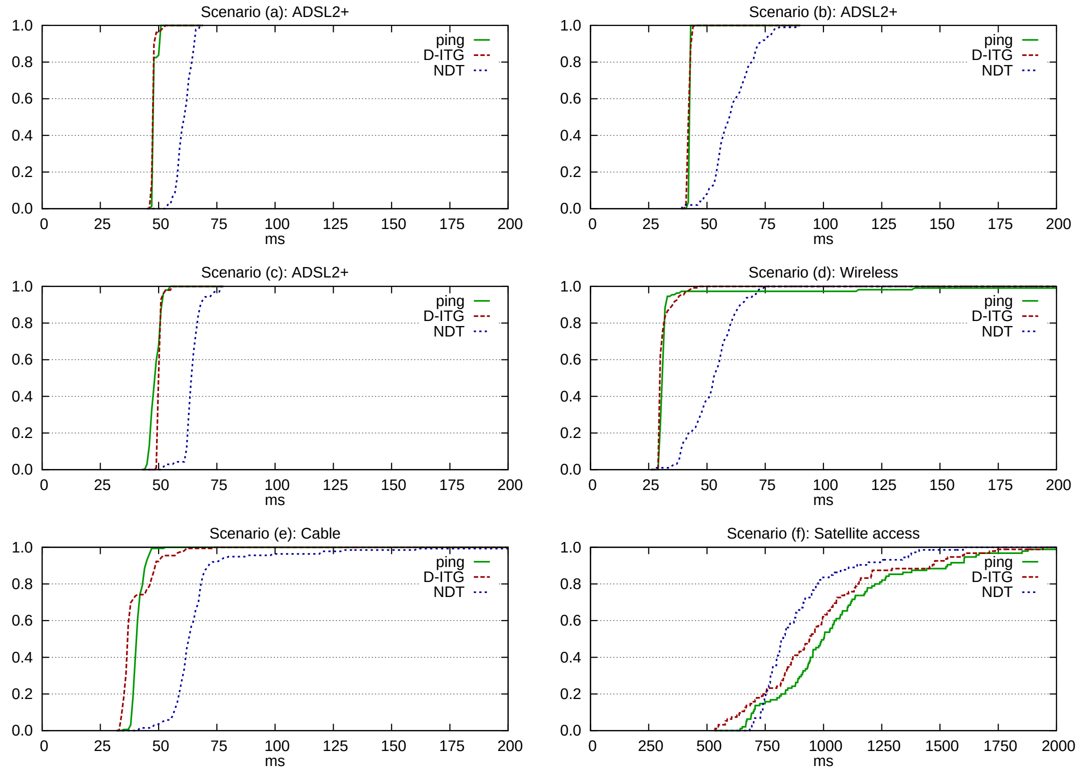


Figure 6.13: *Comparison of round-trip latency measurements with different tools and across different user connections.*

The document is currently being discussed at the Scottish Parliament.

For the purposes of creating an initial broadband census, we mimicked the way ISPs would contribute to the BSense mapping system by trawling through the public websites of different ISPs to determine whether an ISP covers a particular postcode and if so, the estimated download speeds, from the ISP's viewpoint, for each of the 152,000 postcodes in Scotland. This information is then fed into the BSense estimated database via the web service API calls. Figure 6.15 shows the broadband coverage in Scotland for different access technologies based on the estimated data from ISPs collected as described above. For the 3G mobile broadband case, we show data for only one network operator for clarity but the coverage for other mobile network operators is similar. From these maps, we observe that ADSL is the dominant access technology with cable and wireless confined mainly to population centers in the central belt and north east.

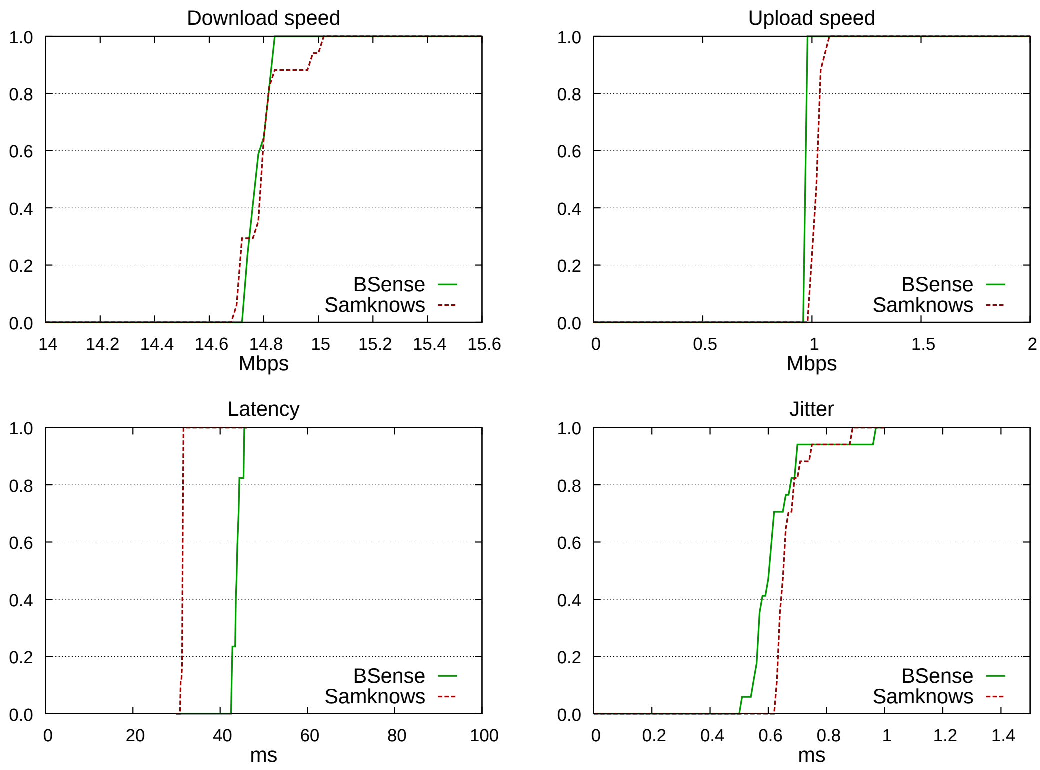


Figure 6.14: *Comparison of software based measurement using* `BSense` *with SamKnows hardware-based measurement for the same broadband connection over a two week period.*

The authors of the ‘Digital Scotland’ report used the reporting features of `BSense` to infer notspots in Scotland by simple fusion of estimated data available. We consider four different threshold values (512Kbps, 1Mbps, 4Mbps, 8Mbps). Resulting notspot maps produced using `BSense` are shown in Figure 6.16. It can be clearly seen that most postcode areas outside of the central belt of Scotland (with the two main cities of Edinburgh and Glasgow and having the largest population concentration) become notspots as the threshold is raised. While it is true that Satellite based broadband covers virtually the whole of Scotland it is not desirable because of the large round-trip latencies as will be clear from the measurement study in the next section.

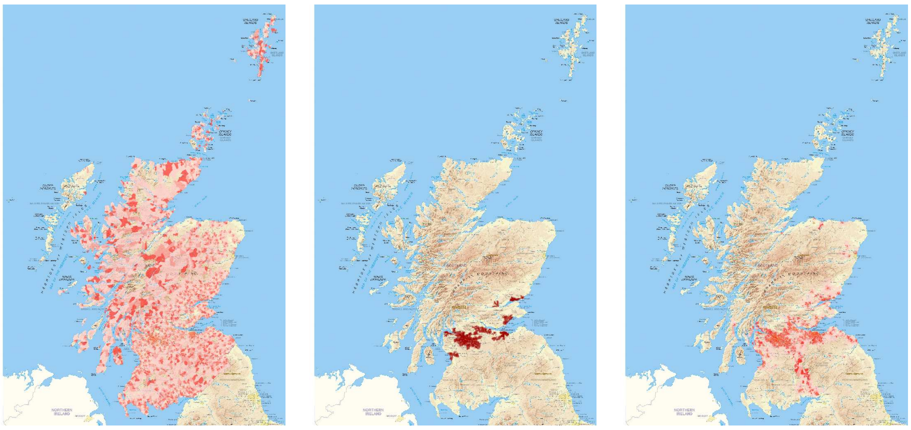
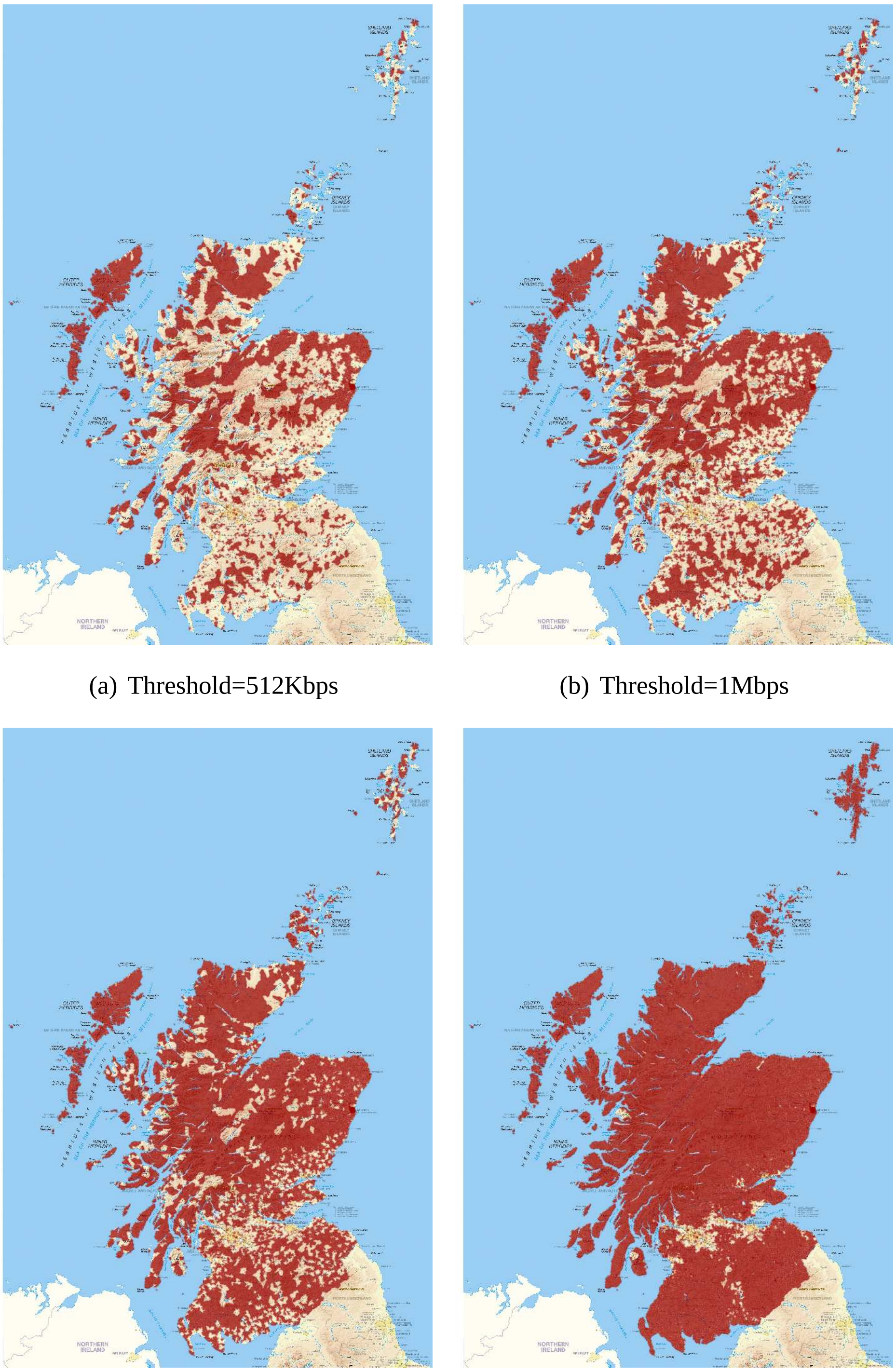


Figure 6.15: *`BSense` generated broadband map for various access technologies and ISPs based on their estimated data: (a) ADSL — BT Wholesale; (b) Cable — Virgin and Smallworld; (c) 3G mobile broadband — Orange. In each of these maps, postcode areas with the corresponding service are colored with darker colors indicating faster expected speeds or better wireless coverage.*

6.4.2 Broadband Quality Measurement: A Pilot Study

In this case study, we assess the broadband quality in a rural part of Scotland. Specifically, we focus on the area located in the western highlands of Scotland and around the Isle of Skye. This region has a population of around 10 thousand people and is demographically quite diverse with a handful of small towns, several small villages and scores of isolated dwellers in the farming lands. We also consider the neighbouring archipelago of the 'Small Isles' and the mainland rural areas of Glenelg and Knoydart peninsulas. The latter is considered the remotest part of mainland Britain with no road access.

Various reasons make this a well suited region for our broadband case study with several different access technologies used for broadband provisioning. In total, 15 phone exchanges are located in the area, and although every resident has access to a landline, broadband quality varies. A few phone exchanges are enabled for ADSL2/ADSL2+, which is available only from the telecom incumbent (BT). Other exchanges offer ADSL (8Mbps download speed) services and a few are enabled only for "Exchange Activate" services, which consists of 512Kbps ADSL lines. There are



(a) Threshold=512Kbps (b) Threshold=1Mbps

(c) Threshold=4Mbps (d) Threshold=8Mbps

Figure 6.16: *BSense generated map of notspots in Scotland that lack a service supporting download speed greater than the indicated threshold. Notspot postcode areas are shaded in red.*

no FTTH deployments in the area, and cable and 3G coverage is also virtually non-existent. Due to a recent broadband reach initiative from the Scottish government, some of the users in rural and remote areas in previously notspot areas now connect via a subsidised yet relatively expensive satellite connections. In addition, residents in a small part of this area connect via the Tegola network.

Through publicity of our pilot broadband quality assessment initiative via email, local press and word of mouth, we recruited 60 volunteers in the area who were willing to install and run our `BSense` agent software. Half of these users are connected to the Internet via ADSL lines to different exchanges and with differing line lengths, whereas 18 users connected via our Tegola network; the remaining volunteers used satellite connections. Over a three month period, we measured the broadband connections of each of the volunteer users, keeping track of median values of download/upload speeds and latency measurements for each user. We collected around 40,000 measurements in total. To study the broadband quality index across users and access technologies, we used the following parameter settings for the individual utility functions (see Section 6.2.3). We justify these values as follows:

- The UK's Universal Service Obligation (USO) dictates that the minimum download speed of broadband connections should be 2Mbps, which becomes our low-end knee.

- The maximum theoretical download speed for ADSL2 services (24Mbps) is taken as high-end knee.

- Latency values above 200ms render the user experience of real-time applications (e.g., two-way video), while latencies below 20ms provide only neglibile difference.

- Modern Internet websites, cloud services and video conferencing require higher uplink speeds than legacy applications. For this reason, we give marginal utility to connections unable to offer at least 1Mbps in uplink.

Overall, we believe that these values are reasonable given the type of broadband connections in the target area.

Download speeds	*Low-end*	2Mbps with a utility of 0.1
	High-end	24Mbps with a utility of 0.9
Upload speeds	*Low-end*	1Mbps with a utility of 0.1
	High-end	5Mbps with a utility of 0.9
Latency	*Low-end*	200ms with a utility of 0.1
	High-end	20ms with a utility of 0.9

Figure 6.17 shows the results. The top graphs show the utility function values for each of the three performance attributes. Each data point in the plots corresponds to a user with the color of the data points indicating the access technology used. Clearly and as expected, satellite users have poor utility values and are cluster together at the worst extreme. Wireless users on the Tegola network, on the other hand, while experiencing high speeds exceeding 20Mbps also face greater variability in terms of speeds because of the shared nature of access. ADSL users also exhibit greater variability in speeds like wireless users but due to different reasons: differences in broadband capabilities of the associated phone exchanges and differences in line lengths. Most ADSL users fall in between satellite users and wireless users in this area. The bottom graph shows the combined effect of the three attributes. Results for different access technologies reflect the top graphs given our simple choice for multi-attribute utility function, which in this case is just an equal weighted sum of individual attribute utilities. However, even this simple function used as a broadband quality index is quite revealing when the index values for different users are geographically rendered on a map (see Figure 6.18). Here, coloring is done at the ward level: all users belong to a ward are aggregated together. We observe that remote parts of Knoydart and the Small Isles (colored in red) fare poorly, whereas adjacent ward above Knoydart has the best index as a result of high-speed wireless connections from the Tegola network. Wards on the Isle of Skye have intermediate index values as it mainly consists of ADSL users.

