TECHNICAL UNIVERSITY OF VALENCIA

DEPARTMENT OF COMPUTER ENGINEERING

PROVIDING RURAL AREAS CONNECTIVITY USING WIRELESS NETWORKS TECHNOLOGY

Jorge Hortelano Otero

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

December 2007

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 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 for rescue operations at disaster sites 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, login on remote machines, and do this from anywhere on land, sea, or air. Furthermore, wireless networks are of great value to fleets of trucks, taxis, busses, and vehicles in general. Eventually, all wireless and fixed-line networks will merge with the Internet.

There are three basic methods 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 [1]. Wireless networks based on the radio interface defined by the IEEE 802.11 [2] standard which is the radio technology this thesis is based on.

The chapter is structured as follows. Section 1.1 explains the IEEE 802.11 standard. Section 1.2 describes the connection problem observed into rural areas, and Section 1.3 explains the objectives of this master thesis.

1.1 Overview of the IEEE 802.11 Standard

The IEEE 802.11 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 centralized access mode optionally used when a point coordinator (PC) is available. When relying on the PCF, contention-free period (CFP) and contention period (CP) alternate over time. The Distributed Coordination Function uses a listen-before-talk scheme named carrier sense multiple access (CSMA) with collision avoidance (CA). 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.

1.1.1 Network Architecture

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

- An IBSS (Independent Basic Server Set), also known as ad hoc network, is a network established from 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) networks are characterized by the existence of multiple access points whose coverage area partially overlaps.

1.1.2 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 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 IEEE 802.11 for operating on the 5 GHz radio frequency. It supports several different data rates 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 with Complementary Code Key (CCK) with 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 the most recent specification available for 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 802.11 standard for wireless local-area network. The real data rate throughput is estimated that reach a theoretical 540 Mbit/s. IEEE 802.11n builds upon the previous 802.11 standard by adding MIMO (multiple-input multiple-output) and orthogonal frequency-division multiplexing (OFDM).

1.1.3 Summary

The main purpose of wireless networks is support computational and communication services while moving. The advantages of wireless networks are ease and low cost of installation. In environments when deploying a wired network is difficult, i.e. at home, in remote areas, or where the cost is high compared to benefits, wireless networks are a validate alternative. Therefore, this technology is an alternative to deploy rural networks.

1.2 Rural Networks

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 [3]. 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 emphasized by the study of Bright [4], who shows the Digital Divide between urban areas and rural areas. Others works [5, 6, 7] also focus on this problem, emphasizing on how serious and difficult to handle it is.

The solution to this problem might to have these characteristics: (a) implementation should be possible at a low cost in areas where population density is low, (b) the system can be easily installed, even in remote and inaccessible locations, (c) system operation and maintenance may 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) long life cycles.

Recently, the development of wireless local area network technologies has made it possible to consider non-traditional approaches to deploy an infrastructure in rural and developing areas. Such solutions were not possible some years ago since the deployment cost was too high. 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 [8, 6, 9].

1.3 Objectives

The objective of this master thesis is to design an architecture to deploy a mesh network to connect distant areas and create a test-bed to develop and validate the routing protocols used in the architecture proposed.

The main contributions of this thesis are summarized below:

Generate a tool to provide access control to a wireless network. We develop a tool to provide access control using an alternative to the standard wireless network encryption like WEP or WPA. The system should allow any public user to register, controlling each user's privileges. We create a new user control system based on a specially-designed web portal. These special type of portals are known as captive portals. A captive portal is a web-based solution that allows a client to subscribe with the system and, in this way, connect to the different services the system offers.

Design and test an architecture to connect rural areas with a low cost. We design an architecture based on wireless networks which extends the capabilities of hotspots to provide wireless connectivity at distant areas and at a low cost. The system combines the promising paradigm of Wireless Mesh Networks (WMNs) [10] with the captive portal technology, and it is based on the use of commercial off-the-shelf wireless devices.

Generate a test-bed to validate the proposed architecture. All routing protocols used in the mesh network need to be tested. We develop a test-bed where we can validate these protocols for mesh networking in real devices. We test the behaviour of these routing protocols with different kinds of traffic (such as UDP or TCP), and in different scenarios (like a video-conference).

1.4 Structure of the Thesis

The rest of this master thesis is organized in 4 chapters with the structure presented next:

Chapter 2 overviews the related work concerning MANETs and mesh networks, explaining the behaviour of the different routing protocols involved in such kind of networks.

Chapter 3 describes the application developed to provide access control to the network, called TocToc, and the proposed architecture to deploy a mesh network in rural areas, called RuralNet.

Chapter 4 presents and evaluates our proposed test-bed designed to validate the architecture and the routing protocols of the mesh network used. This test-bed is called Castadiva.

Finally, Chapter 5 draws the main conclusions inferred from this master thesis, as well as future research lines related to the present work.

