UNIVERSIDAD POLITÉCNICA DE VALENCIA

UNIVERSIDAD
POLITECNICA
DE VALENCIA

DEPARTMENT OF COMPUTER ENGINEERING

Design and Implementation of Architectures for the Deployment of Secure Community Wireless Networks

Jorge Hortelano Otero

Ph.D Advisors:
Dr. Pietro Manzoni
Dr. Juan Carlos Cano Escribá

Valencia, February 2011

Acknowledgements

This Ph. D. Thesis would not have been possible without the support of many people:

The author wishes to express his gratitude to my advisors Dr. Pietro Manzoni and Dr. Juan Carlos Cano, who shared with me their expertise and research insight. I also want to express my complete gratitude to Dr. Carlos T. Calafate, who always has been like the third advisor.

I'd also like to express my gratitude to the members of the GRC research group, those that will continue in the group and those who already completed their studies before me. I much appreciate the exchanges of knowledge, skills and assistance during all these years. I wish you all to complete your thesis succesfully!

The author would also like to convey thanks to the Ministry and the Universidad Politécnica de Valencia for providing the financial means and laboratory facilities. And I also express a gratitude to Massimo Mecella and the "Università La Sapienza" in Rome, where I worked for almost four months.

I wish to thanks the NGOs "TeSo" and "PoliClick". Both associations helped me to focus this research into a social context. Thanks to them, the results of this thesis is now deployed in a developing country and it is not another forgotten thesis into the deep of a bookcase. I proudly say that I am a piece of both associations.

I'd like to say how grateful I am with the people that is closest to me. All have given to me the support needed to continue with this work. Particular thanks, of course, to my parents, for the support they provided me through my entire life; and for my brothers who have always take interest in the activities of their little brother. I also must acknowledge to Paola, without her patience and encouragement I would not have finished this thesis. Finally, I want to express my gratitude to all my friends, who have been close to me for all these years. To all of them, I dedicate this thesis.

Abstract

Recent advances in communication technologies, as well as the proliferation of computing devices, are shaping our surroundings towards an ubiquitous Internet. The Internet offers a global platform for low-cost access to a wide range of communication services such as e-mail, e-commerce, tele-education, tele-health and tele-medicine. However, even in developed countries, a large number of rural areas are poorly equipped even with basic communication infrastructure. Nowadays, there are some efforts to address this lack of infrastructure, but they are still insufficient.

With this aim we present in this thesis *RuralNet*, a community wireless network to provide Internet access on a personalised basis to subscribers in rural areas. The objectives of this study were the development of a new architecture to offer a flexible and secure Internet access in remote rural areas. *RuralNet* combines the paradigm of wireless mesh networks and wireless devices, using cheap off-the-shelf devices and offering a wide range of applications and Internet-based communications services. The solution developed for *RuralNet* can cover large areas at a low cost, and can also be easily implemented and extended in terms of both coverage and services offered.

Since the implementation and testing of *RuralNet* was expensive and labour intensive, we considered that simulation and emulation were a viable alternative to reduce costs. For this purpose we developed *Castadiva*, a flexible test bed designed for testing MANETs and mesh networks. *Castadiva* is a test bed based on low-cost devices, used to test the protocols and applications developed. Through a user friendly interface, *Castadiva* offers the ability to define and test different scenarios and traffic patterns, adding the ability to export to the ns-2 format, in order to compare the results. Ns-2 is the most widely used simulator in research at an international level.

Finally, we have added security features to our proposal. So we made *RuralNet* safe from "malicious" attacks, thus making it more robust. We therefore analysed the community network security, studying the most common attack in this type of networks, namely the *black hole* attack. To detect this attack, we have combined the use of the *watchdog* mechanism, a solution offered in the literature to detect intrusions in wireless networks, and bayesian filtering, a technique for developing robust systems in scenarios with a high noise level.

Resumen

Recientes avances en las tecnologías de la comunicación, así como la proliferación de nuevos dispositivos de computación, están plasmando nuestro entorno hacia un Internet ubicuo. Internet ofrece una plataforma global para acceder con bajo coste a una vasta gama de servicios de telecomunicaciones, como el correo electrónico, comercio electrónico, tele-educación, tele-salud y tele-medicina a bajo coste. Sin embargo, incluso en los países más desarrollados, un gran número de áreas rurales todavía están pobremente equipadas con una infraestructura básica de telecomunicaciones. Hoy en día, existen algunos esfuerzos para resolver esta falta de infraestructura, pero resultan todavía insuficientes.

Con este objetivo presentamos en esta tesis *RuralNet*, una red comunitaria inalámbrica para proveer acceso a Internet de forma personalizada a los subscriptores de un área rural. Los objetivos de este estudio han sido el desarrollo de una nueva arquitectura para ofrecer un acceso a Internet flexible y seguro para zonas rurales aisladas. *RuralNet* combina el paradigma de las redes mesh y el uso de los dispositivos inalámbricos embebidos más económicos para ofrecer un gran número de servicios y aplicaciones basados en Internet. La solución desarrollada por *RuralNet* es capaz de cubrir grandes áreas a bajo coste, y puede también ser fácilmente desplegado y extendido tanto en términos de cobertura como de servicios ofrecidos.

Dado que la implementación y la evaluación de *RuralNet* requiere un alto coste y una gran cantidad de mano de obra, hemos considerado que la simulación y la emulación eran una alternativa válida para ahorrar costes. Con este objetivo hemos desarrollado *Castadiva*, un emulador flexible proyectado para la evaluación de redes MANET y mesh. *Castadiva* es un emulador basado en dispositivos de bajo coste, utilizado para evaluar los protocolos y las aplicaciones desarrolladas. A través de su interfaz amigable, *Castadiva* ofrece la posibilidad de definir y evaluar diferentes escenarios y patrones de tráfico, añadiendo la posibilidad de exportarlos a formato ns-2, el simulador más utilizado por la comunidad científica a nivel internacional, para poder comparar la compatibilidad de los resultados con otras propuestas.

Finalmente, hemos añadido funcionalidades de seguridad a nuestra propuesta. Por lo tanto, hemos hecho *RuralNet* seguro contra ataques "maliciosos", haciéndolo así más robusto. Hemos por tanto analizado la seguridad en redes comunitarias, estudiando el ataque más común en este tipo de redes, es decir, el ataque de *black hole*. Para detectar este ataque, hemos combinado el uso de la herramienta

watchdog, una solución ofrecida en la literatura para detectar intrusiones en redes inalámbricas, y los filtros bayesianos, una técnica para el desarrollo de sistemas robustos en escenarios con un alto nivel de ruido.

Riassunto

I recenti progressi nelle tecnologie di comunicazione, così come la proliferazione di dispositivi informatici, stanno plasmando il nostro intorno verso una Internet omnipresente. Internet offre una piattaforma globale per l'accesso a basso costo a una vasta gamma di servizi di comunicazione come l'e-mail, e-commerce, la tele-istruzione, la tele-sanità e la tele-medicina. Tuttavia, anche nei paesi più sviluppati, un gran numero di zone rurali sono ancora scarsamente dotate di una infrastruttura di comunicazione di base. Al giorno d'oggi, ci sono alcuni sforzi per risolvere questa mancanza di infrastrutture, ma sono ancora insufficienti.

Con questo scopo presentiamo in questa tesi *RuralNet*, una rete wireless comunitaria per fornire accesso a Internet in forma personalizzata agli abbonati di una zona rurale. Gli obiettivi di questo lavoro sono stati lo sviluppo di una nuova architettura per offrire un accesso a Internet flessibile e sicuro in aree rurali isolate. *RuralNet* coniuga il paradigma delle reti wireless Mesh e usa i dispositivi wireless off-the-shelf più economici per offrire una vasta gamma di applicazioni e di servizi di comunicazione basati su Internet. La soluzione sviluppata per *RuralNet* è in grado di coprire vaste aree a basso costo, inoltre, può essere facilmente implementato ed esteso sia in termini di copertura sia in termini dei servizi offerti.

Dal momento che l'implementazione e il collaudo di *RuralNet* comporta costi elevati ed alta intensità di manodopera, abbiamo considerato che la simulazione e l'emulazione fossero una valida alternativa per ridurre i costi. A questo scopo abbiamo sviluppato *Castadiva*, un emulatore flessibile progettato per il testing di reti MANET e mesh. *Castadiva* è un emulatore basato in dispositivi a basso costo, utilizzato per testare i protocolli e le applicazioni sviluppate. Attraverso un'interfaccia user friendly, *Castadiva* offre la possibilità di definire e testare diversi scenari e modelli di traffico, aggiungendo la possibilità di esportarli al formato dell'ns-2, il simulatore più utilizzato nella ricerca a livello internazionale, per poter confrontare la compatibilità dei risultati con altre proposte.

Finalmente, abbiamo aggiunto funzionalità di sicurezza alla nostra proposta. Quindi abbiamo fatto *RuralNet* sicuro da attacchi "maliziosi", rendendolo quindi più robusto. Abbiamo pertanto analizzato la sicurezza nelle reti comunitarie, studiando l'attacco più comune in questo tipo di reti, cioè l'attacco *black hole*. Per il rilevamento di questo attacco, abbiamo combinato l'uso dello strumento *watchdog*, una soluzione offerto in letteratura per rilevare le intrusioni nelle reti wireless, ed i filtri bayesiani, una tecnica per lo sviluppo di sistemi robusti in scenari con un alto livello di rumore.

Resum

Avanços recents en les tecnologies de la comunicació, com també la proliferació de nous dispositius de comunicació, estan orientant el nostre entorn cap a una Internet ubiqua. Internet ofereix una plataforma global per a accedir a un vast rang de serveis de les telecomunicacions, com ara correu electrònic, comerç electrònic, teleeducació, telesalut i telemedicina a baix cost. No obstant això, fins i tot als països més desenvolupats, la majoria de les àrees rurals encara no tenen cap infraestructura de telecomunicacions. Avui dia, hi ha alguns esforços per a resoldre aquesta mancança d'infraestructures, però resulten encara insuficients.

Amb aquest propòsit presentem en la tesi *RuralNet*, una xarxa comunitària sense fil basada en la tecnologia de portals captius (*captive portals*) que proveeix accés a Internet a subscriptors rurals. Els objectius d'aquest treball són desenvolupar una nova tecnologia d'informació i comunicació per a oferir un servei de banda estreta i accés segur a Internet per a zones rurals aïllades. *RuralNet* combina el paradigma de les xarxes mallades (mesh) i l'ús de dispositius sense fils de baix cost encastats per a oferir un gran nombre de serveis i aplicacions basats en Internet. La solució desenvolupada per a *RuralNet* és capaç d'aconseguir àrees distants a baix cost. A més, pot ser fàcilment desplegat i ampliat tant en termes de cobertura com de serveis oferits.

Pel fet que desplegar i avaluar *RuralNet* requereix un alt cost i un esforç intens, simular i emular és una alternativa per a estalviar costos. Tanmateix, per a testar *RuralNet* ens cal un emulador que s'adapte bé a les nostres necessitats. Per aquest motiu, hem desenvolupat *Castadiva*, un emulador flexible dissenyat per a avaluar MANET i xarxes mallades. *Castadiva* és un emulador basat en dispositius encastats de baix cost, usat per a avaluar protocols i aplicacions desenvolupades per a MANET i xarxes mallades. A través d'una interfície amigable, *Castadiva* ofereix la possibilitat de definir i avaluar diferents escenaris i patrons de trànsit, afegint-hi la possibilitat d'exportar-los a format ns-2, el simulador més comunament utilitzat per la comunitat científica, per a comparar resultats o per compatibilitat amb altres experiments.

Finalment, hem volgut afegir característiques de seguretat a la nostra proposta. Per tant, hem fet *RuralNet* segur contra atacs maliciosos i així és més robust. Per això, analitzem la seguretat en xarxes comunitàries, estudiant els atacs més comuns en aquest tipus de xarxes, com per exemple, l'atac de *black hole*. Per a detectar aquest atac, proposem en aquest treball una fusió de l'eina del temporitzador de vigilància (*watchdog*), una solució comuna, que es pot trobar en la bibliografia per

a detectar intrusions en xarxes sense fils; i els filtres bayesians, una tècnica per al desenvolupament de sistemes robustos en escenaris amb un alt nivell de soroll.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Motivation, Objectives and Organisation of the Thesis

1.1 Motivation

In 2005, when the first steps of this thesis were taken, there were still some areas in the Comunidad Valenciana which had no Internet access. This was mostly due to the reluctance of the Internet Service Providers, who hesitated to invest in infrastructure in areas where there is not enough market, or where the physical characteristics of the area makes the investment in infrastructure too expensive.

Nowadays, new choices for Internet connectivity are available, such as using the mobile telephony as an ISP, being the easiest way to provide Internet in urban areas. But there are still large areas in the world where traditional technologies are not deployed, and so new solutions are needed to provide connectivity. The causes for this lack of infrastructure are various, such as the prohibitive cost of deploying a conventional wired infrastructure, or because of national policy issues. These causes are present in almost all countries of the developing world, where the Internet would allow access to all kinds of information to promote trade, education, employment, health, and wealth and therefore, being an important tool for human development.

One solution consists of using other non-conventional technologies such as the IEEE 802.11 standard, which is still a good chance for improving the telecommunications. Wireless Internet may be a very effective and inexpensive connectivity tool, solving the problem of the prohibitive cost of infrastructure. But it does not carry any magic in itself. More work must be done to make this technology useful to the users. New protocols must be developed for these new networks, and new security issues must be addressed before this technology can be widely accepted by users.

1.2 Objectives of the Thesis

In this thesis we propose a new way to provide Internet connectivity among users by combining wireless technologies. Our system is based in a low-cost infrastructure, being available for all rural areas of the developing world. Although the main objective of this thesis is to develop this low-cost infrastructure, and although a lot of work related to wireless ad hoc networks was already done by the research community, we need to do extra research work in order to solve new problems inherent to the wireless networks that affect our work.

The first contribution of this thesis is the design of a community network architecture for providing Internet connectivity at a low cost. The proposed architecture allows us to control the user access, the services provided to the entire network, and the bandwidth of each user.

The second contribution of this thesis is the development of a test bed where we can test our proposed solution. In order to perform a realistic study of the wireless networks, we need a test bed where we can test our real implementation. In the literature we can find some test beds already implemented that provide a good approach to the wireless networks, but we need a more flexible one, which is completely suitable for our task. This test bed must be flexible in the sense of easily allowing to add new protocols or tools for testing.

As a third contribution we study security issues regarding to the wireless networks. In order to deploy a secure infrastructure, that is robust against attacks, we study the watchdog, a basic brick in the Intrusion Detection Systems developed for these kind of cooperative wireless networks. We performed a deep study of it and found some drawbacks that can cause a malicious node to remain undetected by the watchdog.

The fourth contribution is an improvement of the watchdog. We study the bayesian filters, a very effective technique in other scenarios such as detecting SPAM in the e-mail. We integrate the bayesian filter methodology with the watchdog to improve its accuracy in these scenarios. We also perform some tests to measure the obtained improvements.

After describing our contributions in detail, we proceed by making a joint evaluation of all the previous proposals, obtaining a clear picture of the overall improvements achieved.

1.3 Structure of the Thesis

This thesis is organised as follows: in Chapter 2 we present a survey of different community networks, describing the technology and the specific routing protocols used in their deployment.

In Chapter 3 we present *Castadiva*, our test bed emulator created for testing mesh networks and MANETs which has been used along this thesis for evaluating and testing our proposed solution. We also present on this chapter a deep evaluation of this tool to validate the obtained results.

Chapter 4 shows our proposed solution for deploying a low-cost infrastructure to bring Internet connectivity to rural areas. We call this architecture *RuralNet*.

RuralNet is an architecture to strengthen networking support in rural environments, and it allows subscribers to access the Internet. It can also provide a group of free services to all the people within a certain area, and it also allows segregating different kinds of users with different privileges of the network. We also present on this chapter an example of a real deployment of our prototype in Mozambique. We describe the selected scenario and discuss some problems we faced to deploy our community network in a developing country.

Chapter 5 highlights some important security drawbacks on MANETs and mesh networks that *RuralNet* must face. In a cooperative network such as a MANET or a mesh network, it is common that a selfish node refuses to cooperate to save resources, causing a *black hole* in the network. The watchdog technique can be used to detect this abnormal node's behaviour, but it is well known that it has some drawbacks. We propose an improvement of the standard countermeasures deployed in these networks by merging it with bayesian filters, which can detect and reject this kind of attack successfully. We also perform several tests to validate our proposal in different scenarios and to measure the accuracy of our countermeasure.

Finally, in Chapter 6 we present a summary of the main results of this thesis, along with some concluding remarks. We also include a list of the publications related to the thesis, and we comment on possible future research works that can derive from the work here presented.

Chapter 2

Related Work

Within a few years, wireless networking may revolutionise the manner by which we can access the Internet as well as communicate with other people. Although the software behind wireless mesh networking is still evolving, the concept behind this technology, that eliminates the need for a centralised control mechanism, is well thought out and will remain in place.

This chapter presents an overview of the current state-of-the-art and examines the concept behind wireless networking. We also detail how wireless networking is used to deploy community networks at a low cost. Finally, we show the methodology used to evaluate these networks, an important previous work before deploying them in a real scenario.

2.1 Introduction

Mobile computers, such as notebook computers and personal digital assistants (PDAs), are facing a rapid growth, being nowadays everywhere. Owners of these devices also have desktop machines back at the office and want to be connected to their home base even when away. Since having a wired connection is impossible in many scenarios, there is a lot of interest in wireless networks.

Wireless networks have many uses, like the portable office, or rescue operations at disaster areas where the communications infrastructure has been destroyed. People on the road want to use their portable electronic devices to send and receive telephone calls, faxes, and electronic mail, read remote files, and login on remote machines; and all this from anywhere and at anytime. Furthermore, wireless networks are of great value to fleets of trucks, taxis, buses, and vehicles in general, and eventually all wireless and fixed-line networks will merge with the global Internet.

There are three basic technologies to create a wireless network: infrared, radio or laser. Infrared is used for local wire replacements over small distances (mouse to PC). Laser technology offers much greater capability for distance and speed. In networking, lasers typically bridge two network segments between two separate buildings. Radio solutions are different for LAN and WAN uses. WANs use

satellite and microwave portions of the radio frequency spectrum. LANs use radios that operate in the free ISM frequency [Ole05]. Wireless networks are based on the radio interface defined by the IEEE 802.11 [The99] standard, which is the radio technology this thesis is based on.

2.2 Overview of the IEEE 802.11 Standard

The IEEE 802.11 [The99] standard is a technology whose purpose is to provide wireless access to local area networks (WLANs). Stations using this technology access the wireless medium using either the Point Coordination Function (PCF) or the Distributed Coordination Function (DCF). The Point Coordination Function is a centralised access mode optionally used when a point coordinator (PC) is available. When relying on the PCF, contention-free periods (CFP) and contention periods (CP) alternate over time. The Distributed Coordination Function uses a listen-before-talk scheme named carrier sense multiple access (CSMA) [Tob80] with collision avoidance (CA) [ZA02]. The CSMA/CA technology distributes the medium access task among all stations, making every station responsible for assuring the delivery of MAC service data units and reacting to collisions. The collision avoidance scheme is used to reduce the probability of collisions between different stations.

2.2.1 Network Architecture

There are three possible network configurations available within the IEEE 802.11 framework, and they are IBSS, BSS and ESS [ASWZ02].

- An IBSS (Independent Basic Server Set), also known as ad hoc network, is a network established to connect a mesh of mobile stations without any sort of infrastructure.
- BSS (Basic Server Set)-based networks, also known as infrastructure networks, are formed around an access point that typically has a wired connection with a external network infrastructure. Each mobile node communicates directly with the access point.
- ESS (Extended Service Set)-based networks are characterised by the existence of multiple access points whose coverage area partially overlaps.

2.2.2 Physical Level

The bandwidths defined by the standard currently range from 1 to 54 Mbps, but other standards being developed in the 802.11 family shall offer greater bandwidth. The IEEE 802.11 standard defines three physical layers. Two of them were designed for operation at the free ISM (Industry, Scientific and Medical) frequency band (2.4 GHz); these are Frequency-hopping (FH) and Direct-sequency (DS) spread-spectrum frequency techniques. A physical layer using infrared light (IR) was also defined. The 802.11a technology is a physical layer annex to the IEEE

802.11 for operating in the 5 GHz radio frequency. It supports several different data rates ranging from 6 to 54 Mbit/s.

- IEEE 802.11a technology allows achieving good results supporting multimedia applications in environments with several users. The only drawback is that more access points are required to cover a similar area than with IEEE 802.11b or IEEE 802.11g.
- IEEE 802.11b specification enhances the IEEE 802.11's physical layer to achieve higher data rates on the 2.4 GHz band, combining the DSSS (Direct Sequence Spread Spectrum) technique based on Complementary Code Key (CCK) with the QPSK (Quadrature Phase Shift Keying) modulation, which is the key for achieving data rates of 5.5 and 11 Mbit/s.
- IEEE 802.11g is another enhancement to the IEEE 802.11 physical layer. The main advantage of 802.11g is that it maintains compatibility with more than 11 million Wi-Fi products (IEEE 802.11b) already sold.
- IEEE 802.11n is the most recent 802.11 standard for wireless local-area networks. The real data rate throughput is estimated to reach a theoretical 540 Mbit/s. IEEE 802.11n builds upon the previous 802.11 standards by adding MIMO (multiple-input multiple-output) and orthogonal frequency-division multiplexing (OFDM).

2.2.3 Summary

The main purpose of wireless networks is supporting computational and communication services while moving. The advantages of wireless networks are ease and low cost of installation. In environments where deploying a wired network is difficult, i.e. at home, in remote areas, or where the cost is high compared to the benefits, wireless networks are a valid alternative. As a result, this technology is an alternative to deploy community networks and rural networks, which is also of extended use in developing countries.

2.3 Community Networks

Community network is a term broadly used to indicate the use of networking technologies by and for a local community. The primary goals of a community network may include a sustainable, trusted platform for an urban neighbourhood, suburban village or exurban town or region to enhance a vital community, as well as offering easier access to already existing information and services, promotion of local economic development and employment, strengthening of local identity, and/or revitalisation, promotion, and/or maintenance of local communal ties.

Wireless community networks or wireless community projects are the largely hobbyist-led development of interlinked computer networks using wireless LAN technologies, taking advantage of the recent development of cheap, standardised 802.11 (Wi-Fi) devices to build growing clusters of linked, citywide networks. Some are being used to link to the wider Internet, particularly where individuals can

obtain unmetered ADSL and/or cable modem Internet connectivities at fixed costs and share them with friends. When such access is unavailable or expensive, they can act as a low-cost partial alternative, as the only cost is the fixed cost of the equipment.

2.3.1 Examples of Wireless Community Networks

Apart from the technology used to deploy these kind of communities, we can classify the wireless community networks according to the purpose of the community.

2.3.1.1 Private Wireless Community Networks

Private Wireless Community Networks are communities formed by enterprises. They provide a Internet connectivity to any customer in exchange of a fee. Some examples of private wireless communities are:

FON (FON Wireless Ltd.) [FON10] is a company incorporated and registered in the UK. FON was created in Madrid, Spain, and its high-tech investors include Google and Skype. It is a company that runs a system for sharing wireless networks. The business was launched in November 2005. Members need to purchase a low-cost wireless router (called a "Fonera"), which acts as a public Internet access point. FON members can use any other FON access points free of charge and non-FON members can buy Internet access passes or prepaid Wi-Fi from FON for a fee.

Meraki [Mer] is a wireless networking company that provides hardware and software for building large scale wireless networks. These networks are used by businesses, schools, and other organisations that need several access points to offer fill coverage to their users. It uses a centralised control system hosted on Meraki's servers. Meraki was funded by Google and Sequoia Capital in March of 2006.

2.3.1.2 Free Wireless Community Networks

These community networks are open networks where any node can extend the coverage area of the network. Some examples of free wireless community networks are the following:

Guifi.net [RP09] is an open mesh network initially deployed in Catalonia, north of Spain, and it successfully connects almost 8000 nodes. This community also exchanges knowledge among its users about mesh networks and information about devices in the market, allowing them to know the best way to extend their network. Guifi.net calls itself: Open-Free-Neutral because anybody can connect to this network without paying a fee, it has not any imposition of a provider and the traffic is not controlled or limited.

Netsukuku [Pum07] is the name of an experimental peer-to-peer routing system, developed by the FreakNet MediaLab (Italian), born to build up a distributed network, anonymous or censorship-free, fully independent but not necessarily separated from Internet, without the support of any server and with no central authority. It does not rely on a backbone router, neither on any routing equipment other than normal network interface cards.

2.3.1.3 Wireless Community Networks in developing countries and rural areas

Wireless community networks are particularly useful in developing countries or rural areas where commercial telecommunications services are unavailable due to important handicaps that make Internet connectivity a complex and costly task. In rural areas and small towns the Internet Service Providers (ISPs) do not assume the high-cost of technologies designed for the urban market. Moreover, low population density and high deployment costs discourage ISP investments since the estimated return on investment (ROI) is unattractive. It is widely accepted that new information and telecommunication technologies are needed to alleviate a wide range of obstacles for economic and social development in these rural areas. Therefore, some efforts are being done for some non-governmental organisations (NGOs) or governments to connect these areas using wireless networks as a cheap alternative. The main difference with the free wireless community networks is that the last ones are deployed by the users, while the community networks in developing countries or rural areas, are usually deployed by external entities. These communities also have additional problems such as formation of local partners. Another barrier is local telecommunication regulation, which is hindered by limited technical staff, "imperfect" government, and the presence of local incumbent monopolies. Some of the problems we encountered are unregulated wireless usage resulting in significant same-band interference. An example of a wireless community network in developing countries is:

EHAS (Enlace Hispano Americano de Salud) [MVPJ+07] is formed by the *Universidad Politécnica de Madrid* and the non-governmental organisation ISF (Ingenieros Sin Fronteras) in 2004. This Foundation has deployed different community networks in countries such as Peru, Colombia or Cuba. The main objective of this community is to connect different sanitary points across the country to exchange information and formation for its workers. Basically they are using Wi-Fi or VHF-HF technologies for this purpose.

2.3.2 How a Wireless Community Network works

The main advantage of these networks in comparison to classical networks is that wireless community networks do not needs a huge inversion. However, they have some problems that must be addressed: users may need to purchase some hardware to connect to the network as a Wi-Fi card, some specific protocols are needed, user access control is difficult to achieve in an open network, and security is a complex matter in a network where there is no authority.

2.3.2.1 Infrastructure and Hardware

We can divide the community networks' infrastructure in two parts: the wired network and the wireless network.

Wireless Network: The wireless network is commonly composed by the users of the network themselves. Each user connects its device to an access point of the network or to another user. If two nodes are not within radio range of each other, they can communicate with the help of intermediate nodes. These nodes forward data packets from one to the other, thereby transferring packets from the source node to the destination node. In this manner, the coverage area of the network is easily increased. We could think that users' nodes in a wireless community network are usually composed by laptops, but not only. PDAs and mobile phones can be also used to connect to our network.

Wired Network: Independently of how many wireless clients are members of the community network, the wireless network must finally connect to other networks, such as the Internet. This is the main scope of the wired network in a community network. It forms a kind of backbone network over the community to transfer data to a small number of Internet gateways located in areas where wired Internet connectivity is available to route data from the wireless network. The wired network is usually composed by access points, routers or gateways.

Another important point of the community network is that there is no any centralised infrastructure. The community is created using point-to-point or point-to-multipoint links each time a new node joins the network. This encourages the inclusion of new nodes and allows a high scalability of the network.

2.3.2.2 Software

As described in the previous section, there is an absence of a centralised infrastructure. This is paid in terms of client complexity. New protocols are needed for these scenarios, which guarantee the interconnection of wireless devices across the network. However, the routing protocols procedures must be incorporated into all the mobile nodes, and, therefore, we need devices where we can install these protocols or any extra tool we need to deploy in our network. This task that is easy to perform in a computer, can be hard in other devices, specially devices where the manufacturer has installed a specific firmware. Some efforts have already been done to develop generic operative systems which replace the manufacturer's firmware, and where we can install our own tools. The most widely known operative systems for these embedded devices are the OpenWRT system [Ope10] and the DD-WRT one [DD-10]. Both of them are open source operating systems available for a wide range of router manufacturers, where we can install new applications or develop new ones.

2.3.2.3 Security in Wireless Community Networks

As in other scenarios, security is a major issue and it needs to be applied in wireless community networks. On the radio level, since these networks are open, encryption (e.g. WPA) is not usually active and, therefore, it is normal to use encryption at the application layer. But these communities networks are also vulnerable to intrusions and attacks such as other wireless networks. We perform a deeper study of the security strategies used in these networks in Section 2.8.

2.3.2.4 Access Control: Captive Portals

As described in the previous sections, any device can become a member of the network if it has the software and hardware required since there is no encryption. But if the community network is not open to everyone, it can be an undesired situation. Therefore, community networks need new techniques to regulate the access to the network. The most extended solution is the use of Captive Portals.

Captive Portals are used for access control in community networks or other scenarios such as Wi-Fi hotspots or wired access (e.g. apartment houses, hotel rooms, business centres) as well.

The captive portal technique forces an HTTP client on a network to be redirected to a special web page (usually for authentication purposes) before surfing the Internet. A captive portal turns a Web browser into a secure authentication device, which is done by intercepting all packets, regardless of address or port, until the user opens a browser and tries to access the Internet. At that time the browser is redirected to a web page which may require authentication and/or payment, or simply display an acceptable policy and require the user to agree.

In section 4.1 we explain *RuralNet*, our captive portal proposal. *RuralNet* allows any user to connect to a network using devices such as PDAs, mobile phones or laptops. It also provides access control to each user.

2.3.2.5 Non-technical Problems of Wireless Community Networks

Excluding the private community networks, we encountered a variety of non-technical problems. These deployments present much larger installation, maintenance and servicing costs, mainly due to the lack of local technical expertise, equipment availability and logistics. Consequently, there is a need for new quality solutions, and not just research prototypes. The hardware and software must be robust, user friendly, and simple to install, maintain and manage. Another additional problem is the administration of the entire network. For example, if we have a network composed by 200 nodes, it is a non trivial problem to change the radio signal channel if the one used is affected by external interferences. The entire network needs to set the new configuration following an order to avoid fragmentation of the entire network, specially to avoid losing contact with nodes used as repeaters and also with nodes having a difficult physical access.

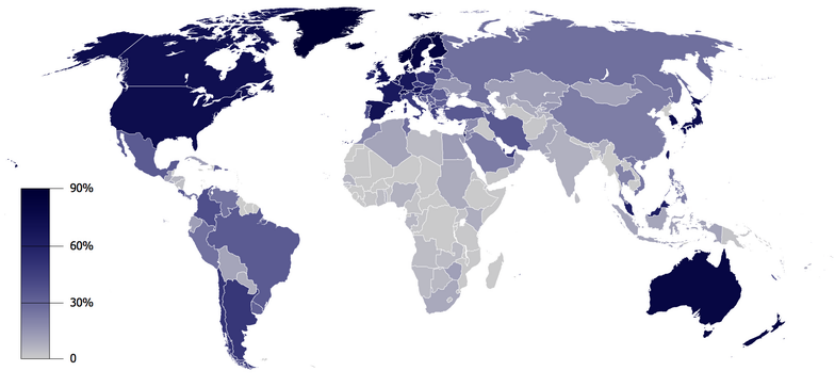

Figure 2.1: Internet penetration on the world.

2.4 Information and Communication Technologies in Developing Countries.

All world countries are faced with substantial numbers of major challenges. Among them we can remark the provision for their inhabitants of good health, adequate education, opportunities for advancement, adequate housing, employment, sufficient income to meet material needs, a sense of personal security within the law, and a sense of security as a nation. Although individual countries may disagree about how to go about achieving those goals, there is agreement in a general sense about what the goals should be [Sad98]. In general, there is a great rush to be part of the network age [Nat01]; the combined result of the technological revolutions and globalisation, that are integrating markets and linking people across all kinds of traditional boundaries, will help us to achieve these goals.

Rapid expansion of the Internet holds substantial promise for developing countries, which can benefit greatly from the Internet's communication and information delivery capabilities to help meet these needs. The accelerating transition of information to electronic media is making information resources of the world available to an increasingly global audience through the Internet. Developing countries have much to gain from that revolution in communication and information access. In contrast to the situation in the developed world, where transport and communications infrastructures for delivery of both physical goods and information services are well established, the alternatives available within developing countries are generally slow, expensive, or nonexistent. Figure 2.1 shows the differences between the richer countries and the poorer ones with regard to Internet penetration.

People in richer countries have better access to information and communication technologies (ICTs), and use it more intensively than do people in poorer countries. It is clear that Internet usage is far more common in richer countries than in poorer countries, and also that the number of users has grown more quickly in richer countries [Wal05]. But, if the development community turns its back on the explosion of technological innovation in food, medicine and information, it risks

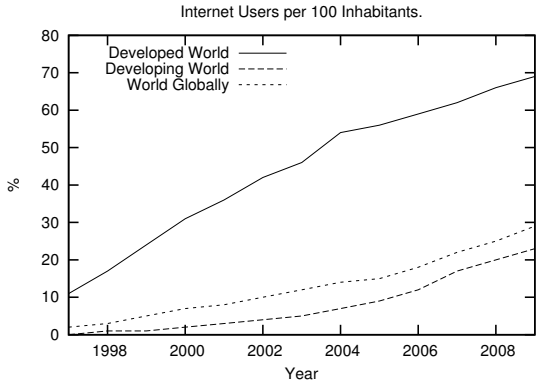

Figure 2.2: Internet Users.

marginalising itself and denying developing countries opportunities that, if harnessed effectively, could transform the lives of poor people and offer breakthrough development opportunities to poor countries.

However, the countries that can take more profit of the new technologies are those which have less capacity of using it. This is due to a phenomenon called the "digital divide". Figure 2.2 shows a comparative of the increase of Internet usage, in developed countries and in developing countries.

2.4.1 The Digital Divide

The digital divide [Rog01] refers to the gap between people with effective access to digital and information technology, and those with very limited or no access at all. It includes the imbalance both in physical access to technology, and the resources and skills needed to effectively take profit of the new technologies. The term is a social construction that emerged in the latter half of the 1990s after the Internet came into the public domain and the World Wide Web exploded into history's largest repository of human knowledge.

The essence of this concept is that, while low incomes cause low ICTs penetration, low penetration may also perpetuate low incomes. In developing countries, information poverty is one of the more significant and insidious obstacles to effective exploitation of information processing and other types of technology. Lack of adequate information regarding developments in other countries and other environments is often not noticed, and in the absence of new information, old techniques and procedures are continued without conscious knowledge of alternatives, affecting the economy of the entire country.

From a global perspective [Vie08], we see that concentration of access to ICTs abounds in the North America, Europe and the Northern Asia Pacific while access is restricted in southern regions of the globe, most notably in Africa, rural India and the southern regions of Asia. The poorer nations, plagued by multiple burdens of debt, disease, and lack of education are those least likely to benefit from Internet

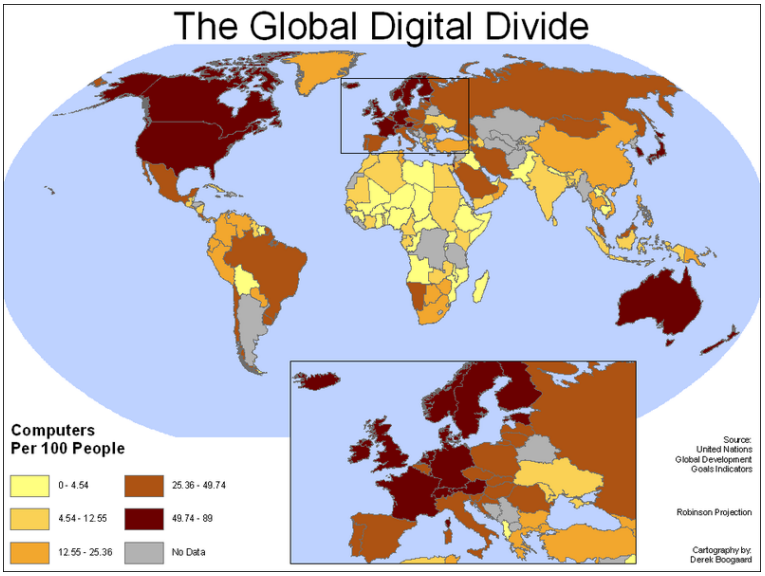

Figure 2.3: Global digital divide.

access. Figure 2.3 shows the global digital divide by country.

But not only Internet could help stimulate economic development, as has been demonstrated in a number of countries recently: access to information affects political democratisation efforts at the global level as well as within nations. In developing countries, where much of the media is controlled by the state and individual access to networks is currently limited, the need to decentralise control over information and over networks themselves is clear.

2.4.2 Causes of the Digital Divide

The state of the physical communications infrastructure is crucial. Adequate international and local links may be neither present nor reliable; equipment may be difficult to obtain, maintain, and repair; electrical power may not be reliable. Computers and the related peripherals required for networking may be absent or inadequate. Transportation and communications links may be weak and retard progress.

Many developing countries are benefiting from direct assistance in transferring computing technology to themselves. But, information poverty, financial poverty, and misperceptions about the costs and benefits of network connectivity have sometimes resulted in decisions to delay investment in networking activities, which may be considered too expensive relative to other needs. Therefore, we can find a lack of investment in ICTs' infrastructure in developing countries, which slows down the development of other sectors such as industry.

2.4.3 Old Solutions for New Problems

In developing countries where local governments cannot solve all the problems presented in the society, population groups usually face their problems by themselves to improve their quality of live. As a consequence, different communities (associations, non-governmental organisations or other groups of people) appear to solve a specific problem. In these communities each member collaborates with the other ones to achieve a solution.

This behaviour is not different from the one presented in community networks, where nodes collaborate with each other to form a platform to provide connectivity. Therefore, deploying a community network can be the solution for developing countries to fight the digital divide in a straight manner thus improving the economy and the human development of the area.

In this thesis we propose a methodology to share Internet connectivity among a local community, following the philosophy of people collaborating to face a certain problem as this has proved to be very effective with other old problems.

2.5 Mesh Networks

Wireless mesh networks (WMNs) [IXW05] are dynamically self-organised and self-configured, with the nodes in the network automatically establishing an ad hoc network and maintaining the mesh connectivity. Wireless mesh networks were originally developed for military applications but over the past decade the size, cost, and power requirements of radios has declined, extending it use to civil applications. WMNs are comprised of two types of nodes: mesh routers and mesh clients. Other than the routing capability for gateway/bridge functions as in a conventional wireless router, a mesh router contains additional routing functions to support mesh networking. Through multi-hop communications, the same coverage can be achieved by a mesh router with much lower transmission power. In spite of all these differences, mesh and conventional wireless routers are usually built based on a similar hardware platform. Mesh routers have minimal mobility and form the mesh backbone for mesh clients. Thus, although mesh clients can also work as a router for mesh networking as a Mobile Ad hoc network (explained in the next section), the hardware platform and software for them can be much simpler than those for mesh routers. Figure 2.4 shows an example of a mesh network architecture.

Mesh networks differ from other networks since all the components can connect to each other via multiple hops, and they are generally not mobile. It employs two different connection arrangements: full mesh topology or partial mesh topology. In the full mesh topology, each node (workstation or other device) is connected directly to each of the other nodes. In the partial mesh topology, some nodes are connected to all the others, while others are only partially connected. Mesh networks are usually used to extend the coverage of access points, and can be seen as a type of ad hoc network.

As we can see, mesh networks are the backbone of community networks, and they are used as the basic infrastructure for deploying them. To date, several

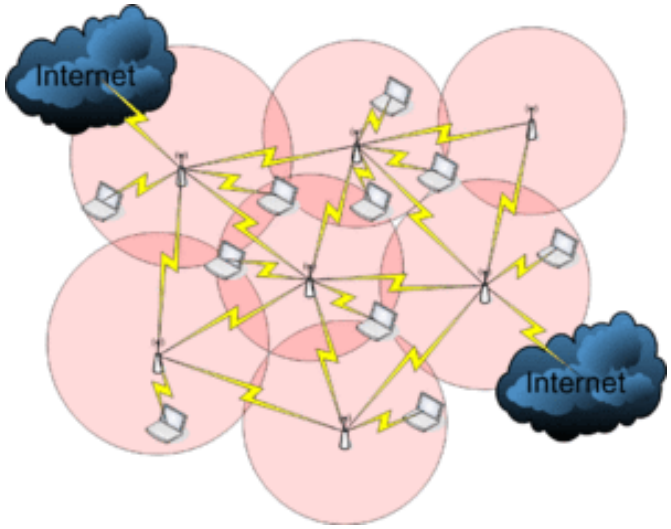

Figure 2.4: Example of a mesh network.

companies have already realised the potential of this technology and offer wireless mesh networking products.