6.5 Discussion

Until very recently, geographical broadband mapping was little more than a term used by ISPs and mobile operators to identify the market research initiatives commissioned to specialised companies with the aim of studying Internet penetration and competitors

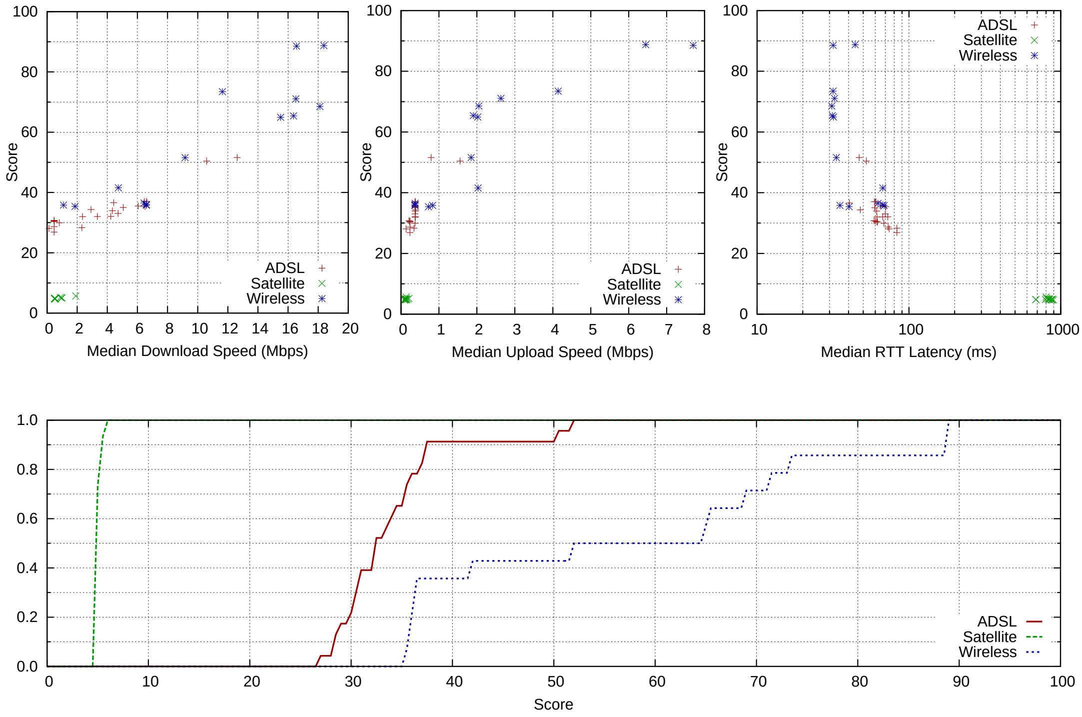


Figure 6.17: *Distribution of the broadband quality scores obtained on the* `BSense` *test deployments around the Isle of Skye, UK.*

share in a given region. The pitch is now changing as an increasing number of public bodies are entering the game. Governmental agencies and local administrations are more familiar with Geographical Information Systems (GISs) than in the past, and knowledge about local broadband availability is considered an important asset.

The main drivers for this change are the increasing need for market regulation and to efficiently pilot financial incentives. For example, in many countries licenses for portions of the wireless spectrum have been awarded to operators with a clause requiring the recipient to cover a minimum region over a period of time. Also, many states are discussing whether broadband access should be considered a “universal service”, providing subsides to regions underserved by the free market, which requires accurate analysis of the market offer currently available.

While broadband mapping is gathering momentum, obtaining a detailed picture is much more complex than in the past. New access technologies, both in the wired

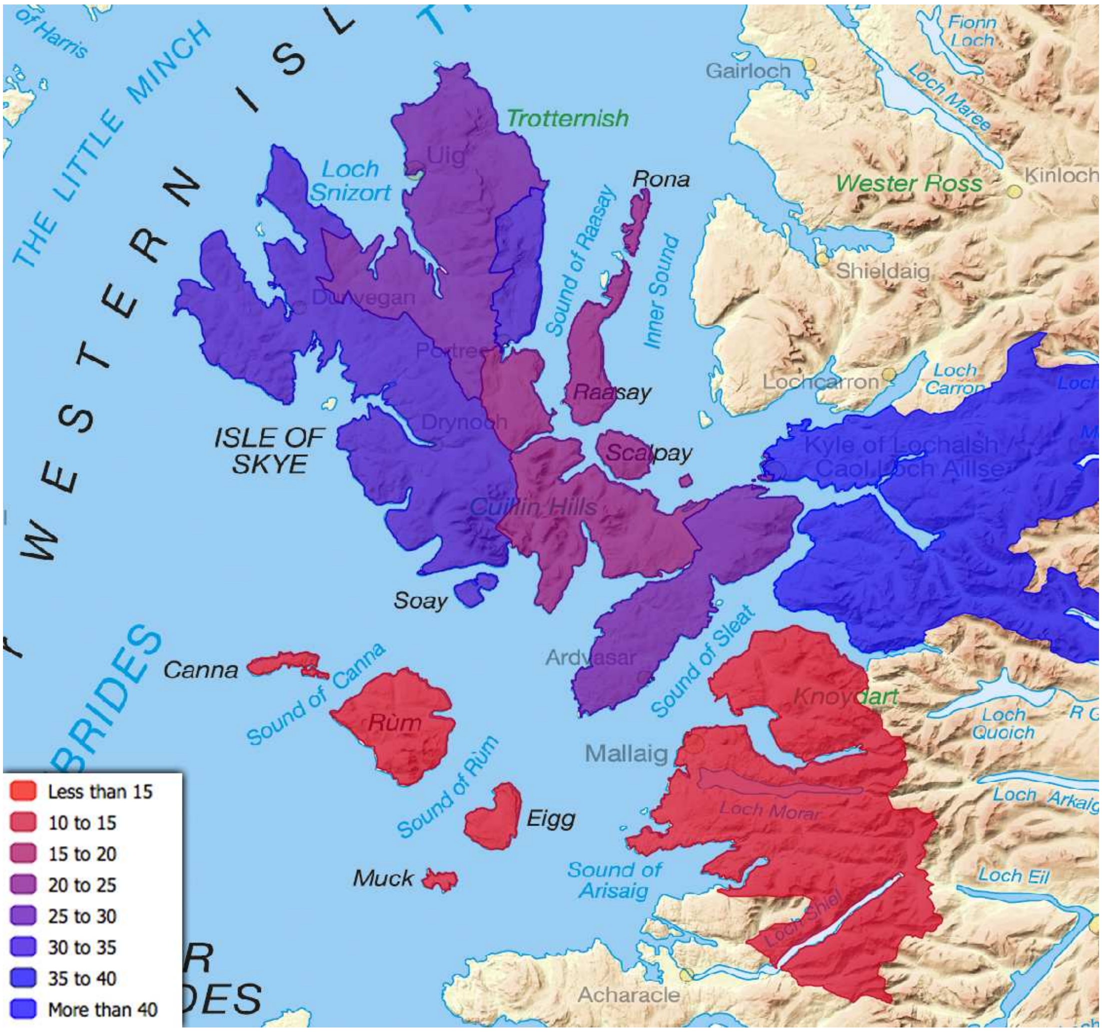


Figure 6.18: *Map generated using the results from the pilot study aggregating users belonging to a ward together and colors representing different ranges for broadband quality index values.*

domain (e.g., FTTB/FTTH roll outs, mixed FTTC+VDSL deployments, new Passive Optical Networks, the latest DOCSIS developments for cable networks, etc.) and in the wireless domain (e.g., LTE, WiMax, etc.) broadly differ as network topologies and for physical level propagation. They are also driven by conflicting economic models, charged for in mismatching ways and scale differently with the number of customers.

Finally, the number of players in the market is constantly increasing with the current expansion of WISPs, most of which operate in the unlicensed spectrum, the proliferation of 3G data plans, new community deployments and municipal fibre initiatives.

This fragmentation and the lack of incentives for the stakeholders to participate in a mapping effort have traditionally been show stoppers for mapping initiatives.

It is for these reasons that we developed `BSense`: we aimed to provide the open-source basis for broadband mapping software anyone can adopt. The current implementation has allowed us to perform a case study of the quality and availability of broadband connections in a given geographical region, and has been used to produce a study which we released to the Government and the local regional development agencies. The `BSense` framework is generic enough to be implemented anywhere, as it does not make assumptions about specific geographical organisation, and has been shown to be sufficiently stable during the launch of the public broadband test.

6.6 Summary

In this chapter, we have considered the problem of broadband mapping, which is fast becoming very important in view of the quest for universal broadband and bridging the broadband digital divide in many parts of the world. We have a developed a flexible software broadband census system called `BSense` based on open source software. We have also discussed how the `BSense` framework can contribute towards generating comprehensive and reliable broadband maps. Compared to existing mapping approaches, the `BSense` framework is more general and we also discuss how various stakeholders (including policy makers and regulators) can be incentivized to contribute to the mapping exercise. The `BSense` framework also incorporates a flexible specification of broadband quality index based on utility functions and multi-attribute utility theory. We implemented `BSense` using open-source tools and use it to demonstrate the value of the `BSense` approach for broadband coverage and quality assessment with two case studies.

`BSense` is unique in that it is a distributed architecture based on multi-platform measurement agents, which can be customised to measure specific network features. The methodology adopted can evaluate how well broadband connections will perform with a given set of IP traffic traces, and produces a ‘broadband score’ as the output of a set of utility functions.

Chapter 7

Other Contributions

In this chapter we present a set of secondary contributions that lie outside the main area of interest of this thesis. We introduce the difficulties of establishing *long distance over-water links* and *power planning* of transmission towers in remote locations. These two issues represent difficult challenges in deploying wireless networks in rural regions. This section presents our initial attempts at understanding these challenges.

Our wireless testbed network is unique in two key respects. First, our environment necessitates radio propagation across long distances over sea water, which has a significant impact on magnitude and variability of the observed received signal strength characteristics because of multi-path reflections off changing water levels, including due to tidal patterns, and signal attenuation due to water absorption. Consequently, the channel and link characteristics from our measurements are in sharp contrast to the conclusions made about rural long distance 802.11 links in the recent literature (Sheth et al., 2007). Second, self-powered wireless masts in our testbed deployed on mountains to obtain line of sight connectivity are powered by a combination of different sources (wind and solar). This is different from other existing deployments, which are situated in areas with tropical climates with plenty of solar radiation and rely solely on solar power. We find that exploiting a diversity of energy sources can significantly cut down the cost and size of the power system (that includes power generators and batteries).

7.1 Planning of Long Distance Over-water Links

In this section, we focus on a very specific yet common scenario: point-to-point wireless links that are deployed at low altitudes over the sea. It is estimated that 90% of the population of the the Scottish Highlands and Islands live in coastal villages: such a scenario then plays a key role when planning rural BWA networks.

Water (especially sea water) presents a very high reflection coefficient, which can present a major obstacle to wireless communications, and in turn to reliable Internet service delivery. We look at the impact of radio propagation over sea water on channel and link characteristics using the longest link in the testbed between nodes *S* and *B* (19Km long, see Figure 3.1).

We begin by looking at the feasibility test and link budget calculation for the link *S* to *B* using the Radio Mobile application. Inputs to this tool are input transmit power, transmit and receive antenna gains, cable losses at sender and receiver, and receiver sensitivity at the desired rate. Given that we had radio hardware that supported high transmit power (600mW) and antennas with high gain (29dBi), we had to lower transmit power to 10dBm in order to comply with the regulations (4W EIRP limit). We assumed cable losses to be a reasonable value of 3dB at each end. Since we are interested in the feasibility of operating the link at the highest rate, we used the receive sensitivity of -74dBm corresponding to 54Mbps obtained from the XR5 card data sheet. Note that Radio Mobile automatically calculates the path loss using the supplied terrain data and relevant climate settings using the Longley-Rice propagation model. With the above information, we predicted a received signal level of -69.7dBm, a link margin around 4dB which is somewhat lower than the typically desired margin of 10-20dB. The snapshot of the output is shown in Figure 7.1.

Our next step was to study the behaviour of the measured receive signal strength value over time for this same *S* to *B* link along with the achievable link capacity seen at the higher protocol layers. For fine-grained sampling of the signal strength at the receiver, we used ping traffic between sender and receiver every 500ms. In addition, we used the widely used `pathrate` application[1] every two minutes in the *S* to *B* direction to estimate the average link capacity. Measurement results are shown in Figure 7.2. Note that the light red lines in Figure 7.2 correspond to the receive sensitivity values

[1]Pathrate: A Measurement Tool for the Capacity of Network Paths, `http://www.cc.gatech.edu/fac/Constantinos.Dovrolis/pathrate.html`

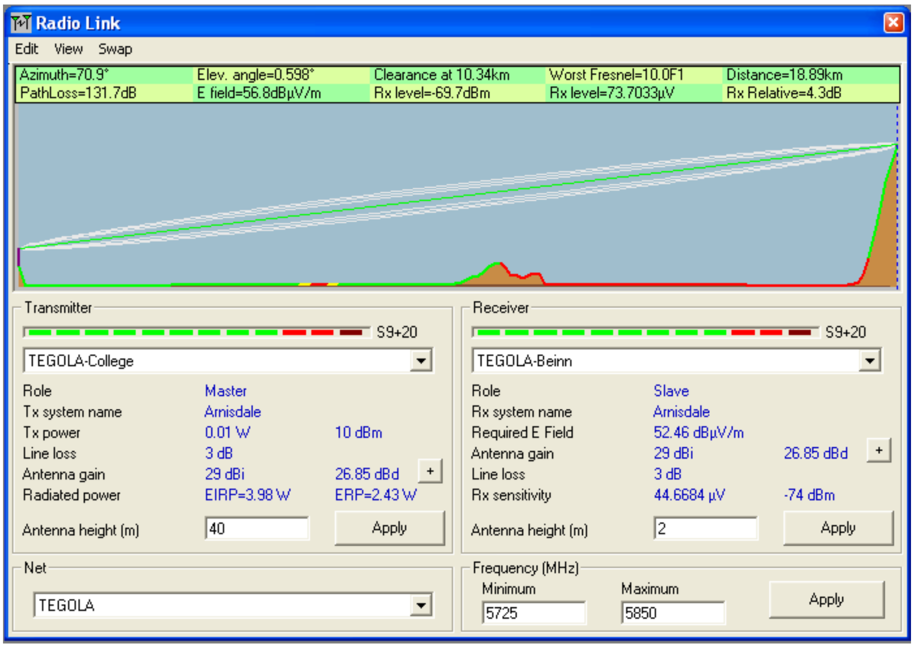


Figure 7.1: *Radio Mobile output for the link S to B in the testbed with a transmit power of 10dBm, transmit/receive antenna gains of 29dBi and 3dB cable loss at each end.*

at different rates as specified in the XR5 card data sheet. Looking at this data, we can make two main observations: (1) While the mean RSSI value from the measurements is close to the predicted value obtained from Radio Mobile, instantaneous RSSI value exhibits significant fluctuations by as much as 20dB in a short span of 1-2 hour period. Such fluctuations are much higher than anything reported in the literature for rural long distance 802.11 links (Gokhale et al., 2008; Sheth et al., 2007). We should mention here that we also monitored the signal strength variations for the other link between nodes *B* and *C* that is relatively shorter and over the land. While RSSI is much higher, as expected, we also noticed that the variations are quite small within a 4dB range and mostly within a 2dB range. (2) Changes in average link capacity estimates are fairly well correlated with the signal strength variations with the capacity dropping by more than half during periods with low signal levels.

The water level changes in the Sound of Sleat according to tidal patterns are a major factor behind the large signal strength variations. In fact, tide heights can vary by as much as 7 meters in a six hour period. While a direct comparison of signal strength

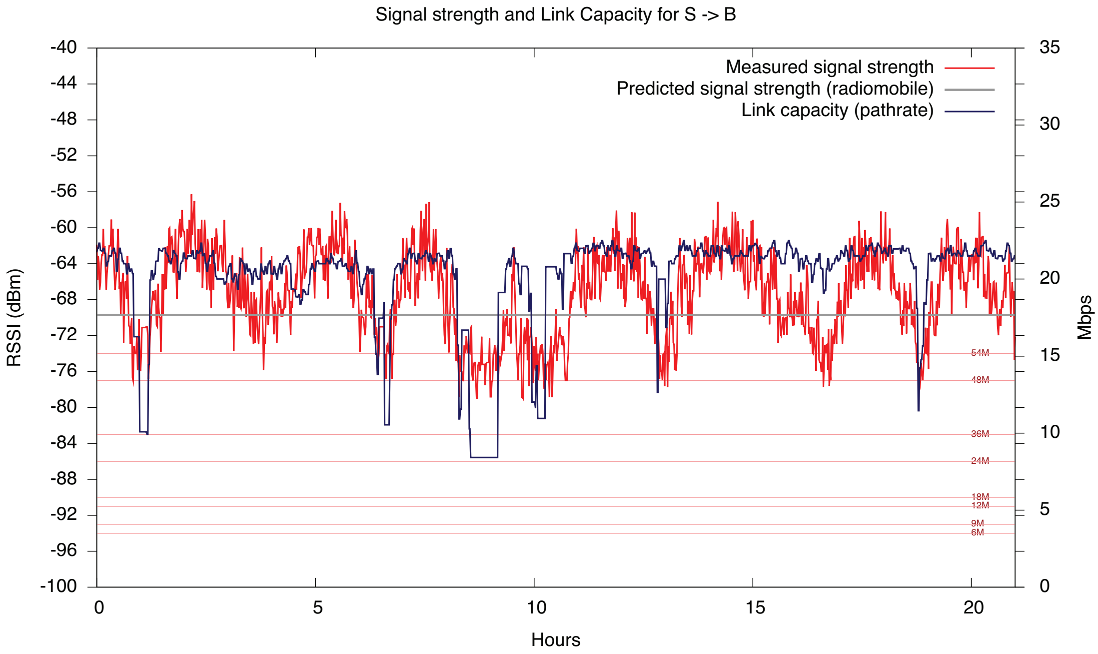


Figure 7.2: *Measured signal strength and link capacity values for the link S to B over a 24 hour period.*

measurements with the tidal level variation data obtained for the same period does not suggest any correlation, a simple two ray reflection model to predict the received signal strength with varying tide levels with identical transmitter and receiver antenna heights as in the testbed does tend to confirm our hypothesis (see Figure 7.3). This model essentially determines the instantaneous difference in length between the direct path and the path of the reflected wave over sea water based on the tide level at that particular instant. Difference in path lengths in turn leads to amplitude and phase changes of the receive signal level. We should mention, however, that other factors not accounted in the current model such as atmospheric pressure may also contribute to the RSSI variations.