Chapter 2

Related Work

As mentioned in the introduction, several rural areas exist where it is impossible or is too expensive to deploy a centralized communications infrastructure. The absence of a centralized infrastructure provides reliability in Mobile ad hoc networks (MANET [11]), eases their deployment and guarantees the interconnection of wireless devices in hostile environments. However, these advantages come at a cost as the routing protocol procedures must be incorporated into all the mobile nodes. 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 how the breaks of the established routes are detected. Developing and testing such routing protocols is mandatory for their success in any mesh network. To do this test, researchers have two options: simulator and emulators.

The rest of this chapter is structured as follows. Section 2.1 defines MANETs and mesh networks. Section 2.2 describes the different routing protocols to deploy these networks. Section 2.3 explain the routing protocol chosen by us to generate a mesh network and Section 2.5 describes the differences between simulators and emulators, as well as the advantages of using the latter to design MANETs and mesh networks.

2.1 MANETs and Wireless Mesh Networks

MANETs became a popular subject for research as laptops and 802.11/Wi-Fi wireless networking became widespread.

MANETs consist of devices that are autonomously self-organizing in networks. In ad hoc networks the devices themselves conform the network, and this allows seamless communication, at a low cost, in a self-organized 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.

A mesh network [10] is a way to route data, voice and instructions between nodes. Mesh networks differ from other networks in that the component parts can all connect to each other via multiple hops, and they generally are not mobile. It employs one of two 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 connected only to other nodes with which they exchange most of the data. Mesh networks are usually used to extend the coverage of access points, and can be seen as a type of ad hoc network. 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 the nodes.

Both MANETs and mesh networks require efficient routing protocols to generate correctly. 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 [12, 13, 14, 15, 16, 17, 18, 19] 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. Section 2.2 provides a short introduction to routing protocols, explaining the different strategies used to generate a MANET and a mesh network.

The importance of MANET and mesh 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 on it.

2.2 Routing Protocols

A routing protocol is required when a packet must go through several hops to reach its destination. It is responsible for finding a route for the packet and making sure it is forwarded through the appropriate path.

2.2.1 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 the basic principles of each of these techniques.

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.

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.

Source Routing Technique where all the data packets have the routing information on 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.

2.2.2 Classification of Routing Protocols

Routing protocols based on algorithms such as distance vector (e.g. RIP [20]) or link-state (e.g. OSPF [21]) 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:

- Centralized 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 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.

2.2.3 Routing in Ad hoc Networks

An ideal routing protocol for ad hoc networks must have certain properties that make it different from the rest. It must be distributed to increase reliability: when all the nodes are mobile, it makes no sense to have a centralized routing protocol. Each node must have enough capabilities to take routing-related decisions with the aid of the rest of the nodes.

Also, a routing protocol should assume that the links detected are unidirectional connections. On a wireless channel an unidirectional connection may be formed due to physical factors, so that bidirectional communication may result impossible. It is also important that an ad hoc routing protocol takes into account issues such as power consumption and security. Obviously, mobile nodes depend on batteries. This means that a protocol that minimizes the total power consumption of network nodes would be ideal. Concerning security, you must 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 these networks they should be, according to the classification of section 2.2.2, 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-cataloged proposals.

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) [22] and the Wireless Routing Protocol (WRP) [23]. Examples of link-state based protocols are the Open Shortest Path First (OSPF) [21], the Optimized Link State Routing (OLSR) [13], the Topology Broadcast Reverse Path Forwarding (TBRPF) [24], the Source Tree Adaptive Routing (STAR) [25], the Global State Routing (GSR) [26], the Fisheye State Routing (FSR) [27] and the Landmark Routing Protocol (LANMAR) [28].

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) [29], the Dynamic Source Routing (DSR) [30], the Associativity Based Routing (ABR) [31], the Signal Stability based Adaptive routing (SSA) [32], the Temporally Ordered Routing Algorithm (TORA) [33], the Relative Distance Micro-discovery Ad-hoc Routing (RDMAR) [34] and the Dynamic On-demand MANET routing protocol (DYMO) [14].

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) [35], there are some protocols based on clustering and hierarchical architectures, such as the Clusterhead Gateway Switch Routing (CGSR) [36], the Distributed Mobility-Adaptive Clustering (DMAC) [37] and the Cluster-based Energy Saving Algorithm (CERA) [38]. The LAR protocol [39] 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 protocol since we decided to use it in our Castadiva architecture (see Chapter 4).

2.3 The Optimized Link-State Routing Protocol (OLSR)

The Optimized Link State Routing protocol [40] 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 to reduce 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.

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 minimizing 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 optimizations 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.

Multipoint Relays The Multipoint Relay concept consists in trying to minimize 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 2.1 illustrates this concept.