2.6 Mobile Ad hoc Networks

MANETs or Mobile Ad hoc Networks [SMSI04] became a popular research topic as laptops and 802.11/Wi-Fi wireless networking became widespread. It consist of devices that are autonomously self-organised. In ad hoc networks the devices themselves conform the network, and this allows seamless communication, at a low cost, in a self-organised fashion and with easy deployment. Users have the opportunity to create their own networks, which can be deployed easily and cheaply. However, a price for all those features is paid in terms of complex technological solutions, which are needed at all layers and also across several layers.

The importance of MANETs is shown in the wide application area where they are involved. Special situations need communication networks without any short of infrastructure, like emergency missions, military operations or meeting rooms. These scenarios require the quick deployment allowed by MANETs. The research spent in this new technology is growing every year, and it is important to have some tools that allow researchers to evaluate their proposals before investing huge amounts of money and developing effort.

MANET and mesh networks are, therefore, closely related, but mobile ad hoc networks also have to deal with the problems introduced by the mobility of nodes. In this work we use MANETs to extend the coverage of our proposed mesh network (generating a partial mesh topology), thus increasing the number of subscribers to the community network with a minimal investment.

2.7 Routing Protocols

Both MANETs and mesh networks require efficient routing protocols to be correctly deployed. If an ad-hoc network consists of only a few nodes that are up and running at all times, don't move and always have stable radio links, it would be possible to write individual routing tables for all nodes by hand.

Unfortunately, those conditions are rarely met in a real mesh network and, far from it, in a real MANET or mesh network. Nodes can fail, Wi-Fi enabled devices roam around, and interference can make radio links unusable at any time. Also no one wants to update several routing tables by hand if one node is added to the network. By using routing protocols that automatically maintain individual routing tables in all nodes involved, we can avoid these issues. Popular routing protocols from the wired world (such as OSPF [Moy98]) do not work well in such an environment because they are not designed to deal with lossy links or rapidly changing topology.

A routing protocol is a protocol that specifies how routers communicate with each other to disseminate information, allowing them to select routes between any two nodes on a network. Many of the academic papers dealing with ad hoc and mesh networks [CES03, TPA+01, IC06, DDJ01, DD96, PAPL02, TGLG01, SCM99] evaluate protocols assuming varying degrees of mobility within a bounded space, usually with all nodes within a few hops of each other, and usually with nodes sending data at a constant rate. Different protocols are then evaluated based on the packet drop rate, the overhead introduced by the routing protocol, and other measures. Although there is a significant number of proposed ad hoc routing protocols, they basically differ in the methods to discover the routes between two distant nodes, and on how the route failures are detected.

2.7.1 Basic Routing Techniques

Independently of how a routing protocol is classified according to those criteria, the routing techniques used can be divided into three families: distance vector, link state and source routing. We now detail briefly the basic principles of each of these techniques.

2.7.1.1 Distance Vector

This technique maintains a table for the communication taking place and employs diffusion (not flooding) for information exchange between neighbours. All the nodes must calculate the shortest path towards the destination using the routing information of their neighbours.

2.7.1.2 Link State

The protocols based on this technique maintain a routing table with the full topology. The topology is built by finding the shortest path in terms of link cost, cost that is periodically exchanged among all the nodes through a flooding technique. Each node updates its routing table by using information gathered about link

costs. This technique is prone to cause loops on networks with a fast changing topology.

2.7.1.3 Source Routing

It is a technique where all data packets have routing information in their headers. The route decision is made on the source node. This technique avoids loops entirely, though the protocol overhead is quite significant. This technique can be inefficient for fast moving topologies due to route invalidation along the path of a packet.

2.7.2 Classification of Routing Protocols

Routing protocols based on algorithms such as distance vector (e.g. RIP [G. 98]) or link-state (e.g. OSPF [Moy98]) were available before solutions were sought in the field of wireless ad hoc networks. These routing protocols generate periodic control messages, a procedure that is not adequate for a large network with long routes since it would result in a large number of control messages. Also, all the conventional routing protocols assume bidirectional routes with a similar quality, something that is not always true on some kinds of networks (e.g. wireless ad hoc networks). Routing protocols can be classified according to three different criteria:

- Centralised or distributed: all the decisions take place at a central node. However, with a distributed routing protocol, all the nodes share the routing decisions.

- Adaptive or static: an adaptive routing protocol can change its behaviour according to the network's state, which can be the congestion on a certain connection or other possible factors, contrarily to a static one.

- Reactive, proactive or hybrid: a reactive routing protocol must act to find routes when necessary, while a proactive routing protocol finds routes before these are required. Reactive routing protocols are also known as on-demand routing protocols. Since these are executed on-demand, the control packets' overhead is considerably reduced. Proactive methods maintain routing tables, being these periodically updated. Concerning hybrid methods, they use a combination of both reactive and proactive techniques to achieve a more balanced solution.

2.7.3 Routing in Ad hoc Networks

An ideal routing protocol for ad hoc networks must have certain properties that make it different from the rest: (a) It must be distributed to increase reliability (when all the nodes are mobile, it makes no sense to have a centralised routing protocol). (b) Each node must have enough capabilities to take routing-related decisions with the aid of the rest of the nodes, (c) it should assume that the links detected are unidirectional connections. On a wireless channel, a unidirectional

connection may be formed due to physical factors, so that bidirectional communication may result impossible. (d) it should take into account issues such as power consumption and security. Obviously, mobile nodes depend on batteries. This means that a protocol that minimises the total power consumption of network nodes would be ideal. (e) it should take into account that the wireless medium is very vulnerable. At the physical level, DoS attacks can be avoided by using frequency hopping or code-based Spread Spectrum techniques. At the routing level, though, both the authentication of neighbours and the encryption of data are required.

Concerning the routing protocols used on ad hoc networks, they should be, according to the classification of Section 2.7.2, both distributed and adaptive.

Relatively to the third category (reactive/proactive/hybrid), there is no consensus over which is the most adequate strategy. Below we present the different proposals that are currently available for each of these protocol families, and we also include other non-catalogued proposals.

2.7.3.1 Proactive Routing Protocols

The concept of proactive routing means that all the nodes (routers) periodically interchange routing information (or upon detecting topology changes) with the aim of maintaining a consistent, updated and complete view of the network. This avoids delays associated with finding routes on-demand. Proactive techniques typically use algorithms such as distance vector or link-state. Both techniques require routers to periodically broadcast information and, based on that information, to calculate the shortest path towards the rest of the nodes. The main advantage of proactive routing schemes is that there is no initial delay when a route is required. On the other hand, these are usually related to a greater overhead and a larger convergence time than for reactive routing techniques, especially when mobility is high.

Examples of routing protocols using distance vector techniques are the Destination Sequenced Distance Vector (DSDV) [CP94] and the Wireless Routing Protocol (WRP) [MGLA96]. Examples of link-state based protocols are the Open Shortest Path First (OSPF) [Moy98], the Optimised Link State Routing (OLSR) [TPA+01], the Topology Broadcast Reverse Path Forwarding (TBRPF) [RFM04], the Source Tree Adaptive Routing (STAR) [GLAS99], the Global State Routing (GSR) [CQS98], the Fisheye State Routing (FSR) [PGC00] and the Landmark Routing Protocol (LANMAR) [PGH00].

2.7.3.2 Reactive Routing Protocols

Reactive routing does not depend, in general, of periodic exchange of routing information or route calculation. Therefore, when a route is required, the node must start a route discovery process. This means that it must disseminate the route request throughout the network and wait for an answer before it can proceed to send packets to the destination. The route is maintained until the destination is unreachable or until the route is no longer necessary. On the other hand, the route discovery process causes a significant start-up delay and causes a considerable

waste of resources. If the network is wide enough, the overhead will be similar or superior to that achieved with proactive routing protocols.

The most common routing algorithms found among reactive routing protocols are distance vector and source routing. Example of reactive routing protocols are the Ad-hoc On-demand Distance Vector (AODV) [PR99], the Dynamic Source Routing (DSR) [DDY04], the Associatively Based Routing (ABR) [C. 97], the Signal Stability based Adaptive routing (SSA) [DRWT96], the Temporally Ordered Routing Algorithm (TORA) [VS00], the Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) [AT99] and the Dynamic On-demand MANET routing protocol (DYMO) [IC06].

2.7.3.3 Other Strategies

There are other strategies proposed for the design of routing protocols. There are, for instance, hybrid solutions such as the Zone Routing Protocol (ZRP) [ZM99], there are some protocols based on clustering and hierarchical architectures, such as the Clusterhead Gateway Switch Routing (CGSR) [Chi97], the Distributed Mobility-Adaptive Clustering (DMAC) [Bas99] and the Cluster-based Energy Saving Algorithm (CERA) [JDP03]. The LAR protocol [YN98] tries to avoid this problem by using GPS information so that only those nodes on a certain geographic area between source and destination must retransmit route requests.

Next we describe the OLSR and AODV protocols since we decided to use both of them as the representation of a proactive protocol and a reactive protocol respectively.

2.7.4 Why different protocols?

A key issue in MANETs is the need for routing protocols to respond rapidly to topological changes in the network. At the same time, due to the limited available bandwidth achieved through mobile radio interfaces, it is imperative that the amount of control traffic generated by the routing protocols is kept at a minimum.

As shown in [Cla04] both classes, reactive and proactive protocols, complement to each other. With TCP traffic or networks with heavy load of traffic, proactive protocols outperform reactive ones; on the other hand, in UDP scenarios or scenarios with a light load of traffic, a reactive protocol achieves better results.

It is clear, that none of the two protocol classes outperforms the other in every domain. Therefore, both solutions should be considered. In this thesis we test both reactive protocol - OLSR - , and a proactive protocol - AODV -. Both protocols are explained in detail below.

2.7.5 The Optimised Link-State Routing Protocol (OLSR)

The Optimised Link State Routing protocol [TP03] is a proactive routing protocol specifically designed for mobile ad hoc networks (MANETs). It is based on the definition and use of dedicated nodes, called multipoint relays (MPRs). MPRs are selected nodes which are responsible for forwarding broadcast packets during the flooding process. This technique allows reducing the packet overhead compared

to a pure flooding mechanism where every node retransmits the packet when it receives the first copy of it. Contrarily to the classic link-state algorithm, partial link-state information is distributed throughout the network.

2.7.5.1 Basic Principles

The OLSR protocol inherits its stability from link-state algorithms. Due to its proactive nature, it offers the advantage that available routes can be used immediately.

Pure link-state algorithms declare and propagate the list of neighbours for each node throughout the network. OLSR tries to improve this solution by using different techniques. To start with, it reduces the size of control packets since it does not declare all of its neighbours, but only a subset of these referred as Multipoint Relay Selectors. A node's Multipoint Relay is in charge of retransmitting its broadcast messages. The use of MPRs serves the purpose of minimising the amount of retransmissions upon a flooding or broadcast event.

Besides periodic control messages, the protocol does not generate additional control traffic in response to failures or association with new nodes. The protocol maintains routes towards all networks destinations, being useful in those situations where a great number of MANET nodes is communicating, especially when source/destination pairs are changing frequently. This protocol is more adequate for large and dense networks, where the optimisations achieved by introducing Multipoint Relays offer important benefits.

The protocol is designed to operate in a distributed fashion, so it does not depend on a central entity. Moreover, it does not require reliable transmission of its control messages: each node sends periodic control messages, being tolerant to sporadic losses of control packets. Packet reordering, a frequent phenomena in ad hoc networks, will not cause OLSR to misbehave since each message carries a different sequence number.

The OLSR protocol uses per-node packet forwarding, which means that each node uses its most recent information to route a packet. The ability to follow a node can be adjusted by setting the interval between consecutive control messages.

2.7.5.2 Multipoint Relays

The Multipoint Relay concept consists in trying to minimise the flooding caused by broadcast traffic by eliminating duplicated transmission on a same region. Each network node selects a subset of those nodes in its vicinity to retransmit its packets. Nodes belonging to this subset are a node's Multipoint Relays (MPRs). The neighbours not part of the MPR subset of a certain node N will still receive packets from it, but will not re-transmit them again. That way, each node maintains a table with the nodes which have selected it as their MPR.

Each node selects its own set of MPRs among their neighbours with a criteria that consists of assuring that all those nodes two hops away from it can be reached with a minimal number of MPRs. Figure 2.5 illustrates this concept.

OLSR trusts on the MPR node selection to calculate routes towards all the destinations having these as intermediate stations. This solution requires each

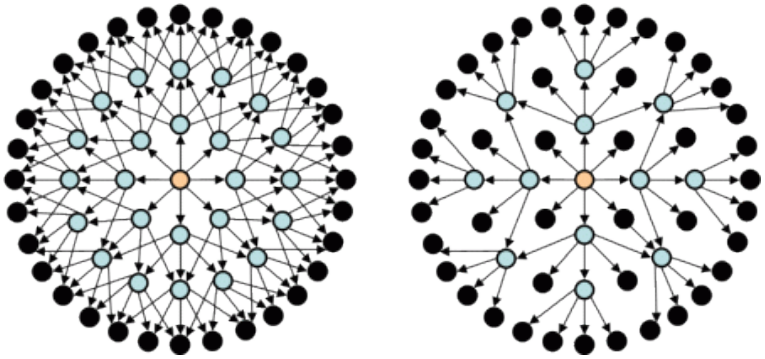

Figure 2.5: Illustration of the multi-point relay concept for node N

node to periodically broadcast the list of Neighbour nodes chosen as its MPRs. When receiving this information, each neighbour node updates the routes towards all known stations.

2.7.5.3 Neighbour Detection

Each node must detect those neighbours nodes towards which bidirectional communication exists. To achieve this purpose a node periodically broadcasts HELLO messages containing information about its neighbours and the state of the channel towards them. These messages are received by all neighbours nodes but not retransmitted.

For adequate operation each node will then maintain a table with a list of all the nodes it can see either directly or indirectly. Links to one hop neighbours are tagged as either unidirectional, bidirectional or MPR. Each table entry has a both a sequence number and a timeout value associated, so that old entries can be removed.

2.7.5.4 Multipoint Relay Selection

Each network node independently chooses its MPR set. Maintaining a list of the two-hop neighbours requires analysing HELLO messages and filtering all the unidirectional links. The MPR set is only altered when a change is detected in terms of one-hop or two-hop neighbours (bidirectional connections only).

2.7.5.5 MPR Information Broadcasting

Each node must broadcast topology control messages (TC) So that all nodes maintain their database updated. These messages are broadcasted throughout the network using a technique similar to the one used for traditional link-state routing protocols, with the only difference that it employs MPRs to improve scalability.

A TC message is sent periodically to each network node to declare its MPR selector set. This means that the message must contain a list with those direct

neighbours that have selected it as their MPR. This list has always a sequence number associated.

The list of addresses on each TC message can be partial, but it must be complete before each refresh period ends. These messages will allow each node to maintain its own table with the network topology. If a node has not been selected as any other node's MPR it does not send TC messages, thereby saving power and bandwidth.

The interval between the transmission of two TC messages depends on whether there have been changes on a node's MPR selector set. If so, the next TC message can be transmitted before the time scheduled, though respecting the minimum inter-message time.

2.7.5.6 Calculation of the Routing Table

Each node maintains a routing table with information on how to access other network terminals. When nodes receive a TC message they store sets of two addresses indicating the last hop before reaching a certain destination node, as well as the destination node itself. By combining the information in these address pairs the node is able to find what is the next hop towards a certain destination node. Minimum distance criteria should be followed to restrict the search options.

Routing table entries are composed of destination, next hop and estimated distance to destination. On this table we only register those entries for which the route towards the destination is known. This means that the routing table must be constantly updated according to the topology changes detected.

In a real implementation the OLSR daemon must update the kernel's forwarding table according to the routing table it maintains, so that packets are sent through valid routes.

2.7.5.7 Versatility of the OLSR

As described in Section 2.3.2.2, for a community network it is imperative that the routing protocol can be installed on any device which is part of the network. The *olsrd* implementation of OLSR can be installed on several devices such as iPhone, Nokia 770, Android, Windows or any Linux System such as Ubuntu, DD-WRT or OpenWRT. Using this implementation ensures that any user can easily connect to the wireless community network without purchasing specific devices.

2.7.5.8 Advantages and Disadvantages

Being a proactive protocol, OLSR uses power and network resources in order to propagate data about possibly unused routes. While this is not a problem for wired access points and laptops, it makes other devices such as mobiles or sensors to spend battery and bandwidth only for maintaining the routes. Nevertheless, all routes are refreshed constantly and, when a node needs to send data through the network, it can use the route immediately.

2.7.6 Ad hoc On-Demand Distance Vector Routing (AODV)

2.7.6.1 Basic Principles

AODV is an on-demand routing protocol for ad hoc networks which uses hop-by-hop routing by maintaining routing table entries at intermediate nodes.

2.7.6.2 Route Discovery

The route discovery process is initiated when a source needs a route to a destination and it does not have a route in its routing table. To initiate the route discovery, the source floods the network with a RREQ packet specifying the destination for which the route process is requested. When a node receives an RREQ packet, it checks to see whether it is the destination or whether it has a route to the destination. If either case is true, the node generates an RREP packet, which is sent back to the source along the reverse path. Each node along the reverse path sets up a forward pointer to the node it received the RREP from. This sets up a forward path from the source to the destination. If the node is not the destination and does not have a route to the destination, it rebroadcasts the RREQ packet. At intermediate nodes duplicate RREQ packets are discarded. When the source node receives the first RREP, it can begin sending data to the destination. To determine the relative out-of-dateness degree of routes, each entry in the node routing table and all RREQ and RREP packets are tagged with a destination sequence number. A larger destination sequence number indicates a more current (or more recent) route. Upon receiving an RREQ or RREP packet, a node updates its routing information to set up the reverse or forward path, respectively, only if the route contained in the RREQ or RREP packet is more recent than its own route.

2.7.6.3 Route Maintenance

AODV maintains routes for as long as the route is active. Because the network nodes are mobile, it is likely that many link breakages along a route will occur during the lifetime of that route. When a node detects a broken link while attempting to forward a packet to the next hop, it generates a RERR packet that is sent to all sources using the broken link. The RERR packet erases all routes using the link along the way. If a source receives a RERR packet and a route to the destination is still required, it initiates a new route discovery process. Routes are also deleted from the routing table if they are unused for a certain amount of time.

2.7.6.4 Neighbour management

Detecting the availability of new neighbours and, more important, detecting that a neighbour has become unavailable is an important issue in mobile ad hoc networks. AODV allows nodes to detect their neighbours in two different ways. When a neighbour receives a broadcast packet from one of his neighbours it updates its internal tables to include that neighbour. Sometimes a node does not send any

packets downstream, which could cause downstream nodes to consider it unavailable. In those cases such nodes broadcast an unsolicited RREP packet within a *hello_interval* time so that downstream nodes become aware of their liveliness. This packet has a TTL of 1, and so it is not re-broadcasted. When a node fails to listen up to *allowed_hello_loss* consecutive packets from a node participating on an active path, it considers the link with the upstream node to be lost and, therefore, generates a route error message to be sent to the source of that stream.

In a real implementation this technique can suffer in those situations where the channel is very congested, causing frequent losses of broadcast packets. Nodes could then erroneously infer that a link is down, when in fact it is not.

2.7.6.5 Advantages and Disadvantages

An important advantage of AODV is that it generates no extra traffic for communication along existing routes. Also, distance vector routing is simple and doesn't require much memory or calculation. However, AODV requires more time to establish a connection and, besides, the initial process required to establish a route provokes more routing overhead than proactive approaches.

2.7.7 Other Protocol specifically used in Mesh Networking: B.A.T.M.A.N.

The Better Approach To Mobile Ad hoc Networking, or B.A.T.M.A.N. [JNA08], is a routing protocol which is currently under development by the "Freifunk"-Community [Fre].

B.A.T.M.A.N.'s crucial point is the decentralisation of the knowledge about the best route through the network: no single node has all the data. Using this technique, the need for spreading information concerning network changes to every node in the network becomes superfluous. The individual node only saves information about the "direction" it received data from, sending its data accordingly. Hereby the data gets passed on from node to node and packets get individual, dynamically created routes. A network of collective intelligence is created.

On a regular basis, every node sends out a broadcast thereby informing all its neighbours about its existence. The neighbours then relay this message to their neighbours and so on and so forth. This carries the information to every node in the network. In order to find the best way to a certain node, B.A.T.M.A.N counts the originator-messages received and logs which neighbour the message came in through.

Like distance-vector protocols, but unlike link-state protocols, B.A.T.M.A.N does not try to determine the whole path, but, by using the originator-messages, only the packet's first step in the right direction. The data is handed over to the next neighbour in that direction, who in turn uses the same mechanism. This process is repeated until the data reaches its destination.

B.A.T.M.A.N consumes less resources than other protocol such as OLSR and causes less disturbances into the network, but it is only useful in networks where the mobility is reduced such as urban networks because it needs more time to

recalculate a new route when the previous one fails. B.A.T.M.A.N. could be also a candidate approach in mesh networking.

2.8 Security on MANETs and Wireless Mesh Networks

Unlike the wired networks, the unique characteristics of mesh networks and mobile ad hoc networks pose a number of nontrivial challenges to security design, such as open peer-to-peer network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology.

Existing protocol specifications, like AODV or OLSR described before, cope well with changes in the network topology. However, defence against malicious attacks is an issue which has not been addressed in the specifications. The majority of these protocols assumes that each node in the network is a peer and not a malicious node. Therefore, a single attacking node can cause the entire network to fail, i.e. flooding the MANET with useless packets or disseminating false routes to prevent the MANET from working.

Then, trustworthiness is essential for the practical exploitation and use of MANETs and mesh networks. In such context, network availability is a minimum requirement. To achieve such requirement, routing protocols should be robust against both topology changes and malicious attacks. Therefore, there is an emerging need of research focused on the provision of practical proposals for securing ad hoc routing protocols [Hao04].

2.8.1 Challenges

One fundamental vulnerability of MANETs comes from their open peer-to-peer architecture. Unlike wired networks that have dedicated routers, each mobile node in an ad hoc network may function as a router, forwarding packets for other nodes. The wireless channel is accessible to both legitimate network users and malicious attackers. As a result, there is no clear line of defence in MANETs from the security design perspective. There is no well defined infrastructure where we may deploy a single security solution.

Moreover, portable devices, as well as the system security information they store, are vulnerable to compromises or physical capture, especially low-end devices with weak protection.

The stringent resource constraints in MANETs constitute another nontrivial challenge to security design. The wireless channel is bandwidth-constrained and shared among multiple networking entities. The computation capability of a mobile node is also constrained. For example, some low-end devices, such as PDAs, can hardly perform computation-intensive tasks like asymmetric cryptographic computation. Because mobile devices are typically powered by batteries, they may have very limited energy resources.

The wireless medium and node mobility pose far more dynamics in MANETs compared to the wire-line networks. The network topology is highly dynamic as nodes frequently join or leave the network, and roam in the network at their own

will. The wireless channel is also subject to interferences and errors, exhibiting volatile characteristics in terms of bandwidth and delay. Despite such dynamics, mobile users may request for anytime, anywhere security services as they move from one place to another.

Therefore, any proposed solution deployed against an attack must consider these constraints: mobility of the network, computation capability, limited energy resources and high level of interferences in the wireless channel.

2.8.2 Routing disruption attacks

As reported in [HP04], one of the main attacks against ad hoc networks affects their routing protocols and are named routing-disruption attacks. Such attacks can be considered as instances of a denial-of-service (DoS) attack since they compromise the routing process, thus affecting the availability of certain (or all) network and application services.

An example of a routing-disruption attack is for an attacker to send false routing packets to cause the routing protocol to misbehave. An attacker might create a routing *black hole* with the objective of dropping data packets. An attacker creates a *black hole* by distributing forged routing information (that is claiming falsified short distance information); the attacker attracts traffic and can then discard it. Figure 2.6 shows how a malicious node perform this kind of attack.

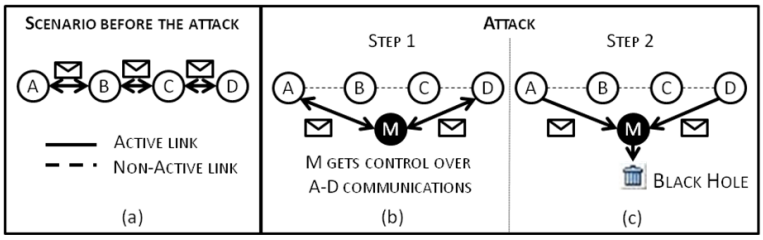

Figure 2.6: Steps for a black hole attack.

1. The malicious node (M) induces a network topology propitious for the attack success (Figure 2.6.b). To cope with that goal (i) M induces a possible routing link between attack targeted devices (call them A and D), then (ii) M emits protocol-compliant messages for leading both A and D to choose such link for their communications.

2. M carries out the attack (Figure 2.6.c). In the case of a black hole attack, M drops (does not retransmit) the packets.

In a special case of a *black hole*, and attacker could create a *grey hole*, in which it selectively drops some packets but not other, for example, by forwarding control packets but not data packets.

A similar kind of routing disruption is the selfish node. A selfish node uses the network but does not cooperate, saving battery life for its own communications; it

does not intend to directly damage other nodes. The selfish attack is very common in community networks, where a node does not cooperate not only to save battery, but also to increase its bandwidth by disrupting other traffic flows if the network is congested. In practise, a selfish node can be considered as a *black hole*.

In some recent publications [R+08], authors have explored to what extent existing ad hoc networks routing protocols, like OLSR or AODV, are sensible to selfish nodes, *black holes* and *grey holes* attacks. Results are clear: although these attacks are well-known, their detection and tolerance remain in most of the cases an optional feature rarely implemented in practise. Therefore an extra effort must be done to develop new tools to react against these malicious behaviours.

2.8.3 Watchdogs

In this context, intrusion detection systems (IDS) aim at monitoring the activity of the nodes in the network in order to detect misbehaviour. A basic brick in the construction of such systems is the watchdog, a sensor that can detect selfish nodes and *black hole* attackers.

The watchdog is an intrusion detection mechanism proposed in previous studies [STKM00, CLY06] that copes well with the challenges described before. It ensures that an intermediate node of a traffic route has the expected behaviour for a standard MANET's node, forwarding the packets to the final destination. The watchdog controls it by listening to the next node promiscuously. If the next node does not forward the packet, then it is termed as misbehaving. The main advantage of the watchdog is to offer a node the possibility of detecting an attacker using only local information, thus avoiding that a malicious node affects the decisions made by the mechanism and reducing the traffic overload in the MANET that other solutions may cause. Therefore, results provided by watchdogs are exploited by other reputation systems, like in the case of the Pathrater [MGLB00] and Routeguard [HZH05] solutions. Such systems then isolate and/or punish misbehaving nodes or routes by decreasing their trustability rates. Although this methodology should be enough to detect malicious nodes, collisions and signal noise make that, in practise, the false detections problem will appear.

False detections is not the only problem of the watchdog. It is well known that the watchdog is unable to detect cooperative attacks. Some previous works [TS08, WF07, Deb08] define techniques for avoiding the problem of cooperative black-holing in MANETs. However, they have some limitations. For example, all of the described methodology is based on the AODV protocol and [TS08, WF07] need to change the implementation of AODV. In Chapter 5 we perform a deeper study of the watchdog and we propose improvements to this strategy to avoid its limitations.

2.9 Methodology Used to Evaluate MANET and Mesh Networks Proposals

Deploying and testing applications and new protocols for MANETs and mesh networks involves high cost and intensive labour. Hence, simulation and emulation

is a useful alternative prior to evaluating actual implementations. In this section we study the differences between emulation and simulation for evaluating a research work.

2.9.1 Importance of Evaluation in Research

Research efforts focusing on ad hoc networks and mesh networks are growing every year, and it is important to have tools that allow researchers to evaluate their proposals before investing huge amounts of money on it. Testing and evaluating a MANET protocol is, therefore, mandatory for its success in any real word application. Researchers in this field have two options for testing new MANET protocols: (a) simulation tools, and (b) emulators.

2.9.2 Simulators

A simulation tool is a program or system used during software verification, which behaves or operates like a given system when provided with a set of controlled inputs.

Currently, several simulators exist, like ns-2 [UBr98], OPNET [OPN10], Seawind [MAJ+01], GloMoSim [XRM98], REAL [Kes88] and JiST/SWANS [RZR04, RZR05, RZ04]. Computer simulation is the most popular way to evaluate MANET routing protocols [JDD+98, PTN+99, SCE00] being ns-2 one of the most extended under the research community.

2.9.2.1 Ns-2

Ns is an object oriented simulator, written in C++, with an OTcl interpreter as a front-end. The simulator supports a class hierarchy in C++, and a similar class hierarchy within the OTcl interpreter. Ns-2 uses two languages because the simulator has two different kinds of tasks it needs to do. On one hand, detailed simulations of protocols requires a systems programming language which can efficiently manipulate bytes, packet headers, and implement algorithms that run over large data sets. For these tasks run-time speed is important and turn-around time (run simulation, find bug, fix bug, recompile, re-run) is less important. On the other hand, a large part of network research involves slightly varying parameters or configurations, or quickly exploring a number of scenarios. In these cases, iteration time (change the model and re-run) is more important. Since configuration runs once (at the beginning of the simulation), the run-time of this part of the task is less important.

Ns-2 is a discrete event network simulator and it is popular in academia for its extensibility (due to its open source model) and plentiful on-line documentation. Ns-2 is popularly used in the simulation of routing and multicast protocols, among others, and is heavily used in ad-hoc research. Ns-2 supports an array of popular network protocols, offering simulation results for wired and wireless networks alike.

2.9.3 Emulators

An emulator is a platform for experimentation. Emulators allow for rigorous, transparent and replicable testing of scientific theories, computational tools, and other new technologies.

2.9.3.1 Comparative among Emulators

The idea of automatic computer-controlled routing in a mesh network is not new; in fact the main idea dates back to Paul Baran in the early 1960s [Bar64]. Since then, several prototypes for generating real ad hoc network experiments were proposed and can be found in the literature:

ORBIT: an indoor radio grid emulator for controlled experimentation and an outdoor field trial network for end-user evaluations in real-world settings [DIM+05]. This emulator needs an expensive noise generator since it emulates higher node distances by reducing the signal-to-noise ratio. It also requires investing a high budget to create the grid of nodes (each computer is a possible position of the node in a simulation), as well as extra support servers for data storage. Thus, deploying the entire infrastructure requires a lot of room.

mLab: an emulator that strikes a balance between desktop simulations and outdoor field tests by allowing users to develop and test ad hoc protocols and applications in a laboratory environment [AE05]. This emulator can only generate network topologies and capture packets.

Carnegie Mellon University Wireless Emulator: supports real devices, applications, MAC and PHY layers on a network-wide scale while maintaining experimental control and repeatability [GP07]. The disadvantages of this emulator are that it does not use commercial off-the-shelf devices, using a FPGA for digital emulation instead.

MobiEmu: an emulator to test an ad hoc network of any scale and with any mobility scenario without actually moving the ad hoc nodes physically [ZL02]. We discarded this emulator for our tests since it relies on expensive clusters to emulate the scenario.

Seawind: another emulator designed for performance studies of real protocols and applications on wireless networks [MAJ+01]. It uses an emulated link and router to generate the network topology and non-commercial off-the-shelf devices, making it difficult to scale up the network. Also, it can only emulate one traffic flow.

WHYNET: this emulator is a hybrid wireless test bed environment targeting realistic, scalable and flexible evaluation of wireless technologies and applications [JZM+06]. These hybrid experimentation modes use physical and simulated elements (e.g., protocol layers, subnets) in different combinations. WHYNET uses a geographically distributed set of physical wireless test beds, making results difficult to be replicated by other researchers.

Name	GUI	Real Devices	Real Code	Flexibility	Size of Emulator	Initial Budget	Scalable	Free
Castadiva	Yes	Routers, PDAs and computers	Yes	Yes	Fits in a desktop	<1000€	Yes	Yes
MobiEmu	No	Computers	Yes	Yes	Fits in a room	>3000€	Yes	Yes
mLab	No	No (is almost a simulator)	No	Yes	Fits in a desktop	<1000€	Yes	Yes
Carnegie Mellon University Wireless Emulator	Yes	Computers, but needs FPGAs	Yes	Yes	Fits in a desktop	>3000€	Yes	No
ORBIT	No	Computers	Yes	Only mobility in grid.	Huge, needs a warehouse	>10000€	Yes, but very expensive	No
Seawind	Yes	Computers	No	No, only emulate two nodes	Fits in a desktop	–	No	No
WHYNET	No	Hybrid simulator	Yes	Yes	Fits in a desktop	<1000€	Yes (by simulation)	No

GUI: a graphical interface for an easy use of the emulator.

Real Devices: which devices can be used in the emulation tests.

Real Code: can users introduce real implementations of applications and protocols in the emulations without doing an extra programming effort?

Flexibility: can any topology be implemented?

Size of the Emulator: can the emulator be deployed on a simple desktop of a lab, or does it need an entire room for its deployment?

Initial Budget: the initial (approx.) inversion for generating a test bed of 10 nodes.

Scalable: is it easy to increase the number of nodes without increasing the deploying cost significantly?

Free: can it be used freely by anyone without purchasing a license?

Table 2.1: Comparative of existing emulators.

Castadiva: We developed *Castadiva* to deploy a cheap architecture that relies on low-cost devices to generate a test bed. The test bed obtained effectively uses low-cost hardware to achieve a complex, yet reliable wireless network simulation.

Table 2.1 compares the different emulators available against *Castadiva*. As can be observed, only three emulators are free: *Castadiva*, MobiEmu and mLab. MLab is almost a simulator since it does not use real devices. *Castadiva* can use any devices with a wireless card, which offers flexibility when purchasing devices, thus reducing the budget to deploy the emulator. MobiEmu is more expensive than *Castadiva* and does not have a GUI. *Castadiva* implements a GUI to guide the users when creating the scenario and the traffic pattern.

Overall, *Castadiva* outperforms the other emulators in terms of: (a) variety of devices that can be use as nodes, (b) initial budged needed to deploy the emulator and (c) ease of use provided by the GUI; this means that it makes a clear contribution to the research community.

In Chapter 3, we explain deeply how *Castadiva* works.

2.9.4 Main Differences between Simulators and Emulators

Simulation offers four important advantages [AGT+04] compared to emulators: First, it enables experimentation with large networks. Second, it enables experimentation with configurations that may not be possible with existing technology. Third, it allows rapid prototyping. Finally, it makes reproducible experiments in a controlled environment. On the other hand, simulations also have some disadvantages that emulators do not have: first, researchers cannot test their own real world implementation of a protocol in a realistic scenario, and second, simulators also need to incorporate realistic models of node mobility and radio propagation. Then, we can use a simulator to test a protocol or tool in a large network with a high number of nodes, but if we want test the final implementation, we should use an emulator. In this thesis, we use *Castadiva*, our own implementation of a MANET emulator, for testing the behaviour of our proposed architecture using real devices. However, for testing some large scenarios used in Chapter 5, we also use the ns-2 simulator, taking advantage of the simulations benefits. In the next chapter we introduce *Castadiva* presenting its novelties in comparison with other emulators.

Chapter 3

Castadiva: a MANET Emulator

In this chapter we present *Castadiva*, a test bed architecture that simplifies carrying out realistic experiments; it relies on low-cost, off-the-shelf wireless routers combined with a Linux platform. *Castadiva* allows generating network topologies, exporting them to real devices and obtaining the test results. It can also generate different types of traffic between nodes, and it offers support for some well-known ad-hoc routing protocols, i.e. AODV and OLSR. It relies on a cheap architecture that includes two different networks: a wired network, called connection network, that connects the server with a group of wireless nodes, and a wireless network where the actual test bed experiments are made. We developed a group of tools for administration purposes, with a friendly user interface designed to help the user to define the scenario of the network and the desired traffic connections between MANET nodes in a simple and straightforward manner. All of these tools were developed with open source software, and they are freely available for the research community at `http://castadiva.sourceforge.net/`.

The rest of this chapter is structured as follows. Section 3.2 explains the objectives of *Castadiva*. Section 3.3 describes the proposed architecture and Section 3.4 describes its implementation. Section 3.5 describes the evaluation made to validate our *Castadiva* test bed. Finally, Section 3.7 draws the conclusions of this chapter.

3.1 Introduction

Research efforts focusing on MANETs and mesh networks are growing every year, and it is important to have tools that allow researchers to evaluate their proposals before investing huge amounts of money on it. Testing and evaluating protocols or tools for these networks is, therefore, mandatory for its success in any real word application.

A test bed is a platform for experimentation in large development projects. test beds allow rigorous, transparent and replicable testing of scientific theories,

computational tools, and other new technologies. Several prototypes for generating a real test bed can also be founded in the literature. But all available test beds have some disadvantages: they use expensive clusters for testing, or their functionality is limited (see Table 2.1).

Therefore, we need to develop a test bed which allows us to perform our evaluation tests to ensure the correct behaviour of our community network. In this sense we create *Castadiva*, a test bed for studying MANETs and mesh networks.

3.2 Objectives of *Castadiva*

The objective of this chapter is to develop a test bed where evaluating our proposed architecture for deploying community networks. This test bed is called *Castadiva*.

Castadiva is designed for emulating MANETs where we can test the behaviour of the network with different routing protocols and different real applications. Our test bed should use a low cost architecture based on commercial-of-the-self devices and free software. We also wish to generate an application with an user-friendly interface that allows defining the network's scenario. *Castadiva* must be compatible with the ns-2 simulator used in our group for comparison purposes and it should have an user-friendly interface that allows the user to generate the entire simulation easily.

3.3 Architectural Overview

Castadiva is a test bed designed to deal with the development and the performance evaluation of protocols and applications for MANETs. The test bed relies on actual IEEE 802.11 wireless interfaces for communication among nodes. *Castadiva* is composed by a server that runs the main application, several wireless nodes, two different networks (wired and wireless), and an application that coordinates all devices.

Castadiva's server executes the application and configures the network devices. Concerning the wireless nodes used, they can be any sort of computing device, like a laptop, a PDA or a wireless router. In our prototype, each node is a Linksys router [Lin10] to minimise costs. The main requirement for a node is that it must have a Linux/Unix operating system installed, and two network cards: an Ethernet card and an IEEE 802.11 card. If the node is a wireless router, the OpenWRT [Ope10] kernel is an excellent solution. OpenWRT is an open source operating system available for a wide range of router manufacturers. This embedded Linux system natively offers SSH connections, along with the possibility of running shell scripts. Moreover, a programmer can develop its own application in a standard Linux distribution and export it to this operating system. In our case, we developed some applications in C for traffic generation and control purposes.

Figure 3.1 shows a schema of the *Castadiva*'s architecture. The main application, developed in Java, controls all devices and dynamically manages the links among them according to the desired network scenario. It also manages traffic generation between pairs of nodes. Since the controlling application requires com-

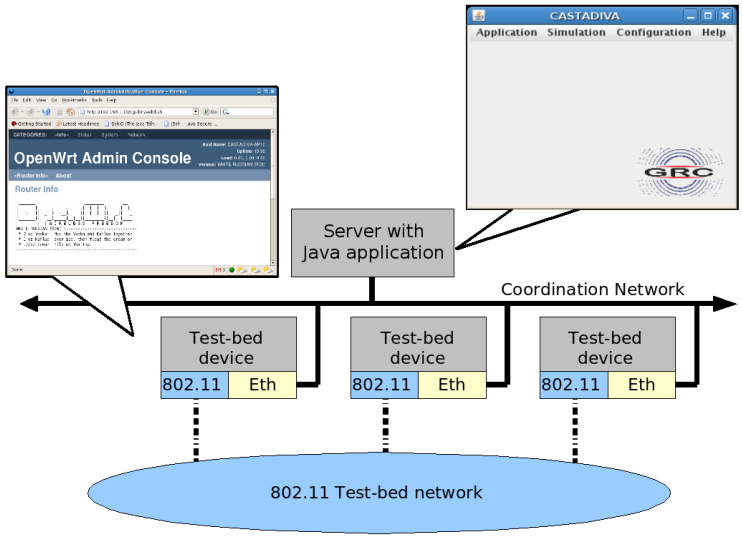

Figure 3.1: Schema of *Castadiva*'s architecture.

municating with nodes to send control packets, *Castadiva* combines two different networks: the coordination network (wired), that connects the *Castadiva* core with the wireless nodes, and the wireless network, where actual tests run.