As signal strength fluctuates significantly at the timescales of an hour because of tide level variations, the effective link capacity varies and it may potentially disrupt performance and service availability to the users. The next natural question is to explore possible remedies to overcome such signal degradations. By experimenting on the Tegola network, Macmillan et al. (2010) analyse various possible approaches. The first and most obvious is to exploit antenna diversity, by having multiple antennas in-

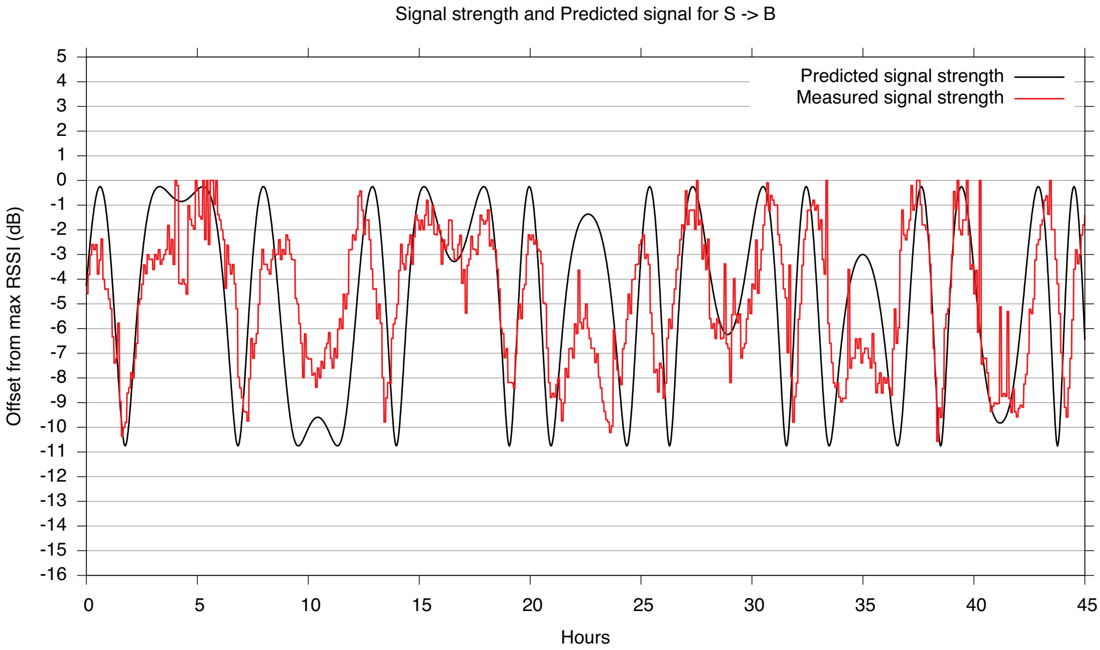


Figure 7.3: *Comparison of measured signal strength values with the values predicted by a simple two-ray reflection model taking into account water level variations over a 45 hour period.*

stalled on the same mast at different height and switching between them depending on the current tidal level. However, such an approach is expensive and complex. Instead, the paper suggest a ‘slow frequency hopping’ technique, which can be implemented in software-only and consists in slowly varying the transmission frequency to exploit the frequency dependence of the two-ray interference pattern, by switching to a frequency that is not currently experiencing fading.

7.2 Power Planning for Self-Powered Wireless Masts

In this section, we look into the question of designing a low cost and small size powering system for self-powered masts, as two backhaul nodes in our testbed (nodes *B* and *I*) are far from the electric grid. We use the mast at node *B* as an illustrative example here to compare two approaches: (i) relying solely on solar power as done in every other deployment; (ii) combining different power sources, such as solar and wind.

To begin with, we start by looking at the power demand of the Gateworks board at node B. Based on the board's data sheet, it consumes 5W without any radios and can take up to an additional 18W to power the 4 mini-PCI sockets, for a maximum power consumption of 23W. Since all four slots are used at the moment — two 802.11a cards for the dual-polarization links between S and B, one 802.11a card for the link between B and C and one 802.11b/g card for local access at B — the power consumption of the board is indeed 23W. For always-on operation, the total energy demand per day is then 552 Watt-hours (Wh).

Suppose that with both approaches we want to ensure always-on operation and continue running up to n "powerless" days. This would mean that in the case of a solar-only approach, the mast will keep functioning for up to n consecutive sunless days. The value of n essentially determines the required amount of battery storage capacity. Given the above per-day energy demand, the total energy demand over a n-day period is $552 * n$ W/h. Assuming the nominal voltage of the board is 12V DC[2], we need $552 * n/12$ Ampere/hour (Ah) of useful battery capacity, corresponding to $46 * n$ Ah. With the two Elecsol 125 Ah batteries we have operating at or below 80% depth of discharge (to ensure they last their full rated lifetime of 1,000 deep discharge cycles), we obtain a useful battery capacity of 200 Ah overall as opposed to the nominal capacity of 250 Ah. With 200 Ah total battery capacity, the node B system can continue to run for $n = 200/46 = 4.35$ powerless days, which is a reasonably sufficient buffer.

Now we consider the first approach of relying solely on solar power. Here we essentially want to determine the number of solar panels required to continuously power a 23W load. But this in turns depends on the solar irradiation data for the worst month. Using the exact GPS coordinates of the node B mast location and the PVGIS solar irradiance data utility[3], we obtained the average irradiation ($Wh/m^2/day$) for each of the twelve months assuming panels are mounted at optimal inclination angle. The worst month for our mast site turns out to be December with an irradiation of $520Wh/m^2/day$. Using the same PVGIS utility, we have obtained the estimated energy production per month for the panel type we have (crystalline silicon with peak power output of 130W) to be 2 KWh over the whole month of December (the worst month). Thus to meet our energy demand of 552 Wh/day (or 17.1 KWh/month), we would need around 8 more such panels or a total of nine 130W panels. With each panel

[2]Note that the Gateworks board can support anywhere from 9-48V DC
[3]PVGIS Solar Irradiance Data Utility, `http://sunbird.jrc.it/pvgis/`

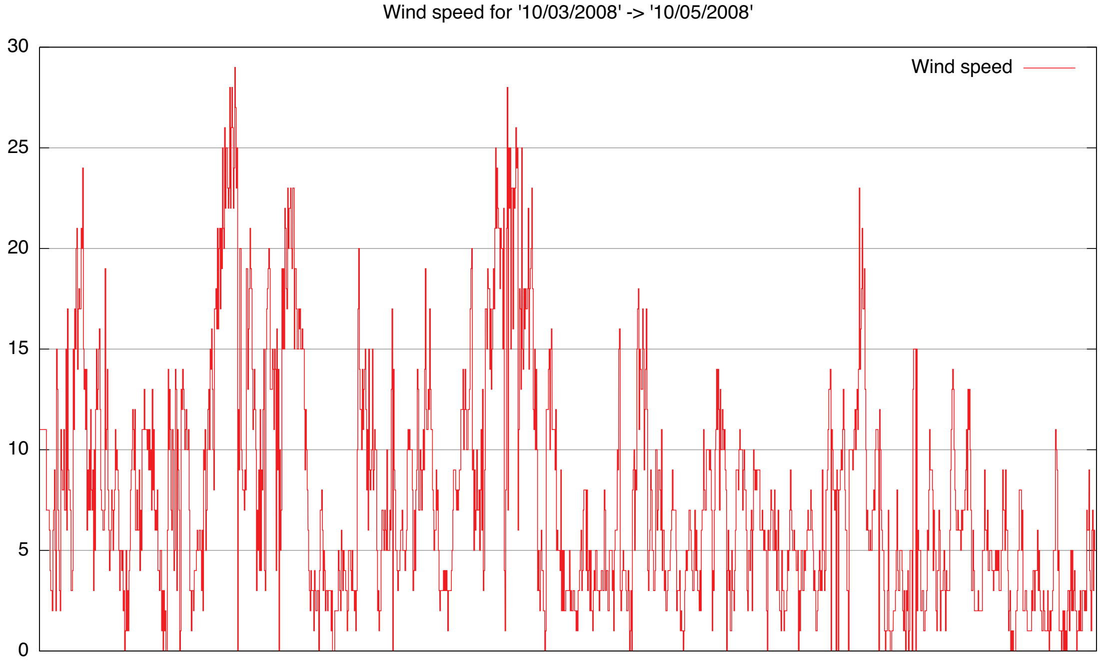


Figure 7.4: *Measured wind speed at the nearest weather station for the period 10 March, 2008 to 10 May, 2008.*

costing around 450 GBP, the total cost of the generation part of the system would be in excess of 4,000 GBP[4].

Let us now turn to the other approach of relying on a combination of energy sources. Specifically, let us consider wind as it is a plentiful resource in regions such as those of our testbed. Wind power generated depends on the wind speed. To get an estimate of the expected wind speed at our mast location, we used the DECC's wind speed database[5], which is fairly accurate for rural areas. We have extrapolated the output from this database to account for the greater height of our mast location (which is 300m above sea level) to get an annual mean wind speed estimate of 9m/s. This would result in 5A current generation to a 12V battery with our Rutland Furlmatic 910-3 wind generator based on the specification data. We can expect about 48W power generation (assuming 20% loss), which is more than sufficient by itself to power the Gateworks board. However, our calculations assume that wind speed remains constant throughout

[4] This costing, however, does not include the components of the powering system common to both cases such as batteries and charge controller, which add a further 400GBP.

[5] UK Department of Energy and Climate Change, Wind Speed Database, `http://www.decc.gov.uk/en/content/cms/meeting_energy/wind/windsp_databas/windsp_databas.aspx`

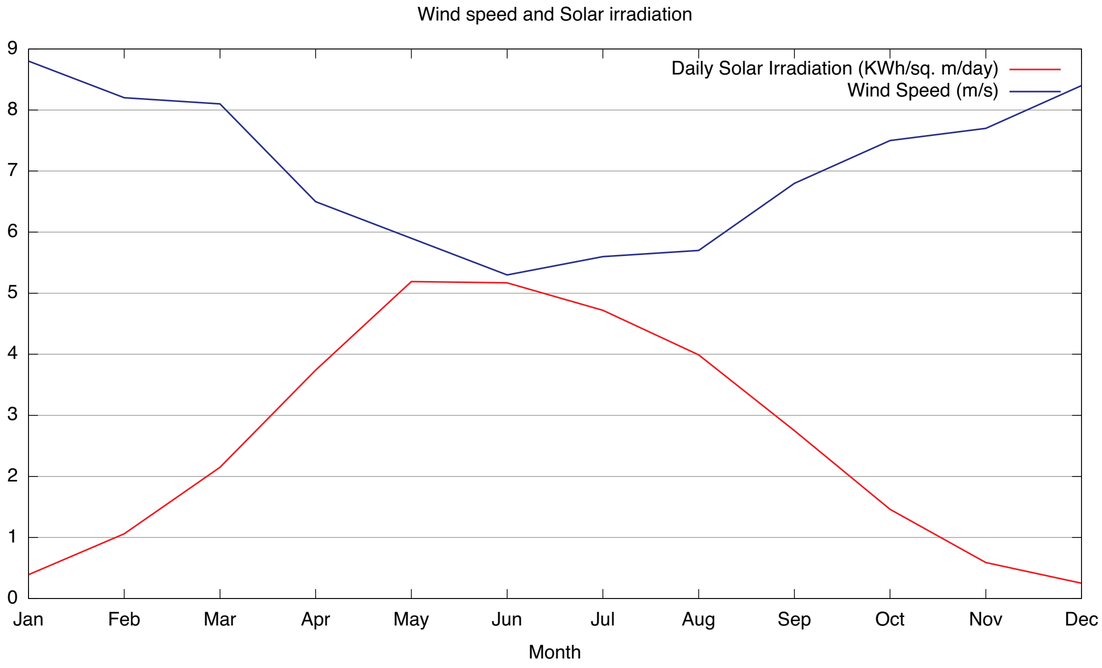


Figure 7.5: *Historical monthly data for wind speed and solar irradiation, obtained from RETScreen climate database.*

the year which is far from the truth. Looking at the measured wind speed data over a two month period at the nearest weather station (see Figure 7.4) shows considerable variability both within a day and across days, ranging from zero to 30m/s, with mean of 7.68m/s. This data also suggests the need for greater battery storage capacity for absorbing the spikes in power generation. Given the above discussion, the question is whether we would additionally need solar power to continuously power our load. To answer this question, we use the historical wind speed and solar irradiation data obtained from a climate database[6] for Portree on the Isle of Skye. This data is presented in Figure 7.5 and shows good agreement with the recent measured wind speeds from the nearest weather station. This data clearly shows that solar and wind energy sources nicely complement each other, with wind power generation peaking in the winter months and solar power peaking in the summer months. Using data sheets for our wind generator and solar panel along with this data, we in fact conclude that we can indeed meet our energy demand of about 0.5KWh/day throughout the year with just one solar panel and one wind generator. Given that both cost about the same, this

[6] RETScreen Climate Database, `http://www.retscreen.net/`

gives a total cost of 900 GBP for the generation part of the powering system which is less than a fourth of the cost compared to the previously discussed solar-only option. The main message from the foregoing discussion is that exploiting a diversity of energy sources can significantly reduce the cost and size of the powering system in self-powered masts.

Even though we have given careful consideration to power planning for the self-powered masts in our testbed, the powering system is still the single most expensive part of the hardware at such masts suggesting that there is motivation for further reducing the cost of the system. Some of the research project spun off from the Tegola deployment (presented in Appendix B.1) has in fact been on power saving techniques for wireless communication. Also the above discussion is based primarily on historical and model data. While such data is quite useful to guide system size planning decisions, it can still deviate from reality, so strictly following this data leaves room for the possibility of occasional power related outages. As a result, we added power and weather monitoring capabilities to our self-powered masts.

Chapter 8

Conclusions

8.1 Summary of Contributions

In this thesis, I focused on three key aspects concerning deployment and operation of Rural Broadband Wireless Access Networks. Looking back at the challenges raised at the beginning of this thesis in Section 1.3, I believe that our contributions can be summarised as follows:

1. **Network Design and Growth.** Through the `IncrEase` tool (Chapter 4), we showed that sustainable long-term strategies for the expansion of BWA networks can be determined by analysing a diversified set of operational metrics and statistics. Our tool uses "heatmaps" to direct the network operator to the areas that would benefit the most by an improvement in coverage. It provides two operational modes: (1) a semi-automated procedure that requires the operator to select a region from a map in order to calculate the best routes to any already-deployed transmission mast; (2) a fully automated mode that relies on a search algorithm to determine the "best" actions (i.e., deployment of new masts) that the WISP should take during the next interval of time.

2. **Network Management.** In Chapter 5 we have introduced `Stix`: a platform that eases the management of BWA networks, especially those operated by community organisations and rural WISPs. Using sample use cases, we showed how our visual event-based programming language (`StixL`) can be used to model network processes and goals using a web GUI. Heterogenous networks

can be effectively controlled by deploying vendor-independent hardware agents (`StixAgent`) at each transmission site. Moreover, we enabled the monitoring and control of remote equipment using low-cost embedded boards (`StixControl`), and implemented realtime reporting based on an overlay data storage network (`StixView`). Delegated operation, support for multi-vendor networks, and the lack of a central point of control make our system suitable for large-scale deployment, as demonstrated in our evaluation.

3. **Broadband Coverage and Quality Assessment.** In Chapter 6 we presented `BSense`, a flexible mapping framework for the generation of a 'census' of broadband availability, choice and quality. `BSense` combines estimated data provided by multiple ISPs with measurement based statistics obtained from a multi-platform software agent, which can be configured to perform a diverse set of network experiments. Moreover, `BSense` incorporates a broadband quality index that summarises the effect of various technical attributes (e.g., download/upload speeds and latency) on user's perceived broadband quality. ISPs and governments can take advantage of maps produced with `BSense` to make informed decisions about public policies and business strategies.

This research was guided by the experience we gained from deploying and operating the Tegola wireless testbed (Chapter 3) which, besides being an experimental rural wireless network, also provides broadband Internet access to one of the remotest regions of Britain. *Tegola* is still operational and has inspired other rural communities across Scotland to deploy similar networks (e.g., see Appendix B.2).