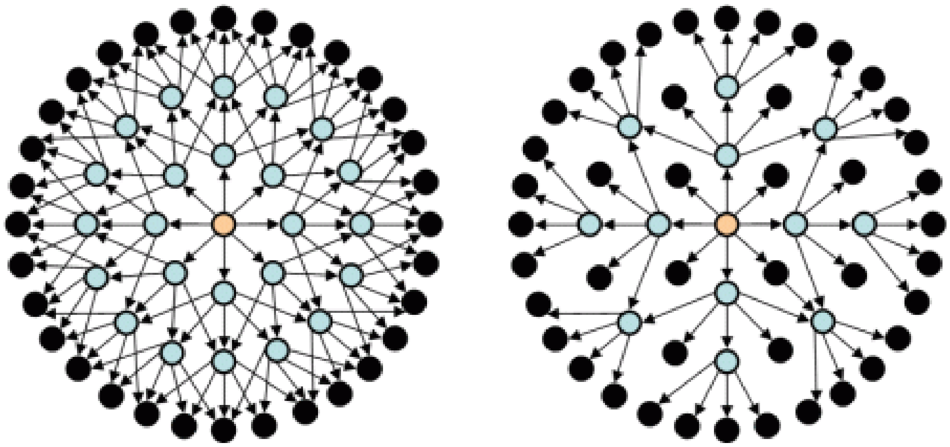


Figure 2.1: Illustration of the multipoint relay concept for node N

OLSR trusts on the MPR node selection to calculate routes towards all the destinations having these as intermediate stations. This solution requires each node to periodically broadcast the list of neighbor nodes chosen as its MPRs. When receiving this information, each neighbor node updates the routes towards all known stations.

Neighbor 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.

Multipoint Relay Selection Each network node independently chooses its MPR set. To maintain a list of the two-hop neighbours requires analyzing 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).

MPR Information Broadcasting Each node must broadcast topology control messages (TC) in order to 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 always has 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.

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 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.

2.4 Captive Portals

Captive portals are gaining increasing use on freely accessible wireless networks like mesh networks, instead of RADIUS servers or other authentication techniques. Captive portals are often employed at Wi-Fi hotspots, and they can be used to control wired access (e.g. apartment houses, hotel rooms, business centers) as well.

The captive portal technique forces an HTTP client on a network to see 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 use policy and require the user to agree.

In section 3.1 we explain TocToc, our captive portal proposal. TocToc 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.

TocToc is not the only captive portal solution available. Other applications like NoCat [41], WifiDog [42] or FirstSpot [43] are available implementations of captive portals.

We decided not to use existing tools like the ones cited above and generate a new one for the following reasons.

FirstSpot: Is a commercial software that costs over 500€ and where developers can not change the source code to fit it to their architecture.

NoCat: Is a powerful captive portal that is free software. NoCat only allow three kinds of traffic. This means that a client can only choose one of these three different connection options to join NoCat, and we need an application completely flexible to control all client traffic.

WifiDog: Was not yet finished when we began to develop TocToc. Therefore, it is not a valid option to integrate with our applications either.

All of these characteristics made us decide to develop our own tool.

2.5 Methodology Used to Evaluate MANET Proposals

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) test-beds.

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 [44], OPNET [45], Seawind [46], GloMoSim [47] and REAL [48], JiST/SWANS [49, 50, 51]. Computer simulation is the most popular way to evaluate MANET routing protocols [52, 53, 54] being ns-2 one of the most extended under the research comunity. Ns-2 is a discrete event network simulator. It is popular in academia for its extensibility (due to its open source model) and plentiful online documentation. NS 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.

Simulation offers four important advantages [55]: 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. Simulations also have some disadvantages: First, researchers can not test there their own real world implementation of a protocol in a realistic scenario. Second, simulators also need to incorporate realistic models of node mobility and radio propagation.

A test-bed is a platform for experimentation. Test-beds allow for 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, like mLab [56] or test emulators like MobiEmu [57]. Test-beds also have some disadvantages: the test-bed called mLab can only generate networks topologies and capture packets, while MobiEmu uses expensive clusters to do their tests.

In Chapter 4 we present our test-bed proposal called Castadiva. Castadiva is a test-bed to emulate MANETs where we can test the behaviour of the network with different routing protocols.

Chapter 3

An Architecture supporting Web-based Services and Authentication

Recent advances in communication technologies, as well as the proliferation of computing devices, are shaping our environments towards an ubiquitous Internet. Each user can access the Internet when going to work using a public transport, in their home or at different public places.

Access control is now a priority to restrict the desired 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 or laptops can connect to Internet. Each year new devices appears that incorporate the technology to connect to a wireless network, likes PDAs or mobile phones. Therefore, our solution must be compatible with all devices on the market.

In this chapter we are going to explain our solution developed to deploy a rural network and the application generated to control users's access.