The coordination network is a wired network that connects *Castadiva*'s core server with the wireless nodes. This network allows the main application to send configuration messages to all the nodes without creating any interference within the wireless network itself. It is based on Fast-Ethernet technology, to avoid large latencies. Basically, the coordination network requires a switch connected to the main server and to all nodes. Through this network the main application sends instructions to nodes, allowing them to re-configure periodically according to the desired network topology, and also to run lightweight traffic-generating applications available on each wireless node. For communication purposes, we rely on the SSH protocol [TC06] to send instructions through this network. Using a fast network means that all nodes participating in a test will start at about the same time, avoiding significant latency effects and maximising result accuracy.

The wireless network is composed by *Castadiva*'s wireless nodes, and the topology of this network is defined by the GUI of *Castadiva*, so that it can change at runtime. Nodes communicate in ad hoc mode using the IEEE 802.11g technology.

Castadiva's core has two main functions: (a) to allow a user to interact with the system so as to define all the test parameters required, and (b) to coordinate the wireless nodes during an experiment. By using *Castadiva*'s GUI a user controls all of *Castadiva*'s functionality, defining the network topology and the traffic flow among nodes. *Castadiva* allows fixing the scenario area where nodes will be deployed. When selecting a node, its location is highlighted and it can be changed according to the desired network topology. When all nodes are deployed the user

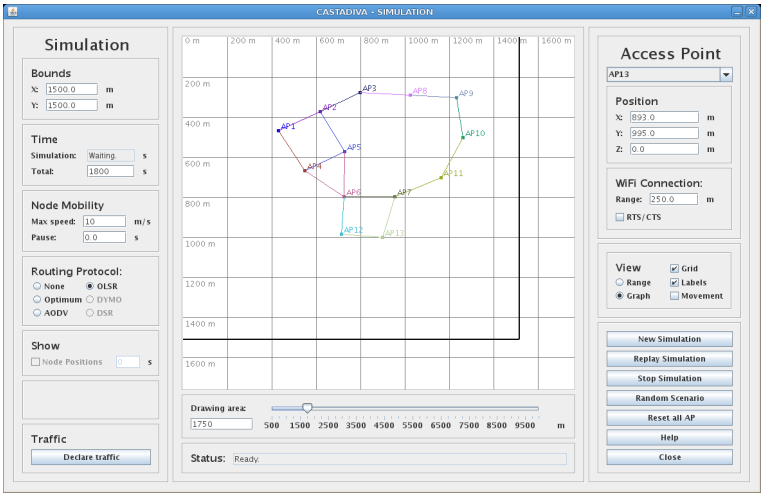

Figure 3.2: Scenario definition with *Castadiva*.

can press the *Simulate* button, and each physical node will be re-programmed so as to enforce the chosen network topology. Figure 3.2 shows the GUI application of *Castadiva*. We describe the whole functionality offered by *Castadiva*'s GUI in Section 3.4.2.

Figure 3.3 shows our test bed, where one switch connects *Castadiva*'s server with all the wireless nodes for coordination purposes. In the middle of the picture the group of wireless nodes being used is shown. It consists of twenty-four Linksys routers (models WRT54G and WRT54GL). The wireless ad hoc network conformed by these nodes is the one used in *Castadiva*'s test bed experiments. The experiments presented in this work required extending *Castadiva* to support traffic injected from outside applications. In particular, we connected laptops running video-conference software to our MANET, allowing us to assess the performance of video conference sessions as experienced by users.

3.4 *Castadiva*'s Implementation Details

In this section we detail the requirements of *Castadiva* on the server and on the wireless nodes. We describe the software tools we have developed to connect all the wireless nodes with the server, and how *Castadiva* allows emulating connections among them. We also explain the process of designing network topologies by using the Scenario Generation tool, an interactive and user-friendly interface that allows defining the network's scenario and the desired traffic connections among nodes.

Castadiva requires some libraries and services to operate. The requirements of *Castadiva* are different for the server and the wireless nodes. The server must be a standard Linux-based system and must have a Java Virtual Machine, an SSH client and an NFS server installed. Concerning nodes, each one must be a Linux based system with an SSH server and an NFS client; besides, they must have the

Figure 3.3: *Castadiva*'s physical network.

libgcc library and have the *Iptables* tool-set installed (see figure 3.4).

The connection between *Castadiva*'s core element (server) and each node is made using both SSH and NFS connections. On *Castadiva*'s server, the user interacts with the GUI application by defining the network topology, the traffic and selecting the desired routing protocol. Then, through SSH, the application sends a *start* instruction to each node through the coordination network (wired). Wireless nodes achieve coordination among themselves by executing the required binaries, which are stored in a server folder shared through NFS. This is an easy way to spread instructions to all nodes, and it also solves storage limitation problems on nodes. When tests start, a group of files with the results are created and stored into *Castadiva*'s server. Each individual router accesses its own configuration file by relying on the NFS file system. We find that Ethernet connections are fast enough to export these files to the routers without significant delays.

The main application parses the results, thus obtaining the different test bed statistics. Finally, the application displays results to the user.

Next we present the *Castadiva*'s implementation, which can be divided into two parts: the lightweight applications running on wireless nodes, and the main application.

3.4.1 Wireless Nodes' Software

Each node has a set of requirements that must be met for successful operation: a Linux-based operating system, a set of special-purpose scripts, some specific applications, and connectivity to the *Castadiva*'s server.

The operating system installed on each router is OpenWRT. OpenWRT allows executing BASH scripts natively; besides, it includes Dropbear, a simple SSH server used to receive instructions from *Castadiva*'s server. Concerning the set of *Castadiva*'s scripts, they are generated automatically by *Castadiva*'s main application. Their purpose is to configure the wireless network topology.

Each node makes use of three applications: Iptables, TcpFlow, and UdpFlow.

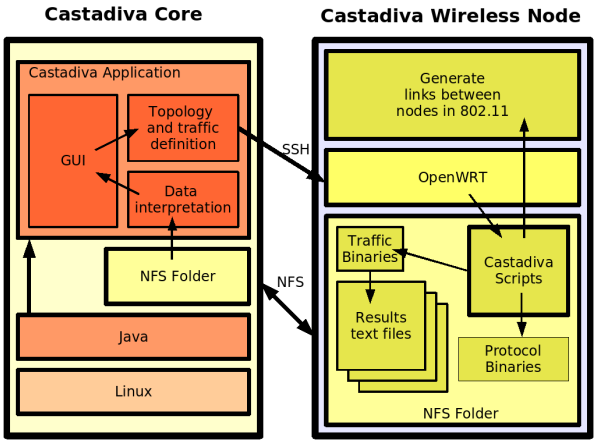

Figure 3.4: Software components for *Castadiva*.

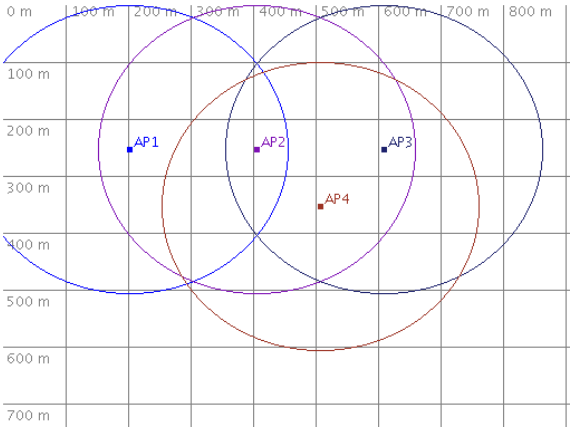

Figure 3.5: Example of a scenario with four nodes.

The first one is open source and exists in most Linux distributions, while the other two were developed for our own purposes.

Network topology configuration relies on the Iptables [Hub03] tool. According to the selected topology, Iptables allows us to dynamically break the network links between pairs of nodes. This tool exists for all Linux distributions, including the OpenWRT embedded system. In Table 3.1 we show an example of Iptables' rules generated from *Castadiva* in the scenario with four nodes shown in Figure 3.5.

To generate traffic we created the UdpFlow and TcpFlow tools. Both tools are designed to generate a traffic flow between two nodes, being that each tool creates a single traffic flow. To create a flow of data we must specify a source/destination pair, the starting and ending times for this flow, and the maximum amount of bytes to be sent.

Node (MAC)	Iptables Rules.
1 (00:13:10:83:99:BE)	/usr/sbin/iptables -I INPUT -m mac –mac-source 00:13:10:83:99:CA -j DROP /usr/sbin/iptables -I INPUT -m mac –mac-source 00:18:39:BC:B5:8E -j DROP
2 (00:0F:66:D9:BE:01)	(none)
3 (00:13:10:83:99:CA)	/usr/sbin/iptables -I INPUT -m mac –mac-source 00:13:10:83:99:BE -j DROP
4 (00:18:39:BC:B5:8E)	/usr/sbin/iptables -I INPUT -m mac –mac-source 00:13:10:83:99:BE -j DROP

Table 3.1: Iptables rules: example of usage in *Castadiva*'s framework.

Castadiva also includes routing agents for well-known routing protocols, such as AODV and OLSR, which are included with the OpenWRT, and initiated according to user settings.

3.4.2 Main Application

Castadiva's core element, a Java application running at the server, includes all the control functions required for test bed experimentation. It is responsible for network topology maintenance, traffic control, as well as result calculation and presentation. A user can define the characteristics of wireless nodes. Each node is deployed at a specific position in a simulated area as chosen by the user. Once the topology is defined, *Castadiva* must configure the wireless nodes according to that topology. The application communicates with each node through SSH connections to send the required instructions. The traffic flow between nodes and the routing protocol used are also set through this technique. When all the experiments are finished, *Castadiva*'s core must calculate the result statistics for the experiment by gathering all the data obtained, and finally show these results to the user.

Castadiva's main application was created using Java's Swing library. We consider that it is a good solution for visual design since most basic components are already created, and can be easily modified by the programmer.

Castadiva is designed to be a test bed where network scenarios and traffic between nodes are generated so as to resemble a real MANET. Therefore, it is expected to be an easy and useful tool for the study of MANETs.

To start a new experiment we only need to define the network topology in the corresponding window and then define the traffic flow and the selected routing protocol. By pressing the start button tests begin, and *Castadiva* returns the test results automatically at the end of the simulation. We now offer more details about the services offered by *Castadiva*.

3.4.2.1 Main Menu

A standard menu allows accessing the different options of *Castadiva*. Basic options were added, allowing a user to save and load a project, or export it to other test

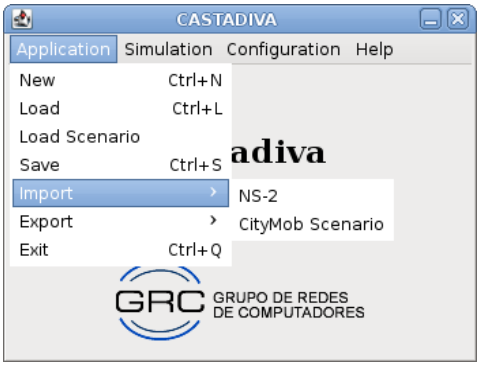

Figure 3.6: Application control menu.

environments such as ns-2. It actually generates all the files required as input to this particular simulator, allowing to compare *Castadiva*'s test results with those obtained through simulation.

Figure 3.6 shows all the options available in *Castadiva*. By selecting the application main menu you may start a new test bed experiment, load a previous one, or save the last one defined. The Simulation menu option opens a window to generate all scenario data. The Configuration option allows a user to adjust server settings, such as defining the NFS folder used. It also allows configuring wireless nodes and adding them to the experiments.

3.4.2.2 Network Scenario Generation

Once all the nodes are defined, they can be configured to create a scenario. *Castadiva* supports both manual and random topology generation, and the scenario is set through *Castadiva*'s blackboard. The blackboard is a representation of a virtual environment where nodes are located. Nodes are differentiated through different colours and labels. If the adequate option is selected, the radio communication range is also shown through a circle of the same colour.

Figure 3.7 shows the topology generation environment. We can see a set of nodes located in a scenario of 1000 x 1000 meters (scenarios bounds are marked with a darker line).

At the right hand we may edit node properties, such as position and signal range. *Castadiva* also can activate or deactivate the RTS/CTS 802.11 option of each node. The group of buttons appearing below allow starting a new test, stopping it, and rebooting nodes to reset all values.

At the left hand, *Castadiva* offers scenario option editing. We can define the scenario bounds, the test time, node mobility and the routing protocol used. The *Declare traffic* button allows setting traffic, as shown later on, and the stop button halts it.

A status bar provides general information to inform the user about what is being done, and the horizontal scroll allows zooming in and out. Finally, the user may alternate between two different views: radio ranges or a graph view. Every

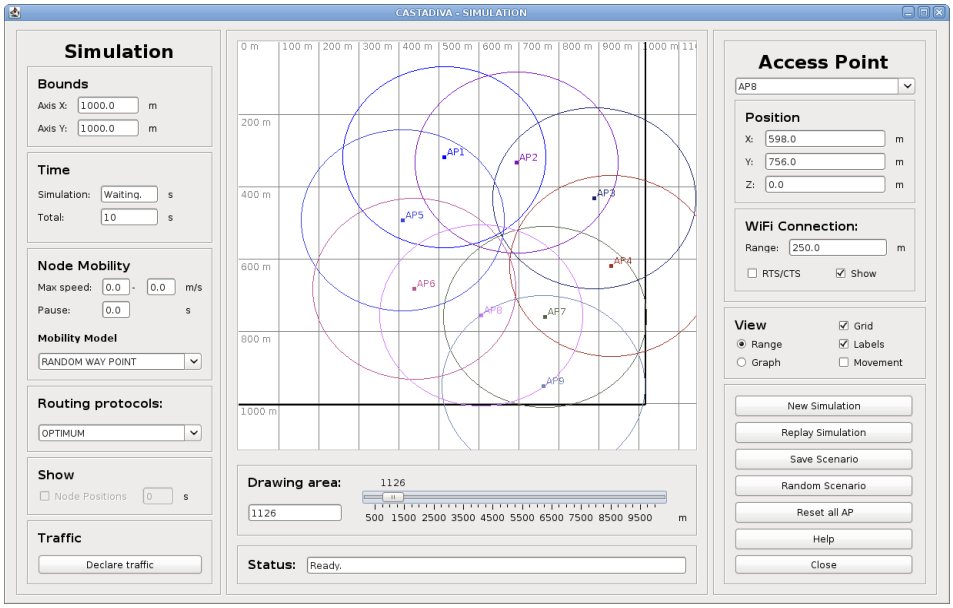

Figure 3.7: Scenario definition with *Castadiva*.

edge of the graph represents an IEEE 802.11 link connection, which is a more intuitive view.

3.4.2.3 Adding Nodes to the test bed scenario

Before starting an experiment the user needs to define the number of participating nodes, along with their configuration. Such information allows *Castadiva* to access nodes and manipulate them to generate a scenario. Figure 3.8 shows an example of the definition of a node in the system.

All the information is defined automatically when the user wishes to add a new one, though it can be changed by the user or can be read from a file. An internal identifier is required to distinguish a node from others in *Castadiva*'s framework. Such identifier is then referenced when defining the network topology and data connections. The remaining parameters will be used by *Castadiva*'s main application to connect nodes among themselves and with the main server. The MAC address is required for *Castadiva* to enforce topology changes.

All the executable files and scripts are stored in an NFS directory that is accessible by all nodes. This way *Castadiva* makes storage capacity independent of wireless nodes' memory.

Castadiva relies on its own tools to generate traffic between nodes. Such tools run at each node, and must be compiled for all types of CPUs used. Currently, tools are compiled for MIPS and Intel processors, though the list can be easily augmented.

The SSH user and password fields are used by the main application to connect

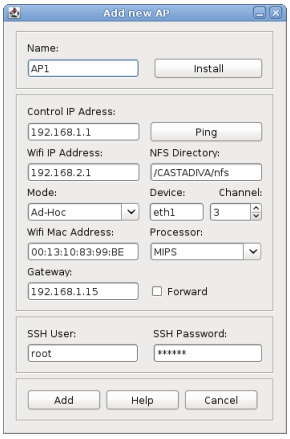

Figure 3.8: Node configuration interface.

to each individual router and submit commands. Also, a Ping button was included to allow testing the connectivity between the server and the routers.

3.4.2.4 Mobility in *Castadiva*

Castadiva offers the user two different approaches for emulating mobility: (a) Any scenario from the ns-2 simulator could be directly used within *Castadiva*, and (b) Users can take advantage of the GUI to select their own mobility model. We already included the random waypoint mobility model [CGP03] for comparison purposes. This behaviour is similar to the one provided by the "setdest" tool embedded in the ns-2 simulator: if a user picks a value greater than zero in the speed option, each node acquires a random movement with a speed between zero and the inserted value. When a node arrives to a destination point, it waits for a selected pause time and then selects a new random destination point to move to. *Castadiva* allows users to easily add new mobility models implementations.

For the emulation of mobility, *Castadiva* generates all node movements required for the test before it starts. In particular it generates, for each node, a mobility vector according to the selected user option in the GUI (see Section 3.4.2.10 to see how to generate a mobility vector), or imports it from a ns-2 scenario. Also, *Castadiva* calculates the variation of the topology continuously in time. Therefore, our emulator obtains the new topology of the network, updating the wireless links on each real device whenever necessary. The granularity used to upgrade the wireless links is of one second, but it can be changed. Since the real devices do not experience real mobility, we emulate the wireless links connectivity between nodes using the Iptables tool. This tool allows each *Castadiva* node to only receive packets from nodes that are within range according to the emulated topology, and thus it blocks the reception of packets when nodes are out of range. All Iptables rules are stored in script files. These scripts will be loaded on each node through NFS when the simulation starts. Figure 3.9 shows the file loaded by AP1. We can see in this figure different Iptables rules combined with *sleep* rules. The sleep

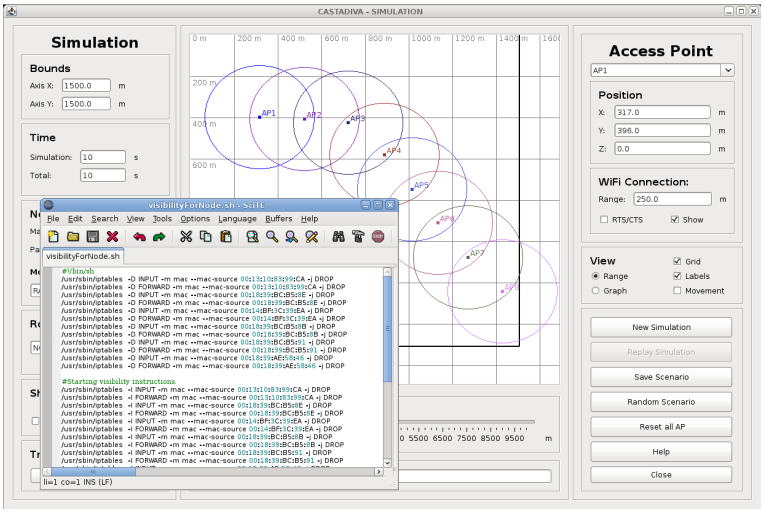

Figure 3.9: Mobility implementation.

Algorithm 3.1 Iptables rules to emulate when a node goes out of range between seconds 15 and 35.

```
sleep 15
iptables -I INPUT -m mac –mac-source 00:14:BF:3C:39:EC -j DROP
sleep 20
iptables -D INPUT -m mac –mac-source 00:14:BF:3C:39:EC -j DROP
```

time allows enforcing Iptables' rules at suitable times. Hence, each rule is loaded only when the emulation requires a node to change its connectivity state towards other node; e.g. Algorithm 3.1 shows the behaviour of *Castadiva* when a node (with MAC 00:14:BF:3C:39:EC) goes out of range at second 15 and comes back into range at second 35.

Castadiva also allows an user to see all node positions at a certain instant of time. When a simulation finishes, the user can activate the *Show* option and pick an instant of time. Immediately *Castadiva* shows the network topology at that time. This option is useful to do a later evaluation of the changes occurred in the network topology when mobility was activated.

3.4.2.5 Network Traffic Declaration

Castadiva's traffic generation tool allows defining different types of traffic flows between pairs of nodes. With that purpose *Castadiva* provides a table where each row defines a connection. Traffic parameters for each connection can be set depending on the type of protocol selected, and invalid values are marked in red. Examples of parameters are: *packet size*, *packets per second*, *start time*, *end time* and *maximum number of packets sent*. Figure 3.10 shows a usage example of this tool, where rows define seven traffic connections. It contains some helpful buttons

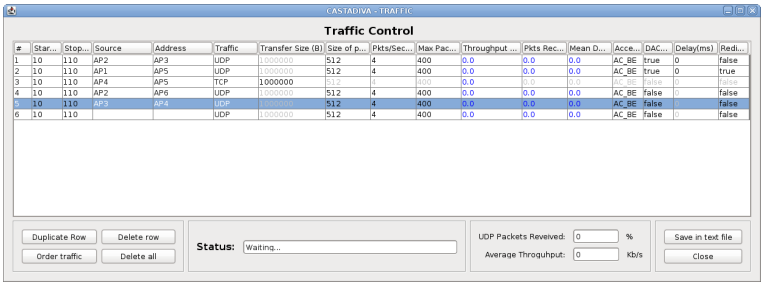

Figure 3.10: Traffic declaration window.

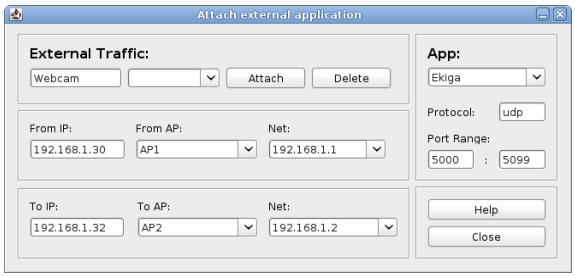

Figure 3.11: External traffic declaration.

that allow making row operations (delete, order by starting time, or copy).

When an experiment finishes, *Castadiva* fills this table with results, including the average throughput, in the case of TCP traffic, or the percentage of packets received in the case of UDP traffic. Note that traffic settings are exportable to NS-2 format also.

3.4.2.6 *Castadiva* Extensions for External Traffic Injection

Such functionality allows external nodes to generate real traffic of any kind and redirect it to specific nodes of *Castadiva*. *Castadiva* also incorporates an extra component that allows attaching a laptop or a computer to a node. Figure 3.11 shows an example of this functionality.

This way we can use laptops to generate any flow of traffic and redirect it to specific nodes of *Castadiva*. For example, you can use real applications like Ekiga to launch a video-conference and study the behaviour of H.323, SIP and video streaming protocols in MANETs.

Figure 3.12 shows an example were two laptops are connected to *Castadiva*. Both laptops have a webcam and run the Ekiga application to generate a videocall. *Castadiva* redirects all traffic related to this video-conference through the emulated MANET, from the entry point to the output point.

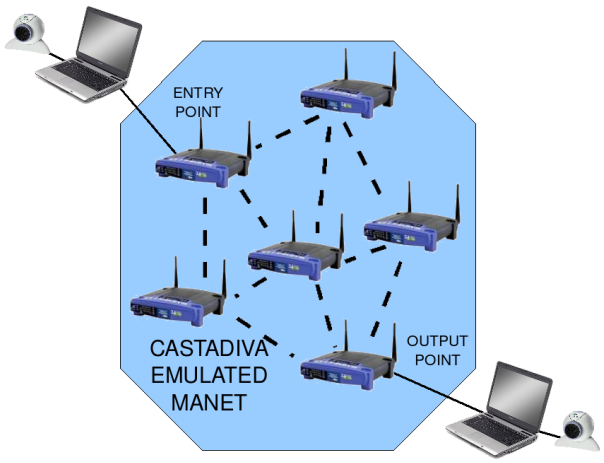

Figure 3.12: Example of how to add external traffic injection.

3.4.2.7 Random Simulation Generator

Sometimes it is useful to automate the test bed evaluation process varying different parameters. With that purpose *Castadiva* includes functionality to generate random tests, where a user can define traffic and automatically test with different number of nodes and randomly-generated network topologies. This is achieved through the *Random test window* shown in Figure 3.13.

The user can specify the bounds of the scenario and the routing protocol used. The minimum and maximum number of nodes for testing must also be defined, along with the granularity interval. (e.g., with a node interval between 4 and 10 nodes and a granularity of 2, *Castadiva* executes four tests with 4, 6, 8, and 10 nodes). *Castadiva* also allows specifying how many times each test will be repeated, averaging the obtained results.

At the top left side the current generated scenario is displayed, though it cannot be modified. Again, all the tests can be stored in either *Castadiva*'s format or NS-2's format.

3.4.2.8 Execution Planner

This window is designed to allow automatic execution of various and possibly different simulation scenarios. The user can generate a list of previously saved scenarios, indicate how many times each simulation should be run and let the system process itself. Results for each simulation are written in a text file.

This window also allows the user to set a timeout, which will delay the start of the simulations. For different reasons, it can be useful to launch the simulation at a specific time, such as during the night or the weekend. For example, if the test are made in a university with a public Wi-Fi, it can cause interferences with the tests, and so performing simulations during the night can palliate these interferences.

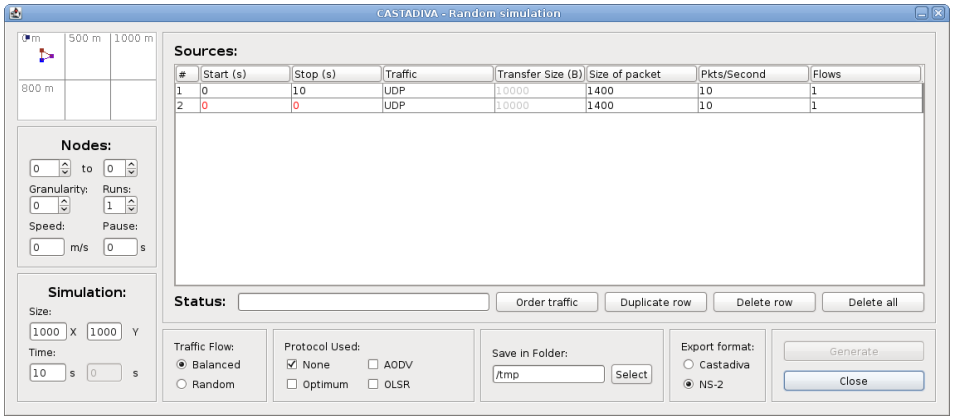

Figure 3.13: Random Simulation window.

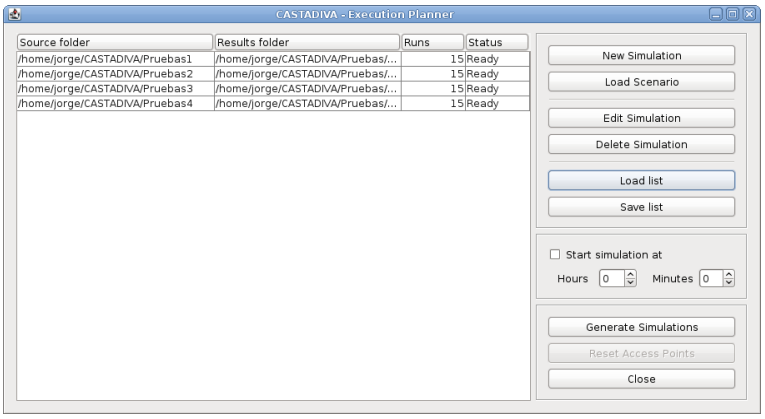

Figure 3.14: Execution Planner.

Figure 3.14 shows an example of use of this functionality. We can see a list of scenarios on the left of the windows, and a group of buttons on the right to add new scenarios or start the simulations group at a specific time.

3.4.2.9 Adding New Routing Protocols

Routing Plugins allow something simple but also very useful. When a routing plugin is configured, it receives the instructions of how to start and stop a specific routing protocol on the Access-Points.

Routing Plugins do not install any routing software on the Access-Points. Thus, it is mandatory to previously install the adequate routing software on each Access-Point. The only thing routing Plugins can do is to send start and stop instructions for the routing protocol, and replace configuration files on the Access-Points. This will allow *Castadiva* to integrate the new routing protocol within the interface.

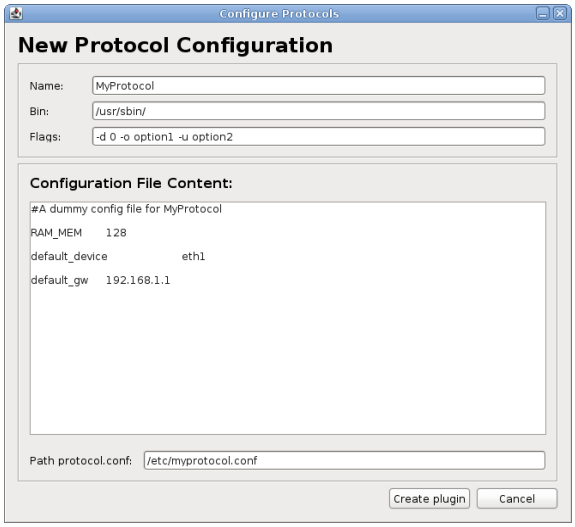

Figure 3.15: New Protocol Window.

Figure 3.15 shows the windows used to add a new routing protocol to *Castadiva*. The *Name* field allows us to identify the protocol and it's configuration. It must be unique and if an existing name is re-used, it will be overwritten. The *bin* field must point to the location of the binary file on the Access-Point, which allows to start the routing protocol. *Configuration file content* is the content of the configuration file that will be written in the Path protocol.conf, on each Access-Point. With this information, *Castadiva* can start the protocol, load the desired configuration of this protocol, and end it when the simulation time is finished.

3.4.2.10 Mobility Plugins Designer

A Mobility Plugin is supposed to enable a custom mobility pattern definition in *Castadiva*. In other words, Mobility Plugins can be configured to add dynamic mobility to *Castadiva*'s scenarios. Controlled mobility allows the usage of an algorithm to define the positions of the nodes during the simulation. The Mobility Plugin enables the system to use several parameters, like the amount of nodes, the maximum speed, and so on.

Figure 3.16 shows the unique user interface that can be used to generate a custom mobility plugin. In the upper part of the figure, we can see the header of a JAVA function. The variables shown in that header are the one that the user can use in order to design his algorithm. The only requirement for the final algorithm is that it entirely fills the *NodeCheckPoint[][][]* nodes matrix. After the Generate Plug-in button is pressed, the program will try to compile the code, and if there is any compilation error, it will be displayed in a special window.

When it is compiled, the Mobility Plugin will be stored and will be available in the Mobility Model drop-list of the Simulation Window for a future use.

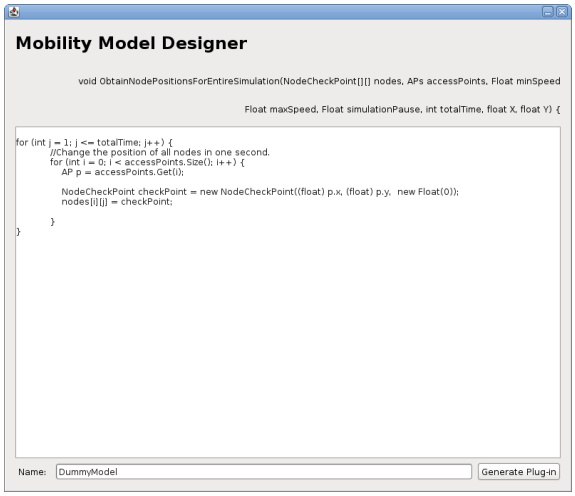

Figure 3.16: Mobility Plugins Designer.

3.4.2.11 Compatibility with the Ns-2 Simulator

Castadiva can also import/export scenarios to/from ns-2 [UBr98], making it compatible with the most widely used MANET simulator. This characteristic offers the possibility of comparing results and reaching more meaningful conclusions.

3.4.2.12 Compatibility with the *Maya* Tool

For evaluation purposes, *Castadiva* can export the topology defined to the *Maya* tool [JJCP07d] (see Section 4.7 for more information about the *Maya* application). This option generates a file that can be used by the mentioned tool to obtain all the information regarding the nodes used in the *Castadiva*'s scenario, and allows *Maya* to use its emulated network.

3.5 Performance Evaluation and Validation of *Castadiva*

In this section we validate the functionality and accuracy of *Castadiva* in different MANET scenarios. We divide our tests in two groups: first, we compare *Castadiva* with the ns-2 simulator using both TCP and UDP traffic in a static scenario. We then proceed with a similar analysis in a mobile scenario.

3.5.1 Evaluation of *Castadiva* with a Static Scenario

To validate the functionality of the proposed tool, and to test its effectiveness, we have chosen a representative scenario where nodes are located in such a way that the maximum number of hops between nodes is of four. The topology used in our

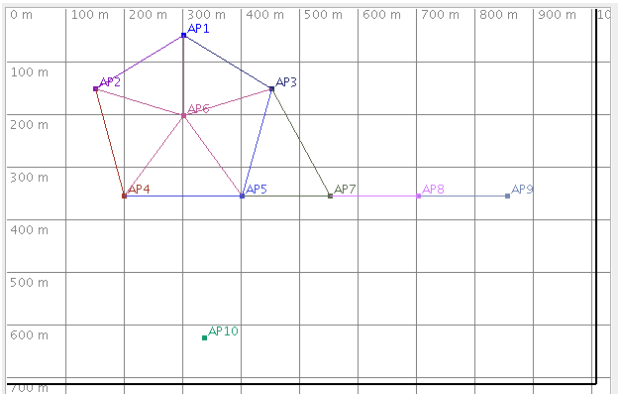

Figure 3.17: Scenario used for evaluation purposes.

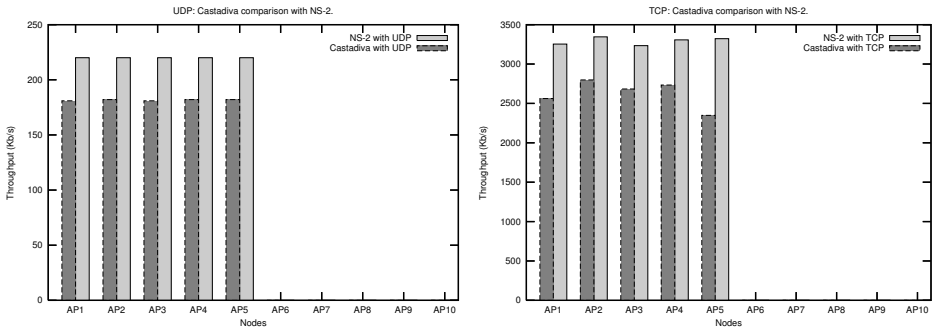

Figure 3.18: Performance comparison between *Castadiva* and ns-2 in a static scenario using CBR/UDP traffic (left) and FTP/TCP traffic (right). Routing disabled.

evaluation is shown in Figure 3.17. The scenario is defined in a 1000m x 700m area, and the test time is of 100 seconds. The selected scenario was generated by ns-2 and imported to *Castadiva*.

Since *Castadiva* is completely compatible with the ns-2 file format, we were able to compare both in a simple and straightforward manner.

We set the *Castadiva*'s wireless nodes' range to 250 meters. In terms of traffic, we define both UDP and TCP connections between each participating node and node 6. For TCP connections, the maximum transfer size is of 100 MB. Each UDP flow generates 55 packets per second, and packet size is fixed at 512 bytes.

In our first evaluation no routing protocol is being used, so that we can check the functionality of our tool. Figure 3.18 shows the obtained results in these tests. We can see that, as expected, wireless nodes that are out-of-range from node 6 are not able to communicate with it. This shows that both network topology and traffic definitions of *Castadiva* are being enforced correctly.

In both tests we observe that *Castadiva* has a lower throughput than ns-2.

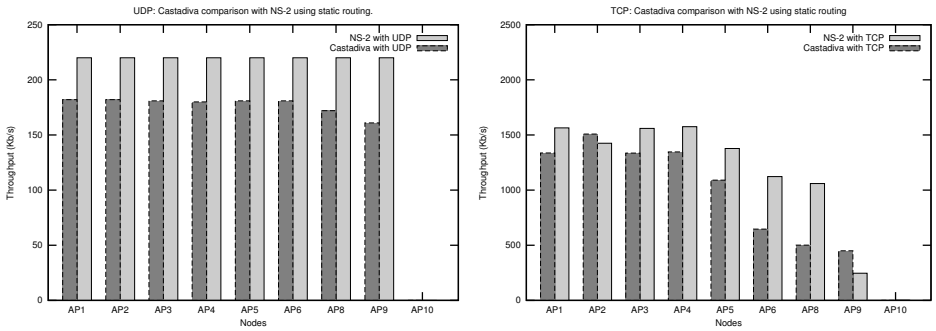

Figure 3.19: Performance comparison between *Castadiva* with ns-2 in a static scenario. Using CBR/UDP traffic (left) and FTP/TCP traffic (right). Routing enabled.

We should consider that the shared wireless media is prone to both transmission errors and contention among stations, which is due to the wireless devices being placed physically close to each other. In the case of ns-2, only contention effects are simulated, which explains the observed discrepancy.

We now repeat the previous experiment with routing and forwarding enabled. We pick the Optimum routing option, so that *Castadiva* is responsible for calculating the best route to reach a destination node and modifying the routing tables of nodes to enforce the chosen topology. Figure 3.19 shows the results for this test.

UDP tests show that traffic from nodes AP7, AP8, and AP9 is now able to reach the destination, while node AP10 remains isolated as intended. Notice that, for the former nodes, the packet loss ratio increases slightly with the number of hops to destination. With ns-2 we don't observe this behaviour for the reasons referred above.

Results with TCP traffic show that the throughput for nodes 1 to 5 is reduced compared to the previous experiment (Figure 3.18), which is due to competing traffic from AP7, AP8 and AP9. For these nodes, throughput decreases as the number of hops increases, as expected. With ns-2 we observe that both AP8 and AP9 suffer from starvation; one of the reasons for this behaviour is that, with *Castadiva*, all nodes share a same medium and so packet collisions between out-of-range nodes do not occur.

Figure 3.20 (left) shows the effect of collisions as the number of hops increases in a 802.11g wireless network. We make different tests, varying the load from 256kb/s to 3Mb/s. As can be seen, the number of hops causes the packet loss to increase. This behaviour is normal in a wireless network due to interferences among devices but, in *Castadiva,* it can be aggravated by the fact that all devices are close to one another. To better study how the proximity of the devices affects performance, Figure 3.20 (right) compares the capacity offered by *Castadiva* with respect to other capacity models proposed in the literature for different scenarios. We select the following scenarios: (a) a grid topology [GK00], (b) a random topology [GK00] and (c) a chain topology [LBDC$^+$01]. We observe that, as expected, the capacity of the network obtained in *Castadiva* is lower than the other ones since all devices

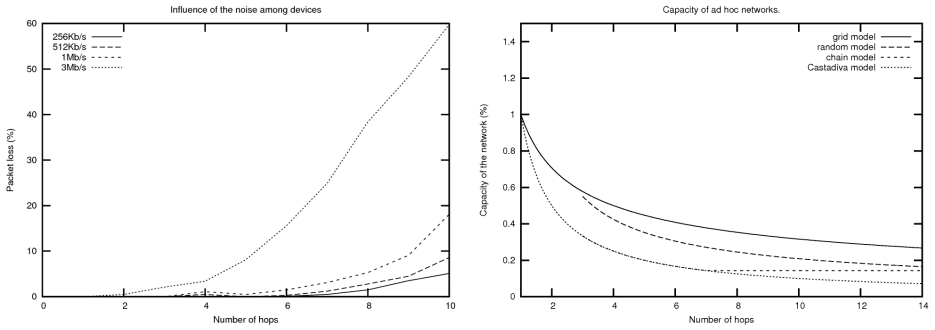

Figure 3.20: Packet loss due to the proximity of the devices in an emulation (left) and capacity of an ad hoc network compared with *Castadiva* (right).

of *Castadiva* interfere with each other. Solutions such as ORBIT try to cope with this problem by mitigating the effect of the noise among nodes; however, it introduces new problems into the emulation as limiting the network topology and limiting the transmission range of all nodes, which is also far from the real behaviour in a MANET. *Castadiva* makes a trade-off between accuracy and price, offering a cheap and portable platform that is good enough for almost all tests.