Tegola provided first-hand experience which helped me to identify the three key research challenges tackled in the thesis. However, when it comes to evaluating the solutions proposed, we exploited our collaboration with NGI SpA which allowed us to test our software tools on a larger network. In particular:

- For `IncrEase`, NGI's network is used exclusively, taking advantage of their database of over 8,900 tower locations to stress-test the algorithm implemented.
- The evaluation of `Stix` includes a use case based on NGI's access network and a realistic simulation on Tegola, where the management dongles have been deployed at each transmission site.

- BSense has been assessed with an experiment involving volunteers from the region where Tegola is deployed.

From an evaluation perspective, NGI and Tegola networks are complementary to each other (e.g., in size, technologies used, ownership, geographical extent) and allowed me to assess the solution proposed in this thesis from multiple viewpoints.

All the software developed in this thesis is made available as *open-source* (see Section 1.4) some of it is currently being used by NGI SpA, a real-world WISP, while other researchers and students are developing it further.

8.2 Suggestions for Future Work

In this section, I summarise future work opportunities in relation to the contributions of this thesis.

8.2.1 IncrEase: Incremental Network Planning

Although I believe that the current implementation of the `IncrEase` tool is sufficiently rich in terms of functionality and stable enough to be used by a WISP, `IncrEase` can foresee a number of improvements to generate more accurate network plans.

First, the current design does not account for redundancy in the core network topology (i.e., the backhaul tier): both the A* and Dijkstra algorithms search for the single lowest-cost route from a newly deployed set of masts to any of the existing ones. In real WISP networks, important transmission sites are often connected to the Internet using multiple independent paths or links. A necessary element of incorporating redundancy is associating a simple metric to each transmission site, e.g., the hop-count to the closest Internet uplink. Similarly, each link can be tagged with a value reflecting its likelihood of failure. When a site is reached by a multi-hop path, the likelihood of it becoming disconnected is equal to the product of the probabilities of each single link going down. A straightforward way to improve the connectivity for sites would be to look for opportunities to add redundant links in the topology generated using the current algorithms, i.e., addition of a link between two sites already deployed but without a link between a them.

Also, to determine whether two locations are within radio reach, the current implementation considers optical line-of-sight. While this is a reasonable approximation at high frequencies (e.g., in the unlicensed 5 GHz band or above) and long distances, more realistic propagation modelling would be needed to account for lower parts of the spectrum (e.g., 'TV white spaces' in the UHF/VHF bands).

`IncrEase` assumes that the data provided in input is static and does not change with time. We also assume that such conditions do not change as the network evolves and that visibility between two sites does not change over time. In reality, the environment changes during the network lifetime: new backhaul locations may become available; the number of towers and their associated rent may change; and the demand patterns may change.

Finally, as pre-existing transmission towers are often limited in rural regions, `IncrEase` could be enhanced to identify suitable locations (e.g., near roads, electricity available, etc.) for new transmission towers and to incorporate extra attributes in our mast cost model (e.g., power, tower height).

As a prototype implementation is now available for WISPs to use, it would be interesting to evaluate `IncrEase` by comparing between incremental and 'all-in-one' strategies over a long-term deployment period.

8.2.2 Stix: a Goal-Oriented Distributed Management System

The current implementation of `Stix` enabled us to start a wider evaluation on larger networks, both operated by WISPs and by community organisations. As of October 2011, `Stix` is being deployed on the NGI SpA network, which has grown to over 600 sites in less than 5 years, connecting over 5,000 managed devices (i.e., base stations, point-to-point radios, network equipment, etc.). Such a rapid deployment pace pushed their network team to optimise the management procedures and evaluate a distributed solution. By studying their management patterns, further use cases can be extrapolated and the usability of both `StixL` and `StixGui` could be improved. For this deployment, experiments are being made to optimise the `StixAgent` memory footprint to make it run on even less powerful, and cheaper, embedded boards. Additional device drivers are being implemented to support a wider set of management protocols and devices and expanding the Task library with useful, and commonly used, tasks.

Leaving the lab for the real world implies solving two other important issues: security and accounting. Widely used management platforms adopt very simple security mechanisms, often relying on simple username-password challenges in cleartext, as WISPs typically use a separate virtualised interface (e.g. 802.1q VLANs) for the network management traffic which is inaccessible to end users. However, security could be addressed in `Stix` in two directions. First, as "task" objects can be written by third parties and imported in the Task Library, the administrator needs a mechanism to enforce trust towards the developer. I suggest the implementation of a Public Key Infrastructure (PKI) scheme in which task developers cryptographically 'sign' their code, which is then verified on the `StixGui`. Such an approach is very popular, being used for example by software repositories (such as the Linux `apt` utility) and by mobile applications markets (as the Apple Iphone Apps[1] and Google Android Marketplace[2]). I also recommend maintaining a "who has done what" accounting log, in which the system keeps track of the workflows designed by each administrator and, subsequently, the actions performed by each workflow. Such an approach is important to solve the deniability problem in a multi-tenant network, that is the possibility of an administrator performing malicious actions on a network that belongs to multiple organisations.

I believe that the introduction of the "network processes" concept and the corresponding workflow modelling notation could be especially beneficial for non-technical staff. However, giving more control is often a threat to network stability, which suggests further research on two important aspects: *automatic workflow validation* performed at design time, in which we check whether a new workflow may somehow harm the network, and *simulation* as a way to run new workflows in a virtualised environment to predict their behaviour once they are deployed on the real network. Simulation could be performed by adding a "dry run" option in the `StixAgent` language interpret that exploits Copy-on-Write (CoW) techniques to allows read access to all the variables, which are then copied to a temporary memory and modified when write-access is performed. To simulate the addition of new devices and sites in the network, an option could be added in the topology GUI (see screenshot in Figure 5.4(a)) to add "virtual" devices.

[1] `http://www.apple.com/iphone/apps-for-iphone/`

[2] `http://www.android.com/market/`

8.2.3 BSense: A Flexible System for Broadband Mapping

`BSense` was used to gather broadband statistics in our Isle of Skye case study (Section 6.4.2) and to generate ‘notspots’ maps for the Digital Scotland report (Section 6.4.1) presented to the Scottish Parliament. The `BSense` framework is generic and flexible enough to be applied anywhere, as it does not make assumptions about particular Internet access technologies or geographical constraints. For this reason we would like to see organisations, such as NGOs and policy makers, adopt our mapping platform in other regions.

Although in our evaluation and case studies network experiments were performed from a single test server, `BSense` is designed to use multiple servers, which are selected at run-time in a plain round-robin fashion. Further research is needed to design more elaborate load-balancing strategies that select between different available servers while removing measurement bias due to a particular server location.

As a further improvement to our broadband quality index, I suggest the introduction of a statistical dispersion metric to capture the variability in measured statistics.

Appendix A

Acronyms and Abbreviations

In this document, we use several acronyms. For ease of reading, they are summarised in the following table.

ADSL *Asymmetric digital subscriber line*, a data communications technology that enables broadband connectivity (up to 24Mbps) over traditional copper telephone lines.

AP *Access Point*, any generic device that has both wired and wireless capabilities.

BTS *Base Transmitting Station*, a device located on a transmitting site which is part of the service provider infrastructure.

BWA *Broadband Wireless Access*, the provisioning of high-speed wireless Internet access using wireless technologies over a wide area.

CPE *Customer Premises Equipment*, a device housed at a user's premises that act as a network endpoint.

DEM *Digital Elevation Model*, a numerical model that describes the altitude of a portion of terrain.

GIS *Geographic Information System*, a software system designed to store, analyse and present geographically referenced data.

ISP *Internet Service Provider*, an organisation that offers access to the Internet.

LOS *Line of Sight*, the condition in which two locations are in view of each other.

NMS *Network Management System*, is any hardware or software framework that is used to configure, monitor and administer a network.

PMP *Point to Multi-point*, a network topology in which multiple endpoints are connected via a central device. Example: a connection from a BTS to a set of CPEs.

PTP *Point to Point*, a network link between exactly two endpoints. Example: the connection between two BTSs.

RFC *Request for Comments*, are memoranda published by the Internet Engineering Task Force (IETF) which describe network protocols and standards.

VPL *Visual Programming Language*, a programming paradigm where programs are created by manipulating elements graphically rather than by specifying textual constructs.

WISP *Wireless Internet Service Provider*, an ISP that uses mostly or exclusively wireless connections to provide user access to the Internet.

Appendix B

Tegola Network Impact

The Tegola deployment has been a stimulus for further research projects on topics related to over-water wireless propagation, power saving schemes and channel-width adaptation. Also, it acted as a catalyst for other communities in Scotland to deploy similar access networks. The following is a brief note about such efforts.

B.1 Other research studies on the Tegola network

Besides the issues of automated wireless network planning, management tools and broadband mapping discussed in this thesis, the Tegola project over the years spurred other researchers to look into other related technical issues relevant to rural broadband wireless access. The following is a selection of publications and reports from such related research[1].

Over water radio propagation:

- Alex Macmillan, Mahesh K. Marina, Jhair Tocancipa Triana. *Slow Frequency Hopping for Mitigating Tidal Fading on Rural Long Distance Over-Water Wireless Links*. In Proc. INFOCOM IEEE Conference on Computer Communications Workshops, March 2010.

[1]Edinburgh University Tech Reports and M.Sc. theses are available online at: `http://www.inf.ed.ac.uk/publications/thesis/msc.html`

Power saving techniques:

- Veljko Pejovic, Elizabeth Belding, Mahesh K. Marina. *An Energy-Flow Model for Self-Powered Routers in Rural Mesh Networks and its Application for Energy-Aware Routing*. In Proc. ACM SOSP 2009 Workshop on Networked Systems for Developing Regions (NSDR'09), October 2009.

- Daniel Tyrode. *On the Predictability of Traffic and Energy Resource Availability for Self-Powered Wireless Network Routers.* Edinburgh University, MSc. Thesis in Informatics, August 2011.

Channel width adaptation:

- Jhair Tocancipa Triana. *Impact of Channel Width on the Performance of Long-distance 802.11 Wireless Links*. Edinburgh University, MSc. Thesis in Informatics, August 2009.

- Theodoros Koletsos. *Modeling Channel Width Adaptation in Rural Wireless Mesh Networks*. Edinburgh University, MSc. Thesis in Informatics, August 2010.

GIS systems for wireless planning:

- Thomas Nelson. *Wirless Mast Place Application for the Scottish Highlands*. Edinburgh University, M.Sc Thesis in Geographical Information Science, August 2008.

B.2 The 'Small Isles' community network

The success of Tegola, covered by the media on various occasions, motivated other rural communities across Scotland to get in touch with us and deploy similar networks to provide Internet access to the local residents.

A notable example is the *Small Isles community network*[2], a joint project involving the communities in the south parth of Skye and the Knoydart peninsula, aimed at

[2] http://www.knoydart-foundation.com/our-work/projects/community-broadband-access/

providing broadband Internet access for the area in and around the Small Isles, a small archipelago of islands in the Inner Hebrides (the four main ones being Canna, Rum, Eigg and Muck), the Sleat Peninsula, Loch Nevis and Loch Hourn.

The project started in early 2010 and, as of October 2011, already reaches around 40 families on the Isles of Eigg and Muck. The project is in expansion, and the tiny Isles of Rum and Canna are being connected. The Internet backhauling point is located on the British mainland at Arisaig, where four ‘business’ ADSL lines have been installed. All the wireless hardware is from Ubiquiti Networks, and operates on the 5 GHz unlicensed spectrum.

After we trained people from the local communities in planning wireless links and installing networking equipment, they completely took over the management of the community network. Users share the connectivity and hardware costs, and the network is operated as no-profit.

Communities in other parts of Scotland are following our lead. The *Digital Scotland* report[3] was published by The Royal Society of Edinburgh and presented to the Scottish Parliament, with the aim of helping rural communities who deploy their own broadband networks with backhaul access to public Internet, by producing an appropriate technical and regulatory framework.

[3] Available at: `http://idea.ed.ac.uk/Digital-Scotland.pdf`

Appendix C

StixL – Grammar

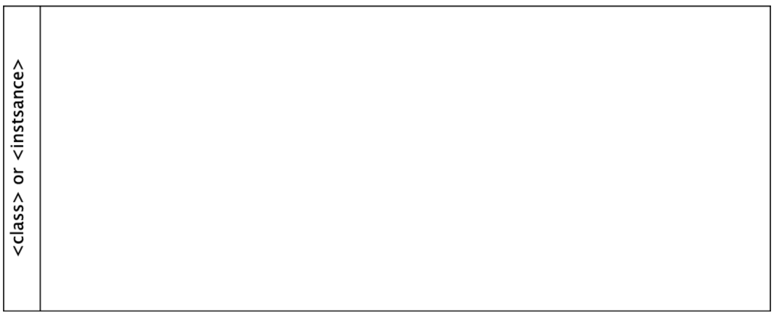
This Appendix formalises the visual grammar of `StixL` for modelling the network management processes. A `StixL` workflow is made up of a set of graphical elements which definition is inspired by the BPMN standard.

C.1 `StixL` Elements

C.1.1 Pool

It acts as a visual container for all the entities belonging to the same workflow. The vertical space on the left hand contains a query expression, which enables the selection of the devices to which the workflow will be applied.

Graphical representation:



Example of XML representation:

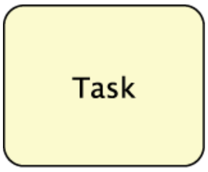
```
<pool query="ON BTS DO" x="100" y="300" ID="stix-100" >
  ...
```

```
</pool>
```

C.1.2 Task

A task is an individual unit of work.

Graphical representation:



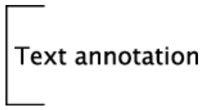
Example of XML representation:

```
<task class="uk.ac.ed.inf.wimo.stix.doSomething" ID="stix-101"
    x="100" y="300">
  <flowMapping>
    <out>stix-100</out>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>sampleParameterOne</name>
        <attribute>var1</attribute>
      </parameter>
      <parameter>
        <name>sampleParameterTwo</name>
        <value>1122334455</value>
      </parameter>
    </in>
    <out/>
  </dataMapping>
</task>
```

C.1.3 Text annotation

Any object can be associated with a text comment, to clarify its use or to provide additional documentation.

Graphical representation:



Example of XML representation:

```
<...>
  <annotation>This is a comment</annotation>
</...>
```

C.1.4 Exclusive Gateway

It routes the sequence flow to exactly one of the outgoing branches based on a boolean test. When merging, it awaits one incoming branch to complete before triggering the outgoing flow. It is the equivalent of an “if ... then ... else ...” instruction in a traditional programming language.

Graphical representation:

Example of XML representation:

```
<exclusiveGateway ID="stix-100">
  <flowMapping>
    <out condition="x==1">stix-101</out>
    <out>stix-102</out> <!-- this is the else branch -->
  </flowMapping>
</exclusiveGateway>
```

C.1.5 Inclusive Gateway

When splitting, one or more branches are activated based on branching other conditions evaluate to conditions. When merging, it awaits all active incoming branches to complete. It is the equivalent of a series of individual “if ... then ...” on each of the outgoing branches.

Graphical representation:

Example of XML representation:

```
<inclusiveGateway ID="stix-100">
  <flowMapping>
    <out condition="x==1">stix-101</out>
    <out condition="x==2">stix-102</out>
  </flowMapping>
</inclusiveGateway>
```

C.1.6 Parallel Gateway

When used to split the sequence flow, all outgoing branches are activated simultaneously. When merging parallel branches it waits for all incoming branches to complete before triggering the outgoing flow. It is the equivalent of a "fork" operation in traditional programming languages.

Graphical representation:

Example of XML representation:

```
<parallelGateway ID="stix-100">
  <flowMapping>
    <out>stix-101</out>
    <out>stix-102</out>
  </flowMapping>
</parallelGateway>
```

C.1.7 Start Messaging Event

It enables the execution of a workflow to be triggered by an incoming message.

Graphical representation:

Example of XML representation:

```
<startMessagingEvent ID="stix-100" message="wakeUp">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
</startMessagingEvent>
```

C.1.8 Start Timer Event

It enables the execution of a workflow to be scheduled at unique or periodic intervals in time. The event caption is a string that takes the form *'every PERIOD from STARTDATE'*, meaning that the event will be triggered every *PERIOD* seconds starting from the *STARTDATE*.

Graphical representation:

Example of XML representation:

```
<startTimerEvent ID="stix-100" timer="2012-05-30T09:00:00"
   period="100">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
</startTimerEvent>
```

C.1.9 Start Condition Event

It enables the execution of a workflow to be started when the specified condition becomes true.

Graphical representation:

Example of XML representation:

```
<startConditionEvent ID="stix-100" condition="config.ath1.snr
   <=10">
```

```
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
</startConditionEvent>
```

C.1.10 Message Throwing Event

It generates and sends a message to another workflow. As soon as the message has been sent and acknowledged, the execution flow resumes.