The chapter is structured as follows. Section 3.1 explains TocToc, a web-based system to authenticate users. Section 3.2 describes an architecture called RuralNet to offer low-bandwidth Internet access to isolated rural areas using the authentication system of Section 3.1. Finally, Section 3.3 presents Dulendué, a proposal of a service searcher based on the physical position of the user.s

3.1 TocToc

TocToc is a architecture based on the captive portal technology that provides access control, control the connexion speed for each user, and give free access to a certain group of servers to everyone if the administrator wants to allow it. A captive portal is a system to control the access into a network using web technology. Today exist several captive portals were developed, like NoCat, WifiDog and Firstpot, but we will not use them since they are not being finished, being commercial software or not having 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 3.1). The main server controls client access depending on whether he is a registered user 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 himself with the captive portal. After a login process the user can use any of the freely available services or purchase others, like Internet access. Concerning the Internet access service, TocToc allows each client to choose among

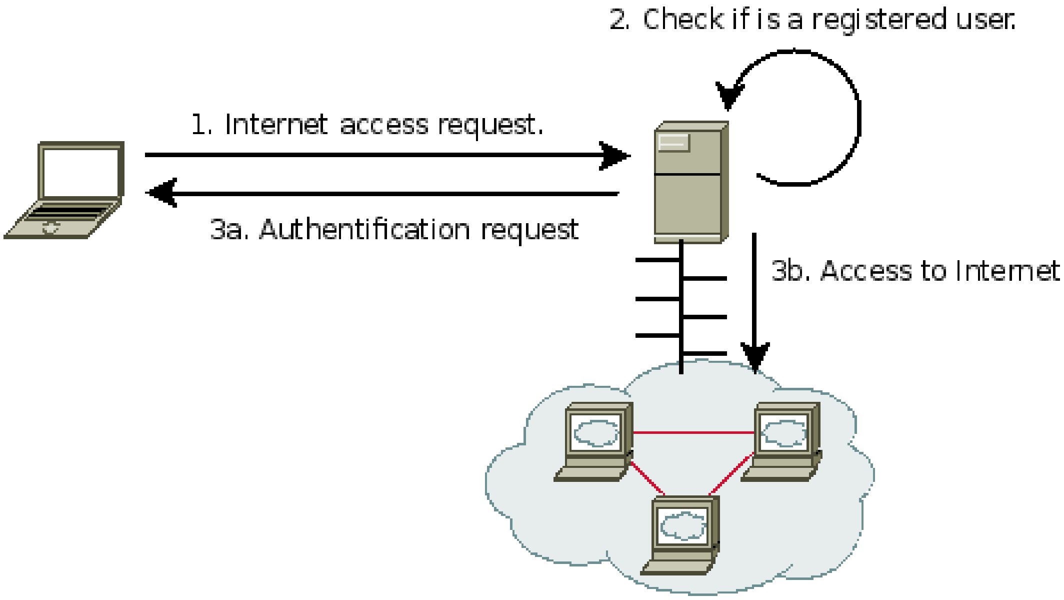

2. Check if is a registered user.
1. Internet access request.
3a. Authentification request
3b. Access to Internet

Figure 3.1: Typical captive portal connection scheme.

multiple connection speeds. Just as an example, Figure 3.2 shows the welcome window for a client that accesses the TocToc system for the first time.

3.1.1 Objectives of the TocToc Proposal

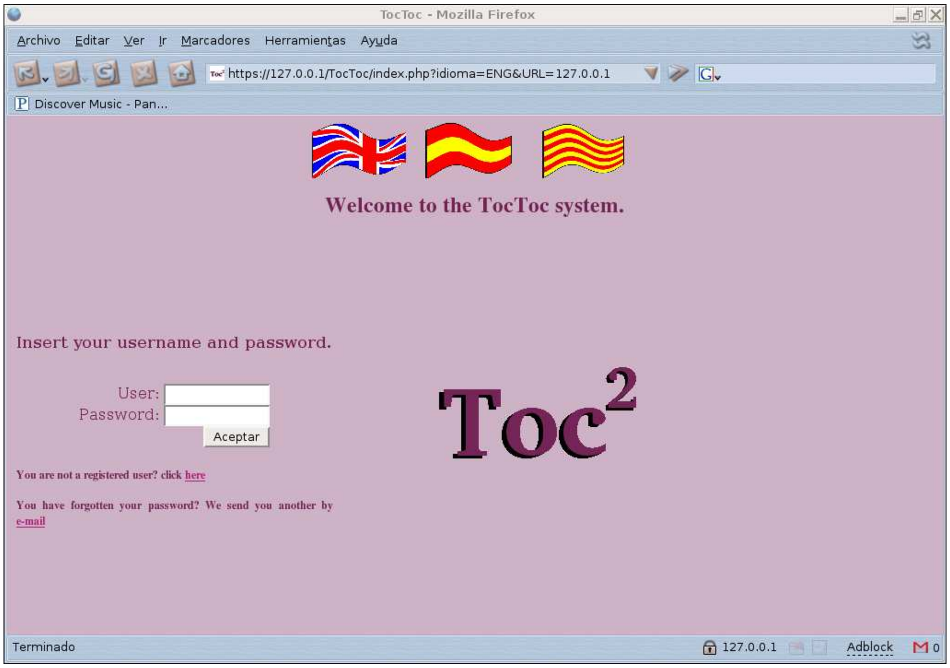
The objective of TocToc is to generate a flexible architecture to easily control the access of each client to a network easily. TocToc simplifies wireless network deployment and lowers costs. It must also be compatible with all devices such as laptops, PDAs and mobile phones that can connect to a network using a Wi-Fi card.