In terms of the control network (wired), we observe that the bandwidth consumption of the SSH protocol over Ethernet is far from approaching saturation, and that latency is low enough to allow adequate coordination of all nodes.

3.5.2 Evaluation of *Castadiva* with a Mobile Scenario

We now define a mobile scenario with an area of 1000m x 700m. Simulation time is of 500 seconds, and we set the wireless nodes' range to 250 meters. In terms of traffic, we define both UDP and TCP connections between each participating node and node AP7. For TCP connections, the maximum transfer size is of 1000 MB. UDP flows generate 4 packets per second, and packet size is fixed at 512 bytes. In the first test, each node has a maximum speed of 5 m/s and no routing protocol is used.

Figure 3.21 shows a node-by-node comparison between *Castadiva* and ns-2 for both UDP and TCP traffic. The selected scenario was generated by ns-2 and imported to *Castadiva* to have the same mobility pattern.

This figure shows that the obtained results are quite similar, which validates *Castadiva*'s implementation. Since we have not selected any routing protocol, transmissions are successful only when the destination is a 1-hop neighbour. In the case of TCP traffic, results are more heterogeneous since it is a stateful, bandwidth-greedy protocol prone to present non-linearities, specially in mobile ad hoc network environments. We also observe that *Castadiva* has, in general, a lower throughput/delivery rate than ns-2. When *Castadiva* is used, the shared wireless media is prone to both transmission errors an contentions among stations. In the case of ns-2, only contention effects are simulated, which explains the observed discrepancy

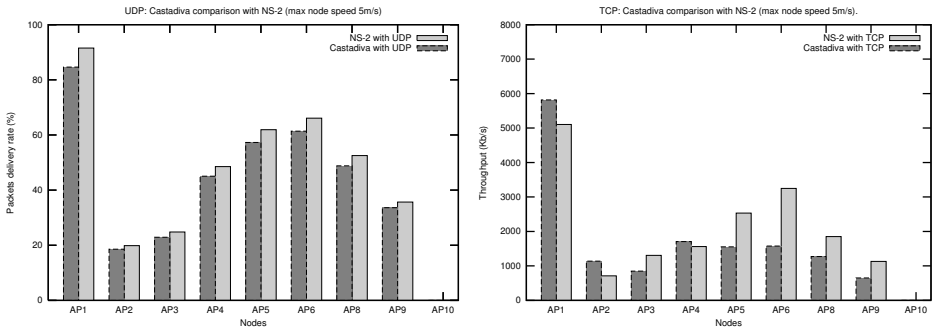

Figure 3.21: Result comparison of *Castadiva* with ns-2 without routing.

Parameter	Value used
HELLO_INTERVAL	2 s
REFRESH_INTERVAL	2 s
TC_INTERVAL	5 s
MID_INTERVAL	TC_INTERVAL
HNA_INTERVAL	TC_INTERVAL

Table 3.2: Default OLSR parameter values.

for UDP traffic.

To evaluate how the routing protocol behaves in *Castadiva*, we used the OLSR protocol. We choose this routing protocol since its implementation is available for both testing environments (simulated and real), being quite similar. Concerning OLSR-related parameters choices, we use the values proposed in the RFC [TP03], shown in Table 3.2 for the reader's convenience.

Figure 3.22 shows the similitude between *Castadiva* and ns-2 for both UDP and TCP traffic. In these tests, those nodes virtually located more than 1-hop away are also able to send traffic to the destination node thanks to the routing protocol. Similarly to the test without a routing protocol, *Castadiva* has a lower throughput/delivering rate than ns-2, for the same reasons explained before. In the case of TCP traffic, results are more heterogeneous as in the test without routing.

We now evaluate the impact of node speed. We vary the degree of mobility in different scenarios, testing with maximum node speeds of 0, 5, 10, 15 and 20 m/s. As for the previous test, each scenario was generated by ns-2 and imported to *Castadiva* to have exactly the same mobility patterns. Figure 3.23 shows the results obtained in this test.

In both tests we observe that *Castadiva* has a lower throughput than ns-2. For TCP we observe that the difference between *Castadiva* and ns-2 is more significant that for UDP. To discover the reason of this effect, we study a controlled scenario with only two nodes, and measure the arrival time of each packet. With this experiment we obtain the mean delay to do a rerouting when OLSR performs a

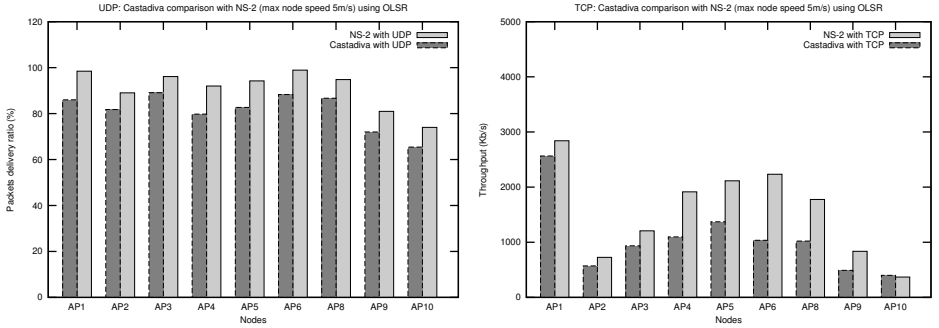

Figure 3.22: Result comparison of *Castadiva* with ns-2 for UDP (left) and TCP (right) traffic with routing.

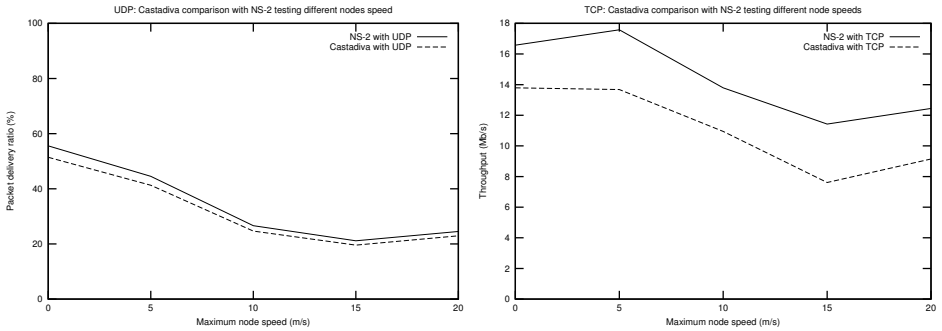

Figure 3.23: Comparison of *Castadiva* and ns-2 at different node speeds with both UDP (left) and TCP (right) traffic. Routing disabled.

topology update. In ns-2 it is of five seconds; however, *Castadiva*'s nodes have an average delay of almost eight seconds. This three-second difference causes TCP agents in ns-2 to achieve a higher throughput.

We now repeat the previous experiment with routing and forwarding enabled. Figure 3.24 shows the average percentage of packets received for maximum node speeds of 0, 5, 10, 15 and 20 m/s using the OLSR protocol.

Again we observe important similarities between *Castadiva* and ns-2's results. For both UDP and TCP traffic the behaviour of these platforms is quite similar. Tests show that the average percentage of UDP packets received is increased when a routing protocol is used, since it allows nodes AP9 and AP10 to reach their destinations. When studying the behaviour of the network using TCP traffic instead, we observe that the average throughput is not increased because, in both simulations, the network is saturated. When looking at the overall trend, though, we find that there is a high degree of resemblance between both sets of experiments.

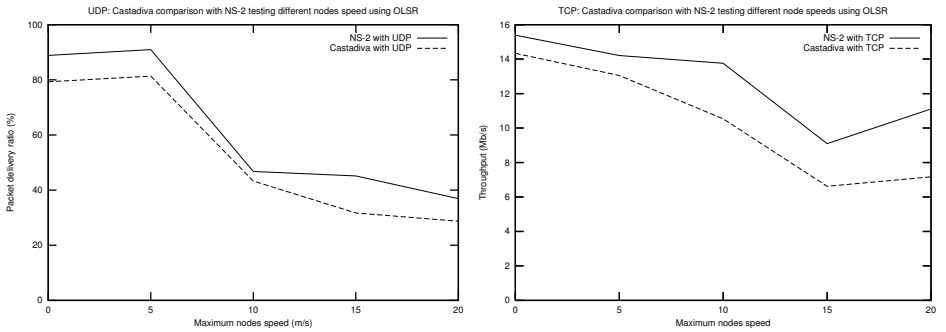

Figure 3.24: Comparison of *Castadiva* and ns-2 using OLSR at different node speeds with both UDP (left) and TCP (right) traffic.

3.6 Assessing the performance of videoconferencing in MANETS with *Castadiva*

We now describe the performance results obtained when transmitting real-time video traffic through *Castadiva*. We test with several different scenarios where the purpose is to stress the application and evaluate the impact of the number of hops in the quality of the received video stream. We are going to employ two different kinds of video streams: a standard videocall, where the video sequence is almost static, and a more dynamic videocall, where we will point the webcams towards a screen displaying a movie. Similarly to the previous section, we will perform tests in both static and mobile scenarios.

3.6.1 Static Scenario

Initially we evaluate the performance of a videocall in a static MANET for reference.

The scenario is defined in a 1500m x 1800m area, and the test time is of 5000 seconds. We set the wireless nodes' range to 250 meters. To monitor the video traffic we use the tcpdump application [SVV89].

To generate the video traffic we have two laptops with a webcam attached and running Ekiga, an open source VoIP and videocall application for Linux.

We generate an emulated MANET topology where all nodes are aligned according to a chain topology. Figure 3.25 shows the topology used when eleven nodes are deployed. In each laptop we redirect all traffic sent to the other laptop through the wireless routers located at both edges of the simulation (AP1 and AP11 in this scenario). We also configure *Castadiva* to route all video traffic through our test bed.

Each node runs the OLSR daemon to obtain consistent routing tables. We initially selected the values proposed in the RFC, but the results obtained in static environments were not good since OLSR was unable to find the routes, or the routes were lost too quickly due to network congestion. Therefore, instead of the

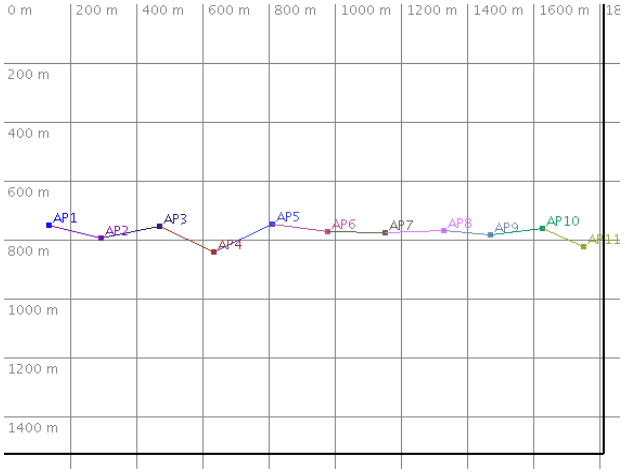

Figure 3.25: Topology for evaluating video traffic delivery.

Parameter	Used Values
HELLO_INTERVAL	5 s
HELLO_VALIDITY	15 s
TC_INTERVAL	2 s
TC_VALIDITY	15 s
MID_INTERVAL	15 s
MID_VALIDITY	300
HNA_INTERVAL	15 s
HNA_VALIDITY	300 s

Table 3.3: OpenWRT parameters values for the OLSR protocol.

default RFC parameters, we used the default configuration of OLSR implemented in the OpenWRT system, which are considered optimum for a standard mesh network. Table 3.3 shows these values.

The effects we want to study are the variations in terms of throughput, delay and jitter when varying the number of hops in the network.

3.6.1.1 Evaluating the Performance with a Standard Videocall

We now present the results obtained when using *Castadiva* to evaluate a standard videocall between two users. Based on traffic traces at source and destination, we measured the throughput and the inter-packet delay (jitter).

Figure 3.26 (left) shows the mean throughput obtained for the generated data rate at different number of hops. As we increase the number of hops, we appreciate that the average throughput is decreased to less than a twentieth compared to the one hop scenario (from around of 700 Kb/s to close to 30 Kb/s). Figure 3.26 (right) shows how the packet loss increases when we vary the number of intermediate hops

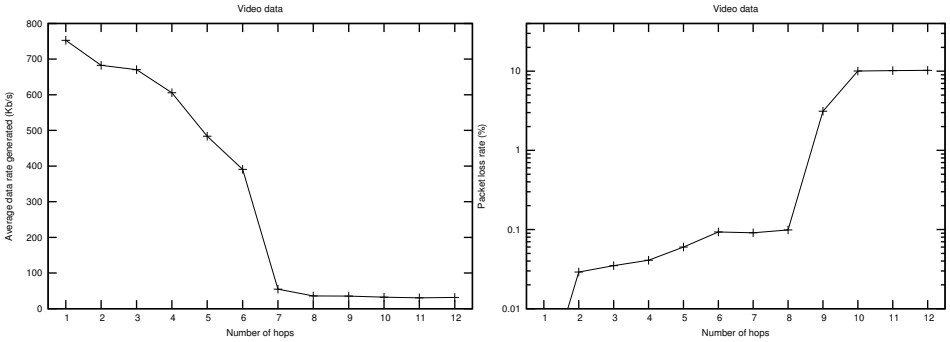

Figure 3.26: Average data rate generated (left) and packet loss ratio (right) for different numbers of hops.

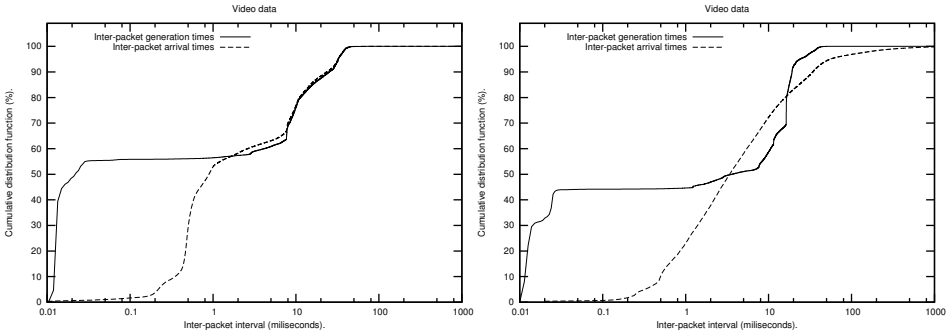

Figure 3.27: Cumulative distribution function for the inter-packet generation interval and inter-packet arrival interval in a scenario with one hop (left) and ten hops (right).

between sender and receiver. In a scenario with more than 9 hops the packet loss rate is significant for a videocall. In particular, for a scenario of 10 hops, the videocall experiences a 10% of packet loss.

Figure 3.27 shows the cumulative distribution function for the inter-packet generation interval and inter-packet arrival interval in two scenarios where source and destination are one and twelve hops away, respectively. In the scenario with one hop, the smallest network possible, we can appreciate the differences between the minimum inter-packet generation time at the source (about 0.01 milliseconds) and the minimum inter-packet arrival time at the receiver (about 0.1 milliseconds). Also, 50% of the packets have an inter-packet generation time lower than 0.03 milliseconds, but the inter-packet arrival time is typically higher than 1 millisecond.

In the scenario with ten hops, if we compare the inter-packet arrival time with the one obtained in the one hop scenario, we can appreciate an increase of more than 2 milliseconds. This is caused by the forwarding time of the additional nodes on the path. This is expected since these packets, usually generated back-to-back by the videoconferencing application, experience significant jitter when traversing

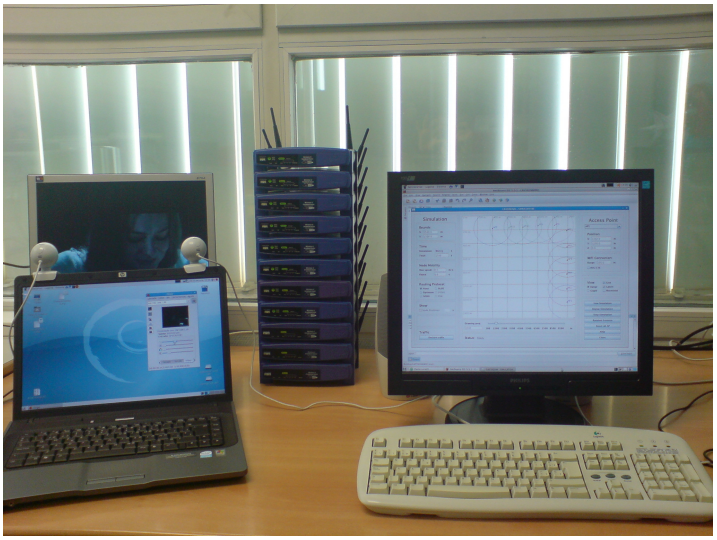

Figure 3.28: Testing a videocall when both webcams point to screen with a movie.

an ad hoc network with a high number of hops. But, in a real scenario, the average number of hops is usually less than five hops.

3.6.1.2 Evaluating the Performance of a Movie Transmission

In the previous section our experiments relied on a standard videocall. The characteristics of such video-conference - low degrees of video motion - do not impose significant demands in terms of network bandwidth. Therefore, we repeated our experiments with a higher motion video. With that purpose we pointed both webcams (each linked to a communication endpoint) to a screen showing a movie. Such strategy also allows running long experiments without the intervention of users. By repeating our experiments and taking long sampling periods we were also able to obtain meaningful values for the end-to-end delay, and so we include them in this section, as well as the measured throughput and jitter for comparison against the previous test. Figure 3.28 shows the working area used in these tests where we can observe both webcams pointing to a screen showing a film.

Figure 3.29 shows two frames obtained in the test for different hops, where we can see the quality of the video received in both scenarios. In the scenario with one hop, there is no significant delay between the real video sequence and the decoded one because the path is too short. In a scenario with ten hops, though, we can verify the poor performance obtained by comparing the received and transmitted sequences. We also appreciate a delay of the received image compared to the real image (notice that both webcams aim at a same target for comparison purposes).

Figure 3.30 (right) shows the mean throughput obtained for the generated data rate obtained for a different number of hops. In a scenario with one hop, we have a mean data rate above 500 Kb/s. In a scenario with 13 hops, the average value of

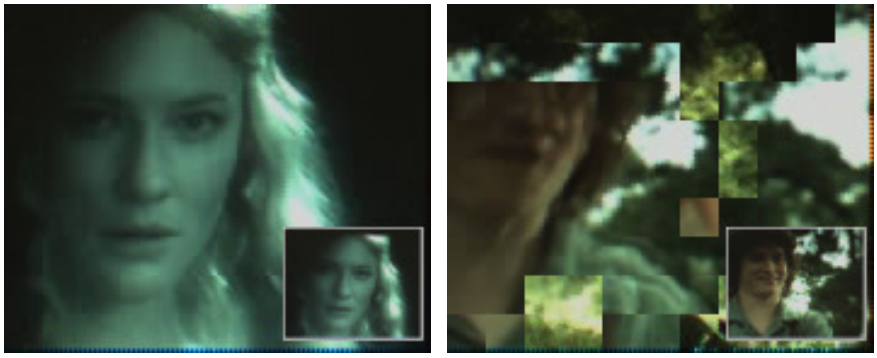

Figure 3.29: Screenshot of the videocall with a scenario of one hop (left) and ten hops (right).

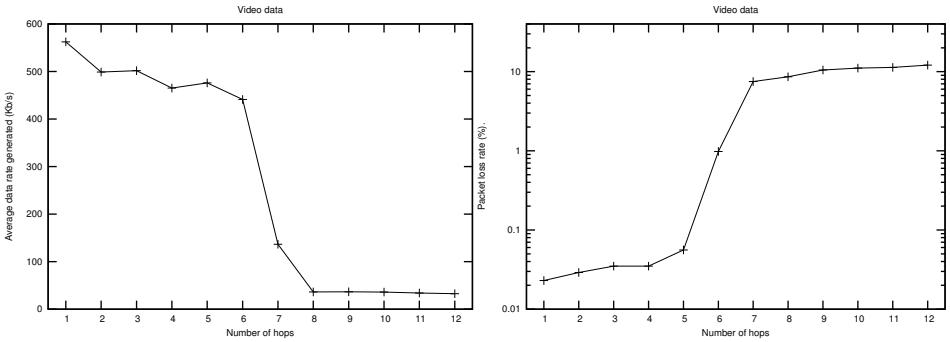

Figure 3.30: Cumulative distribution function for the throughput in a scenario with different hops (left) and packet loss rate in different scenarios (right).

the generated data rate decreases to less than 100 Kb/s due to Ekiga's bandwidth throttling mechanism. We also evaluate the packet loss in each scenario. Figure 3.30 (right) shows how the packet loss increases when we vary the intermediate number of hops between sender and receiver. If we compare this graphic with the one obtained in the evaluation of a real video-conference, we can observe that now the packet loss rate increases more significantly. In fact, for more than 6 hops, the packet loss rate surpasses 10%.

Figure 3.31 shows the cumulative distribution function for the inter-packet generation and arrival times in a scenario with one hop and in a scenario with ten hops. Notice that high delay values are quite prone to occur in this case, often introducing significant jitter (above 100 ms).

3.6.1.3 Evaluating the Round-trip Time in Different Scenarios

We also evaluated the impact of increasing the number of hops on delay by using ping sessions and measuring the impact on round-trip time while maintaining video-conference sessions active. We generate a ping session between the two

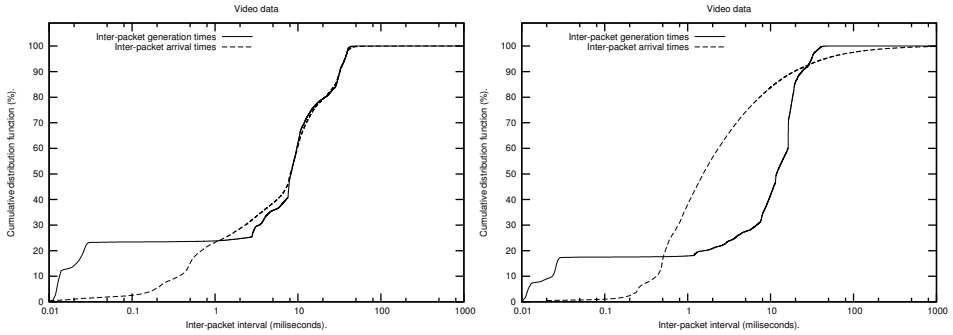

Figure 3.31: Cumulative distribution function for the inter-packet generation interval and inter-packet arrival interval in a scenario with one hop (left) and ten hops (right).

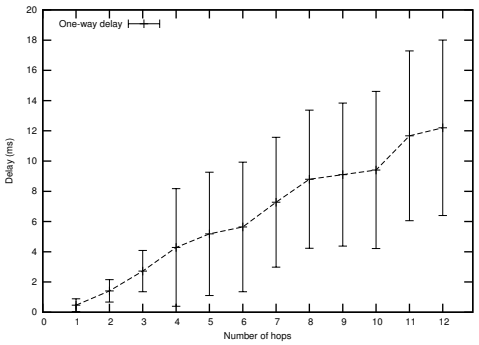

Figure 3.32: Evaluation of the ping sessions in different scenarios.

laptops in the chain scenario described earlier, varying the number of hops from one to ten. We obtain the average time of the ping session used to send a packet and receive the answer. To obtain the one-way delay, we proceed by dividing the results obtained by two.

Figure 3.32 shows the average delay and the standard deviation for each scenario. As expected, delay increases almost linearly with the number of hops between sender and receiver. Of special interest is the increase in terms of standard deviation, which can be quite problematic for real-time video transmission.

3.6.2 Dynamic Scenario

In this section we evaluate different scenarios with a dynamic topology. We change the nodes' mobility by testing with the following speed values: 3, 6, 9, 12, 15 meters per second. The scenario is defined in a 1500m x 900m area, and the test time is of 5000 seconds. We set the wireless nodes' range to 250 meters. In this test, as for the static scenario, we also differentiate between two types of scenarios: a standard videocall and a videocall where the webcams point to a movie being

Parameter	Used Values
HELLO_INTERVAL	2 s
HELLO_VALIDITY	6 s
TC_INTERVAL	5 s
TC_VALIDITY	15 s
MID_INTERVAL	5 s
MID_VALIDITY	15 s
HNA_INTERVAL	5 s
HNA_VALIDITY	30 s

Table 3.4: OLSR values used for the mobility scenarios.

Mobility	% time with route
3 m/s	64.72
6 m/s	65.32
9 m/s	85.70
12 m/s	91.77
15 m/s	93.74

Table 3.5: Percentage of the simulation time when a route between both laptops exists.

displayed. Regarding to OLSR, and according to [CDJ05, YSD06], we need to tune up the protocol when it is used in mobile scenarios. The new configuration of the protocol is shown in Table 3.4.

Since, for these tests, we picked a particular OLSR configuration, the first step is to measure the total amount of time that the route between both laptops is established. For this test we emulate 20 random scenarios for each speed, and we use a ping to determine whether the route is established or not.

Table 3.5 shows the total percentage of time when the route is established.

The percentage of time without route is directly related to the amount of time that OLSR needs to obtain a route when the topology is changing plus the time when the network is split, and there is no possible route between the two laptops.

3.6.2.1 Evaluating the Performance with a Standard Videocall

We first evaluate the behaviour of a standard videocall in a MANET. Since the number of hops is variable in each simulation due to mobility, in this case we study the variation of the throughput with different node speeds, as well as the packet losses.

Figure 3.33 shows the results obtained in this set of tests. As we can observe, higher mobility translates into better performance of the network and less percentage of packets lost. This is due to a the higher route availability, as shown in Table 3.5. Higher speeds avoid lengthy network partitioning effects, which are translated to a better service with a higher average throughput.

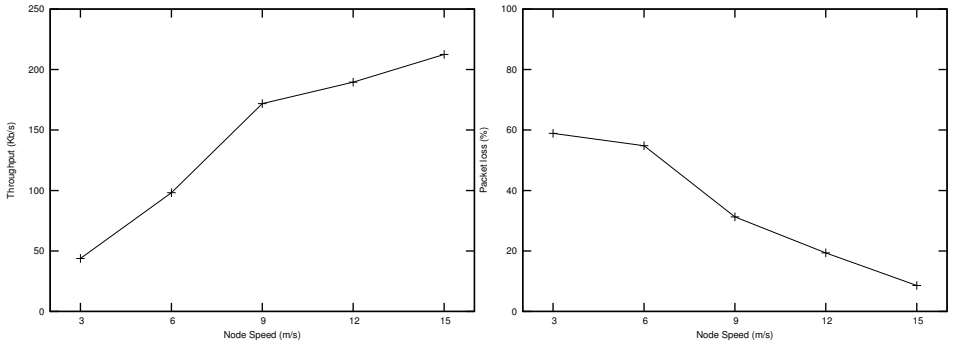

Figure 3.33: Throughput and packet losses with a standard videocall in a scenario with mobility.

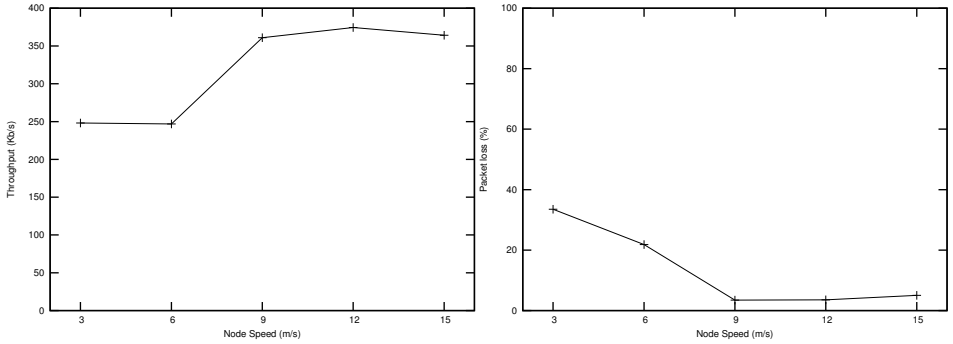

Figure 3.34: Throughput and packet losses with a movie in a scenario with mobility.

3.6.2.2 Evaluating the Performance of a Movie Transmission

In this section we also measure the throughput obtained by the videoconferencing participants when streaming a movie.

Figure 3.34 shows the throughput obtained and the packet losses in a scenario where the webcams point to a screen displaying a movie. As we can see, the throughput is higher than in the previous test: if the webcams capture a movie, the video is more complex than the nearly static image of a standard videocall.

If we compare this figure with the one shown when evaluating a standard videocall, we can see a similar trend. The last test shows a higher throughput, as the transmitted video generates a higher data rate. Packet losses decrease as in the standard videocall because: (1) we have a higher throughput when a route is established due the characteristics of the video, (2) Ekiga's bandwidth throttling mechanism decreases the packet injected when a route is lost, and (3) on average, we have the same time periods without routes between the two laptops as in the previous tests. Jointly, these factors explain why the total percentage of packets lost is reduced at higher node speeds.

3.7 Summary

In this chapter we presented *Castadiva*, a novel architecture to improve research in the MANETs field by allowing to make real test bed experiments. *Castadiva* combines the convenience and productivity of Java with the power of the Linux kernel and accompanying tools to emulate an ad hoc network environment. The system was designed to simplify the tasks of scenario implementation and traffic generation among independent, IEEE 802.11-based, wireless nodes. One of its key advantages is that it is fully compatible with the ns-2 simulator.

The architectural design of *Castadiva* differentiates wireless nodes, used for the actual experiments, from the core application, which has management and control purposes. This core application provides an easy interface to define network topologies and traffic flows between nodes. Those definitions are then translated into run-time instructions sent to test bed nodes while experiments are on-going. We observed that the use of SSH protocols and Fast Ethernet connectivity allows nodes to synchronise the start of an experiment with high accuracy, being all instructions read at once; afterwards, the test bed relies on individual clocks to synchronise instructions throughout the remaining time of an experiment.

By using both TCP and UDP data traffic, and under a variety of static and dynamic MANETs scenarios, we show that *Castadiva* is able to offer confident results while using cheap wireless off-the-shelf devices.

To summarise, this chapter shows that the advantages of using *Castadiva* with respect to other MANET test beds are: (i) it is a very low-cost test bed since each node costs about 50$, (ii) it is fully compatible with the ns-2 simulator, allowing to compare results between both in a straightforward manner and, (iii) does not occupy a lot of physical space.

Now we have a complete test bed where we can test our proposal of a community network. We use *Castadiva* to evaluate our proposal before deploying in a real scenario. In the next chapter we describe *RuralNet*, our architecture proposed for deploying a community network in developing countries.

Chapter 4

An Architecture supporting Web-based Services and Authentication

In this chapter we present *RuralNet*, our captive portal based system providing Internet connectivity to distant areas where deploying a wired-based infrastructure is too expensive. Such an infrastructure can provide the TCP/IP based services required, besides avoiding the problems referred in Chapter 2. *RuralNet* is an architecture to strengthen Internet support in rural environments that allows subscribers to access the Internet and provides a group of free services to all the people within a certain area. *RuralNet* also includes *Maya*, an extension developed for managing the access points of the mesh network which compose the *RuralNet* infrastructure, simplifying the administration of the entire network.

RuralNet began in 2005 as a research project at the *Universidad Politécnica de Valencia*, in Spain. The project intended to develop new information and communication technologies to offer low-bandwidth Internet access to isolated rural areas. With this purpose we developed *RuralNet*, an experimental wireless platform which combines the paradigm of wireless mesh networks with cheap off-the-shelf wireless devices to offer a wide range of Internet-based communication services and applications. *RuralNet* has targeted rural areas of the Comunidad Valenciana, in Spain, with increasing demand for Internet connectivity to support the emerging industrial activity and population demands. In this chapter, we present the system architecture of *RuralNet* and provide details of its implementation and deployment.

The rest of this chapter is structured as follows. Section 4.1 shows the problems detected in rural areas. Section 4.2 explains the objectives of *RuralNet*. Section 4.3 describes the proposed architecture and Section 4.4 describes its implementation. Section 4.5 describes some features designed to improve the *RuralNet* performance in a developing country. Section 4.6 describes the evaluation made to validate our proposal of community network. Section 4.7 shows an extension of *RuralNet* for administrating the entire mesh network. Section 4.8 shows an example of a

real deployment of our prototype in Mozambique. Finally, Section 4.9 draws the conclusions of this chapter.

4.1 Introduction

Besides universal connectivity, the Internet offers a global platform for accessing a wide range of telecommunication services such as e-mail, e-commerce, tele-education, tele-health, and tele-medicine at a low cost. However, outside the main urban areas, there are still important handicaps that make Internet connectivity a complex and costly task. Over 40% of the world's population lives in rural and remote areas of developing countries and have poor or no access to basic telecommunications services [ITU10]. In rural areas and small towns the Internet Service Providers (ISPs) do not assume the high-cost of technologies designed for the urban market. Moreover, low population density and high deployment costs discourage ISP investments since the estimated return on investment (ROI) is unattractive.

This problem is emphasised by the study of Bright [Bri01], who shows the Digital Divide between urban areas and rural areas. Others works [Hud99, Li04, MZ04] also focus on this problem, emphasising on how serious and difficult to handle it is.

The solution to provide Internet connectivity in rural areas have the following characteristics: (a) implementation should be possible at a low cost in areas where population density is low, (b) the system should be easily installed, even in remote and inaccessible locations, (c) system operation and maintenance should be carried out even when qualified technical personnel is scarce, (d) implementation should be possible even when basic infrastructure, such as electricity, running water, paved road networks, etc., is absent; and (e) the infrastructure must be characterised by long life cycles.

The promises of wireless Internet technologies have generated a lot of interest from the international-development community. While in developed nations these technologies have primarily been associated with mobility applications and local area networking in homes and offices, their most intriguing application in developing nations is the deployment of low-cost broadband Internet infrastructure and last-mile distribution. These technologies seem to cope well with the requirements of the solution described above.

As said, the newly available technologies are much cheaper to use, making the infrastructure required to connect a village to a big city affordable at a low-cost [LYHO04, Li04, Ken02]. This new technology allows users to automatically relay radio signals, thus creating a mesh network of wireless connections that could develop a life of its own, reducing the number of required base stations. It lowers the cost of infrastructure while increasing the cost to users only marginally, and provides connectivity redundancy in dense areas.

But not only the wireless technology allows a low-cost Internet connectivity. Nowadays, thanks to this new way of accessing Internet, each user can exchange data when going to work using a public transport, in their home or at different public places. Therefore, access control is now a priority to restrict the use of networks, controlling what users can and cannot access in terms of services offered.

Access control is a main priority in non encrypted wireless networks like MANETs and mesh networks. However, not only computers and laptops can connect to the Internet. Each year new devices appear that incorporate the technology to connect to a wireless network, likes PDAs or mobile phones. Therefore, our proposed solution must be compatible with all the devices on the market.

RuralNet is designed to provide Internet Access to a high number of users taking profit of the wireless technologies as described before, providing access control to the users and extending some services in the covered area. *Maya*, an extension of *RuralNet*, is a management tool for wireless mesh networks. It allows performing critical configuration tasks, such as global ESSID and channel changes, in a simple and efficient manner.

4.2 Objectives of *RuralNet*

The general purpose of *RuralNet* is to strengthen Internet support in rural environments through wireless technologies in different scenarios such as the rural areas of developed countries, or in any scenario of the developing countries. The system should empower mobile users with the ability to access Internet applications on the move.

Rather than creating a new wireless technology, the major challenge has been to demonstrate the practicality of designing and building a system that, by combing existing wireless network paradigms, is able to reach distant areas at a low cost, while offering a wide range of telecommunication services and applications.

4.3 The *RuralNet* System Architecture

The system is designed to cover a wide area, connecting all the clients with a main server that has full control of the system. The overall system architecture for our *RuralNet* telecommunications network is shown in Figure 4.1. The system is composed by different nodes connected to each other, forming a mesh network. This approach allows creating a scalable network which is able to cover a vast area, connecting the main server with all the clients within range of any of the nodes deployed. All the software developed as part of the *RuralNet* project is free software and it can be downloaded at `http://ruralnet.sourceforge.net/`.

We have built a prototype composed by multiple Wi-Fi access points connected to a main server. Our architecture is conceptually organised into three different levels, which are: the management level (main server), the network connection level (called Backbone Network), and the user level.

- **Management level.** The top level of the system is composed by a server that controls user authentication. This level is based on a web server to interact with the subscribers, a database used to store system information, and a control unit that converts management decisions into traffic rules. Besides, the server has also a high-speed connection to the Internet, along with a wireless or an Ethernet connection to link it with the Backbone Network level. It is at this level where the user access control functionality resides.

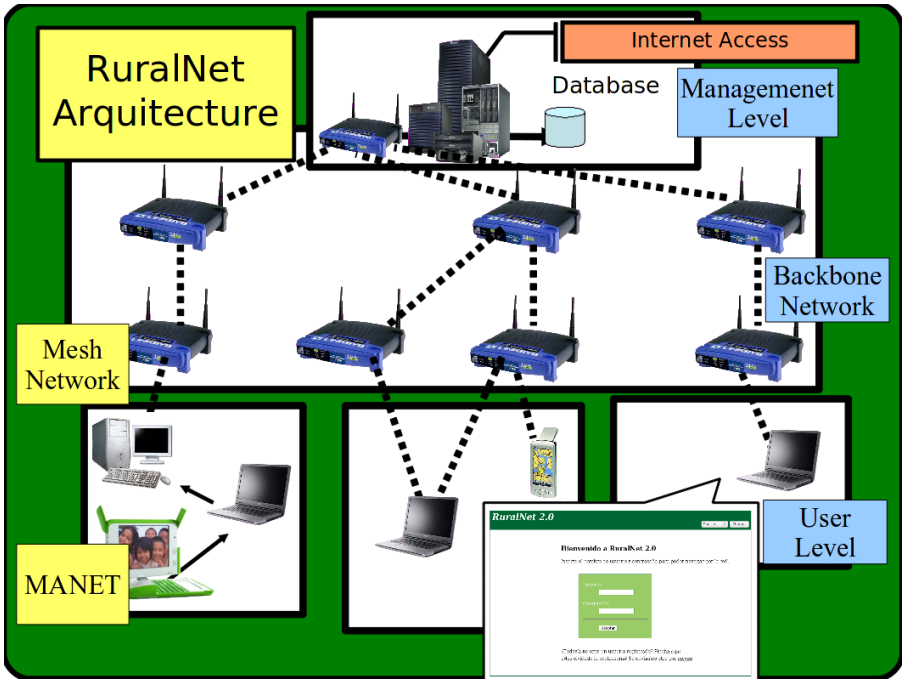

Figure 4.1: The *RuralNet* system architecture.

- **Backbone Network level.** This second level is composed by a group of nodes distributed in a wide area, composing a mesh network. These nodes are connected to the main server and to other nodes through either Ethernet or IEEE 802.11 technology. Wireless connections are preferred since they can benefit from antennas to achieve increased range at little cost. The main purpose of this level is to work as a bridge, connecting subscribers to the main server. Each node runs upon a modified version of the OpenWRT [Ope10] firmware, which allow us to implement access control into the node while increasing the hotspot coverage by implementing a multihop wireless mesh network. The OpenWRT firmware offers us all the functionality of the usual GNU/Linux tools for monitoring, bandwidth shaping, firewalling, and so forth. This firmware also allows us to install the OLSR routing protocol to create the mesh network, used by the client to connect to the server. In our prototype, each node is a Linksys router.