Graphical representation:

Example of XML representation:

```
<messageThrow ID="stix-100">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>messageName</name>
        <value>wakeUp</value>
      </parameter>
      <parameter>
        <name>messageContent</name>
        <value>this is the content</value>
      </parameter>
      <parameter>
        <name>destination</name>
        <value>10.10.10.10</value>
      </parameter>
    </in>
    <out/>
  </dataMapping>
</messageThrow>
```

C.1.11 Message Catching Event

It receives a message sent from another workflow. It is always blocking, in other words the execution flow stops until a message is received.

Graphical representation:

Example of XML representation:

```
<messageCatch ID="stix-100">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>messageName</name>
        <value>wakeUp</value>
      </parameter>
    </in>
    <out>
      <parameter>
        <name>messageContent</name>
        <attribute>var1</attribute>
      </parameter>
    </out>
  </dataMapping>
</messageCatch>
```

C.1.12 Timer Event

When the workflow reaches this object, the execution flow stops for the specified period of time.

Graphical representation:

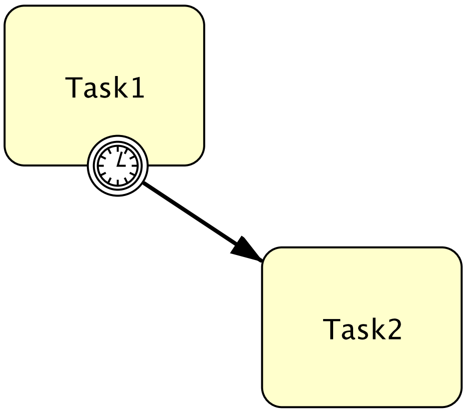
Example of XML representation:

```
<timer ID="stix-100" duration="3600">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
</timer>
```

C.1.13 Timeout event

By attaching this element to a task, it is possible to set a timeout value after which the execution of the task is aborted and the execution continues along this branch.

Graphical representation:



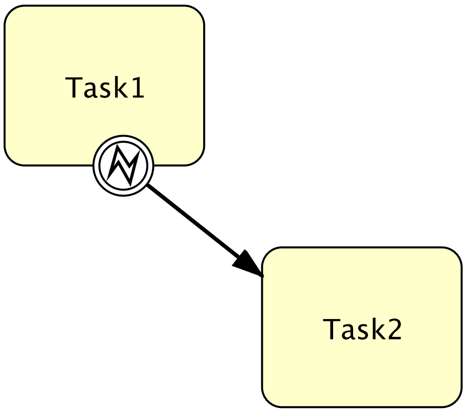
Example of XML representation:

```
<task class="uk.ac.ed.inf.wimo.stix.doSomething" ID="stix-101"
    x="100" y="300">
  <flowMapping>
    <out>stix-100</out>
    <timeout duration="3600">stix-101</timeout>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>coolParameterOne</name>
        <attribute>var1</attribute>
      </parameter>
      <parameter>
        <name>coolParameterTwo</name>
        <value>1122334455</value>
      </parameter>
    </in>
    <out/>
  </dataMapping>
</task>
```

C.1.14 Error event

When it is attached to a task object, it is used to route the execution to a separate flow in case of error.

Graphical representation:



Example of XML representation:

```
<task class="uk.ac.ed.inf.wimo.stix.doSomething" ID="stix-101"
    x="100" y="300">
  <flowMapping>
    <out>stix-100</out>
    <error>stix-101</error>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>coolParameterOne</name>
        <attribute>var1</attribute>
      </parameter>
      <parameter>
        <name>coolParameterTwo</name>
        <value>1122334455</value>
      </parameter>
    </in>
    <out/>
  </dataMapping>
</task>
```

C.1.15 End Plain Event

Denotes the end of a workflow. If there are multiple branches concurrently in execution, they must all reach an end event before the workflow will cease running.

Graphical representation:

Example of XML representation:

```
<endPlainEvent ID="stix-100" />
```

C.1.16 Terminate Event

When the execution flow reaches this elements, it immediately terminates, even if other branches are still executing.

Graphical representation:

Example of XML representation:

```
<terminateEvent ID="stix-100" />
```

C.1.17 Log Event

When the execution flow reaches this elements, a message is written to the log overlay. Anything can be a message: a numeric value representing performance metric, a textual log entry, etc. Storing messages is useful in order to gather informations about the status of the network and to produce reporting via the use of the `StixView` tool.

Graphical representation:

Example of XML representation:

```
<log ID="stix-100">
  <flowMapping>
    <out>stix-102</out>
  </flowMapping>
  <dataMapping>
    <in>
      <parameter>
        <name>message</name>
        <value>this is the message to be logged</value>
      </parameter>
    </in>
    <out/>
  </dataMapping>
</log>
```

C.1.18 Sequence Flow

Defines the execution order of the activities. Specific rules determine which objects can be linked together.

Graphical representation:

C.1.19 Message Flow

Symbolises information flow across pool boundaries, such as between two classes or instances[1].

Graphical representation:

C.2 `StixL` Query Syntax

Workflows are designed within a Pool element which, beside acting as a visual container, it also provides a way for specifying which devices will run the workflow. This

[1]This symbol is not implemented in the current software prototype.

is done by writing a query statement. The accepted query syntax can defined in ABNF format[2] as follows. Note that keywords are not case sensitive and may be written in any lettercase, although the use of uppercase is suggested.

```
query          = 'ON' deviceclass ['WHERE' conditionset] 'DO';
deviceclass    = 'CPE' / 'BTS' / 'DEVICE' / ...;
conditionset   = condition [( 'AND' / 'OR' ) condition];
condition      = ['NOT'] attribute test value;
test           = < / <= / = / >= / > / <>;
```

The device-class "CPE" and "BTS" identify the devices in those two categories while the class "DEVICE" matches all the devices regardless of their type. The 'attribute' field can be any of the internal parameters of the device, such as: model, region, firmware-version, etc.

Some explicatory examples are presented here.

Returns all the BTS devices:

```
ON BTS DO
```

Returns all the devices that are in the "Lothian" region:

```
ON DEVICE WHERE region = 'Lothian' DO
```

Returns all the CPEs that have firmware-version between 4 and 5:

```
ON CPE WHERE firmware-version > 4 AND firmware-version < 5 DO
```

C.3 XML Representation of Workflows

Workflows are created and edited in the Stix GUI. When the administrator issues a deploy command, the workflow is serialized in a XML representation and sent out to the network. The following is an example of the structure of the XML file:

```
<?xml version="1.0"?>
<workflow
  xmlns="http://www.wimo.inf.ed.ac.uk/stix"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.wimo.inf.ed.ac.uk/stix stix.
     xsd">

  <metadata>
    <name>Workflow example</name>
    <author>Author Name</author>
```

[2] Augmented Backus-Naur Form (ABNF) is defined in RFC 5234.

```
    <uuid>550e8400-e29b-41d4-a716-446655440000</uuid>
    <rev>43</rev>
    <notes>This is a workflow example</notes>
    <enabled>1</enabled>
    <validity>
      <notBefore>2012-05-30T09:00:00</notBefore>
      <notAfter>2012-05-20T09:00:00</notAfter>
    </validity>
  </metadata>

  <attributeSet>
    <attribute persistent="true">
      <type>int</type>
      <name>var1</name>
      <value>123</value>
    </attribute>
    <attribute>
      <type>String</type>
      <name>var2</name>
      <value>foo bar</value>
    </attribute>
  </attributeSet>

  <pool query="ON BTS DO" x="100" y="300" ID="stix-100" >

    <startMessagingEvent ID="stix-101" x="123" y="345">
      <flowMapping>
        <out>stix-102</out>
      </flowMapping>
    </startEvent>

    <task class="uk.ac.ed.inf.wimo.stix.doSomething" ID="stix
       -102" x="100" y="300">
      <flowMapping>
        <out>stix-103</out>
      </flowMapping>
      <dataMapping>
        <in>
          <parameter>
            <name>ParameterOne</name>
            <attribute>var1</attribute>
          </parameter>
          <parameter>
            <name>ParameterTwo</name>
            <value>1122334455</value>
          </parameter>
        </in>
        <out/>
      </dataMapping>
    </task>
```

```
    <endEvent ID="stix-103" x="123" y="345" />
  </pool>
</workflow>
```

The `<metadata>` block, contains the following general information about the workflows:

Field:	Type:	Required:	Description:
name	String(255)	Yes	Descriptive name
author	String(255)	No	Author of the workflow
uuid	RFC4122	Yes	Universally Unique Identifier (UUID)
rev	int	Yes	Revision number
notes	String(255)	No	Descriptive notes for the workflow
enabled	boolean	No	Whether the workflow is considered to be active or archived
StartingFrom	Date/time	No	Timestamp from which the workflow will become active
EndingOn	Date/time	No	Timestamp from which the workflow will be cease to be active

The `<attributeSet>` block is used to define the variables used in the workflow. Each `<attribute>` defines the variable type, its name and an optional initial value.

The remainder of the file XML is represented by the `<pool>` object and the elements defined in it. The XML is validated with an XML Schema before being sent out on the network and as soon as it is received by the agents.

Appendix D

StixGUI – Graphical User Interface

The network administrator interacts with the `Stix` management system using a web-based system interface called `StixGUI`. This Appendix explains the intended interaction mechanisms and presents mock-ups of the dialogue windows.

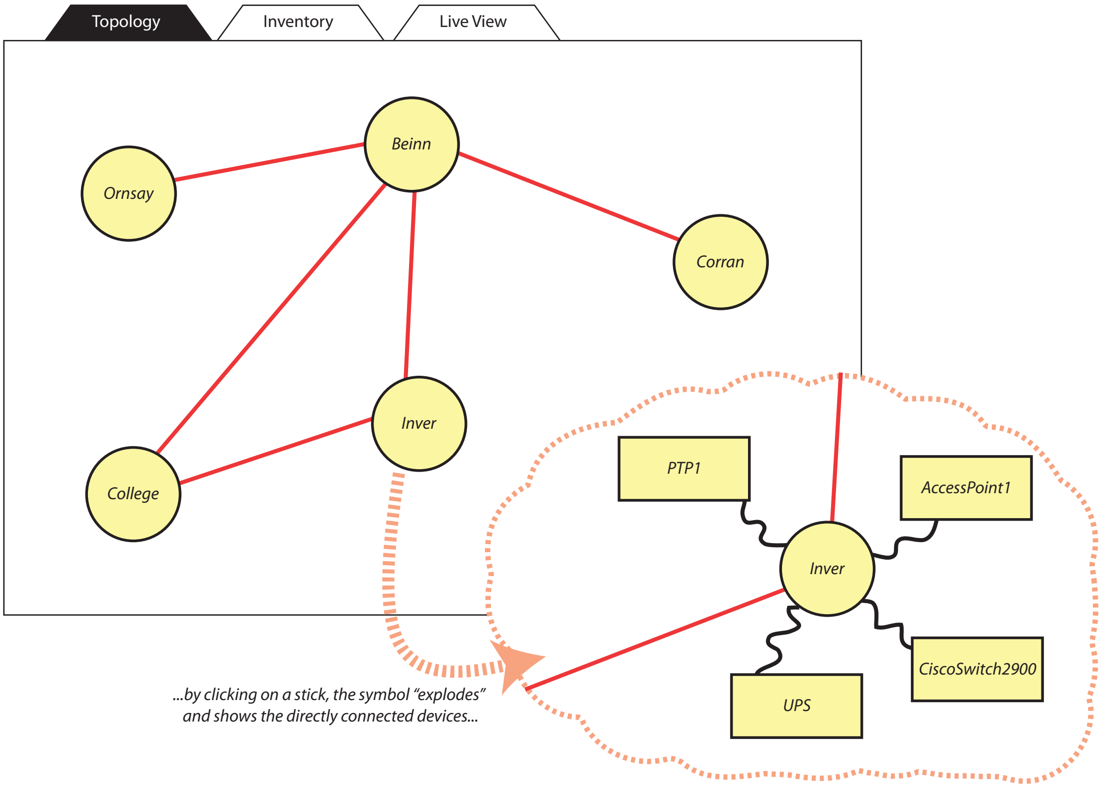


Figure D.1: *Topology tab of the `StixGUI`.*

The first screen, shown in Figure D.1, presents a *topology view* which gives an at-a-glance view of the network layout. A link is shown between two sites when there is a direct network connection (e.g., a backhaul point-to-point link) between them. When the administrator clicks on a site, the view is expanded and all the devices installed at that site shown. Furthermore, clicking on a device opens a menu that enables the manual change of its configuration and to see the list of workflows that are currently running on it.

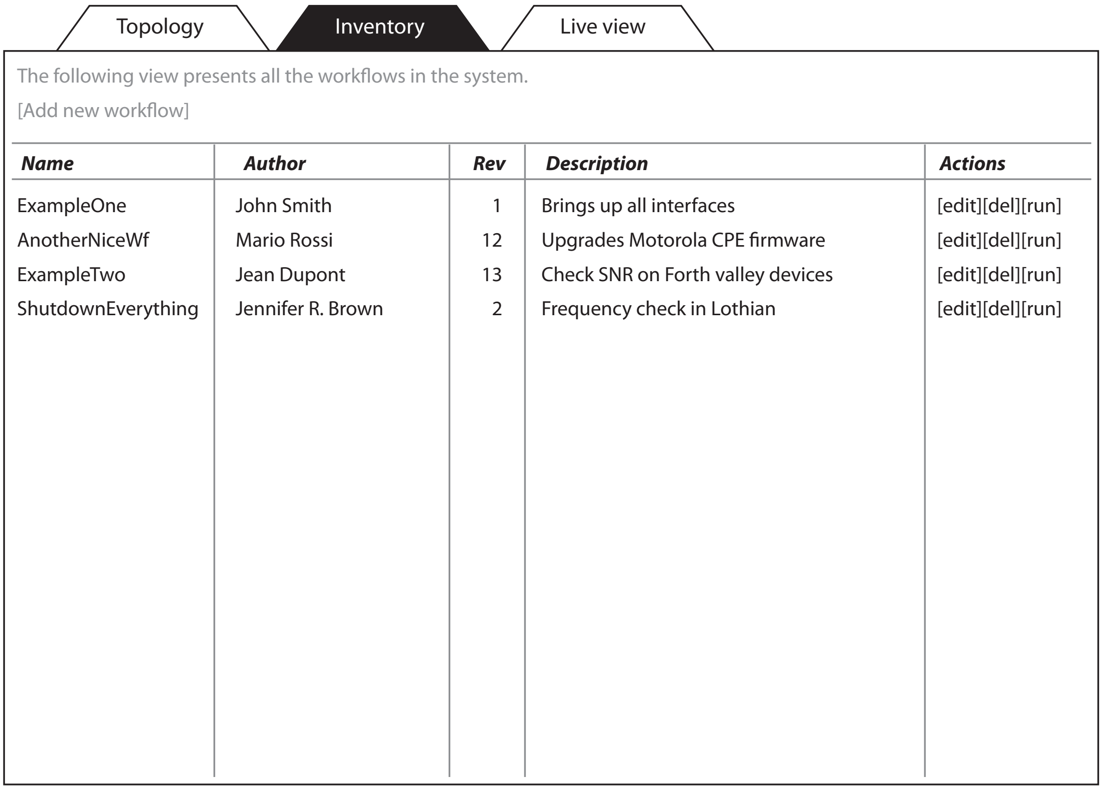


Figure D.2: *Inventory tab of the `StixGUI`.*

The second tab (Figure D.2) is an inventory view, which lists all the workflows defined in the system. Each workflow is described by a row containing its descriptive name, a brief description, its revision number and other optional details. From this screen, the *process edit window* (Figure D.3) can be opened in order to edit, delete and create new workflows. The view is divided into three panels: on the left, the *task library* shows the list of `StixL` events and tasks available in the system; on the top-right, the *flow mapping* pane is used to design workflows by wiring the sequence of events and tasks. As soon as a task is selected in the flow mapping window, the bottom pane shows its current *data mapping*: through this visualisation the network administrator can map workflow variables to input and output parameters.