3.1.2 The TocToc System Architecture.

The system is designed to allow any user to connect to a wireless network. Figure 3.3 shows the overall system architecture of TocToc.

Our architecture is organized into three levels: the main server, the wireless network and the users:

- **Main server.** Is the server where the TocToc applications, the databases and a web server are running. All client authentication and interaction will be done by these applications. This server includes two different networks interfaces: to connect to the wireless network and a high-speed connection to Internet.
- **Wireless network.** A free wireless network that allows users to connect to the system. This wireless network is based on the IEEE 802.11 technology, being composed by one or more wireless nodes. It can not use any encryption, such as WEP or WPA, to allow all clients to freely connect to the system.
- **Users.** The users can connect with different devices, such as PDAs, mobile phones or laptops. The only requirement for these devices is to have a wireless card to connect to the wireless network and a web browser to interact with TocToc.


TocToc - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
https://127.0.0.1/TocToc/index.php?idioma=ENG&URL=127.0.0.1
Discover Music - Pan...
Toc²
Aceptar
Terminado
127.0.0.1
Adblock
0

Welcome to the TocToc system.

Insert your username and password.

User:
Password:

You are not a registered user? click here

You have forgotten your password? We send you another by e-mail

Figure 3.2: TocToc presentation screen.

3.1.3 Technologies Used

The TocToc 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 [58] and IPtables [59], 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 3.4 shows the relationship between different software components. We can see that the main server also offers a centralized DHCP service, making it possible for subscribers to be configured automatically.

Figure 3.4 also evidences the different support files created and the tools that use them. A single arrow line represents a reading action and a double arrow line represents a read/write or execute action. This design provides the required flexibility, allowing to make changes or add new modules in a straightforward manner.

3.1.4 TocToc's Basic Functionality

When a client connects to TocToc he usually does not know any connection information, namely TocToc's URL. Our system re-directs all the unregistered client accesses to TocToc's identification page by typing any URL and trying to access it. The most appropriate solution for this task is developing a captive portal. The enhancements required to make TocToc a captive portal-enabled platform are the following.

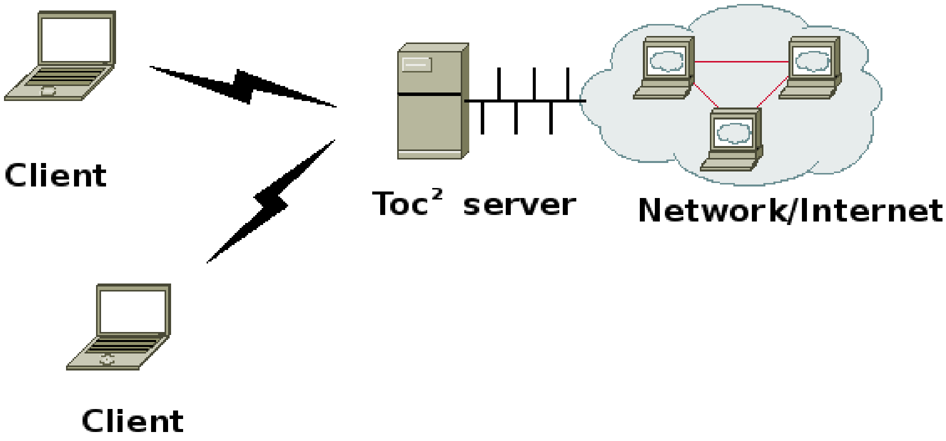


Figure 3.3: Architecture of TocToc.

Capture the subscribers who try to access the Internet through the system.

When an unregistered client tries to access to a web page, the system must re-direct the petition to the registration page instead. Once a client has been authenticated, e.g., introducing his user name and a password, he can access 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 [59].

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 access to 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.