- **User level.** At the lowest level we have the actual subscribers. These can connect directly with the wireless infrastructure using their own Wi-Fi enabled computing devices. Such devices can be quite heterogeneous e.g., cell phones, PDAs, laptops, etc. The only restrictions are that these devices must include a Wi-Fi interface, a web browser, and a OLSR routing protocol to connect to the mesh network. At this level we can deploy a MANET to extend the coverage area of *RuralNet*, using users' devices as repeaters.

Every client within the coverage area of *RuralNet* can have access to all the services offered, which does not mean free access or uncontrolled access. Our system is implemented under a captive portal solution based on the use of wireless access points to provide both an effective user authentication and physical connectivity to the backbone. Therefore, a client only needs a web browser to access the system; further knowledge about wireless networks is not required, neither it is required the use of special software.

4.3.1 Technologies Used

The *RuralNet* system was developed using several programming languages and tools. We split it into three conceptual areas: web interface, system interface and database interface.

- The implementation of the web interface makes use of different programming languages, i.e., PHP, Javascript, HTML and XML. The combination of these languages allows achieving complex solutions, and yet compose the user interface in a simple and straightforward manner.
- The system interface uses PHP to access TC [Hub03] and IPtables [Izu03], and both tools are provided by default on a GNU/Linux system. These tools offer the functionality to control the system's firewall and to regulate the bandwidth for the different user connections.
- The database interface uses PHP technology to access data stored by a MySQL database engine.

Figure 4.2 shows the relationship between the software components of *RuralNet*. We can see that the main server also offers a centralised DHCP service, making it possible for subscribers to be configured automatically.

Figure 4.2 also evidences the different support files created and the related tools that use them. A single arrow line represents a reading action and a double arrow line represents a read/write/execute action. This design provides the required flexibility to make changes or add new modules in a straightforward manner.

4.4 *RuralNet*'s Basic Functionality

When clients connect to *RuralNet* they usually do not know any connection information. Our system re-directs all the unregistered client accesses to the identification page of *RuralNet*. The most appropriate solution for this task is developing a captive portal. The enhancements required to make *RuralNet* a captive portal-enabled platform are the following.

4.4.1 Controlling the Access to *RuralNet*

A captive portal is a system used to control the access into a network using web technology. Nowadays several captive portals exists like NoCat [NoC], WifiDog

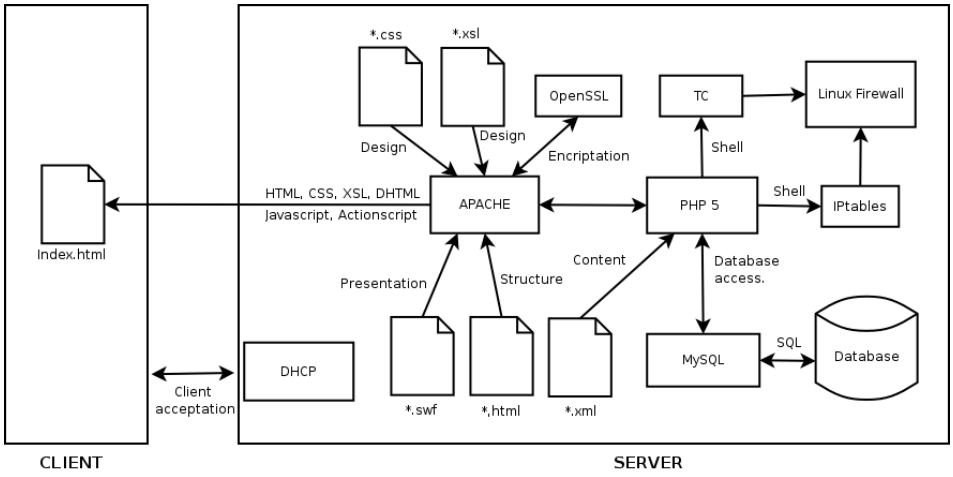

Figure 4.2: Relationship among *RuralNet*'s software components.

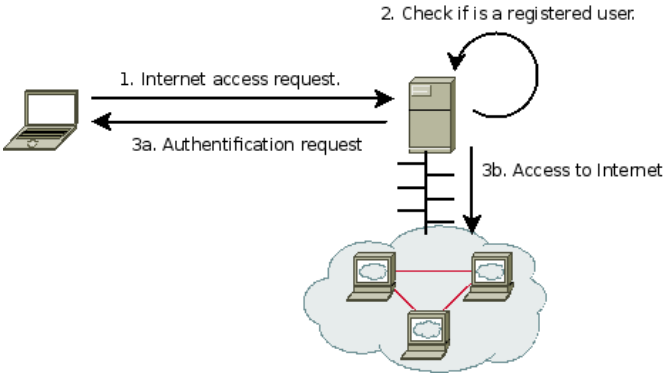

Figure 4.3: Typical captive portal connection scheme.

[Wif] and Firstpot [Pat], but we will not use them since they are either commercial software or they do not have the functionality required to satisfy our requirements.

In a captive portal tool, when a client first connects to the system and opens a web browser, he is automatically redirected to the main page of the portal; this process is completely transparent to the user (Figure 4.3). The main server controls client access depending on whether the user is registered or not. Depending on the client's access level, different services will be provided.

The first time a client accesses the system, he is asked to register with the captive portal. After a login process the user can use any of the freely available services or purchase others. Concerning the Internet access service, *RuralNet* allows each client to choose among multiple connection speeds. Just as an example, Figure 4.4 shows the welcome window for a client accessing the *RuralNet* system for the first time.

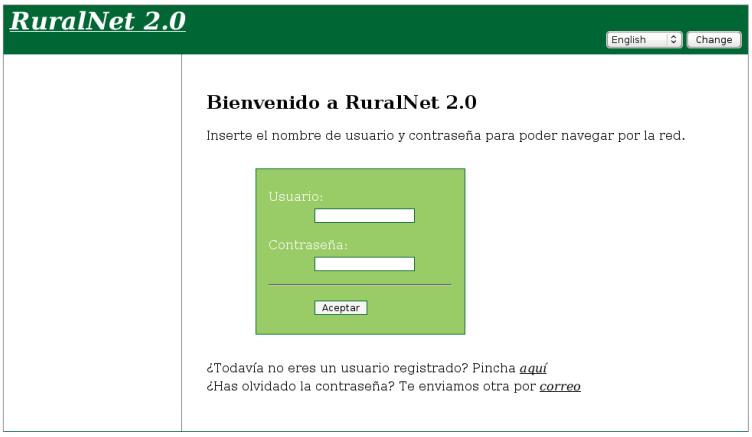

Figure 4.4: *RuralNet* presentation screen.

Thanks to the captive portal technology, users only need a web browser to access *RuralNet*. When they access the network they are automatically redirected through the captive portal system to the main web page. Further knowledge about wireless networks is neither required, nor it is necessary to use special software. Therefore, the only requirement to access the system is that clients run a web browser.

4.4.1.1 Controlling the Access to the System

When an unregistered client tries to access a web page, the system must re-direct the request to the registration page instead. Once a client has been authenticated, e.g., introducing his user name and a password, he gets access to the web page he originally requested (if he has credit to do so). In a GNU/Linux system we implemented the capture process using the IPtables tool [Izu03].

IPtables allows a computer to redirect all the traffic that arrives to a Linux system. Redirecting all the traffic coming from an unknown client to the control server allows the system to show a login page. To allow an user to have access to the Internet, the system must delete the rule that redirects a client to the control server. This option can only be visible for those clients who are allowed to access the Internet.

To know how many clients are connected at any time the system must detect the remaining clients, disconnecting and registering as off-line clients as quickly as possible. To do that, the system uses Algorithm 4.1. This algorithm classifies clients into four groups: (a) clients recently connected to the system, (b) clients connected but without any new connections in the last few seconds, (c) clients that must to be disconnected because they are off-line for a long time, (d) disconnected clients. The algorithm basically decreases periodically the group counter of each user. Users have a window in their web browser that updates their state to the maximum value. If a user does not refresh his state, the algorithm decreases it

Algorithm 4.1 User disconnection from *RuralNet*.

```
#The server runs periodically a script containing:
Do for ever:
   For every IP used by the DHCP:
      read the state of a client:
      If the state is:
         3) The client is connected, update it to level 2.
         2) No news of the client in a few seconds, another chance:
            Update the state to level 1.
         1) No news for a long time. Disconnecting the client:
            Redirect all the Internet connectivities to the server.
            Update the client state to 0. Register the disconnection.
         0) Client already disconnected.
      end if.
   end for.
   sleep X seconds.
end do.

#Each time the client access to the system:
Put the client state to 3.
```

periodically until it arrives to 1, hence disconnecting the user.

4.4.1.2 Controlling the Connection Speed for each Client.

When subscribers get access to the Internet using our system, they could, in theory, use all the available bandwidth. This is an undesirable situation. So, it would be desirable to have, for each user, a flexible control of the connection speed. In *RuralNet* this characteristic has been named City-ticket. Depending on the City-ticket each user has, they acquire the corresponding connection speed. *RuralNet* implements the City-ticket functionality by using the Traffic Control (TC) [Hub03] Linux tool.

Using TC the main server can regulate the bandwidth for each user, by controlling both download and upload packets passing through the interface that provides access to the Internet. TC generates virtual queues assigned to a network device. These queues allow the operating system to control the connection speed associated to this device.

RuralNet uses TC to create a hierarchy where every client has two queues and each of these queues controls the download or the upload connection speed. Figure 4.5 shows this hierarchy. The TC tool is controlled by *RuralNet*'s server when it starts, and generates a couple of new queues every time a client connects to the Internet. These queues are exclusive to that client, controlling the connection speed separately. Depending on the City-ticket assigned to a client, he acquires a different connection speed. Algorithm 4.2 shows the creation of the corresponding queue for each client.

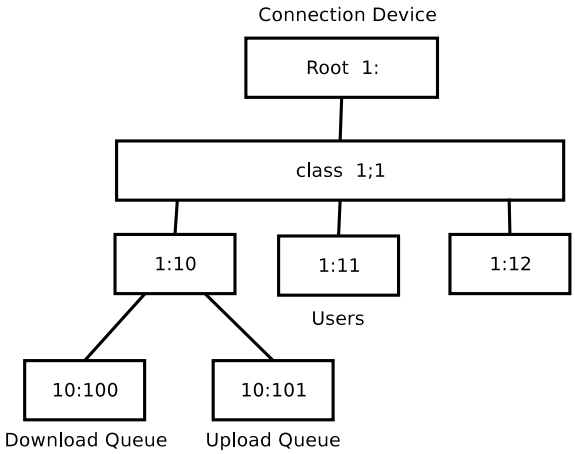

Figure 4.5: TC queue hierarchy.

4.4.1.3 Offering Free Access for Specific Web Servers.

For different reasons, it could be necessary for all clients to have access to a selected group of external web servers, including those clients that do not acquire Internet connectivity. *RuralNet* has the option for an administrator to choose a group of servers that can be accessed by everyone, regardless of the Internet privileges granted. For example, if *RuralNet* is installed in a certain town, a free web server could be the city-hall's web server, which has news addressed to every citizen. *RuralNet* has two ways to add a free web server: by IP address and by connection port.

- Free web server by IP: the system can add new rules to the firewall to allow a specific IP to be reached by all clients connected to the wireless network. If a web server is added here, every request to this server will not be redirected to the *RuralNet* captive portal.
- Free web server by connection ports: the system redirects all the free requests to a specific port to an external web server. *RuralNet* has a list of servers in the main page. If a client selects a server in this list, the system redirects the request to the appropriate port.

The choice of free server based on IP or connection port depends on the desired behaviour of *RuralNet*.

4.4.2 The *RuralNet* Interface Implementation

The *RuralNet* Interface is designed to be maintained mostly unaltered regardless of which web browser, operating system, or device is used. With this objective *RuralNet* is designed using both HTML and XML. Other specialised languages like ActionScript (Flash) and DHTML (a mix of Javascript and HTML) are able to

Algorithm 4.2 Connection speed of each user using *RuralNet*.

#Server algorithm. Its supposed to be one device to connect to Internet (client upload speed) and another to make the WI-FI net (client download speed).

For every device
  Delete all the old rules assigned to the device.
  Generate a *root* queue and assign it to the device.
  #It must not to use the 100% of the bandwidth to avoid saturation.
  Generate a main queue attached to the root queue:
    Select a type of queue (FIFO, stochastic,...).
    Assign 75% of the bandwidth.
  Generate a default queue attached to the main queue for all the traffic no regulated.
    Select a type of queue (FIFO, stochastic,...).
    Assign the desired bandwidth.
    Mark all the default packets to use this queue.
end for.

#Client algorithm. The client must create their own queues for upload and download speed.

For the download and upload connection do:
  Delete all old queues assigned to this client IP.
  Generate a client queue attached to the main queue.
    Select a type of queue (FIFO, stochastic,...).
    Assign the bandwidth of the client.
    Mark all the packets from/to the client IP to use this queue.
end.

offer a higher programming level for the design and implementation of the visual interface. Notice that all these programming languages are available for every client regardless of the system architecture used. By making use of PHP, we are able to adapt the web pages to interact with the *RuralNet* Core, hence allowing every device with web browsing capabilities to access the Internet and connect to the *RuralNet* system.

Once inside the system the interface is rather intuitive. Figure 4.6 shows the developed interface. A left-hand menu provides the user access to all the options made available to him. There are three types of users defined in the system: (a) regular users, (b) users with Internet access, and (c) system administrators. The differences among them are reflected in terms of options appearing in the menu. For a regular user only the most basic options appear, i.e., user profile, Internet access, allowed servers and help. An administrator has all the options available, including the system administration options, i.e., user management, connection properties, and City-ticket administration. By selecting a menu option the user is able to navigate through the system, thereby accessing the desired service.

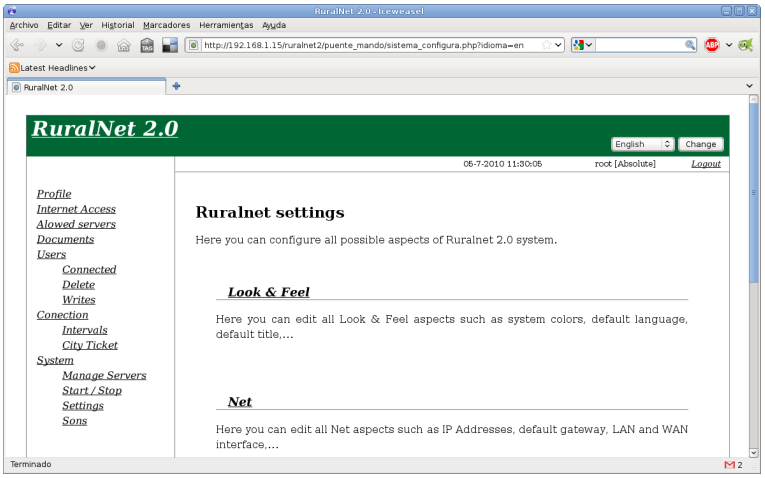

Figure 4.6: *RuralNet* interface.

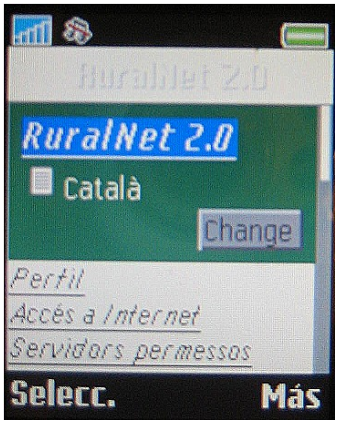

Figure 4.7: Accessing *RuralNet* with a mobile phone.

RuralNet has been implemented to support multiple languages. To implement this functionality, we separate the web structure from the text. This means that web pages have two components: a basic structure written in HTML language and text in the XML format on a separate file. This separation allows an administrator to easily update the system with a new language. It merely requires editing the appropriate XML files and adding to every paragraph the appropriate translation for the new language to be supported. The current prototype version of *RuralNet* supports English, Spanish, and Valencian.

RuralNet also allows any user to connect to a network using devices such as PDAs, mobile phones or laptops. Therefore, the interface implementation must be adapted according to the device used. Figure 4.7 shows an example of a user accessing to *RuralNet* using a mobile phone.

4.5 *RuralNet* for Developing Countries

Nowadays, in the richer countries the use of ADSL or other technologies such as optical fiber to achieve a high bandwidth per user is extended. However, old technologies such as RDSI or 56k Modems are still in use in developing countries. The technology available in the scenario is one of the main factors to think about when deploying a system such as *RuralNet*. Therefore, *RuralNet* needs to cope well with different situations: (i) low bandwidth in the Internet connectivity, (ii) scalability of the system to increase the number of users, and (iii) cope well with the frequent disconnections due to a bad service provided by the ISP.

4.5.1 Using Multiple Internet Connectivities

This feature allows *RuralNet* to connect to Internet with different physical interfaces. Thus, more bandwidth can be used by the system and, if one connection fails, *RuralNet* can still access the Internet using the other connections. For example, if we deploy *RuralNet* in a country where ADSL does not exist, we can purchase two or more Internet connections using different telephone lines and modems, and *RuralNet* will use all of them, sharing the different users among the different connections. With this option we can improve our *RuralNet* system and accept more users into the system.

RuralNet uses *iproute2 [KH10]* to configure the multiple Internet connectivities. Iproute2 is also used to obtain a load balancing among the different connections, since it shares each IP request on each connection.

4.5.2 Scalability

To improve scalability, *RuralNet* can also be a client of other different *RuralNet* system. For example, different villages can use their own *RuralNet* system but, if only one of them has Internet connectivity, each *RuralNet* can share its connection. For example, in Figure 4.8 *RuralNet* A has access to Internet, and *RuralNet* B can use this Internet connection because it is also a client of *RuralNet* A. In this Figure each *RuralNet* in the example has its own users and services. Therefore, we can extend the range of our architecture without modifying the design of the *RuralNet* main core.

4.5.3 Services without Internet Connectivity

Since *RuralNet* is designed for developing countries, in some scenarios it is possible that Internet connectivity is frequently down or without enough bandwidth to support all the users. Then, it can cause the entire infrastructure to be useless. In *RuralNet*, we offer some extra services as an intranet that can be used without Internet connectivity.

RuralNet integrates an FTP server to share documents with the entire network. This is the easiest way to exchange information, documents or other data among the nodes of the networks. Figure 4.9 shows how a user interacts with this FTP service integrated into *RuralNet*.

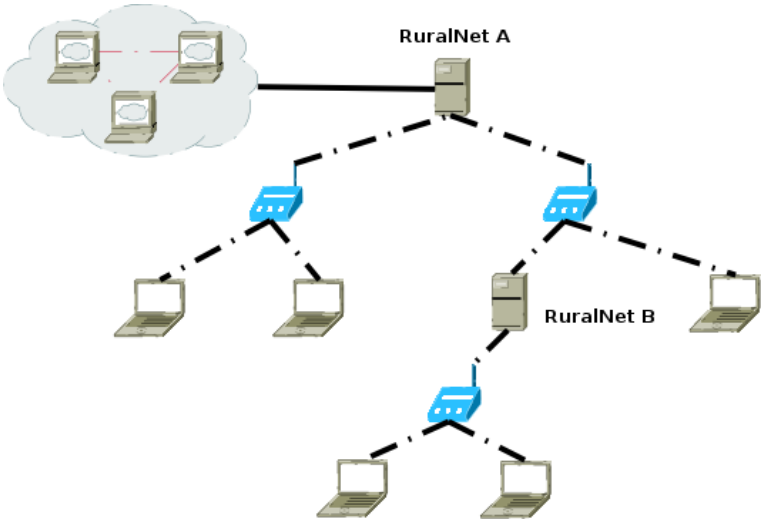

Figure 4.8: *RuralNet* connected to another *RuralNet* system.

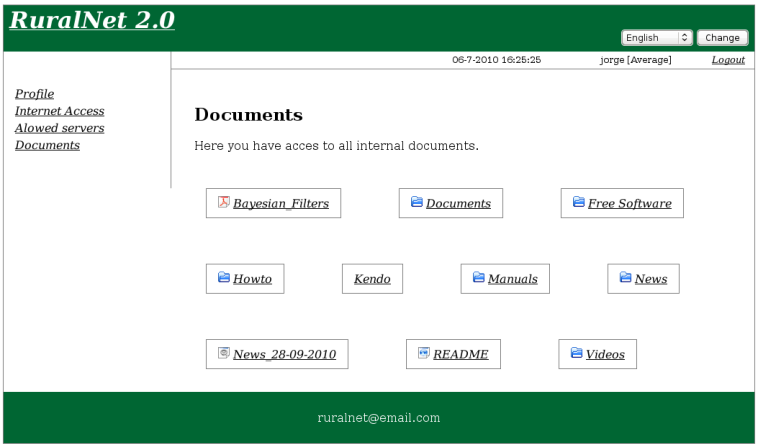

Figure 4.9: Documents in *RuralNet*.

RuralNet is composed by a mesh network and therefore, it can also provide extra applications that are useful in private networks, such as videocall or messaging services.

4.6 Evaluation

In this section we present some experimental results where the purpose is to assess the correct operation of our *RuralNet* system. We will study how bandwidth is shared among different subscribers when we have homogeneous or heterogeneous

types of Internet connectivities and subscribers. We also evaluated the behaviour of the distribution network and the impact on round-trip time in our prototype.

4.6.1 Evaluation with one client

Our first step is testing the connectivity of one client to *RuralNet*. We deploy a test bed using *Castadiva* to emulate a mesh network with the structure of our designed architecture.

The Main Server was an AMD XP 2000+ PC with 512 MBs of DDR RAM. It has a Fast-Ethernet connection to the Internet, and a similar connection to a root Access Point. The operating system used is Debian 3.1.

Concerning the clients, these are simulated using four laptops with different capabilities. The best performing one has an Intel Pentium 4 processor at 1700 MHz, and the worst one has was an Intel Celeron processor at 150 MHz. All of them are equipped with IEEE 802.11b PCMCIA Wireless Cards (maximum data rate of 11 Mb/s). We also used another machine as an FTP server for experimental purposes.

The strategy followed to make several long-run measurements consisted of developing a series of scripts that allowed automating the evaluation process. Every client has a script that starts an FTP connection to our FTP server, which is accessed only through the *RuralNet* system. We control the throughput of the FTP server so that we can model different connection speeds with *RuralNet*'s main server.

We run the tcpdump [SVV89] tool at the Main Server in order to trace all the incoming and outgoing packets. We begin our tests by using only one client, and measure the accuracy of the bandwidth management system offered by the TC tool. So, our client downloads data uninterruptedly from the FTP server, allowing us to compare the requested data rate with the actual throughput received. Figure 4.10 shows the obtained results.

It includes different intervals, where each interval represents the actual range of throughput values experienced by the client; the mean value is also represented. By observing these results we confirm that the requested bandwidth value is close to the throughput provided by the system, except for the last value where 8 Mbit/s are requested and only about 6Mbit/s are finally delivered. When the requested bandwidth surpasses the maximum value achievable with IEEE 802.11b technology, then the Wi-Fi link becomes the bottleneck.

4.6.2 Interactions among different clients

We now proceed with a set of tests where we have all four clients active. We will study the interactions among them when limiting both their bandwidth and the bandwidth towards the FTP server.

We begin by experimenting with different bandwidth values for each client. We set the bandwidth of the four users to 128, 256, 512 and 1024 Kb/s, respectively. Our purpose is to check if the system can indeed make effective the different user privileges. Figure 4.11 shows the obtained results. As can be observed, the *RuralNet* system can offer the requested bandwidth according to the user

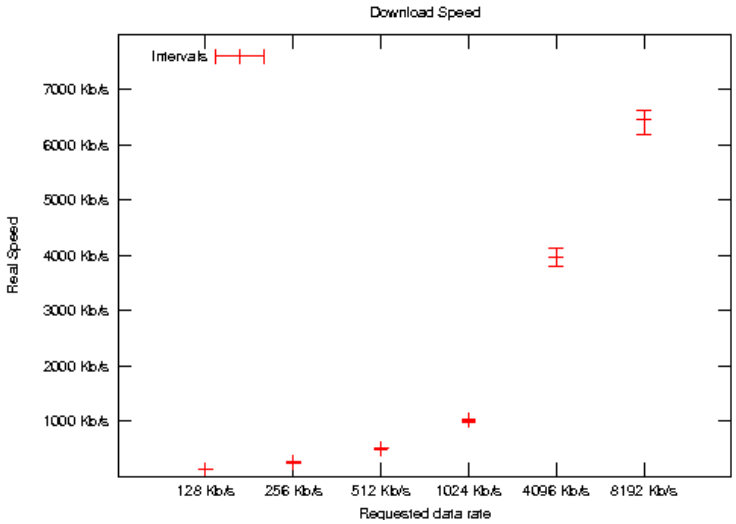

Figure 4.10: Connection speed for one client.

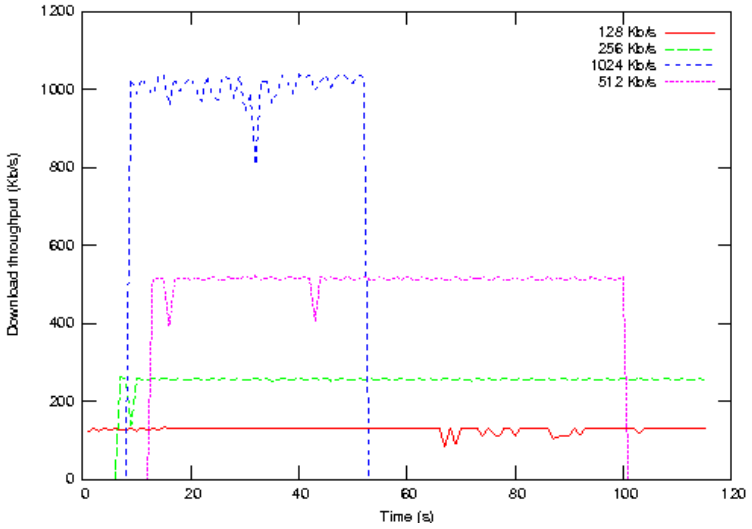

Figure 4.11: Download speed for the 4 clients under analysis.

configuration. We also observed that the throughput values sometimes suffer from variations that deviate them from the requested ones. We consider that these small discrepancies are due to wireless media noise and not to the bandwidth control algorithms used.

We now extend the previous example by acting upon the link speed between *RuralNet*'s server and the FTP server so as to assess the impact on clients' performance. We experiment with two different speeds for this link: 1024 Kb/s and

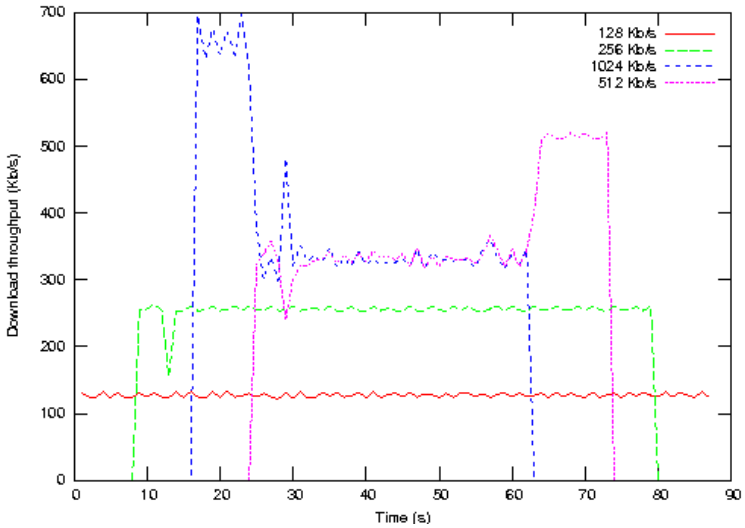

Figure 4.12: Download speed when the bandwidth towards the FTP server is limited to 1024 Kb/s.

256 Kb/s. In neither case there is enough bandwidth to serve the four clients at their requested data rate.

Figure 4.12 shows that, when the maximum aggregated data rate is limited to 1024 Kb/s, the two slowest connections use their requested bandwidth of 128 and 256 Kb/s. However the other two high speed connections tend to reach an equilibrium at a similar data rate value during the period when both are active.

When we further reduce the bandwidth at the bottleneck link to 256 Kb/s (Figure 4.13) we find that the throughput value for all the connections becomes stable at a value close to 64 Kb/s, as expected (fair resource sharing).

Though currently we apply this strategy of evenly sharing available resources when they become much lower than the aggregated bandwidth requested, we could instead apply a different policy. As an example, a solution based on a weighted fair queueing policy, would means that, when the available bandwidth becomes too low, users would receive a channel bandwidth that is proportional to the bandwidth they have paid for.

4.6.3 Round-trip time

We also evaluated the behaviour of the distribution network using ping sessions and evaluating the impact on round-trip time. Figure 4.14 shows the Round-trip time for each packet. We consider two different scenarios: for a clean environment and for a noisy environment. The first one is supposing a network without any interference. The second one is a scenario with other wireless networks causing interference. A real scenario is between both of them.

We can observe that, in a noisy environment, the round-trip time is higher than

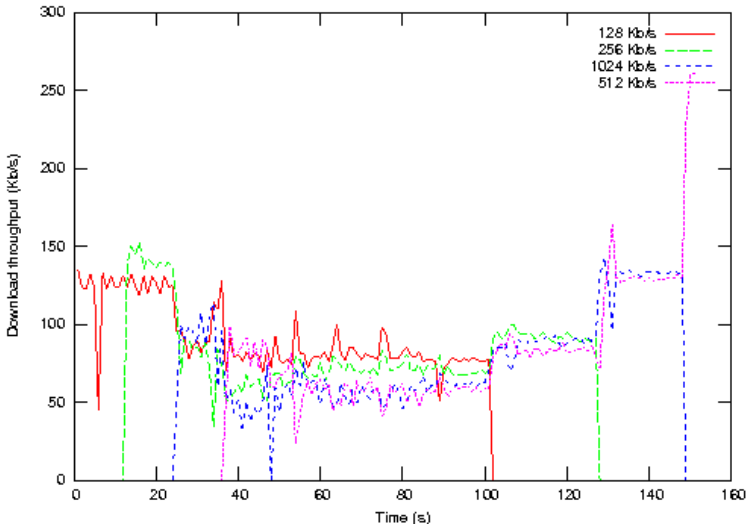

Figure 4.13: Download speed when the bandwidth towards the FTP server is limited 256 Kb/s.

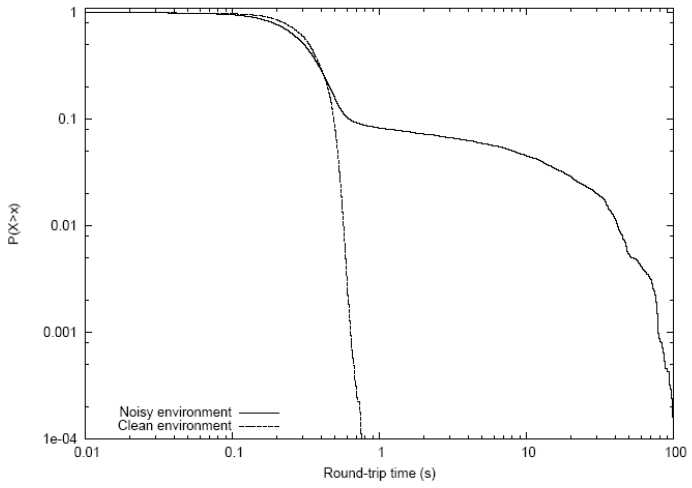

Figure 4.14: Evaluation of the distribution network.

in a clean environment, and the packet loss is also increased. We also observe that 90% of the total packet interchanges are successfully completed for a round-trip time in the interval from 0.1 to 0.8 seconds, (acceptable to a subscriber without causing annoyance) and only 10% of the traffic suffers from poor performance. More specifically, 4% of the traffic experiences too high delays, being considered

lost in practise.

Packet loses are caused by the collision between packets sent by different devices on the wireless network, and the delay time is caused by the retransmissions caused by collisions.

4.7 *Maya*: Our Mesh Networks Management Tool

Now we present *Maya*, a tool designed for network management and configuration that combines simplicity of use with a full set of features. *Maya* is used to manage the *RuralNet* network. It is also compatible with the firmware developed by the OpenWRT [Ope10] team, which offers all the functionality of a Linux distribution for embedded systems. *Maya* builds upon the OpenWRT platform by extending its functionality so as to adapt wireless routers to the needs of the *RuralNet* network.

With *Maya*, an administrator is free to configure the wireless mesh routers whenever he is within the signal range of any of the wireless routers. Obviously, *Maya* has been developed to offer a secure infrastructure avoiding that clients perform administrative tasks, changing the configuration of the devices. Since the traffic generated by *Maya* is very low, it does not affect client connections and the overall network performance.

4.7.1 Implementation and Functionality

We divide the implementation of *Maya* in two different parts: the *Maya* graphical user interface (GUI), and the configuration setup module.

Figure 4.15 shows *Maya*'s management screen, which combines the simplicity of a web-based design with the convenience of integration with the OpenWRT platform. Since *Maya* is completely compatible with the OpenWRT platform we offer new functionality for mesh networking while using the most widely adopted open source firmware for wireless routers.

The system GUI has been implemented using standard web programming languages such as PHP, Javascript, HTML, and CSS. The combination of standard Web languages allows achieving complex solutions, and yet offer a powerful user interface in a simple and straightforward manner. It also uses a MySQL database engine to store configuration parameters.

The configuration setup module is in charge of configuring the mesh network by enforcing that the new configuration is updated on all wireless mesh routers. The setup module has been implemented using C language. Shell scripting has also been used to remotely execute instructions at wireless routers.

The basic operation mode of the proposed application allows to simultaneously manage the wireless routers conforming the distribution network. New infrastructure nodes can manually be added to the mesh network by specifying their IP address, or they can be imported from a file. Once the network has been established, *Maya* offers the network manager all the functionality to enable/disable each of the wireless routers, and to change any of the network's parameters, such as the ESSID or the selected channel in some/all routers at the same time. Moreover, *Maya* includes a security module to protect the network from possible attacks by

Figure 4.15: *Maya*'s management interface.

unauthorised clients. In the following sections we provide implementation details for each of *Maya*'s main features.

4.7.2 The Wireless Router Enabling/Disabling problem

On a wireless mesh network most of the wireless mesh routers are only accessible using a wireless link. Therefore, an administrator can't actually turn them off completely since that would make it impossible to turn them back on. Therefore, alternative techniques are required to offer pseudo on-off switching. With *Maya*, enabled routers are those accepting all types of traffic. On the contrary, disabled routers are those which discard all traffic (routing traffic included), though they remain active. Such option is very useful in several scenarios, such as preventing the access to a group of clients who are connected through a specific router, disabling the router performing gateway functions to isolate the ad hoc network, or reducing the routing traffic by disabling routers that are not being used. We implement this functionality in the *Maya* system using the IPtables [Izu03] tool, a well-known utility for Linux based systems.

Thanks to the firewall functionality of IPtables, we can easily isolate a router from receiving any kind of traffic. However, once a router has been disabled, we need control messages sent by the network manager to get them back up. To enable a previously disabled router we developed a protocol based on the use of UDP broadcast sockets, which basically implements a mechanism to send a message to a network node without using the routing protocol. To do that, on the booting process, each router will create a socket listening on a default management port (we have selected the 10500 UDP port) that is the only port that the firewall never blocks. This socket has the broadcast option enabled, so it also receives broadcasted packets. Note that, in order to avoid excessive flooding, the protocol was designed to ensure that a message would only be rebroadcasted once per node. For that reason, all management messages include sequence numbers and

Algorithm 4.3 OnReceivingaBroadcast() function of the *Maya* tool.

```
for (every received message) do
    Perform a table look-up;
    if (is an old message) then
        delete the message;
    else if (target IP == my IP) then
        execute the corresponding actions and send back an ACK message;
    else
        send the message to the IP broadcast address;
```

each router keeps a table with the last sequence number received for every IP address in the network. Algorithm 4.3 shows the steps followed by each router receiving a broadcast message on the selected UDP port.

4.7.3 Network Parameters Setup

For each router added to *Maya*, a link to the standard OpenWRT configuration web page is generated in *Maya*'s GUI (see Figure 4.15). Through this interface we can manually configure each router in the traditional manner. The *Maya* system offers an advanced functionality, commonly demanded by network managers, which consists of applying a new configuration to all routers, or only to a subset, at once. To do that we just have to pick the target routers, the network parameter to change, and the new value. With only a few clicks we can modify, for example, the ESSID or the channel at all selected routers.

To implement this functionality correctly, and to avoid network partitioning as we change critical parameters, it is important to choose the correct order to update routers and a suitable method to make those changes.

With respect to router parameter updating, *Maya* offers two different methods: an SSH connection if the router is enabled and reachable (which means that a path of enabled routers exists between the source and the target), or a set of UDP messages if the router is disabled or unreachable. As we have mentioned before, the order in which configuration changes are made in the network devices is important. If we want to change a parameter like the ESSID and we begin by applying changes to the nearest routers, all the remaining routers (more than one hop away) become unreachable, and so the updating process cannot continue. The right sequence for this task must be the opposite one: to begin applying changes on far away nodes with respect to the management server, and then gradually move towards routers that are nearer.

To obtain a snapshot of the wireless mesh network topology we build a tree including all active routers through ICMP echo packets sent to all routers, and we then check the TTL (Time To Live) field in the IP header. The lowest the TTL, the highest the number of hops. Hence, the algorithm starts by picking routers with the lowest TTL. Whenever a router is added to the network, and after any enable or disable action, TTL values are recalculated for each router; this value is constantly updated into the router database. If a router is disabled or

Algorithm 4.4 ApplyNetworkConfiguration() function of the *Maya* tool.

```
Get a list of selected routers having TTL=0;
    Apply changes to them;
Get a list of selected routers having TTL>0;
    Sort them in increasing order;
    Go through the list, applying changes one by one;
```

not reachable, ICMP packets will not be able to reach it. In this case we consider a TTL equal to zero.

Algorithm 4.4 shows the steps followed to establish the correct order for applying network configuration parameters.

4.7.4 Security Issues

Changing network configuration parameters is an action restricted to the network administrator. So, we must protect our system to avoid possible client-side attacks. To change the configuration on those mesh routers that are enabled we use SSH connections; since a password or the use of a public key is required, and since all the transmitted information is encrypted, the method is considered secure. However, when trying to perform similar configuration tasks on disabled routers, we are limited to the use of UDP messages. If they are sent as plain text, they can be intercepted and reproduced easily. So, for disabled routers, we use RSA public-key cryptography to provide a digital signature that ensures the authenticity of UDP messages. *Maya*'s public key is distributed to all nodes, and each node sends its public key back to the management application. A message signed with a sender's private key can be verified by anyone who has access to the sender's public key, confirming that the sender signed it and that the message has not been tampered with. Public keys are exchanged when a new router joins the network, and this process is carried out using an SSH connection. The management server stores the public key of each router in a database.

4.7.5 UDP Message Issues

The UDP protocol does not provide the reliability and data ordering that TCP does. So, datagrams may be lost or arrive out of order. Nevertheless, since it introduces no extra overhead to offer complex functionalities, it is faster and lightweight, being optimal for time-sensitive applications.

In the scope of *Maya* we cannot use TCP to access disabled routers since bidirectional communication channels are not available. The technique used is to broadcast UDP packets throughout the network, hoping that the target router is able to listen to the message. However, when changing the parameters of several routers, it is important to have a feedback about the failure or success of each individual operation. Hence, some sort of acknowledgement message is required. For that reason, when the destination router receives an UDP message, it sends another message back to report to the management server about the operation's

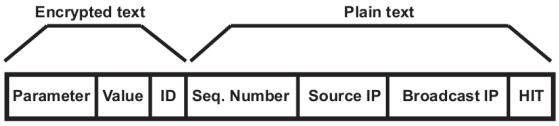

Figure 4.16: Format of the management UDP messages.

successfulness. In case that the operation fails for any reason, the UDP message must be transmitted again.

We found experimentally that three retries and three seconds of waiting time between retries is a setting which works fine in moderate sized mesh networks. However, default settings can be configured by the user. Figure 4.16 shows the UDP message fields used in *Maya*. Both the command message and the acknowledgement share the same format.

The message is only partially encrypted since some fields are required by intermediate routers along the path, and it is not efficient to decrypt and encrypt messages at every router. The encrypted fields include the parameter to change, the new value and an identifier (ID). The ID represents the last action executed, and it is used by the management server to create a system log for security reasons. With respect to the plain text fields, we used the sequence number, the source IP, the broadcast IP and a HIT field.