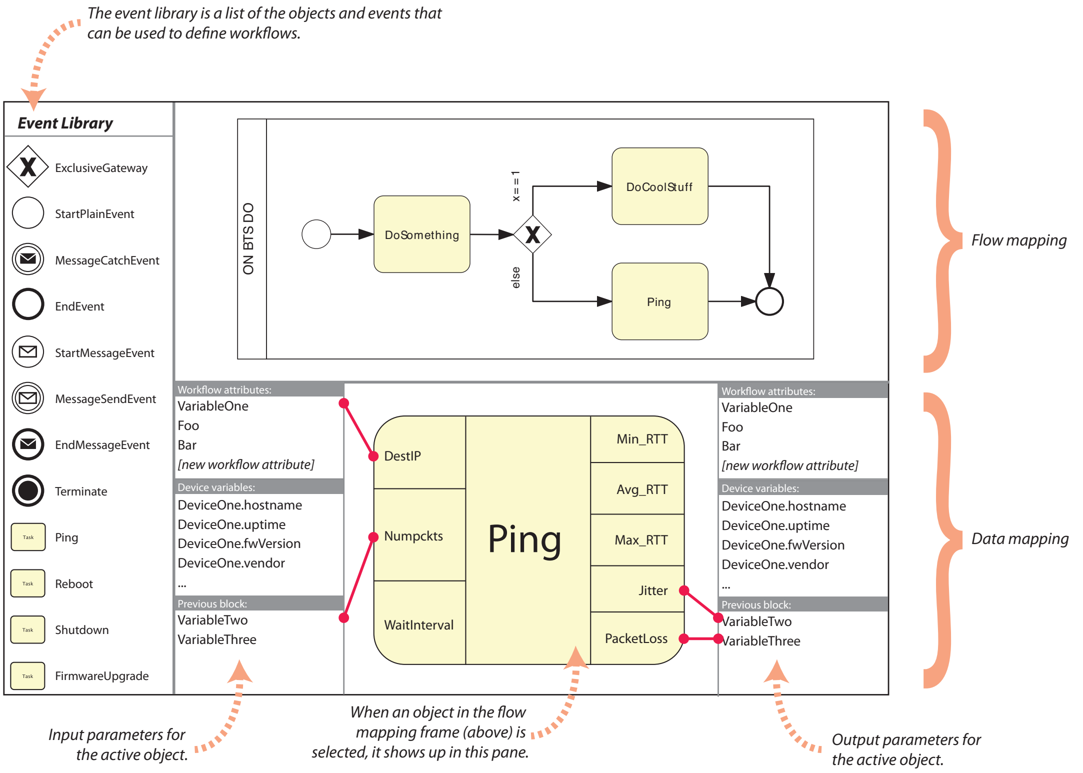


Figure D.3: *The process edit window in* `StixGUI`.

Finally, Figure D.4 shows the *StixView* component: a realtime reporting system that can be used to document network processes and to investigate the current status of the network. Reports are called *perspectives* and can contain static text and images together with queries to the 'log overlay', which are rendered as text, graphs or tables. On the left pane, perspectives can be selected, created and deleted. When a new perspective is selected, the right side of the window shows a blank page which can be edited in a wiki-like fashion using a common markup syntax. The following are examples of what the query syntax could specify:

- select the last 10 log entries for any PTP device in a network region, and show them in a table.
- plot a pie graph of the percentage of completed and failed firmware upgrade over the last day.

- print a number representing the average SNR value of a link over the last hour.

Perspectives can be edited by any user, and the network administrator can browse a history of all the previous revisions.

Perspectives are defined by using a specific syntax, which includes the following markup codes:

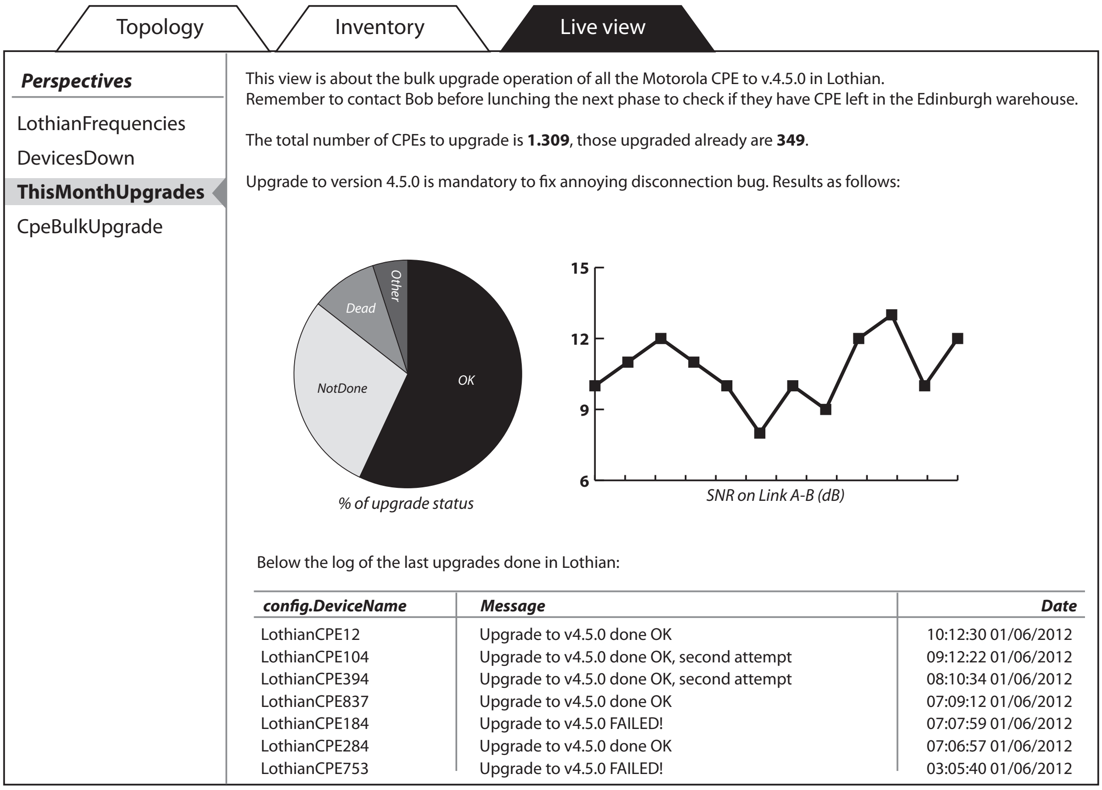


Figure D.4: *The StixView interface.*

Markup:	**Meaning:**
`''text''`	Italic text
`''''text'''`	Bold text
`= Text =`	Section header
`== Text ==`	Subsection header
`=== Text ===`	Sub-subsection header
`[[Text]]`	Link to another perspective

`* Text`
`** Text`
`* Text`
`* Text`
	Unordered lists. More stars indicate a deeper level.

`# Text`
`## Text`
`# Text`
`# Text`
	Numbered lists. More hash symbols indicate sub-numbering.

{{{ `QUERY` }}}	Insert query and present the result as field, table or plot.

By using the notation {{{`ON ... SELECT ...`}}}, dynamic data can be integrated in the visualisation. The query syntax allowed follows this Augmented Backus-Naur Form (ABNF) notation:

```
query             = 'ON' (list-of-agents / '*')
                    'SELECT' (field / list-of-fields / '*')
                    ['WHERE' condition-set]
                    ['ORDER BY' field ['ASC' / 'DESC']]
                    ['LIMIT' 1*DIGIT]
                    'PRESENT AS' ('FIELD' / 'TABLE' / 'PIEGRAPH' /
                        'BARGRAPH' / 'LINEGRAPH');
condition-set     = condition [( 'AND' / 'OR' ) condition];
condition         = ['NOT'] field test value;
test              = < / <= / = / >= / > / <> / 'LIKE';
field             = agentid / workflowid / name / value /
    timestamp
list-of-fields    = 1*(field ',')
```

For example, the following query renders a line graph of the signal-to-noise value recorded on a link:

```
ON siteOne
SELECT value
WHERE name = 'SNR' AND workflowid = 'test123'
```

```
ORDER BY timestamp DESC
PRESENT AS LINEGRAPH;
```

At render-time, the server interprets the markup syntax of the page and runs the queries specified. For each query, `StixView` determines whether the data has to be collected from a single agent (i.e., if a `ON agent` is specified) or from several agents. Consequently, `StixView` connects to the communication manager of the appropriate remote agents. Results are aggregated and rendered as a field, a table or a plot. In case an agent is not responding, `StixView` tries to collect the same data from the neighbours of the unreachable agent, following the 'Sprinkle' paradigm described in Section 5.3.2.4.

Appendix E

StixControl – Board Diagrams

`StixControl` is an add-on board to the `Stix` management system, which can be used to provide "eyes and hands" access to remote devices (e.g., base stations, point-to-point equipment, Ethernet switches, router boards, etc.) in three ways:

- by providing control of the power that feeds the device, thus enabling it to be powered on, off or rebooted.
- by enabling voltage measurement.
- by enabling current measurement.

Up to eight `StixControl` boards, each controlling a single piece of equipment, can be daisy-chained to a `Stix` dongle using its I^2C bus, as illustrated in Figure E.1. The `StixControl` boards is powered up directly from the I^2C bus, and it is opto-isolated

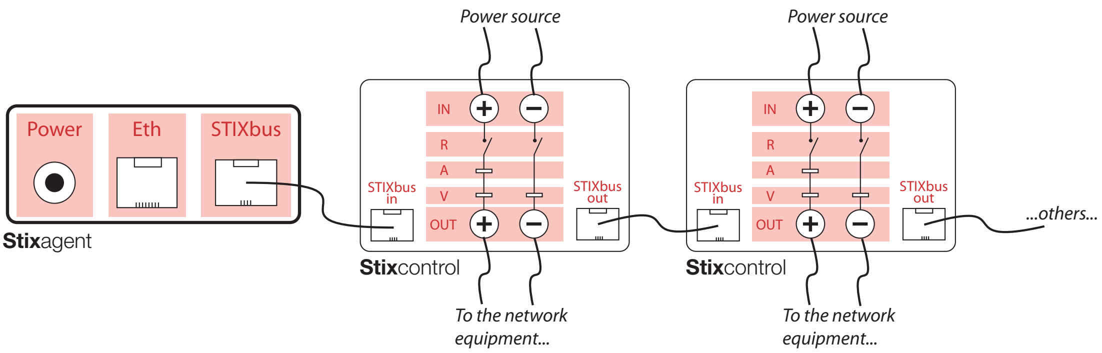


Figure E.1: *Connection layout of StixAgent and StixControl* from the surrounding environment. In this Appendix, we include the diagrams needed for assembling the `StixControl` board: Figure E.2 is the suggested component layout, and Figure E.2 presents the circuit diagrams including type and model of each component.

The central piece of the `StixControl` board is the *PCF8591* chip, which is an I^2C 8-bit A/D and D/A converter. For measuring current, we used a *LTS 6-NP* sensor, which takes up to 19.2A with an accuracy of below 0.7%. The total cost of all the components is estimated at around 15 USD. has various on-board settings, which can be set by operating on the onboard jumpers (marked with *J* in the Figures). The board address can be configured, from 0 to 7, by setting the 3-jumper block J12 (a jumper closed is a logical one, opened is a zero). The voltage and current sensors can be configured to provide different tradeoffs between admissible input range and the measurement granularity, as detailed in the following tables.

Voltage sensor settings:

Configuration	**Range**	**Granularity**	**Jumpers**
0	0 to 16V	60mV	J11 on left
1	0 to 22V	90mV	J11 on right

Current sensor settings:

Configuration	**Range**	**Granularity**	**Jumpers**
0	0 to 6A	26mA	J1-J16, J3-J6, J5-J7
1	0 to 9A	40mA	J1-J16, J2-J3, J6-J7, J5-J7
2	0 to 19A	80mA	J1-J16, J2-J3, J3-J5, J6-J7, J8-J9

The board is to be connected on the power line between the device to be controlled and the power source: input is J15 (positive) and J19 (ground), output is J10 (positive) and J20 (ground).

The `Stix` source code includes a device driver for controlling the `StixControl` board. For troubleshooting, we developed a small C utility called `stixcontrol`, which is included in the source repository: it allows to turn on and off the relay of a given `StixControl` board, and to read its voltage and current levels.

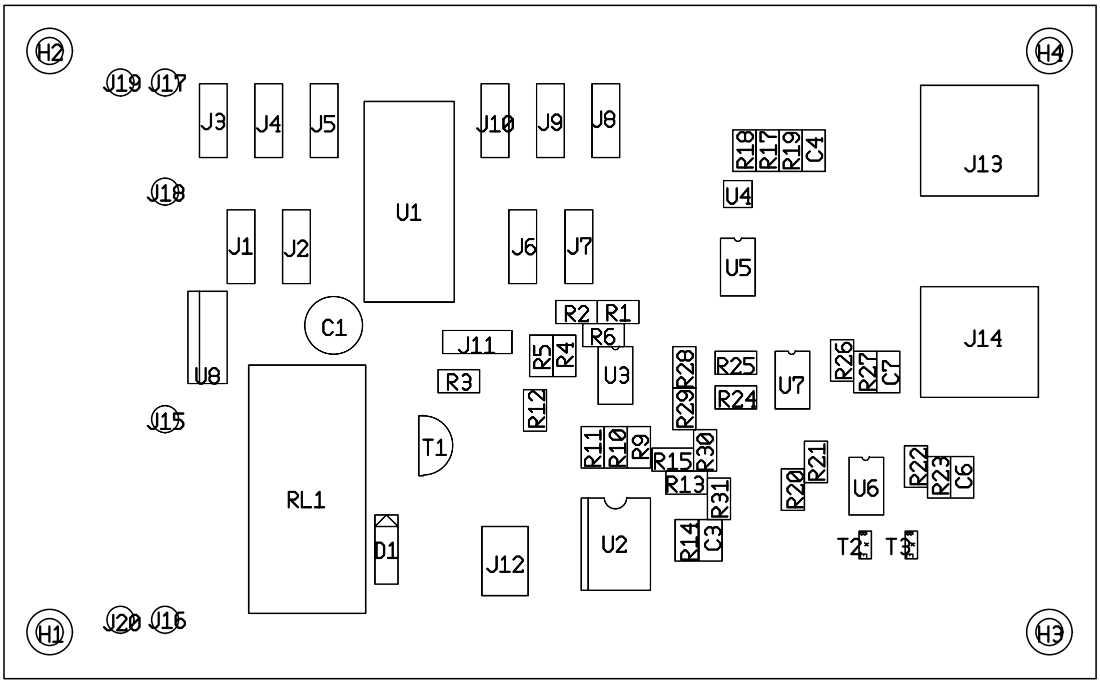


Figure E.2: *Suggested component layout for the* `StixControl` *printed circuit board.*

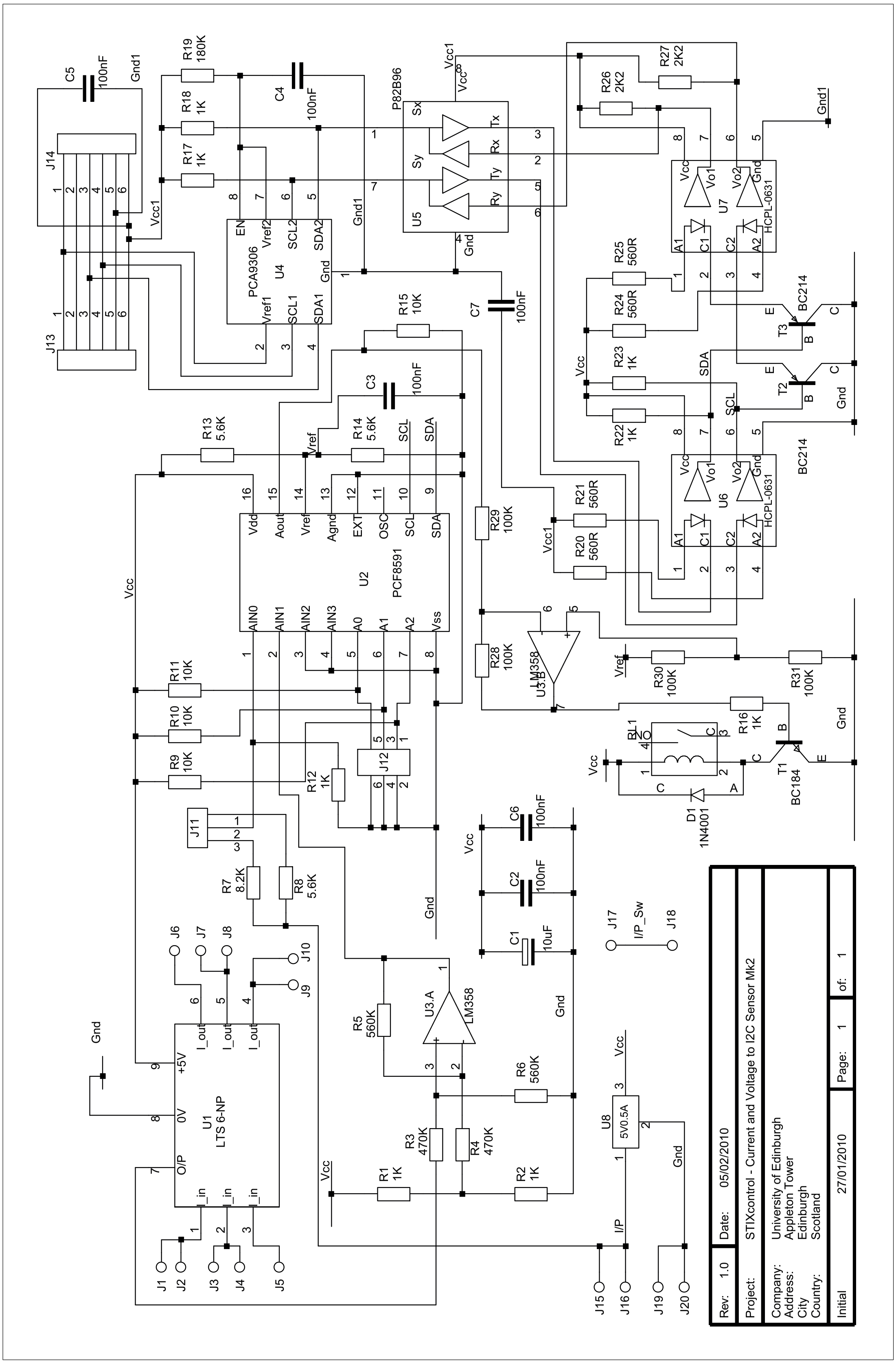


Figure E.3: *Circuit diagram of the* `StixControl` *board.*

Appendix F

Management Use Cases

In this Appendix, we presents a set of use cases we gathered from interviewing BWA operators.