Algorithm 1 User disconnection from TocToc.

```
#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 connections 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.
```

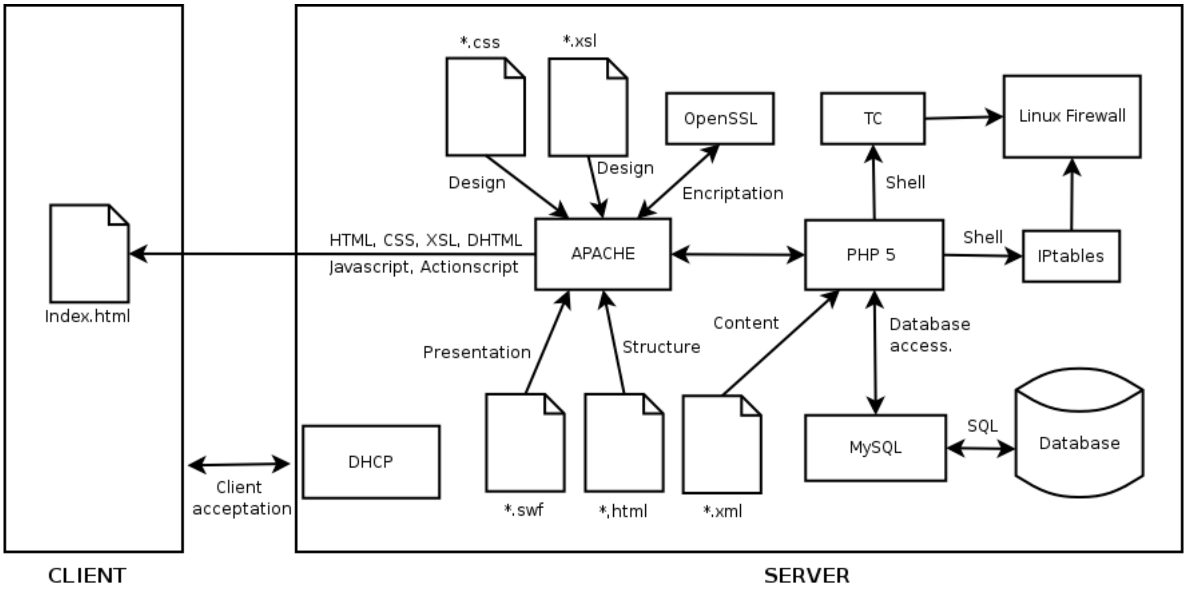


Figure 3.4: Relationship among TocToc's software components.

To know how many clients are connected at any time the system must detect the remaining clients, disconnect and register off-line clients as quickly as possible. To do that, the system uses Algorithm 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 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 periodically until it arrives to 1, hence disconnecting the user.

Control the connection speed of each client.

When subscribers access the Internet using a system as described above, 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 TocToc this characteristic has been named Cityticket. Depending on the Cityticket bought by a client, he acquires the corresponding connection speed. TocToc implements the Cityticket functionality by using the Traffic Control (TC) [58] Linux tool.

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

TocToc 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 3.5 shows this hierarchy. The TC tool is controlled by TocToc'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, and this allows to control the connection speed separately. Depending on the Cityticket assigned to a client, he acquires a different connection speed. Algorithm 2 shows the the creation of a queue for each client.

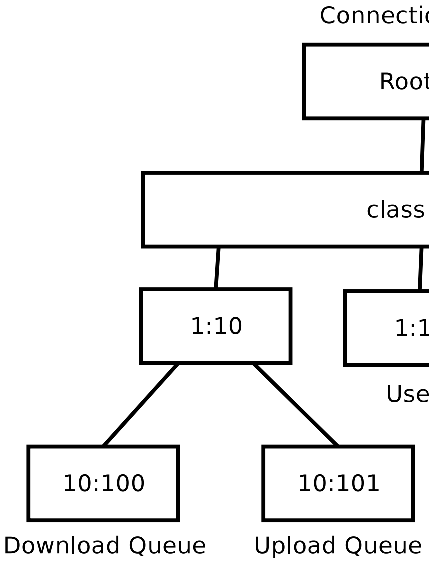


Figure 3.5: TC queue hierarchy.

Offer 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 purchase an Internet connection. TocToc 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 TocToc 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. TocToc 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 RuralNet.
- Free web server by connection ports: the system redirects all the free requests to a specific port to an external web server. TocToc has a list of servers in the main page. If a client selects a server in this list, the system redirects the petition the appropriate port.

The choice of free server based on IP or connection depends on the desired behaviour of TocToc.

3.1.5 The TocToc Interface Implementation

The TocToc Interface is designed to be maintained mostly unaltered regardless of which web browser, operating system or device is used. This means that TocToc must be designed with extended languages such a HTML and XML. Other specialized languages like ActionScript (Flash) and DHTML (a mix of Javascript and HTML) are able to 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 technology we are able to adapt the web pages to interact with the TocToc Core, hence allowing every device with web browsing capabilities to access the Internet and connect to the TocToc system.