When an ACK message is not received after issuing a command and a retry is required, the ID remains the same but the sequence number increases. The sequence number and the source IP are the fields required to keep the routers' tables updated. Broadcast IP is the address where messages are sent to, and it represents the broadcast network address. The last parameter is the Host Identity Tag (HIT), a concept introduced by the Host Identity Protocol (HIP) [Gur08]. Basically, we use the SHA-1 algorithm to transform the public key into an unique identifier for a router. This parameter replaces the target IP.

4.7.6 Evaluation

We evaluated *Maya* using the *Castadiva* emulator (see Chapter 3) to confirm its correct behaviour. We studied the system performance under different traffic conditions to prove its reliability.

4.7.6.1 Performance Evaluation of *Maya*

Initially we test the functionality of *Maya* without any type of traffic: the only data flowing in the mesh network corresponds to the activity of the routing protocol. We then perform some tests by activating TCP traffic flows between randomly selected nodes. We evaluate the delay of the SSH connections and of the secure key exchanging, the effectiveness of the UDP messages, and the packet loss rate.

Our emulator uses four WRT54G Linksys routers for the distribution network. The main management server is a PC with a 2000+ XP AMD Athlon processor, 512MB of DDR RAM and an IEEE 802.11g Conceptronics USB card. It uses

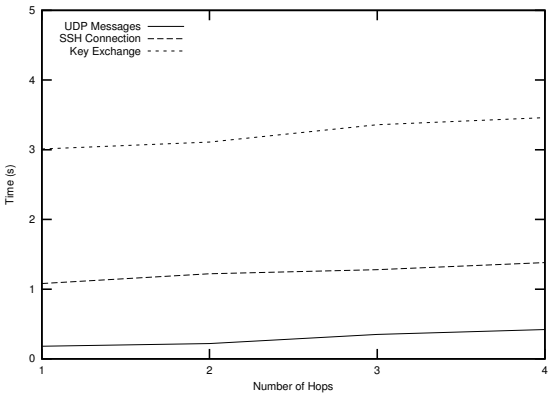

Figure 4.17: Comparison of the latency associated to *Maya*'s management tasks when varying the number of hops.

a Linux based operating system with wireless extensions v19 and wireless tools v28. We distribute the wireless routers in a linear topology where each router only reaches the previous and next router along the path. The management server is located at an edge of the network; hence, the maximum number of hops for a message is of four.

4.7.6.2 Evaluation of *Maya*'s Functionality

We first evaluate the functionality of *Maya*. We select a range of scenarios with different number of hops between source and destination and evaluate the time that *Maya* requires to perform the three following actions: (a) change network parameters of active routers using SSH connections, (b) change network parameters of disabled routers using UDP messages, and (c) do a secure key exchange. We evaluated two different cases: with an unloaded scenario where only the traffic is due to the AODV routing protocol, and a loaded scenario where the traffic generated by *Maya* has to compete with several bandwidth-hungry TCP flows.

Figure 4.17 shows the results obtained under no load. The results confirm that lowest times correspond to UDP messages since UDP is not a connection-oriented protocol. The two other functions are implemented using SSH-based commands, which implies additional traffic to create a secure connection. The key exchange process requires transferring files between the two endpoints. This operation consumes more resources than merely executing a set of commands from a remote node, which explains its higher cost. We also observe that distance does not significantly affect performance.

We now evaluate *Maya*'s effectiveness in a scenario where a set of TCP flows compete with *Maya*'s management traffic. This scenario is more realistic since typically the network is loaded when management and maintenance tasks take place. We varied the number of TCP flows from 1 to 20, and configured TCP connections to use frames of maximum size (1500 bytes). Figure 4.18 shows that, as the number of intermediate hops and the number of competing TCP connections

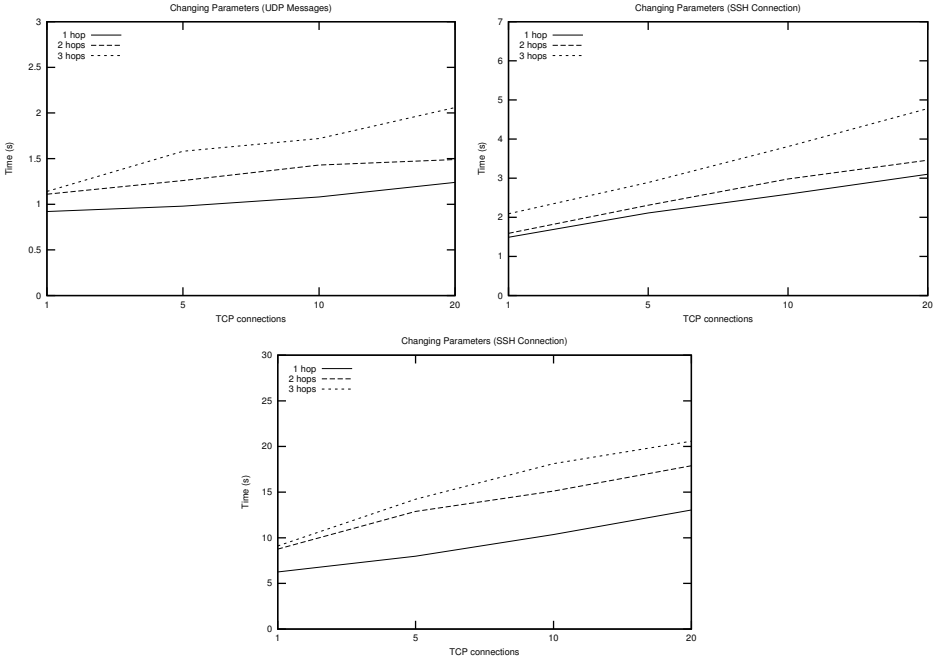

Figure 4.18: Overhead of management tasks requiring UDP messages, SSH connections and key exchanges when varying the number of TCP flows, at different hop distances.

increase, the delay in performing management operations grows steadily. Results also confirm that the operation suffering from higher performance degradation is key exchange. To complete a key exchange between the management server and a wireless router when setting the number of active TCP flows to maximum (20), *Maya* requires about 10s, 15s, and 20s, for distances of 1, 2, and 3 hops, respectively. Notice that these are worst case results that apply in extreme cases of congestion.

Comparing parameter updating through SSH and UDP, we again observe that SSH imposes a greater overhead due to both TCP connection setup delay and the authentication process. On the contrary, when selecting UDP broadcasts to communicate with disabled routers, datagrams may arrive out of order, appear duplicated, or be lost. In the event of losses, our application will re-send UDP messages for up to three times, which increases the overall delay to complete the corresponding operation, but also improves reliability.

Figure 4.19 shows the probability that an UDP message arrives correctly with respect to the number of hops and the network traffic. We observe that, as we increase the number of hops and the number of TCP connections, the probability of success decreases. For a distance of three hops the probability that the UDP message arrives correctly is of 50%, 47%, 39%, and 38% when we have a scenario with 1, 5, 10, and 20 TCP flows, respectively.

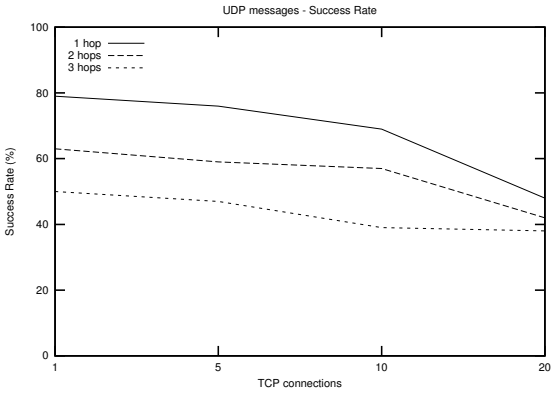

Figure 4.19: UDP message arrival probability.

4.8 Deploying *RuralNet* in Mozambique

Information and communications technology can provide rapid, low-cost access to information about almost all areas of human activity [Nat01]. From distance learning in technological Turkey and long-distance medical diagnosis in the Gambia, to information on market prices of grain in India, the Internet is breaking barriers, making markets more efficient, creating opportunities for income generation and enabling increased local participation. Sadly, this utopian promise applies only to an elite. While information poverty is rarely blamed as a direct cause of human suffering, the digital divide raises ethical questions of universal access. The entry costs to secure equipment and to set up services are far beyond the means of most third-world communities. Startup costs and expenses of technical maintenance compete with resources needed for essential human survival. But nowadays, policy makers are challenged to find justification for investment in ICTs when local and national resources are limited, and where the urgent needs of people for basic nutrition, health care, and education remain unsatisfied.

If the access of information is a basic need of the world's population, new ways of accessing the Internet are required to achieve this goal. *RuralNet* is an architecture for enhance the access to the Internet of an ISP, and extend the coverage are using a low-cost infrastructure, increasing the number of users that can directly benefit from the use of Internet. It is formed by off-the-self devices easy to find in the markets of a developing country, the architecture is easy to maintain, and the system is easy to use and administrate. Therefore, we believe that *RuralNet* accomplishes all the requirements to be a candidate solution to fight the digital divide.

4.8.1 Why we chose Mozambique as our scenario?

Since we work in the *Universidad Politécnica de Valencia*, we make contact with other associations hosted in this university which also fight the digital divide. In this way, we started a collaboration with the university association called *PoliClick,*

HDI rank		Telephone mainlines (per 1,000 people) 1990	1999	Cellular mobile subscribers (per 1,000 people) 1990	1999	Internet hosts (per 1,000 people) 1995	2000	Cost of a three-minute local call PPP US$ 1999	Index (1990 = 100) 1999	Waiting list for mainlines (per 1,000 people) 1990	1999
151	Malawi	3	4	0	2	0.0	0.0	0.12	122	1	3
152	Rwanda	2	2	0	2	0.0	0.1	..	..	(.)	1
153	Mali	1	..	0	..	0.0	(.)	..	..	..	..
154	Central African Republic	2	3	0	..	0.0	(.)	..	..	..	..
155	Chad	1	1	0	..	0.0	(.)	..	..	(.)	..
156	Guinea-Bissau	6	..	0	..	0.0	(.)	..	..	..	..
157	Mozambique	3	4	0	1	0.0	(.)	..	..	2	2
158	Ethiopia	3	3	0	(.)	(.)	(.)	0.15	47	2	4
159	Burkina Faso	2	4	0	(.)	0.0	(.)	0.37	..	..	..
160	Burundi	2	3	0	(.)	0.0	0.0	..	..	(.)	..
161	Niger	1	..	0	..	0.0	(.)	..	..	(.)	..
162	Sierra Leone	3	..	0	..	0.0	0.1	0.10	21	4	..

Figure 4.20: Diffusion of Technology.

and also with the non-governmental organisation *Telecomunicaciones Solidarias* (TeSo). These associations have contact with three scenarios in three different developing countries where we can take profit of using *RuralNet*. These three possible scenarios are: the *Universidad de Guinea Ecuatorial* (UNGE) in Equatorial Guinea, the *Universidad de Pinar del Río* of Cuba and the *Missionarios Vicentinos* of Mozambique. All these three countries are a priority target to the Spanish cooperation as showed by the *Agencia Española de Cooperación Internacional para el Desarrollo [AEC10]*.

Figure 4.20 shows an extract of the Human Development Report presented in 2001 by the United Nations [Nat01] regarding to the information and communication technologies (ICTs). As we can see, Mozambique is the number 157 of 163 countries analysed, rated as a country with low human development. In 1999 only 0.4% of its population has a cellular phone, and 0% of them have Internet connectivity. Equatorial Guinea (not shown in this figure) is in position number 110 (average human development), and Cuba is not rated in this document, but has a better economical situation than Mozambique. Therefore, Mozambique would be the country which has the greatest needs for investment in new technologies to decrease the lack of infrastructure of the ICTs.

We also chose Mozambique because there we met the community *Missionarios Vicentinos* of Nacala, a religious community involved in the development of the country with more than 20 years of experience. They also have the ideal scenario for testing the first prototype of *RuralNet*: different schools in a limited area, with the basic infrastructure needed for deploying our architecture.

Other problems have been found in the project evaluation to discard the other two countries, e.g., some policy problems were found in Cuba, or lack of trust problems with the counterpart, as found in the university of Equatorial Guinea.

4.8.2 Objectives in Mozambique

The objective of this project consists of deploying *RuralNet* in a real scenario, allowing the access to different TCP/IP services for different users located in rural areas without any affordable connectivity access to the Internet. Specifically,

Figure 4.21: Region of Nampula

our objective focused on connecting to Internet different high schools and occupational training schools of different rural areas, all managed by the *Missionarios Vicentinos*.

4.8.3 Deploying *RuralNet*

This project is designed using the knowledge of the research group *Grupo de Redes de Computadores* (GRC) of the *Universidad Politécnica de Valencia,* and the collaboration with the non-governmental organisation *Telecomunicaciones Solidarias* (TeSo), the university association *PoliClick* of the *Universidad Politécnica de Valencia* and the community *Missionarios Vicentinos* of Nacala (Mozambique). Basically, the GRC provides the technical knowledge and the software to deploy the infrastructure, while *TeSo* and *PoliClick* provide the computers, the experience in cooperation projects and other material needed in different places to deploy this project; the *Missionarios Vicentinos* provide the place and the manpower.

We sent three volunteers to Mozambique between years 2008 and 2010 for coordination purposes, and to teach the use, administration and maintenance of the network. As a result, each volunteer has done its master thesis in Computer Engineering about this activity [Bat08, GÍ0, Mar09].

4.8.4 Scenario

Our scenario is the city of Nacala, in the region of Nampula. Nacala, also known as *Cidade de Nacala* or *Nacala-Porto,* is on the northern coast of Mozambique. The city had 207.894 inhabitants in 2007, being the fourth city of the country by number of population. Its major industries were cement, sisal and cashew. Other

Figure 4.22: Selected schools of Nacala.

Node	Latitude	Longitude	Altitude
San Juan Bautista	14° 33'15.40"S	40° 41'32.20"E	138m
Mokone	14° 32'45.30"S	40° 41'35.00"E	109m
Miramar	14° 33'30.90"S	40° 40'59.80"E	126m
Cristo e Vida	14° 32'59.00"S	40° 42'12.90"E	139m

Table 4.1: Coordinates of each node deployed in Nacala.

important employers were the seaport, its small modern hospital, and services (banking, insurance and administration).

The *Missionarios Vicentinos* [Mis10] work in this city. They are a charitable organisation ruled by the church. Their main objectives are the evangelising of the population while improving their life-quality.

We focus on four different schools of Nacala: the *Escola Mokone*, the *Escola Taller "San Juan Bautista"*, the *Escola Miramar* and the *Escola Cristo e Vida*.

The selection of these schools was motivated because of the following characteristics: (i) all of them are managed by the *Missionarios Vicentinos* (ii) the *Missionarios Vicentinos* is composed by missionary people without any interest on taking profit of our system and (iii) these schools are public places where all the people can go to use the service provided.

The *Escola Taller "San Juan Bautista"* has Internet access using an ADSL connection. We can deploy a mesh network using the different centres as nodes and with it, allow this connection to be shared using *RuralNet*.

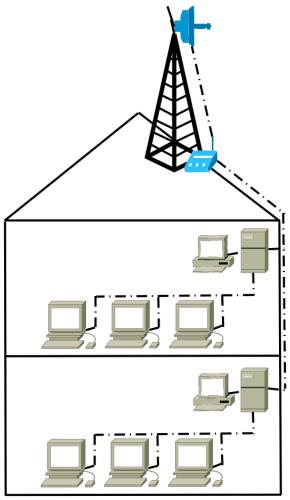

Figure 4.23: Infrastructure of a school.

4.8.5 Infrastructure

In this section we describe the infrastructure used for deploying *RuralNet*. As *RuralNet* is deployed to connect different schools, we differentiate among the structure inside a school, used to connect all computers among them and with an antenna, and the infrastructure used for connecting the schools among themselves.

4.8.5.1 School architecture

To make the infrastructure of each school cheaper, we mix wireless technology with Ethernet technology. Since we finally use desktop computers, we can use Ethernet technology to connect all of them avoiding the interferences produced in a wireless medium.

Figure 4.23 shows the schema of a school. Basically, each room has a proxy which is the teacher's computer. Each student's computer is connected to the proxy using a switch, which in turn is connected to an access point. The access point is also connected to an antenna used to extend its signal range, allowing to connect this school with other ones. This schema is repeated on each school.

4.8.5.2 Scenario architecture

Each school is connected with the other ones using an antenna, as being a node of a mesh network. The *Escola Taller "San Juan Bautista"* is in the centre of the network, being no more than one kilometre away from any of the other schools, and, therefore, we use an omnidirectional antenna. The other schools use a directional antenna to avoid interferences with among the other ones. Figure 4.24 shows the distance between each school.

With this infrastructure, each computer may take part of *RuralNet*. Also, all of them can have access to the main server, which controls the access to the Internet.

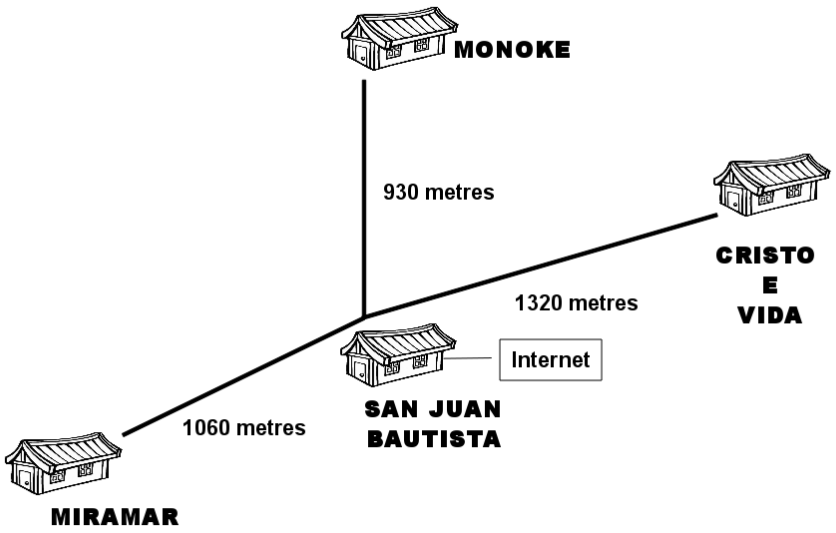

Figure 4.24: Distance between the *RuralNet* nodes.

Also, each computer can use the shared resources of *RuralNet,* such as the FTP server.

4.8.6 Final result

Now, more than 200 students are using *RuralNet* for accessing Internet in the different schools of Nacala. The deployed system also allows information sharing among the schools, as well as publishing documents and other files, and the use of VoIP phones to be in touch among them. Inside the *Escola Taller "San Juan Bautista"* we have installed the *Maya* application, which allows the control of the entire mesh network.

The success of this prototype is also reflected since the *Missionarios Vicentinos* are in contact with us, discussing the possibility of extending the system to others schools located in the same city.

4.9 Summary

In this chapter we presented *RuralNet* and *Maya*. *RuralNet* is a captive portal based architecture especially designed to provide Internet access in rural areas. *Maya* is the management tool for the infrastructure.

Figure 4.25: In order: deploying an antenna for *RuralNet* (left up), one of our members with an antenna (right up), a classroom of the school (left bottom), and one of our router before being installed (right bottom).

We first analysed the architecture of the system, putting into evidence the most important elements that conform *RuralNet*'s infrastructure. We then described three core elements of the system, which allow us to capture clients, perform a per-user regulation of bandwidth, and offer access to an administrator-defined group of free external web servers. We conclude with a experiment were we validate the mesh network used by the system, finding that the service offered to clients is the one we expect.

Maya simplifies the wireless mesh networks management process by allowing to perform critical tasks, such as global ESSID and channel changes, in a simple and efficient manner. We also presented a set of experiments where we evaluated the overhead of each type of management operation in two different scenarios: under no load and under heavy load. The obtained results indicate that *Maya*'s performance is stable under all tested scenarios.

Figure 4.26: In order: An antenna deployed for *RuralNet* (left up), another antenna (right up), one of our members installing an access point (left bottom), one of the schools of the project (right bottom).

We finally present a prototype of *RuralNet* deployed in Mozambique. This prototype successfully connects four schools in the city of Nacala, sharing the Internet connectivity of one of them, and allowing more than 200 students to have access to both local and global information.

Chapter 5

Security Improvements for Community Wireless Networks

In this chapter we present some security issues regarding to any MANET based network or mesh network like *RuralNet*. There are several kinds of attacks that can take place in these networks, but in this thesis we will focus solely on the attacks that are specific to the data transmission process, which are the more specific attacks of these networks. One of the main attacks against ad hoc networks affecting their routing protocols are named routing-disruption attacks. An example of these kinds of attacks is the selfish node, which uses the network but does not cooperate, saving battery life for its own communications. Another similar attack is the blackhole, which intends to disrupt the communication with its neighbourhood by attracting all traffic flows in the network and then dropping all packets received without forwarding them to their final destination. The existence of these attacks makes the network availability quite unpredictable. Notice that network availability is a minimum requirement for developing any commercial system, such as *RuralNet*.

We also present in this chapter the most common countermeasure used to avoid this kind of attack, called the watchdog technique [STKM00], and we perform a deep study of it. We found some drawbacks in this technique in scenarios with high degrees of mobility. Therefore, we develop a version of the watchdog based on bayesian filters [Ber93] which improve the accuracy of the standard watchdog, palliating its drawbacks.

The rest of this chapter is structured as follows. Section 5.1 shows an introduction of security on MANETs. Section 5.2 explains the objectives we want to achieve in this chapter. Section 5.3 describes how the standard approach of the watchdog works and presents some of its drawbacks. Section 5.4 explains how bayesian filters work, and how they can help us to improve the watchdog technique. Section 5.5 describes the evaluation made to validate the bayesian filters used in a MANET. Section 5.6 compares both watchdogs to show the improvements obtained with the new proposed mechanism. Finally, Section 5.7 draws the main conclusions of this chapter.

5.1 Introduction

As commented on Section 2.8, community wireless networks based on MANETs and mesh networks use specific routing protocols, like AODV or OLSR, where it is assumed that each node in the network is a peer and not a malicious node. However, this assumption does not reflect a real scenario, where nodes can have a malicious behaviour. The open and dynamic nature of ad hoc networks make them very sensitive to attacks. When routing protocols are affected by an attack, the whole availability of the network is compromised.

Attacks against ad hoc routing protocols basically follow the manipulation of the sensitive information exchanged among nodes to establish communication routes. Accordingly, adversaries may inject erroneous routing information, replay old routing information, or distort routing information. These actions may partition the network or introduce a certain traffic overload, thus causing retransmission and inefficient routing. The main objectives of this attack are resource saving and bandwidth appropriation. The malicious node disturbs the traffic on the network to save energy, not forwarding other node's packets and stealing the network bandwidth for its own use.

There is therefore an emerging need for research focused on the provision of practical proposals for securing ad hoc routing protocols. This research is essential to enable the exploitation of the potentials of such networks in commercial products such as *RuralNet*.

5.1.1 Black holing Ad hoc Networks

As reported in [HP04], one of the main attacks against MANETs affects their routing protocols, being named routing-disruption attack. An example of a routing-disruption attack is for an attacker to send forged routing packets to create a routing loop, causing packets to traverse nodes in a cycle without reaching their destinations, thus consuming energy and available bandwidth. An attacker might similarly create a routing black hole, which attacks and drops data packets.

Black hole attacks are one of the most common ad hoc network attacks [MS04]. In this attack, a malicious node uses the routing protocol to advertise itself as having the shortest path to the node whose packets it wants to intercept. This malicious node then can choose whether to drop the packets to perform a denial-of-service attack, or to use its position on the route as the first step in a man-in-the-middle attack. In a special case of a *black hole*, and attacker could create a *grey hole*, in which it selectively drops some packets but not others, for example, by forwarding routing packets but not data packets.

Other similar kind of routing disruption is the selfish node. A selfish node uses the network but does not cooperate, saving battery life for its own communications; it does not intend to directly damage other nodes. In practise, this is another kind of black hole attack.

In [R+08] authors have explored to what extent existing MANET routing protocols, such as OLSR or AODV, are sensible to selfish nodes, *black holes* and *grey holes* attacks.

In the literature, the typical countermeasure against a black hole, is to isolate the node from the entire network. This can be accomplished by preventing nodes to handle packets containing the MAC address of the malicious node, for instance, by inserting a new rule in each node's firewall. However, such type of solutions are only effective if the malicious node is detected by the other nodes of the network. Therefore, the development of Intrusion Detection Systems for detecting this kind of attack is mandatory to ensure the correct behaviour of the entire network.

5.2 Objectives to achieve

The objective of this chapter is to study the intrusion detection systems offered by the literature for detecting *black hole* attacks. Search if they fit our requirements for *RuralNet* and, if not, develop a new tool or improve an existing one. The solution developed should be robust in scenarios with some degree of mobility, which is an important characteristic in MANETs. Also, it should have a limited number of false detections to ensure the effectiveness of the solution developed.

Finally, the proposed solution should be tested in different scenarios with different number of nodes, and also should be tested in different devices, to ensure its compatibility with any device that forms the MANET.

5.3 Watchdog-based Intrusion Detection Systems (IDS)

It is widely-known that imperfectness of preventive security measures lead computer systems to embed vulnerabilities that can be exploited by malicious adversaries [DR03]. The existence of a vulnerability per se does not cause a security hazard, and in most of the times they can remain dormant for years. An intrusion is the consequence of an attack that succeeds in the exploitation of an existing vulnerability. This section defines how watchdogs can lead to the detection of such activities, and it is also justified why such mechanisms can be considered today as the basic bricks for the development of more sophisticated IDSs for MANETs.

5.3.1 Watchdogs and their Importance for MANETs IDSs

The watchdog is an intrusion detection mechanism proposed in previous studies [PC03, CLY06]. The main idea behind this IDS is that, because a node can listen to the packets traversing its neighbourhood, it can monitor their activity. Therefore, watchdogs act in promiscuous mode, thus overhearing all next nodes forwarding transmissions. With the information about the neighbourhood behaviour, the watchdog can deduce if nodes are acting as selfish, *black hole* or *grey hole* routers. This technique is independent of the technology and routing protocols used. They cope well with these kinds of attacks in any kind of ad hoc network such as MANETs or mesh networks.

Figure 5.1 shows a basic example of the watchdog behaviour. Node "A" can send packets to node "D" using either route "{A-B-C-D}" or "{A-M-D}". The

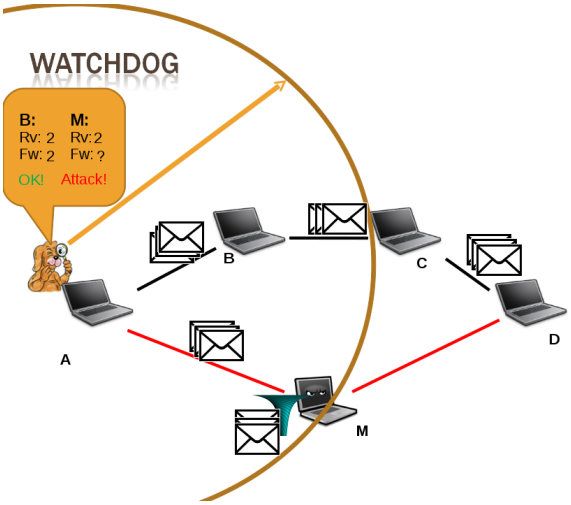

Figure 5.1: The watchdog technique.

watchdog can listen to the packets forwarded by B and M who are in range. "B" forwards all packets to "C" but "M" performs a black hole attack and drops all received packets. When "M" does not send the packet, the watchdog becomes aware of it and marks it as an attacker.

Several proposals use the watchdog as the basic brick of IDS solutions. In the Pathrater approach [MGLB00], each node uses the information provided by watchdogs to rate neighbours. The Routeguard mechanism [HZH05] combines the watchdog and Pathrater solutions to classify each neighbour node as Fresh, Member, Unstable, Suspect or Malicious. Other approaches like hob-by-hop signing [ZRCK03] and Patwardhan [PPJ+05] extend the detection capabilities provided by the watchdog with public key encryption and signatures. As can be seen, watchdogs are at the core of the most important types of IDS solutions for ad hoc networks.

5.3.2 Design Approach

As previously established, the main functional requirement of the watchdog in a particular node is to supervise the activity of the node's neighbours in order to determine whether or not they follow the routing rule imposed by the considered routing protocols. The goal is to meet this requirement while providing a portable solution that can be installed in devices with resource constraints such as the intermediates nodes that compose *RuralNet*'s backbone network. Portability refers here to the ability of the watchdog component to remain protocol and platform independent in order to be usable in the context defined by different routing protocols and different runtime supports.

The next subsections focus on the design decisions that must be considered to produce a portable watchdog component.

5.3.2.1 Detection Approach

To determine whether a node exhibits a malicious behaviour, our watchdog counts all the packets received from its neighbours, and the packets that must be forwarded (those that are not addressed to the node where the watchdog is under execution). A *neighbour trust level* can be defined as the ratio between the received packets for forwarding and those effectively forwarded by the neighbour node. A node forwarding all received packets has a neighbour trust level of 1 (100%). When a node does not forward the received packets, the watchdog changes its state to *untrusted*, and marks it as malicious node. Although an ideal *neighbour trust level* is 1 (100%), collisions and signal noise make that, in practise, such value level is rarely attained.

5.3.2.2 On minimising false watchdog detections

The work published in [MGLB00] reports on the effects of false detections of watchdogs; as it is mentioned in this work, mobility of nodes and collisions limit the detection accuracy of watchdogs, leading them to provide false detections. The effect of such aspects is not considered in the evaluation of the watchdog finally provided, which limits the representativeness of such study in practise.

It is difficult for a watchdog to differentiate whether the loss of a packet is due to an attack or a collision. In this latter case, if an alert is generated, this may lead to the generation of a false positive. In order to mitigate to some extend that problem, our proposal is to introduce in the watchdog design the definition of a *tolerance threshold*. This threshold defines a certain packet loss tolerance. As a result, a node is considered as being malicious if the degree of packet loss of such node exceeds the established watchdog threshold. In other words, an alert is generated whenever the *trust level* of a particular neighbour becomes smaller than one minus the considered threshold.

It is worth nothing that, despite the interest that the inclusion of *tolerance thresholds* may have for facing the problem of the generation of false positives, it may lead to the introduction of another one: the generation of false negatives. A false negative appears when an attack is not detected by the IDS. If intrusion detection time increases, false negatives can appear, and both intermittent and temporal attacks may remain undetected. Temporal attacks are those perpetrated by nodes that exhibit a malicious behaviour during a limited amount of time. The existence of false negatives is neither reported in the previously mentioned paper, although they exist, as this work demonstrates.

It is important to note, that due the high mobility of the nodes in a MANET, most attacks are temporal. The solution designed to face this problem is the use of *devaluation* techniques that consist in decreasing the weight of the oldest received packets along the time, and thus their influence in the computation of the tolerance threshold. We claim that this design decision is more interesting than an "Amnesia" technique (only use present information) strategy since such amnesia may increase the number of generated false positives.

5.3.2.3 Considerations for detecting *Grey holes*

Although the previously identified design decisions are well-suited for *black hole* attack detection or selfish nodes, some additional considerations are required when facing *grey hole* attacks. As introduced in Section 5.1.1, the goal of *grey hole* attacks is to block a particular kind of network traffic. If the traffic blocked by a *grey hole* is only a little portion of the entire traffic, it is not enough to reduce the neighbour trust level for coping with its detection. In addition, the watchdog must be able to differentiate the different types of packets handled by each node in the neighbourhood. So, different thresholds must be computed for each type of packet traversing the network. A node is thus considered as malicious if the number of lost packets of the same kind exceeds the established *tolerance threshold*.

5.3.3 Implementation Trade-offs

In this section we detail the characteristics of the implementation of the watchdog. We implement this component using the C programming language to ease portability to devices such as laptops, PDAs or routers.

The implemented Watchdog performs five steps: (i) reads the packets from the wireless card, (ii) generates the neighbourhood, (iii) detects the black hole attack, (iv) releases consumed resources if they will not be used any more, and (v) sleeps for a random time for resource saving purposes.

The first action of our watchdog is to read each packet that arrives at the wireless card. The card is set to the promiscuous mode to listen to all neighbours' packets in range. Here we can choose between using the Netfilter library or implementing our own socket. We decided to implement our own socket to obtain a standalone application avoiding libraries dependence and, therefore, avoiding dependence with a specific Linux distribution or kernel version.

With the information provided by each obtained packet, the watchdog can define its neighbourhood and decide if any neighbour is performing an attack. To define the neighbourhood list, the node running the watchdog must read each packet received. This includes not only the packets addressed to it, but also all those using the promiscuous mode of the wireless card. It compares the IP address of the sender to know if it is a packet sent by itself, by an already stored neighbour or by a new one. If it is a new neighbour, it stores the neighbour identity by using its IP address and then trying to discover its MAC listening to all the ARP packets.

To detect an attacker, for each listened packet the watchdog distinguishes if it must be forwarded or not. If it must, the packet is stored in a buffer until the packet is sent to the next hop. When the packet is sent, it is removed from the list and the watchdog marks the behaviour of the listened node as normal. If it is not forwarded, when a timeout expires, it is considered a lost packet and the watchdog decrease the *neighbour trust level* of the node in charge of forwarding it. If the number of unforwarded packets is higher than the percentage of *tolerance threshold* of the watchdog, the node is tagged as malicious node and an alarm to other nodes or to a logger is sent, depending to the security policy applied on the network.

For resource saving, the program searches for expired data stored and deletes

them to avoid large memory consumption. If the watchdog does not receive any kind of packet from a given neighbour for a long time, the neighbour is considered as out of range, and it is deleted. This timeout is defined by the user when the watchdog is executed.

Finally, our implementation also has the option to enter in a sleep mode randomly, saving energy and CPU consumption.

We must point out that the implementation is independent of the routing protocol used since it does not modify the protocol behaviour and only listens to the packets forwarded by nodes.

This implementation of the watchdog is released as open source software and it can be downloaded from `http://safewireless.sourceforge.net`.

5.3.4 Countermeasures proposed

When an attack is detected, a node can apply its own policy to block the attack. However, it is also convenient that this information is spread to all the neighbourhood to inform the rest of the nodes.

To send an alarm, there are several options available in literature. We propose that the watchdog uses the common message structure of the common message loggers of Linux system called Syslog-ng [Sec08], thus allowing compatibility with other applications. We chose this implementation for the alerts and excluded other standards more extended on IDS such as IDMEF [MM07], for simplifying the integration of the watchdog with limited devices such as mobile nodes.

Syslog-ng allows nodes to send messages to their neighbourhood only by knowing the IP of the other nodes, which is obtained by the Watchdog. The alarm message has several fields: the severity of the alert, a timestamp, the IP of the node that sent the alert, the PID of the watchdog, an explicative alert with the IP and MAC of the attacker, and the routes affected by the attack. With this information, the neighbourhood can launch some countermeasures. Next, we list an example of them.

5.3.4.1 Isolating the malicious node

The simplest way to avoid a malicious node is that each node detecting the attack or receiving the information about the attack, isolates the malicious node. This task is easily performed by the firewall of each node. As we use Linux based systems, our firewall is the Iptables [Izu03] tool. Algorithm 5.1 shows the set of rules required to perform this action.

The MAC in the iptables rule to be applied is the malicious node's MAC. First the node blocks all packets of the malicious node, then it waits for a short time (i.e. five minutes) and then it finishes the isolation. The isolation must finish to avoid denial of service attacks if the malicious node clones a MAC address.

5.3.4.2 Select an alternative route

Another way to avoid the attack is to force the routing protocol to choose another router. This can be achieved changing the routing table. Algorithm 5.2 shows the

Algorithm 5.1 Isolating a malicious node.

```
iptables -A INPUT -p udp -m mac –mac-source neighbour's mac -j DROP;
iptables -A OUTPUT -p udp -m mac –mac-source neighbour's mac -j DROP;
sleep 300;
iptables -D INPUT -p udp -m mac –mac-source neighbour's mac -j DROP;
iptables -D OUTPUT -p udp -m mac –mac-source neighbour's mac -j DROP;
```

set of rules required to perform this reaction.

Algorithm 5.2 Selecting an alternative route.

```
route -del host destination_node_ip;
route -add host destination_node_ip gw new_neighbour_ip;
```

In these rules, the *destination_node_ip* is the node starting the Ekiga call, and the *new_neighbour_ip* is another neighbour that is used to forward the packets. Only one node must apply these rules.

5.3.5 Evaluation of our watchdog using *Castadiva*

In this section we present a case study for our watchdog using real devices.

5.3.5.1 Experimental Setup

We deployed a small prototype of *RuralNet* using the *Castadiva* emulator presented on Chapter 3. Our studied network integrates five nodes, named A, B, C, D, E and initially has the topology shown in Figure 5.2. The *RuralNet* core does not affect this scenario since the Internet connectivity is not necessary in this case. Nodes A and D acts as users of *RuralNet* and are Ubuntu-based laptops running a VoIP application (called Ekiga [San10]). Thus, an Ekiga client runs on each of these laptops. The other intermediate nodes (B, E and C) act as the backbone network of *RuralNet*. The malicious node M also runs as a process in another access point. We start an Ekiga conversation (that generates about 145 packets/s) between the laptops at both extremes. Node "A" runs the watchdog and monitors all the packets forwarded. The computer running the watchdog is a Pentium IV Core Duo 3Ghz with 1GB of RAM. We test the watchdog in different scenarios using a proactive routing protocol (OLSR) and a reactive routing protocol (AODV). For the OLSR we use the OLSR-Unik implementation [AAO04], and for the AODV we

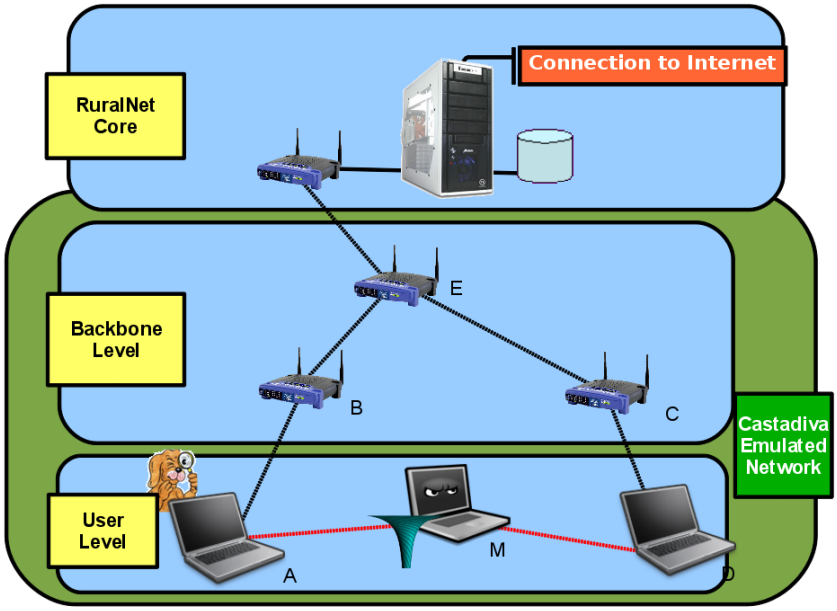

Figure 5.2: Experimental setup: A watchdog in *RuralNet*.

use the implementation of the Uppsala University [oU10]. We use both protocols to check the independence of our component against the routing protocol used.

Malicious node M performs a black hole attack [HP02] on all VoIP-related packets exchanged between A and D. If the attack meets its objective, then the video and voice flows between A and D will be dropped, thus freezing the VoIP call. Initially M is not part of the network. The attack approach proposed in this paper copes with the aforementioned challenge.

5.3.5.2 Validating our Implementation of the Watchdog

The first step is to validate our implementation of the watchdog. For this reason we generate the simplest possible scenario in a MANET and perform a *black hole* attack. We use *Castadiva* for emulating this scenario, shown in Figure 5.3. In this test, a previous communication between node A and node D is hijacked by node M, who later performs a *black hole* attack. Node A is running a watchdog, which detects the attack in the 100% of the cases. This is an optimum scenario where there is no noise or other factors that can affect the watchdog's accuracy. Therefore, in the next section, we perform a deeper study of the watchdog and the presence of false positives and false negatives caused by environmental noise.