Wireless networks are complex systems. The first lesson learned from the following scenarios is that BWA deployments needs to be managed differently from traditional wired networks, as they pose completely new challenges. Moreover, wireless networks are dynamic structures, with new sites being deployed and upgraded. The management system must be able to model large parameters sets, often vendor-specific, and to react to varying conditions in the local RF environment.

The ideal “actor” behind these use cases is the network administrator of a BWA network, which could range from a small community-driven deployment to a large WISP.

In developing the `Stix` system (see Chapter 5), we took into consideration further sources of management use cases, such as the list of self-configuring scenarios proposed by the 3GPP consortium[1], which includes six use cases (specific to LTE) and highlight possible solutions. Other strategies to tackle those scenarios are also proposed by Nomor Research[2]. Similarly, the Next Generation Mobile Network forum (NGMN) has published its own list of 32 use cases[3], which are organised in four

[1] Gathered in the 36.902V0.0.1(2008-02) Technical Report: “Self-configuring and self-optimising network use cases and solutions”.

[2] “Self Organizing Networks in Long Term Evolution”. Technical report, available from `http://www.nomor.de/home/technology/white-papers`.

[3] See “Next Generation Mobile Networks Informative List of SON Use Cases. An Annex Deliverable by the NGMN Alliance.”. Whitepaper released on April 17th, 2007. Available from: `http://www.ngmn.org`.

categories: Planning, Deployment, Optimisation and Maintenance.

F.1 Scenario 1: Remote configuration tasks

Network environment:

In modern carrier-grade wireless devices, an increasingly large part of the functionality is implemented in firmware. Vendors rely on firmware upgrade mechanisms to improve performances, fix bugs or add new features to devices that are already on the market without incurring the cost of a hardware redesign.

An growing number of configuration parameters is an unavoidable characteristic of increasingly intelligent devices. As new functions are added and novel services supported, more and more variables are introduced to control the device. This increases the demand for seamless configuration management mechanisms.

Key Challenges:

The main challenge in changing the software running on remote wireless devices is the lack of out-of-band channels to reach and control the device. Operations such as firmware upgrades or reconfigurations perform critical operations on the onboard memory so there is an inherent risk of loosing control of the device. Even worse, in complex network topologies, the unavailability or misbehaviour of a single device can disrupt wider regions of the network or interfere with seemingly unrelated services.

The network scale poses a strong operational challenge, as the number of devices to be upgraded or reconfigured can be very large. For example, upgrading all the CPEs installed by an Internet Provider may require downloading a binary image of a few megabytes on the devices prior to the operation. If the upgrade is performed on large partitions of the network, the firmware repository may undergo a very heavy load and become unresponsive.

Detailed use case description:

It is often necessary to upgrade the firmware of network devices in order to fix critical bugs or implement new features. The upgrade procedure is different depending on the role of the device: in the case of backhaul devices, the network administrator has to first re-route the network traffic to other nodes that will not be upgraded immediately.

It is then possible to start the upgrade procedure, for example by sending a binary image of the embedded memory or by modifying the filesystem, which are operations that present an inherent risk of loosing control of the remote device. The last step is to reboot, reload or apply the modifications and to check whether the upgrade has completed successfully. If that is the case, the procedure is repeated for subsequent devices.

For CPE upgrades, the upgrade task can become even more complex as these devices are housed at the subscriber site so they can be powered off at any time and can remain off for an arbitrary period (e.g.: customer on holiday). The task of programming all the devices thus becomes indefinitely long, and the administrator has to periodically check whether any CPE with the previous version of the firmware joins the network in order to upgrade them.

The series of steps described is certainly not scalable in relation to the number of devices, especially when it involves manual actions (e.g.: traffic re-routing, backups, etc). It, however, offers some degree of parallelization as upgrades can be performed on several devices in different areas of the network at the same time.

Further use cases:

Scenario 1.1: Remote hardware control

When sensitive operations, such as firmware upgrades, have to be performed, it may prove useful to be able to remotely control the devices, for example by performing "power operations" (i.e., reboot or shutdown) or by connecting to a debugging interface (i.e., serial or JTAG). Enabling such functionalities in a management framework is like providing "remote eyes and hands" to the network administrator, giving him the ability of restore a remote device as if it was handled locally. This quality is of extra importance for WISPs operating in rural regions, where reaching the deployed devices can be problematic.

F.2 Scenario 2: Regional frequency management

Network environment:

The scarcity of the radio spectrum makes it impossible to allocate a unique radio channel to each wireless sector, thus a frequency reuse policy is needed. When two or more radio devices within reach are transmitting on the same frequency, interference is generated and can cause performance degradation. The need for valid frequency planning is always true, regardless of the regulatory state of the spectrum: if the operator is using a licensed band, he will have to carefully divide it and to assign it to devices; the same is true for unlicensed spectrum, which requires devices belonging to other operators to be considered as well.

Key Challenges:

Frequency planning has to take into account a wide number of factors including device location, geographic characteristics, and the number of customers in the area. Traffic demands also vary according to the time of the day, the day of the week and for how long the region has been served.

Any planning mechanism must support network growth, for example by automatically identifying when a new base station is added, and consequently adjusting the frequency plan. When equipment from different vendors is used, the problem becomes even more complex, as the set of performance metrics and reporting functionalities of each may differ. Operators of multi-vendor networks often have to develop adaptation layers to translate statistics and measurements obtained from different devices so that a common optimisation platform can be used. In general, any planning tool should have a holistic view of the operator network.

Detailed use case description:

We consider a WISP operating in the unlicensed spectrum, which has to subdivide the available frequency range into a number of channels, allocating them to radio sectors. Typically, it is necessary to implement a frequency reuse technique and assign the same channel to multiple sectors that do not mutually interfere, but predicting outdoor wireless propagation is difficult and results can significantly change over time, so the operator has to periodically check that there are no geographic locations covered by more than one base station operating at the same frequency, as these would represent sources of interference. A possible solution to carry out this task is to gather the list of BTSs (and their frequencies) that can be seen from each CPE, verifying that they are all operating on distinct channels.

Further use cases:

Scenario 2.1: Automatic channel monitoring

When a network is operating in the unlicensed spectrum, interference is more likely to happen at any time because other operators start transmitting on the same frequencies. In this case, the "best practice" for PTP links is periodically verify the signal level on each of the frequencies supported, changing the operating channel if needed. Obviously, a "channel hop" technique of this kind should be negotiated with the remote end-point, and could trigger a frequency change on other co-hosted devices.

Scenario 2.2: Forced CPE handover

During the normal operations of a WISP network, some base stations may become crowded, for example as a result of marketing campaigns targeted to a specific geographic region. Over time, such cells tend to receive a significant share of the network traffic and to approach saturation. The management system should be able to automatically handle these situations by identifying the CPEs that can be relocated to different base stations, and to force them to perform a radio 'hand-off'.

Scenario 2.3: Channel width adaptation

In some circumstances it may be appropriate to adjust the channel width of existing links: this is the case of long-distance PTP links operating in adverse condition such as operating over water, in harsh weather or, if self-powered, with limited energy left. Varying the channel width is an effective technique to control the Signal-to-Noise level, while saving power, despite limiting throughput. A management system could exploit this notion, adapting the channel width of PTP links according to the current needs.

F.3 Scenario 3: Alarm visualisation and escalation

Network environment:

Monitoring a large scale wireless deployment can be a daunting task: the set of parameters to control is large, and the same 'metric' can be given different meanings by competing vendors. Also, radio link conditions fluctuate because of external causes (e.g., interference) and weather effects. Effective BWA Operation&Management (O&M) must consider all these unique challenges.

Key Challenges:

This scenario focuses on three important network management concepts: alarm detection, alarm escalation and alarm correlation. While there are many known techniques to handle the former even in a feature-rich domain such as wireless networks, alarm escalation and correlation are challenges that require the management framework to have a holistic knowledge of the whole network. Alarm management is a business-critical tool for commercial WISPs: it allows for stricter Service Level Agreements (SLAs), and can generate savings due to lower downtime costs.

Operational challenges of this scenario include the complications of multi-vendor networks and deployment scale. Indeed, network size pushes the limits of centralised solutions using ‘polling’ techniques (where the management system periodically pulls statistics from the devices) as it is implausible to have a single point of collection.

Detailed use case description:

It is not uncommon to find wireless networks composed of hundreds or thousands of base-stations, and orders of magnitude more CPEs. In this context, it is very useful to generate a realtime overall map of the network, to point out anomalies at a glance. This functionality, which is sometime called a “weather map” in commercial software, provides the network administrator help in troubleshooting. In general, the ability to effectively present a very large parameter space is itself an invaluable tool in the hands of network administrators.

Further use cases:

Scenario 3.1: Power monitoring

WISPs operating in rural regions are often faced with the challenge of deploying devices in strategic locations which are far from energy sources. The energy subsystem of self-powered masts can be the most significant part of the financial expenditure, so it is important to determine the right scale to avoid over-provisioning. An effective monitoring system should take self-powered masts into consideration and control the battery health status (e.g., comparing to historical data), the remaining power, and provide the network administrator feedback when actions have to be taken. Additionally, the management framework could exploit knowledge about the local weather, based on sensors or local forecasts, to foresee the amount of energy that will be available in the environment and the consequent battery charge rate.

F.4 Other suggested scenarios

Scenario 4: Bulk configuration change

The network administrator may decide to change a set of configuration parameters in bulk on all or a specific subset of devices. For example, he may need to reboot all the base stations running a specific software version. The management software should provide a flexible way to select devices, such as depending on the device type, brand and model, the subscriber contract details, network addresses, and the geographical area. Also, since CPEs can be powered off at any time, the management system needs to ensure that the configuration is applied automatically once a CPE device comes online.

Scenario 5. Warehouse logistics in the deployment of new devices

When the network coverage of a specific region is expanded or improved, there are significant consequences in the equipment logistics: new BTSs are installed and CPEs are sent out to the new subscribers. The network administrator has to allocate network resources to each of these devices, such as frequencies and IP addresses, and configure them either prior to their installation or at their first connection attempt. The management system should track these myriad components and configure them when needed.

Scenario 6. Historical performances query

When a budget has been allocated for network improvements, the WISP has to determine which point-to-point links would benefit the most from an upgrade. The management system should enable the administrator to view the performance metrics (e.g., the average daily peak load over the last year) to understand the load on each network region and to decide whether or not an upgrade investment is needed.

Scenario 7: Wet connectors

In rural deployments, outdoor CPE devices often have a “detached” antenna, which is connected to the radio with coax RF cables. In harsh weather, cables and connectors can sometimes be damaged. The management system should be able to monitor the

SNR/RSSI of each CPE for their whole lifetime: a sudden drop in the signal strength could mean that the antenna, cable or connectors are faulty or wet.

Bibliography

Adya, A., Bahl, P., Chandra, R., and Qiu, L. (2004). Architecture and techniques for diagnosing faults in IEEE 802.11 infrastructure networks. In *Proc. of ACM Int. conference on Mobile computing and networking (MobiCom)*, pages 30–44, New York, NY, USA.

Alfonsi, B. (2006). Bridging the Digital Divide. *IEEE Pervasive Computing*.

Amaldi, E., Capone, A., and Malucelli, F. (2001). Optimizing base station siting in UMTS networks. In *Proc. of IEEE Vehicular Technology Conference (VTC)*, volume 4, pages 2828–2832.

Amaldi, E., Capone, A., and Malucelli, F. (2003). Planning UMTS base station location: optimization models with power control and algorithms. *Proc. of IEEE Transactions on Wireless Communications*, 2(5).

Amaldi, E., Capone, A., and Malucelli, F. (2008). Radio planning and coverage optimization of 3G cellular networks. *Wireless Networking*, 14:435–447.

Andrews, M. and Zhang, L. (1998). The access network design problem. In *Proc. of Annual Symposium on Foundations of Computer Science*, pages 40–49.

Angelakis, V., Genetzakis, M., Kossifidis, N., Mathioudakis, K., Ntelakis, M., Papadakis, S., Petroulakis, N., and Siris, V. A. (2007). Heraklion MESH: an experimental metropolitan multi-radio mesh network. In *Proc. of ACM workshop on Wireless network testbeds, experimental evaluation and characterization (WinTECH)*, pages 93–94, New York, NY, USA.

Awerbuch, B. and Azar, Y. (1997). Buy-at-bulk network design. In *Proc. of Annual*

Symposium on Foundations of Computer Science, pages 542–, Washington, DC, USA.

Badonnel, R., State, R., and Festor, O. (2008). Self-Configurable Fault Monitoring in Ad-hoc Networks. *Ad Hoc Networks*, 6(3).

Baliosian, J., Matusikova, K., Quinn, K., and Stadler, R. (2008). Policy-based self-healing for radio access networks. In *Proc. of IEEE Network Operations and Management Symposium (NOMS)*, pages 1007–1010.

Baran, P. (2000). Narrowing the gap to Internet access. *Internet Computing, IEEE*, 4(1):49–50.

Bari, F. and Leung, V. (2007). Use of non-monotonic utility in multi-attribute network selection. In *Proc. of IEEE Wireless Telecommunications Symposium (WTS)*, pages 1–8.

Bauer, S. and Lehr, W. (2010). *Understanding broadband speed measurements Understanding broadband speed measurements*. Massachusetts Institute of Technology. Available online at: `http://mitas.csail.mit.edu/papers/Bauer_Clark_Lehr_Broadband_Speed_Measurements.pdf`.

Belton, V. and Stewart, T. J. (2001). *Multiple Criteria Decision Analysis: An Integrated Approach*. Kluwer Academic Publishers.

Bhagwat, P., Raman, B., and Sanghi, D. (2004). Turning 802.11 inside-out. *Proc. of ACM SIGCOMM Computer Communication Review*, 34:33–38.

Bicket, J., Aguayo, D., Biswas, S., and Morris, R. (2005). Architecture and evaluation of an unplanned 802.11b mesh network. In *Proc. of ACM Int. conference on Mobile computing and networking (MobiCom)*, pages 31–42, New York, NY, USA.

Bosio, S., Capone, A., and Cesana, M. (2007). Radio Planning of Wireless Local Area Networks. *IEEE/ACM Transactions on Networking*, 15(6):1414–1427.

Botta, A., Dainotti, A., and Pescapè, A. (2007). Multi-protocol and multi-platform traffic generation and measurement. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM) DEMO Session*.

Brewer, E. (2005). Technology Insights for Rural Connectivity. In *Proc. of Workshop on Wireless Communication and Development: Global Perspective*.

Bu, T., Chan, M. C., and Ramjee, R. (2005). Designing wireless radio access networks for third generation cellular networks. In *Proc. of Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 68–78 vol. 1.

Camp, J., Robinson, J., Steger, C., and Knightly, E. (2006). Measurement driven deployment of a two-tier urban mesh access network. In *Proc. of ACM Int. conference on Mobile systems, applications and services (MobiSys)*, pages 96–109, New York, NY, USA.

Carbone, M. and Rizzo, L. (2010). Dummynet revisited. *ACM SIGCOMM Computer Communication Review*, 40:12–20.

Charzinski, J. and Walter, U. (2006). Optimized Incremental Network Planning. *GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB)*, pages 1–14.

Chebrolu, K. and Raman, B. (2007). FRACTEL: a fresh perspective on (rural) mesh networks. In *ACM Workshop on Networked systems for developing regions (NSDR)*, pages 8:1–8:6, New York, NY, USA.

Cheng, Y.-C., Afanasyev, M., Verkaik, P., Benkö, P., Chiang, J., Snoeren, A. C., Savage, S., and Voelker, G. M. (2007). Automating cross-layer diagnosis of enterprise wireless networks. In *Proc. of ACM Conference on Applications, technologies, architectures, and protocols for computer communications*, pages 25–36, New York, NY, USA.

Cherry, S. (2004). Edholm's law of bandwidth. *Spectrum, IEEE*, 41(7):58–60.

Clemm, A. (2007). *Network Management Fundamentals*. Cisco Press.

Croce, D., En-Najjary, T., Urvoy-Keller, G., and Biersack, E. W. (2008). Capacity estimation of ADSL links. In *Proc. of ACM CoNEXT Conference*, pages 13:1–13:12, New York, NY, USA.

Daskin, M. S. (1995). *Network and Discrete Location: Models, Algorithms, and Applications*. Wiley.

Decker, G., Overdick, H., and Weske, M. (2008). Oryx - An Open Modeling Platform for the BPM Community. In *Proc. of Int. Conference on Business Process Management*.

Dischinger, M., Haeberlen, A., Gummadi, K. P., and Saroiu, S. (2007). Characterizing residential broadband networks. In *Proc. of ACM SIGCOMM conference on Internet measurement (IMC)*, pages 43–56, New York, NY, USA.