Once inside the system the interface is rather intuitive. Figure 3.6 shows the 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: regular users, users with Internet access, and system administrators.

Algorithm 2 Connection speed of each user using TocToc.

```
#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.
```

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 Cityticket administration. By selecting a menu option the user is able to navigate through the system, thereby accessing the desired service.

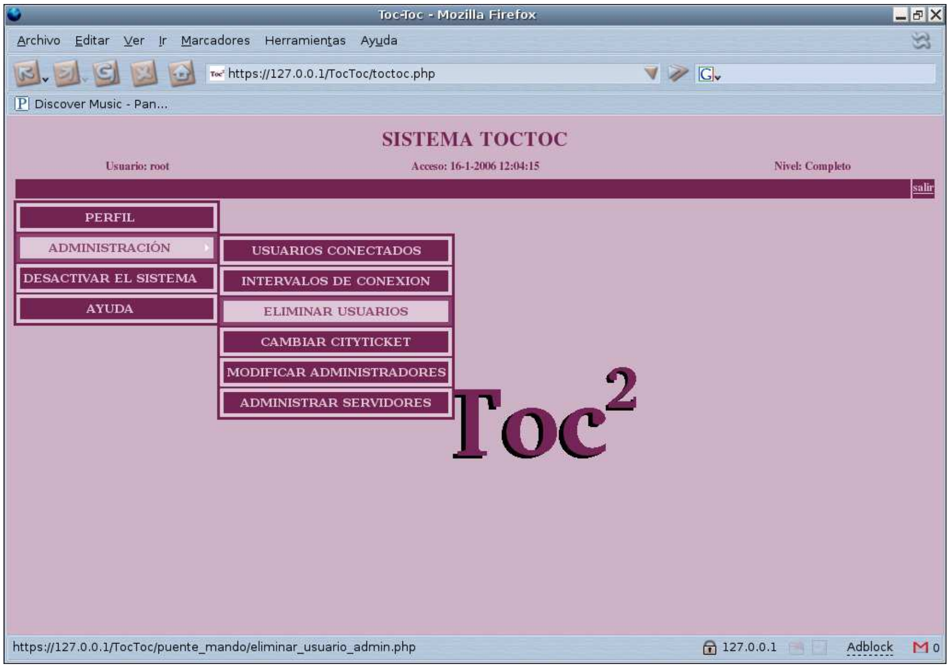
TocToc has been implemented to support multiple languages. To implement this functionality, we chose to separate the web structure from the text. This means that web pages have two components: a basic structure written in HTML language and text in the XML format on a separate file. This separation allows 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 TocToc supports English, Spanish, and Valencian.

3.1.6 Evaluation

In this section we present some experimental results where the purpose is to assess the correct operation of our TocToc system. Also, we will study how the bandwidth is shared among different subscribers when these have homogeneous or heterogeneous types of Internet connections.

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



Toc-Toc - Mozilla Firefox

https://127.0.0.1/TocToc/toctoc.php

SISTEMA TOCTOC

Usuario: root

Acceso: 16-1-2006 12:04:15

Nivel: Completo

https://127.0.0.1/TocToc/puente_mando/eliminar_usuario_admin.php

Figure 3.6: TocToc interface.

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 automate the evaluation process. Every client has a script that starts an FTP connection to our FTP server, which is accessed only through the TocToc system. We control the throughput of the FTP server so that we can model different connection speeds with TocToc's main server.

We run the tcpdump [60] 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 measuring 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 3.7 shows the results obtained.

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 around 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.

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 3.8 shows the obtained results. As can be observed, the TocToc system can offer the requested bandwidth according to the user configuration. We also observed that the throughput values sometimes suffer from variations that deviate them from