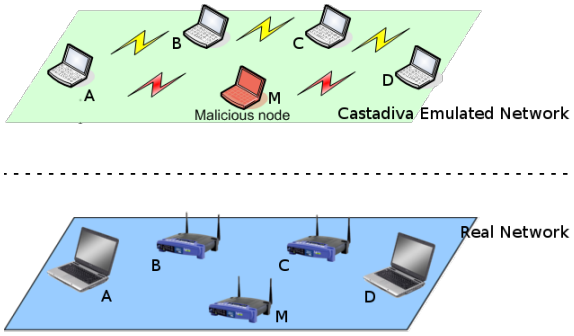

Figure 5.3: Throughput with different levels of noise.

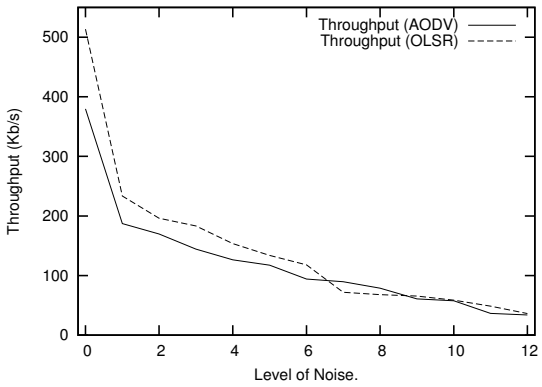

Figure 5.4: Network's throughput affected by noise.

5.3.5.3 Watchdogs in Scenarios with Noise

In this section we evaluate the false positives and the false negatives obtained by the watchdog due to environmental noise.

We consider a scenario with *minimum interference* where only Ekiga is running. A scenario with *maximum interference* is defined as the maximum noise tolerated by a videocall where the image is not frozen for more than one second. We call *noise level* to the number of traffic flows used to generate noise. Each traffic flow is composed by 200 UDP packet per second, and the size of each packet is 1024 bytes. We select 200 packets for each traffic flow because less packets make no significant visual changes on the video test, and so there are no significant changes on the test. Our test shows that a noise level of 12 traffic flows, is the maximum interference that does not freeze the videocall for more than 1 complete second for every 3 conversation seconds. We consider this interference as the maximum tolerated by an user. Figure 5.4 shows analytically how the different levels of noise affect the throughput of the network.

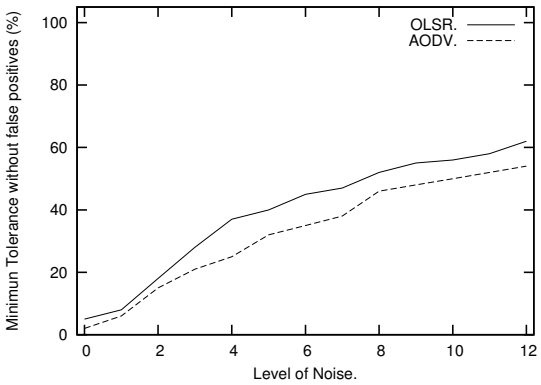

Figure 5.5: Relation between the minimum *tolerance threshold* needed to avoid false positives using OLSR and AODV.

False positive Figure 5.5 describes the results of the test performed when varying the background noise from 0 to 12 traffic sources. This figure shows the *tolerance threshold* needed by the watchdog to avoid a false positive using both OLSR and AODV protocols. We can observe that, when we increase the noise, a higher *tolerance threshold* is needed by the watchdog and less packets are received by the users of the videocall. This need of increasing the tolerance threshold is due to the packet loss due to the collisions caused by the background traffic.

Concerning the different routing protocols, we observe that, when the watchdog is used in a MANET or a mesh network with AODV, the *tolerance threshold* required is smaller than in a network using OLSR. This is explained because, in our tests, the videocalls with AODV have less average traffic throughput. This means that less traffic is forwarded on each hop and, therefore, there is less probability of collisions that prevent the watchdog from listening the packet forwarding.

False negatives Figure 5.6 (up) shows an example of the interval when the watchdog can generate a false negative when the *tolerance threshold* is set to 50% (a very high value only for testing purpose) and has stored previously 10.000 successful forwarded packets without any *devaluation* technique. The traffic analysed is the one generated by the Ekiga videocall tool. In this case we can see an interval of almost 90 seconds where an attack can be performed without any notification by the watchdog. This detection time is proportional to the number of packets previously stored.

Figure 5.6 (bottom) shows the impact of different *devaluation* values in the time required to detect a malicious node for the same test. We make different tests changing its value from 2000, 5000, 10000 or 20000 maximum packets stored. Using only the last 2000 or 5000 packets stored, the detection time of the attack is increased and the watchdog has a worse performance. For values of 10000 and 20000, we can see the reduction of the detection time for the attacker, especially when there is a huge amount of packets stored. The more packets we devalue, the

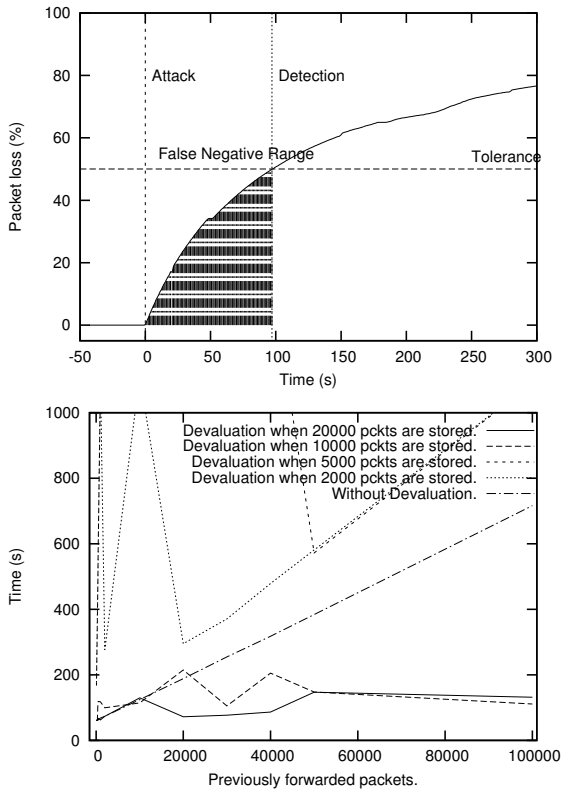

Figure 5.6: False Negative interval when the *tolerance threshold* is set to 50% (up) and relation between the time needed for detecting an attacker and the number of packets forwarded previously when using different values of the *devaluation* option (bottom).

faster we detect the attack. But if we set a value for the *devaluation* option that is lower than the number of packets received per second multiplied by the seconds that the watchdog stores a packet before decides if it is lost, the watchdog is unable to detect any attack in a short time. In Figure 5.6 (bottom) we can appreciate a comparison of this mechanism with different tests. As we can see, for values lower than 10000 packets the time to detect an attack is increased. But for a value of 10000 or 20000 packets we obtain a substantial improvement. We can summarise that *devaluation* is an improvement to reduce the attack detection time.

5.3.6 Evaluation using ns-2

In this section we perform several tests using the ns-2 [UBr98] simulator. We chose this simulator instead of *Castadiva* because we want to test scenarios with a high number of nodes. In this case, the use of a simulator is recommended. In order to use the ns-2, we implemented a specific watchdog module for this

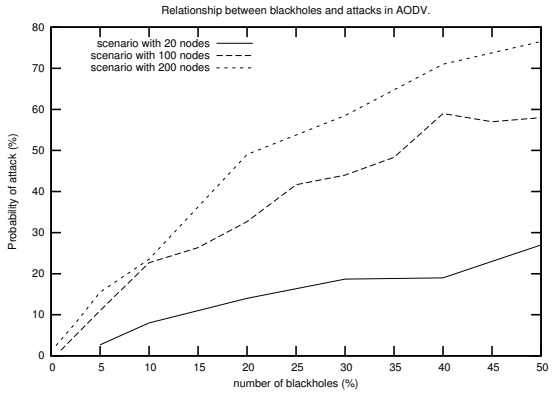

Figure 5.7: Probability of an attack when varying the number of nodes and the percentage of attackers.

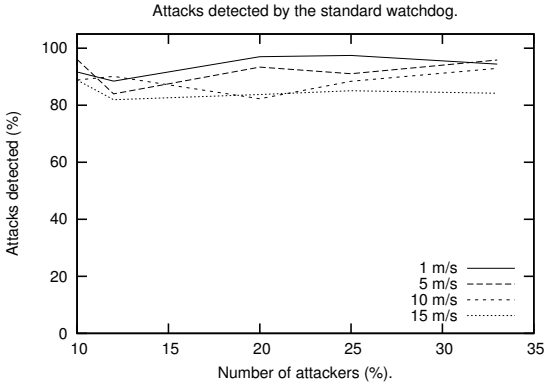

Figure 5.8: Attacks detected by the watchdog when changing the mobility of a scenario.

simulator (available at `http://safewireless.sourceforge.net/`). Using this simulator we can test networks with a large number of nodes, changing the number of attackers and their mobility. Figure 5.7 shows a preliminary study of how the percentage of malicious nodes and the total number of nodes of the scenario affects the probability that an attack is performed in a traffic flow. As we can see, not only the percentage of attackers affects the probability of finding an attack in one test, but also the total number of nodes in the scenario.

Afterwards we implemented the watchdog mechanism for this simulator and performed several tests varying the mobility of the nodes and the number of attacks to assess the effectiveness of the watchdog. Figure 5.8 shows the results obtained with different parameters. We can see that mobility clearly affects the number of attacks detected; however, it decreases as mobility increases. With a mobility of 1 m/s, nearly 100% of the attacks are detected. However, when we increase mobility

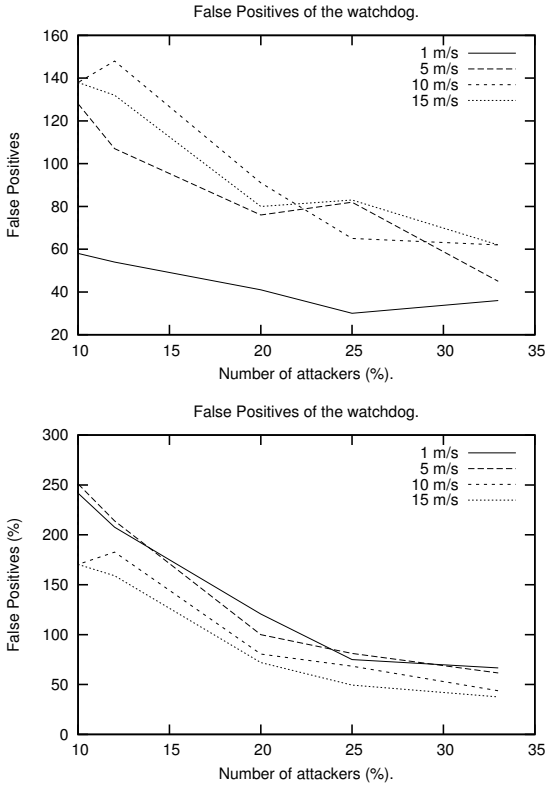

Figure 5.9: Number of false positives generated (up) and false positive ratio (bottom) when changing the mobility of a scenario.

up to 15 m/s, the detection is reduced to 80%. These results are independent of the number of malicious nodes deployed in the simulation. It affects o the total number of attacks, but not the ratio of the attacks detected.

Another studied effect is the false positives problem. Figure 5.9 presents a ratio between false positives and attacks in the simulation. Here we can see that, when the degree of mobility and the number of nodes increases, the ratio of false positives decreases. Despite the fact the number of false positives is increased when we increase mobility, the number of attacks is also increased, causing the number of attacks detected to increase too. Therefore, the total ratio of false positives is decreased.

Overall, we conclude that the watchdog does not cope well with mobility, especially at high node speeds. In fact, the higher the node speed is, the more false positives and false negatives the watchdog incurs in. A deeper study about the relationship between watchdog performance and mobility is discussed in Section 5.3.7.

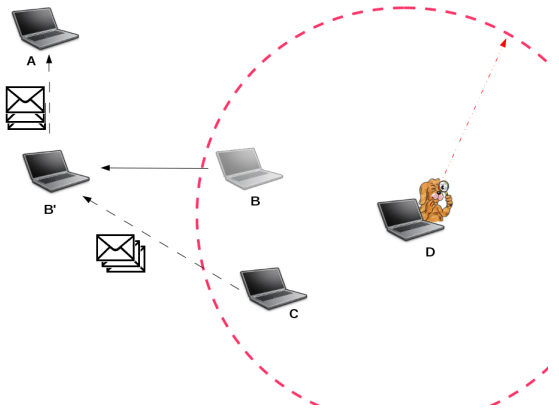

Figure 5.10: False positives due to watchdogs timeouts.

5.3.7 Detected drawbacks of the watchdog mechanism

The main problems detected for a watchdog based mechanisms are: (i) how the environmental noise affects the watchdog and the difficulties to cope with it, and (ii) how the watchdog can infer whether a node is in range or not when nodes have a high degree of mobility.

Although the watchdog methodology should be enough to detect malicious nodes, packet collisions and signal noise cause, in practise, the false positives and false negatives problem to emerge. It is difficult for a watchdog to differentiate whether the loss of a packet is due to an attack or a collision. In this latter case, if an alert is generated, it may lead to the generation of a false positive. This effect is palliated by the use of a tolerance threshold. This tolerance means that a node will ignore a percentage of packet loss. Hence, the value of this parameter represents a trade-off between detection speed and false positives. If we pick a low tolerance value, noise would cause benevolent nodes to be marked as malicious. If the tolerance value is set to high, the watchdog will need too much time to detect an attack. In fact, when it is performed in MANETs or wireless mesh networks with a high degree of mobility, the possibility of detecting an attack becomes minimal.

The previously presented Figure 5.6 (up) shows a schema of the watchdog response in the presence of an attack. In this experiment there is an interval of almost 90 seconds where an attack can remain unnoticed by the watchdog. As shown in this figure, the watchdog methodology requires sniffing enough data packets to decide whether a node is an attacker. This means that more time is needed to make a decision compared to a network without a tolerance threshold. If the attacker is moving, there is a possibility that the malicious node moves outside the watchdog's signal range, and thus it would not be detected. Therefore, false negatives can appear, and both intermittent and temporal attacks may remain undetected.

The second problem is how a watchdog can determine whether a neighbour

is in range or not. As we remarked before, the watchdog has the advantage of using only local information, but this has also some disadvantages, such as the watchdog not knowing when a neighbour goes out of range. This problem is solved by using timeouts: when the time that passes after the last neighbour packet listened surpasses a certain value, the watchdog considers this neighbour to be out of range and will not consider it for future tests. The main problem of this strategy is how to find the best timeout. A low value forces the watchdog to restart all calculations for a neighbour before a decision about it being malicious or not is made, possibly not detecting a malicious node and thus causing false negatives. A high value causes that, when a neighbour goes out of range, the watchdog would consider it to be in range for a long time. In that case, the watchdog would expect retransmissions from this neighbour, but would not listen to any. As a consequence, it would decide that this neighbour is a malicious node, thus causing false positives. Figure 5.10 shows an example of a false positive caused by a high timeout value. When node B moves to position B', the watchdog of node D thinks that B is within range until the timeout is triggered. As a consequence, D would expect to listen to the packets forwarded by B, and, otherwise, it would mark it as being a malicious. This is the main reason why, in Section 5.3.6, the false positives are slightly increased when we increase the mobility of the nodes. We can also consider this as another type of scenario affecting the watchdog's performance.

5.4 Adapting Bayesian Filters to IDS of MANETs: The Bayesian Watchdog

We now propose to use a bayesian filtering technique to filter the noise caused by node mobility in the watchdog monitoring process. This watchdog based on bayesian filters, can fade the false positives and false negatives problem using historical information obtained by the watchdog in the previous time. The obtained proposal is protocol independent, allowing it to be easily adopted by any kind of MANET.

5.4.1 Bayesian Filtering.

Bayesian filters [Ber93] probabilistically estimate a dynamic system's state from noisy observations. At time t, the state is estimated by a random variable θ, which is unknown and this uncertainly is modelled by assuming that θ itself is drawn according to a distribution that is updated as new observations become available. It is called *belief* or $\mathrm{Bel}_t(\theta)$. To illustrate this, let's assume that there is a sequence of time-indexed observations $z_1, z_2, ..., z_n$. The $\mathrm{Bel}_i(\theta)$ is then defined by the posterior density over the random variable θ conditioned on all sensor data available at time t:

$$Bel_t(\theta) = p(\theta|z_1, z_2, ..., z_t)$$

In our approach, the random variable θ belongs to [0,1]. Then we use for the *belief* the distribution $Beta(\alpha, \beta)$ that is suitable for this interval:

$$Bel_t(\theta) = Beta(\alpha_t, \beta_t, \theta)$$

where α and β represent the state of the system, and it is updated according to the following equations:

$$\begin{cases} \alpha_{t+1} = \alpha_t + z_t \\ \beta_{t+1} = \beta_t + z_t \end{cases}$$

The Beta function only needs two parameters that are continuously updated as observations are made or reported. In our approach, observation z_t represents the information from the watchdog about the percentage of non-forwarded packets obtained in time interval $[t, t+1]$.

5.4.2 Why Bayesian Filters?

Working on the protocol behaviour or studying the topology changes is not the only way for detecting misbehaviour in a system. In the literature we can find a reliable and extensive set of tools for detecting abnormal behaviours considered malicious in other fields, such as the SPAM filters. A SPAM filter can segregate illegitimate spam email from legitimate email. These email filters are usually based on bayesian filters [SDHH98], which allow the mail client to learn about the user decisions. Bayesian filters are not only useful for detecting SPAM. Other works such as [BB04, dLHMR08] also successfully use bayesian filters for predictions of abnormal behaviour. Buchegger et al. [BB04] uses them for implementing reputation systems for P2P and MANETs, while Leoni et al. [dLHMR08] use bayesian filters for predicting disconnections in a MANET. Thus, bayesian filters seem to be a useful tool for detecting abnormal behaviour in noisy environments and, therefore, a good tool for improving our intrusion detection system. Our proposal is to combine bayesian filters with the information obtained by our watchdog to design a tool capable of segregating malicious nodes from benign ones using historical information, thereby preventing both false positives and false negatives.

5.4.3 Assumptions

In a few words, each node on the network has some assumptions that can be summarised as follows:

1. Every node installs a watchdog, thus allowing for detecting misbehaviour (e.g., the number of packets that nodes should forward but that they do not do).

2. The percentage of packets that nodes do not correctly forward is used as input for the bayesian filters in order to predict the percentage of non-forwarded packets in the near future. If such a percentage is higher than a certain threshold, then the node is considered a malicious node. Note that it is not possible to assume that "good" nodes forward correctly all packets because of radio noises, packet losses and other similar characteristics of the aerial medium, which cause some delivery attempts to fail.

3. Every node that detects this malicious behaviour enables consequently appropriate actions to avoid malicious nodes from influencing the network's functionality. Every device can take its own recovery actions, or, conversely, all nodes can reach a consensus on the collaborative actions to deal with the situation. However, we focus on signalling malicious nodes, assuming another component to take care of mitigating the consequences of such attacks.

5.4.4 Bayesian Filtering Adapted for our IDS

Our approach is based on the information of the incoming packets that devices have not forwarded, and that nonetheless they should have done. Our bayesian watchdog relies on some basic assumptions:

1. Every device is equipped with a wireless card that allows for promiscuous mode: any device can listen to the packets traversing its neighbourhood and, hence, monitor the activity of one-hop distant nodes.

2. Each node has an implementation of a watchdog sensor (let's indicate as i). The i-th watchdog of a given node monitors the incoming and outgoing traffic of every neighbouring node. In this way, by analysing the packet headers, it is able to count the packets that nodes did not forward.

3. Each node has at least three neighbours. We assume a density of the network that makes different paths possible for reaching a destination, and each node is monitored by different neighbours.

The watchdog of device i is in charge of listening the traffic in its neighbourhood and calculating the percentage of packets not correctly forwarded by every neighbouring device j. If a given j forwards less than a given percentage of packets, the watchdog considers j as misbehaving. Device i does not know a priori such a percentage for each neighbouring node j and, therefore, it defines a random variable $\theta_i(j)$ to estimate it for j. In fact, $\theta_i(j)$ is the viewpoint of device i for what concerns device j. It is worthy highlighting that taking only the last observation is not sufficiently reliable since this could be effected by noise. So, old observations should be considered.

Our watchdog makes use of bayesian filtering. Variable $\theta_i(j)$ complies with the Beta distribution with parameters $(\alpha^{(i,j)}, \beta^{(i,j)})$. These parameters are continuously updated with new incoming observations of the percentage of non-forwarded packets. Node i makes periodical observations each t seconds (with t constant) of the behaviour of node j. Let s be the percentage of packets observed by i that are not forwarded by node j in this observation period. Parameters $\alpha^{(i,j)}$ and $\beta^{(i,j)}$ are updated as follows:

$$\begin{cases} \alpha^{(i,j)} := u \cdot \alpha^{(i,j)} + s \\ \beta^{(i,j)} := u \cdot \beta^{(i,j)} + (1 - s) \end{cases} \tag{5.1}$$

Values $\alpha^{(i,j)}$ and $\beta^{(i,j)}$ are initially set to 1.

Variable u is a fading mechanism for past experiences. This fading mechanism allows for redemption of a neighbour if its behaviour changes to a correct one along

Algorithm 5.3 Pseudocode of the bayesian algorithm for predicting *black hole* attacks.

BayesianBuffer[x,y]: a bi-dimensional square matrix storing (α, β) for each couple of nodes X and Y.
watchdog: is the observation obtained by the sensor. Represents a % of the packet loss.

BAYESIAN FILTERING:
for node ϵ MANET **do**
  start new watchdog sniffing.
  after expiring of TIMER()
    BayesianBuffer[x,y].alpha ←u * BayesianBuffer[x,y].alpha + watchdog
    BayesianBuffer[x,y].beta ←u * BayesianBuffer[x,y].betha + (1-watchdog)
done

the time. This fading mechanism will be useful if there are false positives due to the environmental noise. Greater values for u correspond to considering the old observations more significantly.

5.4.5 Watchdog Reputation Rating

With the beta function defined previously we can define the reputation function of node j on node i $R_i(j)$ using the estimated distribution $Beta(\alpha_i(j), \beta_i(j))$ of variable $\theta_i(j)$:

$$R_i(j) := \begin{cases} 1 & P(\theta_i(j)) < \gamma \\ 0 & P(\theta_i(j)) \geq \gamma \end{cases} \tag{5.2}$$

where

$$P(\theta_i(j) < \gamma) = \int_0^{\gamma} Beta(\alpha_i(j), \beta_i(j))$$

If $R_i(j) = 0$, node i reputes j as malicious. This means that node j is malicious if the estimated percentage of packets that are not correctly forwarded is more than a given value γ, named *tolerance threshold*. This tolerance threshold may depend on the environmental noise and must be defined for each scenario.

Figure 5.11 shows an example of a reputation function using a beta distribution when alpha is equal to 1 and beta is equal to 3. If we set γ to 0.6, the probability of being a benign node is the area of the beta function highlighted in the figure; the probability of being a malicious one is the complementary area of the function. In this example we can assume that the estimated node is non malicious because more than 60% of the function's area is in the quadrant defined by a value gamma of 0.6.

5.4.6 Implementation trade-offs

The bayesian filtering used in our watchdog approach is based on some parameters that, in several cases, need to be tuned for any specific scenarios through a

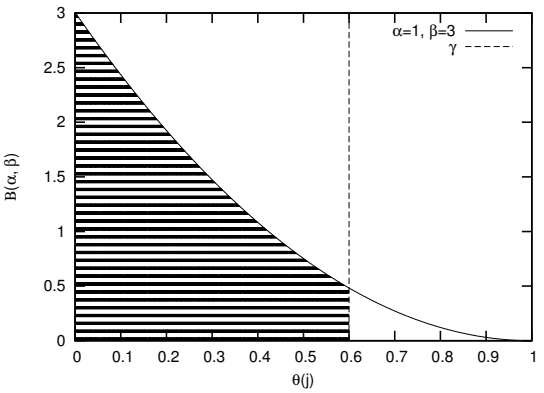

Figure 5.11: Example of a reputation function.

previous training procedure. In particular, the bayesian filtering uses the following parameters:

5.4.6.1 Tolerance Threshold

A higher value for the tolerance threshold (γ) requires more time for the watchdog to detect an attack, as a higher value of alpha is needed in function 5.2 to set the reputation as malicious; to achieve this higher value of alpha, the bayesian filter needs to perform more observations. But, on the other hand, the watchdog is more robust against environmental noise (less false positives). Conversely, if we set a low value for gamma, some nodes affected by noise would be declared as malicious ones and false positives would appear.

5.4.6.2 Fading Mechanism

This parameter (value u) indicates the weight of the old information obtained by the bayesian watchdog. When closer to 1, the old observations weight similarly to the new ones. In case of a change in the behaviour of a given node, since the misbehaviour detected in the latest observation periods is as relevant as the good behaviour observed in the past, the bayesian filters require more time to learn that the node has changed to a bad behaviour. On the other hand, the effects of noise in the filter become less relevant and they are mitigated by the past observations. Therefore, the percentage of false positives is lower. Clearly, for smaller values of u, the opposite behaviour should be observed.

5.4.6.3 Updating Time

This is the period between two subsequent updates of parameters α and β according to the observations harvested about the packets that, mistakenly, are not forwarded. Too frequent updates can cause problems to the bayesian filters if the noise is relatively high. If the filter's parameter is updated frequently and

the packet losses are high, it is likely that the number of packets received in the update period is nearly 0%, thus causing nodes to be wrongly considered as malicious. Conversely, if the observation time is too long, parameters are updated too infrequently and, hence, the time to detect a malicious node may become unacceptable.

5.4.6.4 Neighbour Timeout

This parameter is an implementation issue, and it is also present in the standard watchdog. Since the watchdog only uses local information, it is impossible to know exactly when a node goes out of range. This is solved by using a timeout. When a certain period of time passes by without receiving any packet of a neighbour, the watchdog considers that this neighbour has moved out of range. The main problem is how to find the optimal timeout. A low value forces the watchdog to restart all calculations of a neighbour before a decision about it being malicious or not is made. This creates the possibility of not detecting a malicious node, thus causing false negatives. A high value means that, when a neighbour goes out of range, the watchdog would consider it to be in range for a long time. In that case, the watchdog would expect retransmissions from this neighbour, but would not listen to any. As a consequence, it would decide that this neighbour is a malicious node, thus causing false positives.

5.5 Evaluation

As for the standard watchdog, we have performed several tests using the ns-2 simulator [UBr98] in order to evaluate the approach in large scenarios. In our simulation we considered a network of 50 nodes moving in an area 870x870 m. wide.

We have performed our experiments considering both static and dynamic scenarios. Mobility affects the accuracy of the watchdog due to two important aspects: *(i)* routes used for the traffic flow need to be recalculated each time the topology changes, causing packet losses; and *(ii)* if the attacker is moving, there is a possibility that the malicious node moves outside the watchdog's signal range before it is detected. Both characteristics of these scenarios cause false negatives and false positives to be increased.

5.5.1 Static Scenario

We use random scenarios for validating the implementation of our bayesian watchdog. In the first place we perform several tests to evaluate the behaviour of the watchdog's module in a static scenario. Figure 5.12 shows how the bayesian watchdog detects the 100% of the attacks independently of the number of attackers that there are in the network, therefore achieving 0% of false negatives. The absence of mobility makes the number of false positives negligible, but that is not a realistic setting in MANETs and wireless mesh networks.

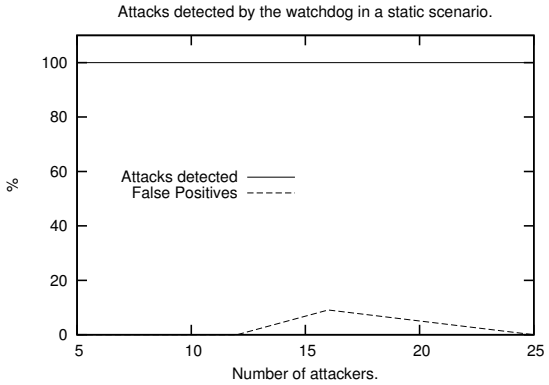

Figure 5.12: Actual detections and false positives in a static scenario.

5.5.2 Dynamic Scenario

The experiments described below are targeted at finding the best tuning of parameters in order to improve the effectiveness. The experiments have been conducted for different motion speeds of the MANET devices, thus verifying how the speed can affect the detection of malicious nodes.

5.5.2.1 Evaluation of the Gamma Value

We perform different tests in scenarios with different mobility speeds and changing the tolerance threshold. Figure 5.13 shows the obtained results. For both diagrams, the x axis represents the various thresholds tested. In Figure 5.13.a and Figure 5.13.b, the y axes measure the percentage of actual attacks detected and false negatives, respectively. For the results analysis, it seems that any threshold between 0.75 and 0.85 decreases the false positives while keeping a good detection rate.

A survey of the ns-2 trace shows that a higher value of gamma (closer to 1) causes the bayesian watchdog to be more strict when detecting an attack, decreasing the false positives but also decreasing the percentage of detection. This is caused by the fact that, as discussed in Section 5, the watchdog needs a higher value of alpha to decide if a neighbouring node is malicious, and therefore, more time is needed to detect the attack.

5.5.2.2 Evaluation of the Fading Value

The next step is to evaluate what is the influence of the fading value upon the accuracy of detection. Figure 5.14 shows the results obtained when varying the fading value of the bayesian watchdog. We use a gamma value of 0.85 for these tests, as it seemed the most suitable according to the results of the previous experiments for the tolerance threshold.

As shown in Figure 5.14, we can see how a high value of fading is more robust

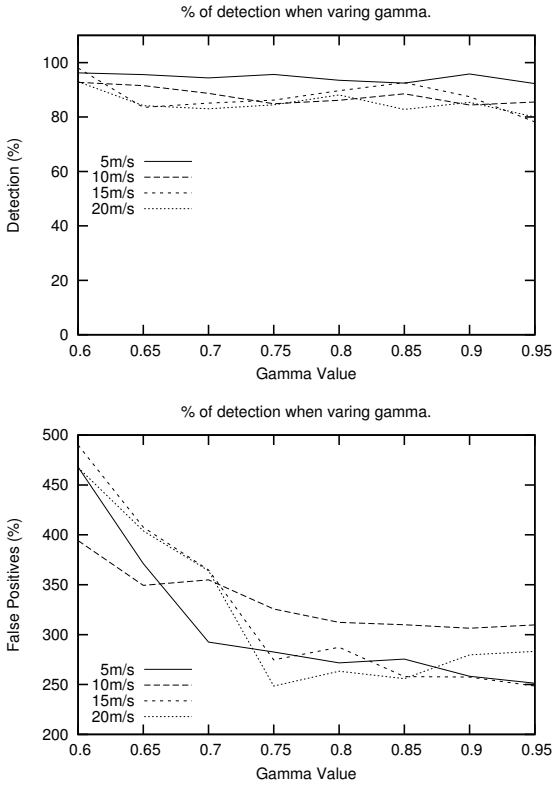

Figure 5.13: Percentage of (a) actual attacks detected and (b) false positives for different tolerance threshold and for different devices' speed.

against false positives. However, when a node starts behaving maliciously, it takes longer to detect the attack. Therefore, such a longer time decreases the accuracy of detecting actual attacks. As a result, the optimal fading value may depend on the needs of the network; e.g., if the routing protocol needs to recalculate new routes frequently due to a high value of the node's speed, a higher value of fading is recommended. However, if a malicious node performs intermittent attacks, a lower value of fading is needed.

5.5.2.3 Evaluation of the Updating Time

As commented in Section 5.4.6.3, a short interval for the updating time would cause a high value of false positives, but would also increase the detection accuracy. Figure 5.15 confirms this statement and also shows us that a value of 1 second is optimal in our scenarios.

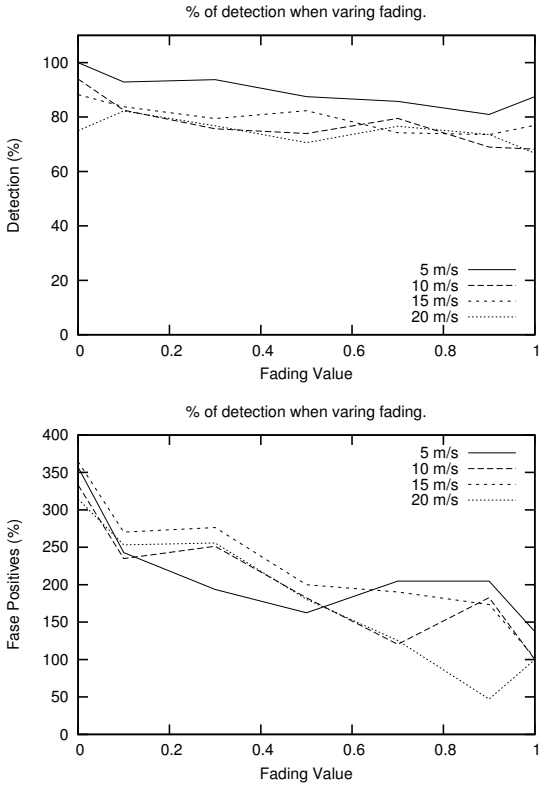

Figure 5.14: Percentage of (a) actual attacks detected and (b) false positives for different fading values and different mobility speeds.

5.5.2.4 Evaluation of the Neighbour Timeout

The higher the neighbour timeout is, the more information about a neighbour is stored in the watchdog for a longer time, and so, the more robust the system is against noise. But this also causes more false positives to appear, as described in Section 5.4.6.4. Figure 5.16 shows how much the increase of the false positives is as we increase the neighbour timeout. In this figure we can see that, with a low neighbour time, the detection is decreased, and with a value greater than 0.2 seconds, the detection is stabilised. With regard to false positives, they are increased when the neighbour timeout value is also increased. Therefore, a value of 0.2 seconds for the neighbour timeout value is the optimal trade-off that we are looking for in these scenarios. Note that 0.2 seconds double the interval between packets in our simulation. Thus, in other scenarios where the traffic changes, this value will be different, and it would be updated accordingly.

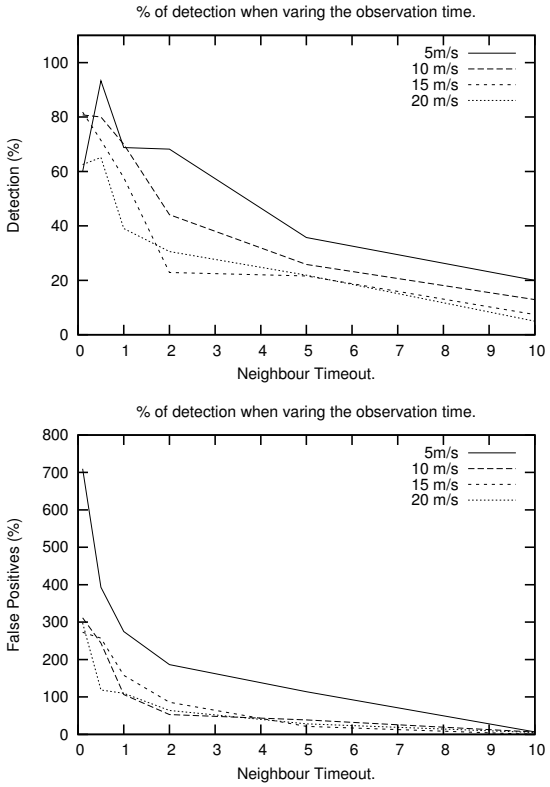

Figure 5.15: Attacks detected (up) and false positives produced (bottom) when varying the neighbour timeout.

5.6 Comparison between the Bayesian Watchdog and the Standard Watchdog

Figure 5.17 shows a comparison between the bayesian watchdog and the standard watchdog in a set of scenarios where we vary the degree of mobility. For these tests we set up the bayesian watchdog with a value of 0.85 for gamma, a value of 1 for the fading, 2 seconds for the observation time and 10 seconds for the neighbours' timeout. These are the optimal values obtained in the evaluation section. In the standard watchdog we set a tolerance threshold of 20%. This value was empirically obtained for these scenarios decreasing the false positives and false negatives with an optimal detection rate. As shown in these figures, the bayesian watchdog is more accurate when detecting attackers in scenarios with high mobility. This is due to the bayesian filters being more robust against noise, and so the number of false positives decreases.

Not only does the bayesian watchdog improve accuracy in detection, but in almost the 95% of the tests, the bayesian watchdog also detects the attacker quicker

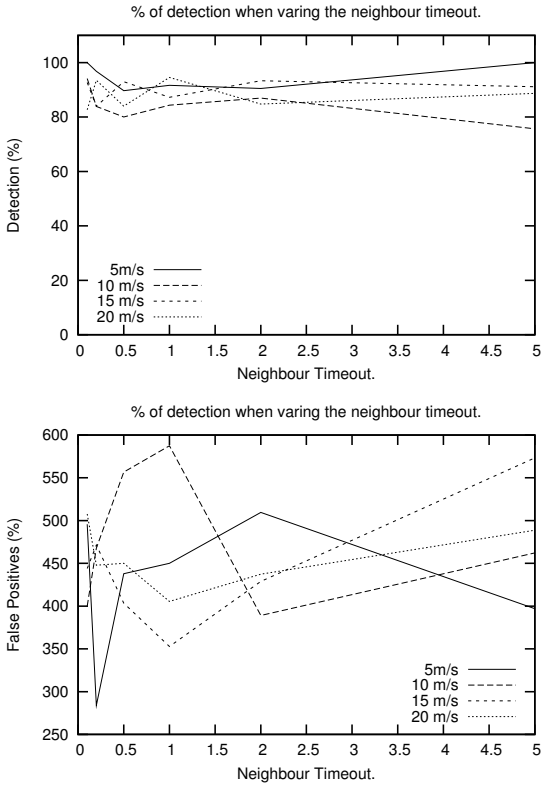

Figure 5.16: Attacks detected (up) and false positives produced (bottom) when varying the neighbour timeout.

than the standard one. This increase of the detection speed is one of the causes for the higher accuracy of detections, since the bayesian watchdog is less affected by the problems introduced by mobility, as explained in Section 5.5.2.

5.7 Summary

In this chapter we have presented a study of one of the most common attacks in ad hoc networks: the black hole attack. A black hole attack is caused when a node does not cooperate to forward the neighbours' packets. In the literature, the technique used to detect this attack is called the watchdog. In this chapter we have demonstrated that this technique is efficient in static scenarios, but it has a high number of false positives and false negatives in scenarios with a high degree of mobility. Therefore, we have proposed a modification of the watchdog which includes bayesian filters. Bayesian filters are broadly used in several scenarios due to their ability to reduce the influence of the noise on the measurements. Here we have proposed a new class of watchdogs that rely on bayesian filters, and we

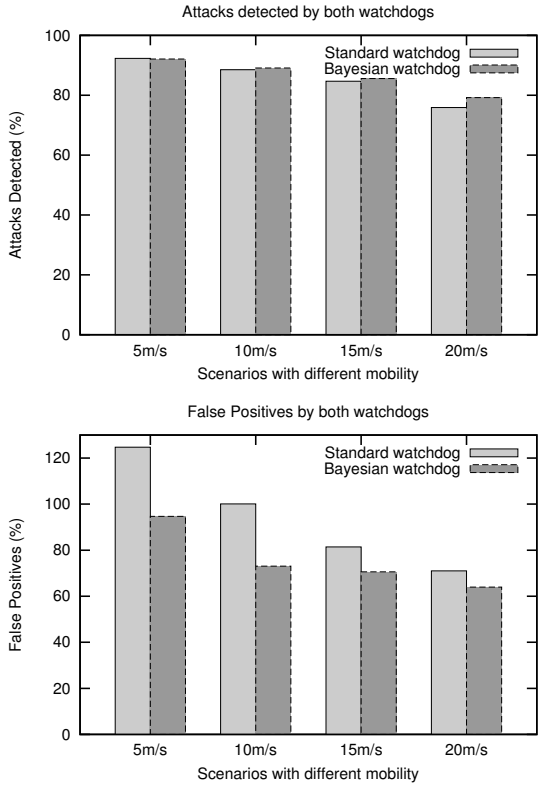

Figure 5.17: Comparison between both watchdogs with different degrees of mobility: detections (up) and false positives (bottom).

called it bayesian watchdog. We have conducted some experiments using our ns-2 based implementation to verify the approach. The integration of bayesian filtering with the watchdog technique has reduced the number of false detections, while the percentage of the detection of the actual attacks has been slightly improved.