Dudkowski, D., Brunner, M., Nunzi, G., Mingardi, C., Foley, C., de Leon, M. P., Meirosu, C., and Engberg, S. (2009). Architectural principles and elements of in-network management. In *Proc. of the IFIP/IEEE international conference on Integrated Network Management (IM)*, pages 529–536, Piscataway, NJ, USA. IEEE Press.

Dyer, J. S. (2005). MAUT - Multiattribute Utility Theory. In Figueira, J., Greco, S., and Ehrgott, M., editors, *Multiple Criteria Decision Analysis: State of the Art Surveys*, pages 265–285. Springer Verlag, Boston, Dordrecht, London.

Farquhar, P. H. (1980). Advances in multiattribute utility theory. *Theory and Decision*, 12:381–394.

Farrell, S., McMahon, A., Meehan, E., Weber, S., and Hartnett, K. (2011). Report on an Arctic summer DTN trial. *Wireless Networks*, 17:1127–1156.

Fiedler, M., Chevul, S., Radtke, O., Tutschku, K., and Binzenhöfer, A. (2005). The Network Utility Function: A practicable concept for assessing network impact on distributed services. In *Proc. of Int. Teletraffic Congress (ITC)*, Beijing, China. Publishing House, BUPT.

Fiedler, M., Tutschku, K., Chevul, S., Isaksson, L., and Binzenhöfer, A. (2006). The Throughput Utility Function: Assessing Network Impact on Mobile Services. In *Wireless Systems and Network Architectures in Next Generation Internet*, volume 3883 of *Lecture Notes in Computer Science*, pages 242–254. Springer Berlin / Heidelberg, Berlin, Germany.

Gokhale, D., Sen, S., Chebrolu, K., and Raman, B. (2008). On the Feasibility of the Link Abstraction in (Rural) Mesh Networks. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 61–65.

Goldszmidt, G. and Yemini, Y. (1995). Distributed management by delegation. In *Proc. of the International Conference on Distributed Computing Systems*, pages 333–, Washington, DC, USA. IEEE Computer Society.

Gordejuela-Sanchez, F., Juttner, A., and Zhang, J. (2009). A multiobjective optimization framework for IEEE 802.16e network design and performance analysis. *Proc. of IEEE Journal on Selected Areas in Communications*, 27(2).

Grubesic, T. H. (2008). Spatial data constraints: Implications for measuring broadband. *Telecommunications Policy*, 32(7):490–502.

Harmatos, J. (2002). Planning of UMTS core networks. In *Proc. of IEEE Int. Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 740–744 vol.2.

Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proc. of the National Academy of Science*, 1021:16569–16572.

Hurley, S., Allen, S., Ryan, D., and Taplin, R. (2010). Modelling and planning fixed wireless networks. *Wireless Networking*, 16.

Ishmael, J., Bury, S., Pezaros, D., and Race, N. (2008). Deploying Rural Community Wireless Mesh Networks. *Internet Computing, IEEE*, 12(4):22–29.

ITU-T, editor (1996). *Recommendation M.3010. Principle for a Telecommunication Management Network*. International Telecommunication Union, Geneva, Switzerland.

ITU-T, editor (2011). *Measuring the Information Society*. International Telecommunication Union, Geneva, Switzerland.

Johnson, D. (2007). Evaluation of a single radio rural mesh network in South Africa. In *Proc. of Int. Conference on Information and Communication Technologies and Development (ICTD)*, pages 1–9.

Johnson, E. M. and Huber, G. P. (1977). The Technology Utility Assessment. *IEEE Transactions on Systems, Man and Cybernetics*, 7(5):311–325.

Kahani, M. and Beadle, H. W. P. (1997). Decentralised approaches for network management. *SIGCOMM Computer Communication Review*, 27(3):36–47.

Kanuparthy, P. and Dovrolis, C. (2010). DiffProbe: Detecting ISP Service Discrimination. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 1–9.

Keeney and Raiffa, H. (1976). *Decisions with multiple objectives: Preferences and value tradeoffs*. J. Wiley, New York.

Keller, A. (1996). Service-Based Systems Management: Using CORBA as a Middleware for Intelligent Agents. In *Proc. of the IFIP/IEEE Int'l. Workshop on Distributed Systems Operations and Management*.

Kohler, E., Morris, R., Chen, B., Jannotti, J., and Kaashoek, M. F. (2000). The Click Modular Router. *ACM Transactions on Computer Systems (TOCS)*, 18(3).

Kooijman, R. (1995). Divide and Conquer in Network Management Using Event-Driven Network Area Agents. In *Proc. of the ASCI Conference*, Netherlands.

Kramer, R. D. J., Lopez, A., and Koonen, A. M. J. (2006). Municipal broadband access networks in the Netherlands - three successful cases, and how New Europe may benefit. In *Proc. of ACM Int. conference on Access networks (AcessNets)*, New York, NY, USA.

Kreibich, C., Weaver, N., Nechaev, B., and Paxson, V. (2010). Netalyzr: illuminating the edge network. In *Proc. of ACM SIGCOMM conference on Internet measurement (IMC)*, pages 246–259, New York, NY, USA. ACM.

Kubat, P. and Smith, J. (2001). A multi-period network design problem for cellular telecommunication systems. *European Journal of Operational Research*, 134(2):439–456.

Lamparter, S., Oberle, D., and Eberhart, A. (2005). Approximating service utility from policies and value function patterns. In *Proc. of IEEE Int. Workshop on Policies for Distributed Systems and Networks*, pages 159–168.

Lim, K.-S. and Stadler, R. (2003). Weaver: realizing a scalable management paradigm on commodity routers. In *IFIP/IEEE International Symposium on Integrated Network Management*, pages 409–424.

Macmillan, A., Marina, M., and Triana, J. (2010). Slow Frequency Hopping for Mitigating Tidal Fading on Rural Long Distance Over-Water Wireless Links. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 1–5.

Maier, G., Feldmann, A., Paxson, V., and Allman, M. (2009). On dominant characteristics of residential broadband internet traffic. In *Proc. of ACM SIGCOMM conference on Internet measurement (IMC)*, pages 90–102, New York, NY, USA.

Mandviwalla, M., Jain, A., Fesenmaier, J., Smith, J., Weinberg, P., and Meyers, G. (2008). Municipal broadband wireless networks. *Communications of the ACM*, 51:72–80.

Martinez, A., Villarroel, V., Seoane, J., and Pozo, F. (2004). Rural telemedicine for primary healthcare in developing countries. *Technology and Society Magazine, IEEE*, 23(2):13–22.

Martynova, D. M. and Renneboog, L. (2010). A Corporate Governance Index: Convergence and Diversity of National Corporate Governance Regulations. Discussion Paper 2010-012, Tilburg University, Tilburg Law and Economic Center.

Mason, editor (2010). *The costs and capabilities of wireless and satellite technologies – 2016 snapshot*. Analysys Mason.

Matthee, K., Mweemba, G., Pais, A., van Stam, G., and Rijken, M. (2007). Bringing Internet connectivity to rural Zambia using a collaborative approach. In *Proc. of Int. Conference on Information and Communication Technologies and Development (ICTD)*, pages 1–12.

Meusburger, C., Schupke, D., and Eberspacher, J. (2008). Multiperiod Planning for Optical Networks - Approaches Based on Cost Optimization and Limited Budget. In *Proc. of IEEE Int. Conference on Communications (ICC)*, pages 5390–5395.

Meyerson, A., Munagala, K., and Plotkin, S. (2000). Cost-distance: two metric network design. In *Proc. of Annual Symposium on Foundations of Computer Science*, pages 624–630.

Meyerson, A., Munagala, K., and Plotkin, S. (2001). Designing Networks Incrementally. *Proc. of Annual IEEE Symposium on Foundations of Computer Science*, 0:406.

Moscibroda, T., Chandra, R., Wu, Y., Sengupta, S., Bahl, P., and Yuan, Y. (2008). Load-aware spectrum distribution in Wireless LANs. In *Proc. of IEEE Int. Conference on Network Protocols (ICNP)*, pages 137–146.

Mountzia, M.-A. and Dreo-Rodosek., G. (1996). Delegation of Functionality: Aspects and Requirements on Management Architectures. In *Proc. of the IFIP/IEEE Int'l. Workshop on Distributed Systems Operations and Management*.

Musolesi, M., Hailes, S., and Mascolo, C. (2005). Adaptive routing for intermittently connected mobile ad hoc networks. In *Proc. of IEEE Int. Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM)*, pages 183–189.

Ng, E. K.-K., Mui, A. K.-H., Ng, M., Fung, J. M.-Y., Kam, A. Y.-T., and Chan, B. S.-P. (2007). Customizable Wireless Management System (CWMS) for Broadband Wireless Access Networks. In *Proc. of Wireless Communications and Networking Conference (WCNC)*, pages 2741–2745.

OECD, editor (2008). *Broadband Growth and Policies in OECD Countries: Main Findings*. Ministerial Background Report. Available online at: `http://edition.cnn.com/2009/TECH/10/15/finland.internet.rights/index.html`.

OFCOM, editor (2009). *UK Broadband Speeds 2008*. OFCOM. Available online at: `http://www.ofcom.org.uk/research/telecoms/reports/bbspeed_jan09/bbspeed_jan09.pdf`.

OFCOM, editor (2011). *UK Fixed Broadband Speeds*. OFCOM. Available online at: `http://stakeholders.ofcom.org.uk/market-data-research/telecoms-research/broadband-speeds/speeds-nov-dec-2010`.

Ormont, J., Walker, J., Banerjee, S., Sridharan, A., Seshadri, M., and Machiraju, S. (2008). A city-wide vehicular infrastructure for wide-area wireless experimentation.

In *Proc. of ACM workshop on Wireless network testbeds, experimental evaluation and characterization (WinTECH)*, pages 3–10, New York, NY, USA.

Panigrahi, D., Duttat, P., Jaiswal, S., Naidu, K., and Rastogi, R. (2008). Minimum Cost Topology Construction for Rural Wireless Mesh Networks. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 771–779.

Pavlou, G. (2007). On the Evolution of Management Approaches, Frameworks and Protocols: A Historical Perspective. *Journal of Network and Systems Management*, 15(4).

Portoles-Comeras, M., Cabellos-Aparicio, A., Mangues-Bafalluy, J., Banchs, A., and Domingo-Pascual, J. (2009). Impact of transient CSMA/CA access delays on active bandwidth measurements. In *Proc. of ACM SIGCOMM conference on Internet measurement (IMC)*, pages 397–409, New York, NY, USA.

Pras, A., Schonwalder, J., Burgess, M., Festor, O., Perez, G., Stadler, R., and Stiller, B. (2007). Key Research Challenges in Network Management. *IEEE Communications*, 45(10).

Purakayastha, A. and Mohindra, A. (1998). System Management With NetScript. In *Proc. of the USENIX conference on System administration (LISA)*, pages 37–48, Berkeley, CA, USA. USENIX Association.

Qiu, L., Bahl, P., Rao, A., and Zhou, L. (2006). Troubleshooting Wireless Mesh Networks. *ACM SIGCOMM Computer Communication Review*, 36(5).

Raghavendra, R., Acharya, P. A., Belding, E. M., and Almeroth, K. C. (2008). Antler: A multi-tiered approach to automated wireless network management. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 1–6. IEEE.

Raisanen, L. and Whitaker, R. M. (2005). Comparison and evaluation of multiple objective genetic algorithms for the antenna placement problem. *Mobile Networks and Applications*, 10.

Ramachandran, K., Belding-Royer, E., and AImeroth, K. (2004). DAMON: a distributed architecture for monitoring multi-hop mobile networks. In *Proc. of IEEE Sensor and Ad Hoc Communications and Networks (SECON)*, pages 601–609.

Raman, B. and Chebrolu, K. (2007). Experiences in using WiFi for rural internet in India. *Communications Magazine, IEEE*, 45(1):104–110.

Reininger, P., Iksal, S., Caminada, A., and Korczak, J. (1999). Multi-stage optimization for mobile radio network planning. In *Proc. of IEEE Vehicular Technology Conference (VTC)*, volume 3, pages 2034–2038 vol.3.

Russell, S. J., Norvig, P., Candy, J. F., Malik, J. M., and Edwards, D. D. (1996). *Artificial intelligence: a modern approach*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Salman, F. S., Cheriyan, J., Ravi, R., and Subramanian, S. (2000). Approximating the Single-Sink Link-Installation Problem in Network Design. *SIAM J. on Optimization*, 11:595–610.

Schoenwaelder, J. (1996). Using Multicast-SNMP to Coordinate Distributed Management Agents. In *Proc. of the IEEE International Workshop on Systems Management (SMW)*, pages 136–, Washington, DC, USA. IEEE Computer Society.

Schuetz, S., Zimmermann, K., Nunzi, G., Schmid, S., and Brunner, M. (2007). Autonomic and decentralized management of wireless access networks. *Proc. of IEEE Transactions on Network and Service Management*, 4(2):96–106.

Schwartz, B., Jackson, A. W., Strayer, W. T., Zhou, W., Rockwell, R. D., and Partridge, C. (2000). Smart packets: applying active networks to network management. *ACM Transactions on Computer Systems*, 18:67–88.

Sen, S. and Raman, B. (2007). Long distance wireless mesh network planning: problem formulation and solution. In *Proc. of ACM Int. conference on World Wide Web (WWW)*, pages 893–902, New York, NY, USA.

Sheth, A., Doerr, C., Grunwald, D., Han, R., and Sicker, D. (2006). MOJO: a distributed physical layer anomaly detection system for 802.11 WLANs. In *Proc. of*

ACM Int. conference on Mobile systems, applications and services (MobiSys), pages 191–204, New York, NY, USA. ACM.

Sheth, A., Nedevschi, S., Patra, R., Surana, S., Brewer, E., and Subramanian, L. (2007). Packet Loss Characterization in WiFi-Based Long Distance Networks. In *Proc. of IEEE Int. Conference on Computer Communications (INFOCOM)*, pages 312–320.

Skiena, S. S. (1998). *The Algorithm Design Manual*. Springer-Verlag.

St-Hilaire, M. (2009). Topological planning and design of UMTS mobile networks: a survey. *Wireless Communications and Mobile Computing*, 9.

Steenekamp, P. and Roos, P. J. (1996). Implementation of Distributed Systems Management Policies: A Framework for the Application of Intelligent Agent Technology. In *Proc. of the IEEE International Workshop on Systems Management (SMW)*, pages 127–, Washington, DC, USA. IEEE Computer Society.

Stewart, T. J. and Scott, L. (1995). A Scenario-Based Framework for Multicriteria Decision Analysis in Water Resources Planning. *Water Resources Research*, 31:2835–2843.

Subramanian, L. (2006). Rethinking Wireless in the Developing World. In *Proc. of ACM Workshop on Hot Topics in Networks (HotNets)*.

Sundaresan, S., de Donato, W., Feamster, N., Teixeira, R., Crawford, S., and Pescapé, A. (2011). Broadband Internet Performance: A View From the Gateway. In *Proc. of ACM SIGCOMM conference on Internet measurement*, pages 134–145.

Surana, S., Patra, R., Nedevschi, S., and Brewer, E. (2008a). Deploying a Rural Wireless Telemedicine System: Experiences in Sustainability. *Computer*, 41(6):48–56.

Surana, S., Patra, R., Nedevschi, S., Ramos, M., Subramanian, L., Ben-David, Y., and Brewer, E. (2008b). Beyond pilots: keeping rural wireless networks alive. In *Proc. of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 119–132, Berkeley, CA, USA. USENIX Association.

Suzuki, M., Kiriha, Y., and Nakai, S. (1996). Dynamic Script Binding for Delegation Agent. In *Proc. of the IFIP/IEEE Int'l. Workshop on Distributed Systems Operations and Management*.

Tanenbaum, A. S. (2003). *Computer Networks*. Prentice Hall, 4 edition.

Tennenhouse, D. L. and Smith, J. M. (1997). A survey of active network research. *IEEE Communications Magazine*, 35:80–86.

Tennenhouse, D. L. and Wetherall, D. J. (1996). Towards an active network architecture. *Computer Communication Review*, 26:5–18.

Vicente, M. and de Bernabè, F. G. (2010). Assessing the broadband gap: From the penetration divide to the quality divide. *Technological Forecasting & Social Change*, 77(5).

Whitaker, R. and Hurley, S. (2003). Evolution of planning for wireless communication systems. In *Proc. of Annual Hawaii Int. Conference on System Sciences*, page 10 pp.

Winterfeldt, D. V. and Edwards, W. (1973). *Evaluation of Complex Stimuli Using Multi-Attribute Utility Procedures*. Michigan University.

Wu, D., Gupta, D., and Mohapatra, P. (2007). Quail Ridge Wireless Mesh Network: Experiences, Challenges and Findings. In *Proc. of Int. Conference on Testbeds and Research Infrastructure for the Development of Networks and Communities (TridentCom)*, pages 1–6.

Zhang, J., Guo, L., and Wu, J. (2004). An integrated approach for UTRAN planning and optimization. In *Proc. of IEEE Conference on Vehicular Technology Conference (VTC)*, volume 4, pages 2360–2364 Vol.4.