Chapter 6

Conclusions, Publications and Future Work

6.1 Conclusions

Throughout this thesis, several contributions have been made to the area of community networks, MANETs and mesh networks. The main purpose of this thesis was to design a secure low-cost architecture to provide Internet connectivity in rural areas of the developing world. We called this architecture *RuralNet*.

We first analysed the architecture of the system, putting into evidence the most important elements of *RuralNet*'s infrastructure. We then described the core elements of the system, which allow us to capture clients, perform a per-user regulation of bandwidth, and offer access to an administrator-defined group of free external web servers. The *RuralNet* application here described is free software, and it can be downloaded for free at `http://ruralnet.sourceforge.net/`

We assessed the viability of our system by providing a set of subscribers with Internet connectivity in our test bed. A performance evaluation was made focusing on the throughput achieved and on the overhead imposed on the devices used. We found that the service offered to clients is the one we expected, while the system can be gradually scaled up as the number of subscribers increases.

For this evaluation we created a platform for testing our proposed architecture before deploying it in a real environment. Hence, we developed a small test bed in our laboratory to do a preliminary evaluation of the feasibility and performance concerning our mesh networking architecture, including testing the capability of the hardware devices used and documenting all the software packages required to tune the system. This test bed is called *Castadiva*.

Castadiva is a novel architecture to improve research in the MANETs field by allowing to make real test bed experiments in a simple and straightforward manner. This tool is optimal to test and evaluate our *RuralNet* prototype.

Castadiva combines the convenience and productivity of Java with the power of the Linux kernel and accompanying tools. The system was designed to simplify the tasks of scenario generation and starting traffic flows among independent, IEEE

802.11-based, wireless nodes.

The architectural design of *Castadiva* differentiates wireless nodes, used for the actual experiments, from the core application, which has management and control purposes. This core application provides an easy interface to define network topologies and traffic flows between nodes. Those definitions are then translated into run-time instructions sent to test bed nodes when experiments are on-going.

An important issue when designing *Castadiva* was that of ns-2 compatibility. We consider it an important goal since we wish to compare results obtained with both platforms. The results obtained in both platforms evidence the correct behaviour of *Castadiva*.

The performance evaluation of *Castadiva* was made focusing on coordination among nodes. We observed that the use of the SSH protocol with the support of an Ethernet allows nodes to synchronise the start of an experiment with high accuracy, being all instructions read at once; afterwards, the test bed relies on individual clocks to synchronise instructions throughout the remaining time of an experiment.

To summarise, the advantages of using *Castadiva* with respect to other MANET test beds are: (1) it is a very low-cost test bed since each node costs about 50$, (2) it is fully compatible with the ns-2 simulator, allowing to compare results between both in a straightforward manner and, (3) does not occupy a lot of physical space.

Castadiva is free software developed under the GNU GPL licence, and can be downloaded at `http://castadiva.sourceforge.net/`

Another aspect studied in this thesis was security of mesh networks and MANETs, and, therefore, it also affects *RuralNet*. The main attack on this networks is the *black hole* or selfish node, where a node does not cooperate with the network to send others nodes' traffic. In the literature we can find the watchdog as the most widely adopted solution. We evaluate the advantages and disadvantages of it. As the main advantage we can say that the watchdog only uses local information and, therefore, it is avoids bad influences from other nodes. In contrast, it has the disadvantage of not being so accurate in networks with high mobility, where false positives and negative detections appear. Most of the false positives and negatives are caused by the erroneous measurements of the packets that nodes should forward, but this is not what occurs. The erroneous measurements are mostly caused by the unreliability of the wireless medium and the mobility of the nodes. Then, we must improve this mechanism if we want to use it in MANETs or mesh networks. Moreover, if we think that the watchdog is a basic brick on several different IDS, we realised the importance that any improvement on it would have.

Therefore we have devised a technique to integrate bayesian filtering techniques inside the watchdogs, and we have conducted some experiments inside an ns-2 implementation to verify the approach. The integration of bayesian filtering inside watchdogs has decreased the number of false positives detected while the percentage of detection of the actual attacks has been kept quite high (or, even slightly improved). This implementation of the watchdog is available under the GNU/GPL licence, and can be downloaded from `http://safewireless.sourceforge.net`.

Finally, we deployed a prototype of *RuralNet* in some schools in Mozambique, where our architecture is now used by more than 60 computers to access Internet,

providing this service to hundreds of students.

The experience acquired made evident that, by combining both wireless and web technologies, we are able to offer a cheap and efficient solution to provide Internet services to rural areas where users are sparsely located.

6.2 Publications Related with this Thesis

The research work related to this thesis has resulted in 16 publications; among them we have two journal articles, and 14 conference papers (4 in national conferences and the others in international conferences, some of them indexed by the Computer Science Conference Ranking or the Computing Research and Education (CORE)). We now proceed by presenting a brief description of each publication. We have organised the different publications based on the chapter were the contents have been discussed.

6.2.1 *Castadiva*

[JJCP07b] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "Castadiva: a test bed architecture for mobile ad hoc networks," in *Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007), Athens, Greece, September 2007*. IEEE Communications Society.

In this paper we present the *Castadiva* tool, presenting its architecture and showing some tests with synthetic traffic. This conference is indexed as CORE B.

[JMJ+07b] J. Hortelano, M. Nácher, J. C. Cano, C. T. Calafate, and P. Manzoni. "Performance evaluation of a mobile ad hoc network test bed architecture," in *XVIII Jornadas de Paralelismo*, pages 253-261, Zaragoza, España, September 2007. CEDI

This paper compares *Castadiva* with ns-2. Several tests compare our test bed with the most spreaded simulator are performed, explaining the little differences between booth tools.

[JMJ+07a] J. Hortelano, M. Nácher, J. C. Cano, C. T. Calafate, and P. Manzoni. "Evaluating the goodness of MANETs performance results obtained with the ns-2 simulator," in *First International Workshop on Network Simulation Tools 2007*, Nantes, France, October 2007. ACM/ICST.

In this paper we add external traffic functionality and present the usefulness of a tool that can use real devices to generate simulations. For example, we use web cams to generate a demonstration of this extra functionality.

[JJCP08] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "Evaluating the Performance of Real Time Videoconferencing in Ad Hoc Networks Through Emulation" in *Principles of Advanced and Distributed Simulation*

(PADs 08), Rome, Italy, June 2008. 22nd ACM/IEEE/SCS Workshop.

In this paper we present the results obtained when performing a videocall in a MANET, and how the numbers of hops inf the network affects this videocall. This conference is indexed as CORE A.

[HCCM09] J. Hortelano, J. C Cano, C. T. Calafate and P. Manzoni. "Testing Applications in MANET Environments through Emulation (2009)" in *EURASIP Journal on Wireless Communications and Networking*, December 2009.

This Journal includes the information presented in the publications showed below and an extensive evaluation of the use of media traffic in different scenarios with different levels of noise. The ISI of the Journal Citation Reports (JCR) of the year 2008, grants this journal with an index impact of 0,976; and, ordering all the publications by this index impact, this journal is listed in the position 29 of a total of 67 in the category of "TELECOMMUNICATIONS".

[WAJ+10] W. Vossen, A. Torres, J. Hortelano, J. C Cano, C. T. Calafate and P. Manzoni. "Extending an emulation platform for automatised and distributed evaluation of QoS in MANETs" in *XXI Jornadas de Paralelismo*, Valencia, España, September 2010. CEDI

This works add extra functionality to *Castadiva* for evaluating the QoS of wireless devices. It also improves the automation of *Castadiva* for performing a sets of tests.

6.2.2 *RuralNet*

[JJCP06b] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "RuralNet: a captive portal based system supporting wireless Internet in rural areas," in *International IFIP Workshop on Wireless Communications and Information Technology in Developing Countries, WCIT 2006*, Santiago de Chile, Chile, August 2006. IFIP.

In this first work we present *RuralNet*, including its architecture, functionality and a preliminary evaluation of its behaviour.

[JJCP06a] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "Providing low-cost wireless connectivity to rural communities," in *XVII Jornadas de Paralelismo*, pages 115-120, Albacete, Spain, September 2006. Universidad de Castilla La Mancha.

In this paper we study the behaviour of *RuralNet* when multiple users access to the network, and how the system shares the bandwidth among them when it is not enough. We make several tests with different users and present the validity of the entire system.

[JJCP07a] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "A wireless mesh network-based system for hotspot deployment and management," in *The Third International Conference on Networking and Services*, Athens, Greece, June 2007. IEEE Computer Society.

In this paper we extend the evaluation to a mesh network. The latency of the entire system and the repercussion of this for the users.

[JJCP07d] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "The RuralMaya project: strengthening Internet support in rural environments through wireless technologies," in *WITFOR 2007 Symposium*, Addis Ababa, Ethiopia, August 2007. IFIP.

This paper discusses the possibility to extend the functionality of *RuralNet* to a developing country. In particular, we focus on the usefulness of this architecture to rural areas of Africa. We also focus on the low cost of the chosen devices and the reduced cost of the total deployment. Additionally, we add to *RuralNet* the application *Maya*, an application to configure the access points of a mesh network. *Maya* was developed by another research colleague.

[JJCP07c] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "RuralMaya: Internet de bajo coste para zonas en desarrollo," in *Simposio Internacional por el XXXV Aniversario de la Institucionalización de los Estudios Superiores en Pinar del Río*, Pinar del Río, Cuba, October 2007. Universidad de Pinar del Río.

This paper about *RuralNet* discusses the possibility of deploying *RuralNet* in a real scenario in a developing country like Cuba. A study of the area, with the advantages and disadvantages of this technology in a specific country like Cuba are analysed.

[HMC+10] J. Hortelano, J. Márquez, J. C Cano, C. T. Calafate and P. Manzoni, "RuralMaya: Internet de bajo coste para países en vías de desarrollo" in Diálogos transdisciplinarios en la sociedad de la información, Era Digital i/2010. May 2010.

This journal includes all the information shown above about *RuralNet* and *Maya*; also new improvements are included into *RuralNet* to be used in a developing country.

6.2.3 Standard Watchdog and Bayesian Watchdog

[HRM09] J. Hortelano, J. Ruiz and P. Manzoni. "The Watchdog: a Black hole Intrusion Tolerance implementation for Ad hoc Networks" in *XX Jornadas de Paralelismo*, a Coruña, Spain, September 2009.

In this work we perform a first analysis of the standard watchdog as a basic

brick for a Intrusion Detection System. We focus on how it can be used to detect an attacker in a MANET, and how to deploy countermeasures against such attack.

[HRM10] J. Hortelano, J. Ruiz, and P. Manzoni. "Evaluating the usefulness of watchdogs for intrusion detection in VANETs". In ICC'10 Workshop on Vehicular Networking & Applications, Cape Town, South Africa, 2010.

This paper performs an analysis of how mobility affects the standard watchdog's accuracy. We use VANETs as the studied scenario. This work reflects that the watchdog must be improved if we want to use it in scenarios with high nodes' mobility.

[JJCP10] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. "Watchdog intrusion detection systems: Are they feasible in MANETs?" in XXI Jornadas de Paralelismo, Valencia, Spain, September 2010. CEDI

This paper performs an analysis of why mobility affects the watchdog accuracy. We perform a study of the false positives and false negatives produced by mobility in several scenarios, and we analyse their cause.

[JJC+10] J. Hortelano, J. C. Cano, C. T. Calafate, M. de Leoni, P. Manzoni, and M. Mecella. "Black hole attacks in P2P mobile networks discovered through bayesian filters". In P2P Collaborative Distributed Virtual Environments (P2P CDVE 2010), Crete, Greece, October 2010. Springer.

In this work we present the bayesian watchdog and perform a comparative with the standard one, showing the improvements obtained using bayesian filters in scenarios with high degrees of mobility.

6.3 Future Work

In the development of this thesis several issues emerged which deserve further scrutiny in the future. The ones we consider more relevant are the following:

- To improve *Castadiva* by adding other technologies like Wi-Max to the test bed.
- To improve *RuralNet* with the experience obtained with the prototype deployed in Mozambique, adding some extra characteristics to convert *RuralNet* in an suitable tool for developing countries.
- To extend the coverage of our prototype deployed in Mozambique to another school, the *Instituto Agrario "Martir Cipriano"*, located 50km far away of our network. For this purpose, we will deploy different nodes to generate a path between our system and this school.

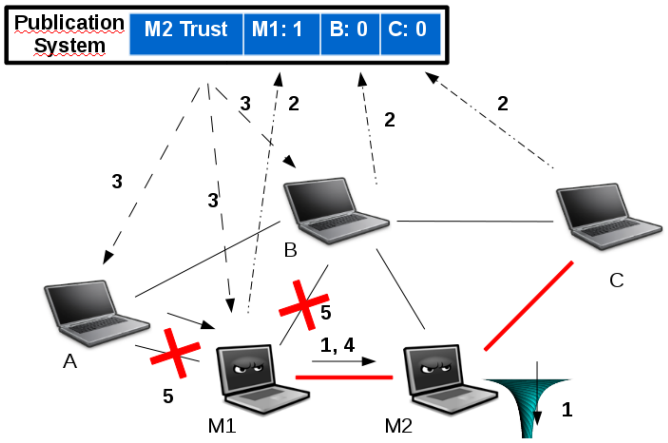

Figure 6.1: Detecting cooperative attacks.

- To decrease the false positives rate and false negatives rate of the bayesian watchdog by sharing information among nodes. We can use a publish/subscribe [DQA04, LO07, LHAS09, LO08, LS08] system for sharing information among nodes in an ad hoc network, and generating with this information a voting system.

- To improve the bayesian watchdog to be robust against cooperative attacks. A cooperative attack is caused when two nodes work together causing a *black hole,* and the other nodes of the network can only reach the signal range of the first one. Hence, the second one can cause an attack dropping the packets obtained by its accomplice. We can use the voting system and the publish/subscribe system to detect a cooperative attack. Figure 6.1 shows how a node can deduce a cooperative attack using this method. (1) First, a node *M2* performs a black hole attack. Node *M1* is an accomplice and continues forwarding packets to *M2*. (2) All neighbours send the vote about *M2* to the subscription system. The majority said that *M2* is a malicious node. (3) This information is shared with all nodes. (4) *M1* ignores the voting system and still sends data to *M2* to perform the collaborative attack. (5) Nodes A and B can see that *M1* still sends packets ignoring the voting system. Therefore A and B consider *M1* as an accomplice and change the vote of *M1,* preventing it from using the network. Then the route used by A for sending to C will be corrected to pass through hop B.

- To study other attacks related to MANETs and mesh networks, and develop countermeasures for these attacks.

Bibliography

[AAO04] Andreas Tonnesen, Andreas Hafslund, and Oivind Kure. The UniK - OLSR plugin library. In *OLSR Interop and Workshop*, Berlin, Germany, June 2004.

[AE05] A. Karygiannis and E. Antonakakis. mLab: a mobile ad hoc network test bed. In *1st International Workshop on Security, Privacy, and Trust in Pervasive and Ubiquitous Computing*, 2005.

[AEC10] AECID. Agencia española de cooperación internacional para el desarrollo (AECID). Further information at: http://www.aecid.es/, 2010.

[AGT+04] A. Lagar, G. Baron, T. W. Hart, L. Litty, and E. Lara. Simplified simulation models for indoor MANET evaluation are not robust. In *Proceedings of the First IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON)*, Santa Clara, California, October 2004.

[ASWZ02] W.A. Arbaugh, N. Shankar, Y.C.J. Wan, and Kan Zhang. Your 80211 wireless network has no clothes. In *Wireless Communications, IEEE*, volume 9, pages 44 – 51, december 2002.

[AT99] George Aggelou and Rahim Tafazolli. RDMAR: A bandwidth-efficient routing protocol for mobile ad hoc networks. In *Proceedings of the WOWMOM*, pages 26–33, 1999.

[Bar64] Paul Baran. On distributed communications. In *The RAND Corporation*, 1964.

[Bas99] S. Basagni. Distributed clustering for ad hoc networks. In *Proceedings of the IEEE International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN)*, pages 310–315, Perth, Western Australia, June 1999.

[Bat08] Borja Roig Batalla. RuralNet 2.0, mejoras a una herramienta para el despliegue de redes en zonas rurales. Master Thesis. In *Facultad de Informática, Universidad Politécnica de Valencia*, 2008.

[BB04] Sonja Buchegger and Jean-Yves Le Boudec. A robust reputation system for p2p and mobile ad-hoc networks. 2004.

[Ber93] James O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer, 1993.

[Bri01] Matthew Bright. Federal strategies encouraging rural broadband access: Intelligent options to minimize the digital divide. In *Washington Internships for Students of Engineering, IEEE*, August 2001.

[C. 97] C. K. Toh. Associativity-Based Routing for Ad-Hoc Mobile Networks. *Wireless Personal Communication*, 4(2):1–36, March 1997.

[CDJ05] C. Gomez, D. Garcia, and J. Paradells. Improving performance of a real ad-hoc network by tuning olsr parameters. 10th IEEE Symposium on Computers and Communications (ISCC05), June 2005.

[CES03] C. Perkins, E. Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. In *Internet Draft*, February 2003.

[CGP03] C. Bettstetter, G. Resta, and P. Santi. The node distribution of the random waypoint mobility model for wireless ad hoc networks. In *IEEE Transactions on Mobile Computing*, volume 2, pages 257 – 269, July-September 2003.

[Chi97] C.-C. Chiang. Routing in clustered multihop, mobile wireless networks with fading channel. In *Proc. IEEE SICON 97*, pages 197–211, April 1997.

[Cla04] Thomas Clausen. Comparative Study of Routing Protocols for Mobile Ad-hoc networks. Research Report RR-5135, INRIA, 2004.

[CLY06] C. Obimbo, L.M. Arboleda C., and Y. Chen. A Watchdog Enhancement to IDS in MANET. In *IASTED conference on Wireless Networks*, July 2006.

[CP94] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM Computer Communication Review*, 24(2):234–244, October 1994.

[CQS98] X. Chen, L. Qi, and D. Sun. Global and superlinear convergence of the smoothing Newton method and its application to general box constrained variational inequalities. *Mathematics of Computation*, 67(222):519–540, 1998.

[DD-10] DD-WRT. Available at: http://www.dd-wrt.com/, 2010.

[DD96] D. B. Johnson and D. A. Maltz. *Dynamic source routing in ad hoc wireless networks.* Kluwer Academic Publishers, 1996. Mobile Computing.

[DDJ01] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: the dynamic source routing protocol for multi-hop wireless ad hoc networks. 2001. Carnegie Mellon University.

[DDY04] David B. Johnson, David A. Maltz, and Yih-Chun Hu. The dynamic source routing protocol. Internet Draft, MANET Working Group, draft-ietf-manet-dsr-10.txt, July 2004. Work in progress.

[Deb08] Moumita Deb. A cooperative blackhole node detection mechanism for adhoc networks. In *World Congress on Engineering and Computer Science*, October 2008.

[DIM+05] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, and K. Ramachandran. Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols. In *Wireless Communications and Networking Conference, 2005 IEEE*, volume 3, pages 1664–1669, 2005.

[dLHMR08] M. de Leoni, S. R. Humayoun, M. Mecella, and R. Russo. A bayesian approach for disconnection management in mobile ad-hoc network. In *Ubiquitous Computing and Communication Journal*, 2008.

[DQA04] Anwitaman Datta, Silvia Quarteroni, and Karl Aberer. Autonomous gossiping: A self-organizing epidemic algorithm for selective. In *In International Conference on Semantics of a Networked*, pages 126–143. World, 2004.

[DR03] D. Powell and R.J. Stroud. Conceptual model and architecture of maftia. Project MAFTIA Deliverable D21, January 2003.

[DRWT96] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang, and Satish K. Tripathi. Signal stability based adaptive routing (ssa) for ad-hoc mobile networks. Technical report, 1996.

[FON10] Official FON website. Available at: http://fon.com, 2010.

[Fre] The freifunk international project for free wireless networks and frequencies. http://start.freifunk.net/.

[G. 98] G. Malkin. RIP Version 2. IETF RFC 2453, November 1998.

[GÍ0] Diego Gómez Gómez. Identificación e implementación de soluciones TICs para el desarrollo de la educación en la provincia de nampula, mozambique. Master Thesis. In *Escuela Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Valencia*, 2010.

[GK00] P. Gupta and P.R. Kumar. The capacity of wireless networks. In *Information Theory, IEEE Transactions on*, volume 46, pages 388–404, Mar 2000.

[GLAS99] J. J. Garcia-Luna-Aceves and Marcelo Spohn. Source-tree routing in wireless networks. In *ICNP*, pages 273–282, 1999.

[GP07] Glenn Judd and Peter Steenkiste. Design and implementation of an rf front end for physical layer wireless network emulation. In *IEEE 2007 IEEE 65th Vehicular Technology Conference (VTC2007)*, April 2007.

[Gur08] Andrei Gurtov. *Host Identity Protocol (HIP): Towards the Secure Mobile Internet*. Wiley Publishing, 2008.

[Hao04] Hao Yang. Security in mobile ad hoc networks: challenges and solutions. In *IEEE Wireless Communications*, volume 11, pages 38–47, February 2004.

[HCCM09] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Testing applications in manet environments through emulation. In *EURASIP Journal on Wireless Communications and Networking*, december 2009.

[HMC+10] J. Hortelano, J. Márquez, J. C Cano, C. T. Calafate, and P. Manzoni. Ruralmaya: Internet de bajo coste para países en vías de desarrollo. In *Diálogos transdisciplinarios en la sociedad de la información, Era Digital i/2010*, pages 130–135, May 2010.

[HP02] W. Li H. Deng and P. Dharma. Routing security in ad hoc networks. *IEEE Communications Magazine, Special Topics on Security in Telecommunication Networks*, 40(10):70–75, October 2002.

[HP04] Yih-Chun Hu and Adrian Perrig. A survey of secure wireless ad hoc routing. In *IEEE Security and Privacy*, volume 2, pages 28–39, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[HRM09] Jorge Hortelano, Juan-Carlos Ruiz, and Pietro Manzoni. The watchdog: a black-hole intrusion tolerance implementation for ad hoc networks. In *Actas de las XX Jornadas de Paralelismo*, A Coruna, Spain, September 2009. Publicaciones de la Universidad de la Coruna.

[HRM10] Jorge Hortelano, Juan-Carlos Ruiz, and Pietro Manzoni. Evaluating the usefulness of watchdogs for intrusion detection in VANETs. In *ICC'10 Workshop on Vehicular Networking & Applications*, Cape Town, South Africa, 2010.

[Hub03] Bert Hubert. *Linux advanced routing & traffic control HOWTO*. http://lartc.org/, 1.43 edition, 10 2003.

[Hud99] Heather E. Hudson. Access to the digital economy: Issues for rural and developing regions. In *Understanding the Digital Economy Conference*, 5 1999. U.S. Department of Commerce, Office of Science and Technology Policy.

[HZH05] Hasswa, A., Zulkernine, M., and Hassanein, H. Routeguard: an intrusion detection and response system for mobile ad hoc networks. In *Wireless And Mobile Computing, Networking And Communications, 2005. (WiMob'2005)*, volume 3, pages 336–343. IEEE, August 2005.

[IC06] I. D. Chakeres and C. E. Perkins. Dynamic MANET on-demand routing protocol. In *IETF Internet Draft*, June 2006.

[ITU10] International telecommunication union. http://itu.org, 2010.

[IXW05] I.F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey, computer networks and ISDN systems. 2005.

[Izu03] Pello Xabier Altadill Izura. *IPtables, manual práctico*. http://www.pello.info/filez/firewall/iptables.html, 1.2 edition, 8 2003.

[JDD+98] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparision of multi-hop wireless ad hoc network routing protocols. In *4th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Dallas, TX, October 1998.

[JDP03] Juan Carlos Cano, Dongkyun Kim, and Pietro Manzoni. CERA: Cluster-based Energy Saving Algorithm to Coordinate Routing in Short-Range Wireless Networks. The International Conference on Information Networking (ICOIN) 2003, Jeju Island, Korea, February 2003.

[JJC+10] J. Hortelano, J. C. Cano, C. T. Calafate, Massimilliano de Leoni, P. Manzoni, and Massimo Mecella. Black-hole attacks in p2p mobile networks discovered through bayesian filters. In Robert Meersman, Tharam Dillon, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems: OTM 2010 Workshops*, volume 6428 of *Lecture Notes in Computer Science*, pages 543–552. Springer, 2010.

[JJCP06a] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Providing low cost wireless connectivity to rural communities. In *XVII Jornadas de Paralelismo*, pages 115–120, Albacete, Spain, September 2006. Universidad de Castilla La Mancha.

[JJCP06b] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. RuralNet: a captive portal based system supporting wireless Internet in rural areas. In *International IFIP Workshop on Wireless Communications and Information Technology in Developing Countries, WCIT 2006*, Santiago de Chile, Chile, August 2006. IFIP.

[JJCP07a] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. A wireless mesh network-based system for hotspot deployment and management. In *The Third International Conference on Networking and Services*, Athens, Greece, June 2007. IEEE Computer Society.

[JJCP07b] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Castadiva: a test-bed architecture for mobile ad hoc networks. In *Symposium on Personal Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, September 2007. IEEE Communications Society.

[JJCP07c] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. RuralMaya: Internet de bajo coste para zonas en desarrollo. In *Simposio Internacional por el XXXV Aniversario de la Institucionalización de los Estudios Superiores en Pinar del Río*, Pinar del Río, Cuba, October 2007. Universidad de Pinar del Río.

[JJCP07d] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. The RuralMaya project: strengthening Internet support in rural environments through wireless technologies. In *WITFOR 2007 Symposium*, Addis Ababa, Ethiopia, August 2007. IFIP.

[JJCP08] J. Hortelano, J. C. Cano, C. T. Calafate, and P. Manzoni. Evaluating the performance of real time videoconferencing in ad hoc networks through emulation. In *22nd ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation*, Rome, Italy, June 2008. ACM/IEEE/SCS.

[JJCP10] J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Watchdog intrusion detection systems: Are they feasible in manets? In *XXI Jornadas de Paralelismo*, Valencia, Spain, September 2010. CEDI.

[JMJ+07a] J. Hortelano, M. Nácher, J. C. Cano, C. T. Calafate, and P. Manzoni. Evaluating the goodness of MANETs performance results obtained with the ns-2 simulator. In *First International Workshop on Network Simulation Tools 2007*, Nantes, France, October 2007. ACM/ICST.

[JMJ+07b] J. Hortelano, M. Nácher, J. C. Cano, C. T. Calafate, and P. Manzoni. Performance evaluation of a mobile ad hoc network test-bed architecture. In *XVIII Jornadas de Paralelismo*, pages 253–261, Zaragoza, Spain, September 2007. CEDI.

[JNA08] David Johnson, Ntsibane Ntlatlapa, and Corinna Aichele. Simple pragmatic approach to mesh routing using batman. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries, CSIR, Pretoria, South Africa*, October 2008.

[JZM+06] J. Zhou, Z. Ji, M. Varshney, Z.Su, and Y. Yang. Whynet: a hybrid testbed for large-scale, heterogeneous and adaptive wireless networks. In *International Conference on Mobile Computing and Networking, Proceedings of the 1st international workshop on Wireless network testbeds, experimental evaluation & characterization*, volume 3, pages 111–112, 2006.

[Ken02] C. Kenny. Information and communication technologies for direct poverty alleviation: Costs and benefits. In *Development Policy Review*, volume 20, pages 141–157, 2002.

[Kes88] S. Keshav. REAL: a network simulator. University of California, Berkeley, December 1988.

[KH10] Alexey Kuznetsov and Stephen Hemminger. Iproute2: a collection of utilities for controlling tcp / ip networking and traffic control in linux. http://linux-net.osdl.org/index.php/Iproute2, 2010.

[LBDC+01] Jinyang Li, Charles Blake, Douglas S.J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 61–69, New York, NY, USA, 2001. ACM.

[LHAS09] Lukasz Juszczyk, Harald Psaier, Atif Manzoor, and Schahram Dustdar. Adaptive query routing on distributed context - the cosine framework. In *In International Workshop on the Role of Services, Ontologies, and Context in Mobile Environments (ROSOC-M)*. 10th International Conference on Mobile Data Management (MDM 09). IEEE, may 2009.

[Li04] Yang Li. *Models and applications of wireless networks in rural environments*. PhD thesis, University of the Western Cape, November 2004.

[Lin10] Linksys. Linksys WRT54G. Further information at: http://home.cisco.com/, 2010.

[LO07] Doug Lundquist and Aris Ouksel. An efficient demand-driven and density-controlled publish/subscribe protocol for mobile environments. In *DEBS '07: Proceedings of the 2007 inaugural international conference on Distributed event-based systems*, pages 26–37, New York, NY, USA, 2007. ACM.

[LO08] Doug Lundquist and Aris Ouksel. Dynamic subscription permission: Extending the depth of demand-controlled flooding. In *Asia-Pacific Conference on Services Computing. 2006 IEEE*, volume 0, pages 211–216, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[LS08] Lukasz Juszczyk and Schahram Dustdar. A middleware for service-oriented communication in mobile disaster response environments. In *In 6th International Workshop on Middleware for Pervasive and Ad-Hoc Computing (MPAC)*. 9th Middleware Conference. ACM/IFIP/USENIX, december 2008.

[LYHO04] Jing-Hong Liew, Alvin W. Yeo, Khairuddin Ab Hamid, and Al-Khalid Othman. Implementation of wireless networks in rural areas. In *Damai Sciences Sdn Bhd*, 2004. Universiti Malaysia Sarawak.

[MAJ+01] M. Kojo, A. Gurtov, J. Manner, P. Sarolahti, T. Alanko, and K. Raatikainen. Seawind: a wireless network emulator. In *P.O.Box 26, FIN-00014 University of Helsinki*, Finland, September 2001.

[Mar09] Víctor M. García Martínez. Diseño, simulación e implementación de una red de área local con acceso a internet para cuatro escuelas en el distrito de nacala, mozambique. Master Thesis. In *Escuela Superior de Ingenieros de Telecomunicación, Universidad Politécnica de Valencia*, 2009.

[Mer] Enterprise wireless lan networks: Indoor and outdoor wireless networks. Available at: http://www.meraki.com.

[MGLA96] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[MGLB00] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 255–265, New York, NY, USA, 2000. ACM.

[Mis10] Missionarios vicentinos. Further information at: http://www.nacalavicenciana.org/, 2010.

[MM07] M. Wood and M. Erlinger. Optimized link state routing protocol (olsr). Request for Comments 4766, Network Working Group, http://www.ietf.org/rfc/rfc4765.txt, March 2007.

[Moy98] John T. Moy. *OSPF Anatomy of an Internet Routing Protocol.* Addison-Wesley Professional, 1998.

[MS04] Mohammad Al-Shurman and Seong-Moo Yoo. Black hole attack in mobile ad hoc networks. In *ACMSEÓ4, Huntsville, USA*, April 2004.

[MVPJ+07] Andres Martinez, Valentin Villarroel, Jaume Puig-Junoy, Joaquin Seoane, and Francisco del Pozo. An economic analysis of the EHAS telemedicine system in Alto Amazonas. In *J Telemed Telecare*, volume 13, pages 7–14, 2007.

[MZ04] Richard S. Wolff Mingliu Zang. Crossing the digital divide: Cost-effective broadband wireless access for rural and remote area. In *IEEE Communications Magazine*, Febr 2004. Montana State University.

[Nat01] United Nations. Making new technologies work for human development. In *Human Development Report*, 2001.

[NoC] Nocat. http://nocat.net.

[Ole05] Ron Olexa. *Implementing 802.11, 802.16 and 802.20 wireless network.* Elsevier, 2005.

[Ope10] OpenWRT, wireless freedom. Available at: http://openwrt.org, 2010.

[OPN10] OPNET Technologies Inc. OPNET making networks and applications performs. Available at: http://www.opnet.com/, 2010.

[oU10] University of Uppsala. AODV-UU. Available at: http://core.it.uu.se/core, 2010.

[PAPL02] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance of multipoint relaying in ad hoc mobile routing protocols. *Networking 2002, Pise (Italy)*, 2002.

[Pat] PatronSoft. Firstspot. http://www.patronsoft.com/firstspot/.

[PC03] Paul Brutch and Calvin Ko. Challenges in Intrusion Detection for Wireless Ad-hoc Networks. In *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium*, pages 368– 373, Jannuary 2003.

[PGC00] Guangyu Pei, Mario Gerla, and Tsu-Wei Chen. Fisheye state routing: A routing scheme for ad hoc wireless networks. In *ICC (1)*, pages 70–74, 2000.

[PGH00] G. Pei, M. Gerla, and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility, 2000.

[PPJ+05] Anand Patwardhan, Jim Parker, Anupam Joshi, Michaela Iorga, and Tom Karygiannis. Secure routing and intrusion detection in ad hoc networks. In *Proceedings of the 3rd International Conference on Pervasive Computing and Communications*, Kauai Island, Hawaii, March 2005. IEEE. Main Conference.

[PR99] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100, February 1999.

[PTN+99] P. Johansson, T. Larsson, N. Hedman, B. Mileczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad hoc networks. In *5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, August 1999.

[Pum07] Andrea Lo Pumo. The netsukuku network topology. In *CoRR*, volume abs/0705.0819, 2007.

[R+08] J.-C. Ruiz et al. Towards Measuring the Security of Ad-hoc Routing Protocols. In *AMBER Workshop*, Italy, 2008.

[RFM04] R. Ogier, F. Templin, and M. Lewis. Topology dissemination based on reverse-path forwarding (TBRPF). Request for Comments 3684, MANET Working Group, http://www.ietf.org/rfc/rfc3684.txt, February 2004. Work in progress.

[Rog01] Everett M. Rogers. The digital divide. In *Convergence: The International Journal of Research into New Media Technologies*, pages 96–111. SAGE Journals, December 2001.

[RP09] Eduard Duran Rosich and Roc Meseguer Pallarès. Wifi mesh network nodes on guifi.net. Universitat Politècnica de Catalunya, September 2009.

[RZ04] R. Barr and Z. J. Haas. JiST/SWANS. Available at: http://www.cs.cornell.edu/barr/repository/jist/, 2004.

[RZR04] R. Barr, Z. J. Haas, and R. Van Renesse. JiST: embedding simulation time into a virtual machine. In *Proceedings of EuroSim 2004*, September 2004.

[RZR05] R. Barr, Z. J. Haas, and R. Van Renesse. Scalable wireless ad hoc network simulation. In *Handbook on Theoretical and Algorithmic Aspects of Sensor*, pages 291–311, 2005.

[Sad98] George Sadowsky. The internet society and developing countries. In *International Electronic Publication of the Internet Society*, November 1998.

[San10] Damien Sandras. Ekiga (GnomeMeeting. Available at: http://ekiga.org/, 2010.

[SCE00] S. R. Das, C. E. Perkins, and E. E. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *19th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, Tel Aviv, Israel, March 2000.

[SCM99] S.-J. Lee, C.-K. Toh, and M. Gerla. Performance Evaluation of Table-Driven and On-Demand Ad Hoc Routing Protocols. In *Proceedings of IEEE PIMRC'99, Osaka, Japan*, pages 297–301, September 1999.

[SDHH98] M Sahami, S Dumais, D Heckerman, and E Horvitz. A bayesian approach to filtering junk e-mail. In *In AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[Sec08] BalaBit IT Security. Syslog-ng. Further information at: http://www.balabit.com/, 2008.

[SMSI04] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile ad hoc networking*. IEEE Press, 2004.

[STKM00] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. 6th MobiCom, Boston, Massachusetts, August 2000.

[SVV89] S. McCanne, V. Leres, and V. Jacobson. The tcpdump manual page. Lawrence Berkeley Laboratory, 6 89.

[TC06] T. Ylonen and C. Lonvick. The secure shell (ssh) authentication protocol. Request for Comments 4252, MANET Working Group, http://www.ietf.org/rfc/rfc4252.txt, January 2006. Standards Track.

[TGLG01] T.H. Clausen, G. Hansen, L. Christensen, and G. Behrmann. The optimized link state routing protocol, evaluation through experiments and simulation. *IEEE Symposium on "Wireless Personal Mobile Communications"*, September 2001.

[The99] The Institute of Electrical and Electronics Engineers, Inc. Ieee/iec std 802.11, wireless LAN medium access control (MAC) and physical layer (PHY) specifications, August 1999.

[Tob80] F. Tobagi. Multiaccess protocols in packet communication systems. In *Communications, IEEE Transactions on*, volume 28, pages 468 – 488, April 1980.

[TP03] T. Clausen and P. Jacquet. Optimized link state routing protocol (olsr). Request for Comments 3626, MANET Working Group, http://www.ietf.org/rfc/rfc3626.txt, October 2003. Work in progress.

[TPA+01] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. *International Multi Topic Conference, Pakistan*, 2001.

[TS08] Latha Tamilselvan and Dr. V Sankaranarayanan. Prevention of cooperative black hole attack in manet. In *Journal Of Networks (JNW)*, volume 3, pages 13–20, may 2008.

[UBr98] USC/ISI UC Berkeley, LBL and Xerox PARC researchers. Network Simulator - ns (Version 2). Available at: http://www.isi.edu/nsnam/ns/, 1998.

[Vie08] Stephanie Vie. Digital Divide 2.0: 'Generation M' and Online Social Networking Sites in the Composition Classroom. In *Computers and Composition*, volume 25, pages 9 – 23, 2008. Media Convergence.

[VS00] V. Park and S. Corson. Temporally-ordered routing algorithm (TORA) version 1 - functional specification. Internet Draft, MANET Working Group, draft-ietf-manet-tora-spec-03.txt, November 2000. Work in progress.

[WAJ+10] Wannes Vossen, Alvaro Torres, J. Hortelano, J. C Cano, C. T. Calafate, and P. Manzoni. Extending an emulation platform for automatized and distributed evaluation of qos in manets. In *XXI Jornadas de Paralelismo*, Valencia, september 2010. CEDI.

[Wal05] Scott Wallsten. Regulation and internet use in developing countries. In *Economic Development and Cultural Change*, volume 53, pages 501–523, 2005.

[WF07] H. Weerasinghe and Huirong Fu. Preventing cooperative black hole attacks in mobile ad hoc networks: Simulation implementation and evaluation. In *Future Generation Communication and Networking (FGCN 2007)*, volume 2, pages 362–367, Dec. 2007.

[Wif] Wifidog. http://dev.wifidog.org/.

[XRM98] X. Zeng, R. Bagrodia, and M. Gerla. GloMoXim: a library for parallel simulation of large-scale wireless networks. In *In Proceedings of the 12th Workshop on Parallel and Distributed Simulations (PADS '98)*, May 1998.

[YN98] Y.B.Ko and N.H.Vaidya. Location aided routing (lar) in mobile ad-hoc networks. In *The Annual International Conference on Mobile Computing and Networking (MOBICOM)*, Dallas, Texas, USA, October 1998.

[YSD06] Yang Cheng Huang, Saleem Bhatti, and Daryl Parker. Tuning olsr. The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC06), September 2006.

[ZA02] Eustathia Ziouva and Theodore Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. In *Computer Communications*, volume 25, pages 313 – 321, 2002.

[ZL02] Yongguang Zhang and Wei Li. An integrated environment for testing mobile ad-hoc networks. In *MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*, pages 104–111, New York, NY, USA, 2002. ACM.

[ZM99] Z. Haas and M. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. Internet Draft, MANET Working Group, draft-ietf-manet-zone-zrp-02.txt, June 1999. Work in progress.

[ZRCK03] Wensheng Zhang, R. Rao, Guohong Cao, and George Kesidis. Secure routing in ad hoc networks and a related intrusion detection problem. In *In IEEE Military Communications Conference*, pages 735–740. IEEE, 2003.