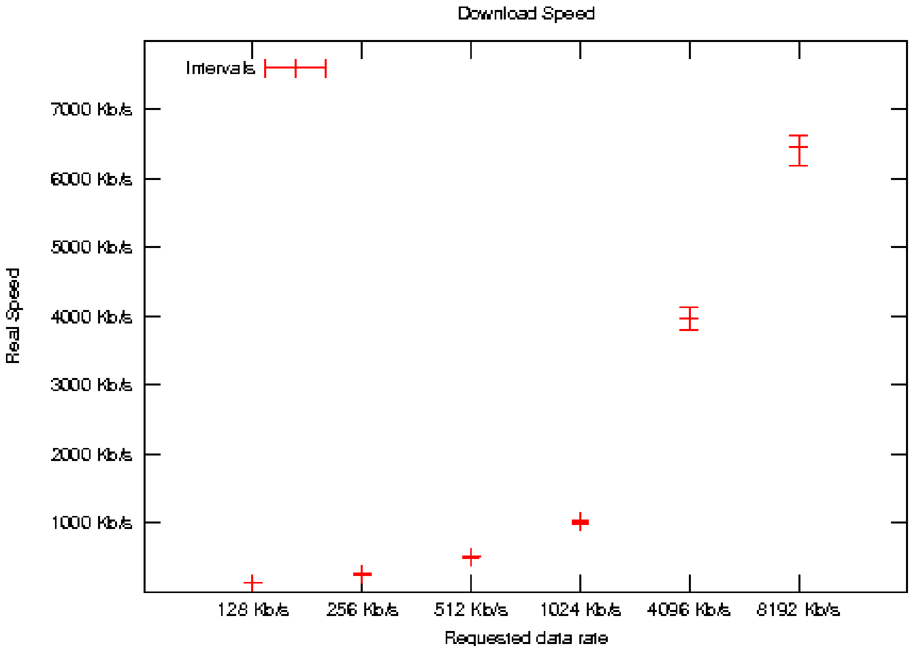


Figure 3.7: Connection speed for one client.

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

Figure 3.9 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 3.10) 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 solution of evenly sharing available resources when they become much lower than the aggregated bandwidth requested, we could instead apply a different policy. To accomplish that we would require a solution based on a weighted fair queueing policy. That way, when the available bandwidth becomes too low, users would receive a share of the channel's bandwidth that is proportional to the bandwidth they have paid for. However, this enhanced solution is outside the scope of this work.

3.1.7 Summary

In this section we describe a new captive portal called TocToc, a architecture based on the captive portal technology that provides access control. This captive portal allows 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 continue explaining the implementation of this captive portal and its Interface. We conclude with a set of experiments were we validate the traffic-shaping tools used by the system, finding that the service offered to clients is the one we expect.

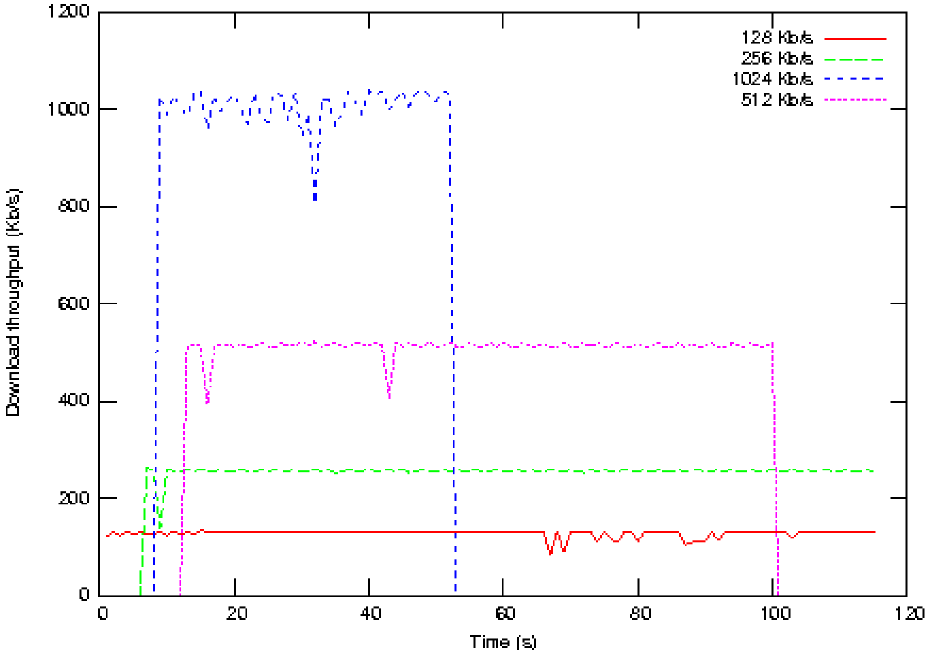


Figure 3.8: Download speed for the 4 clients under analysis.

In the next section we explain RuralNet, an architecture to deploy wireless networks in rural areas with the user's authentication based on TocToc.

3.2 RuralNet

In the previous section we showed an architecture to deploy a wireless network and give Internet access to a group of subscribers. In this section we explain RuralNet, a captive portal based system providing Internet connection 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 before. RuralNet is an architecture to strengthen Internet support in rural environments and allows subscribers to access the Internet. It can also provide a group of free services to all the people within a certain area.

RuralNet began in 2005 as a research project at the Technical University of Valencia, Spain. The project intended to develop new information and communication technology to offer low-bandwidth Internet access to isolated rural areas. With this purpose we developed RuralNet, an experimental wireless platform which combines the promising paradigm of wireless mesh networks and 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. To date, we deployed the proposed system in a small-scale project on a rural area located on the south of the Comunidad Valenciana which encompasses about 50 subscribers including local industry and villagers.

3.2.1 Objectives of RuralNet

The general purpose of RuralNet is to strengthen Internet support in rural environments through wireless technologies in the Comunidad Valenciana, Spain. The system should empower mobile users with the ability to access Internet applications on the move.

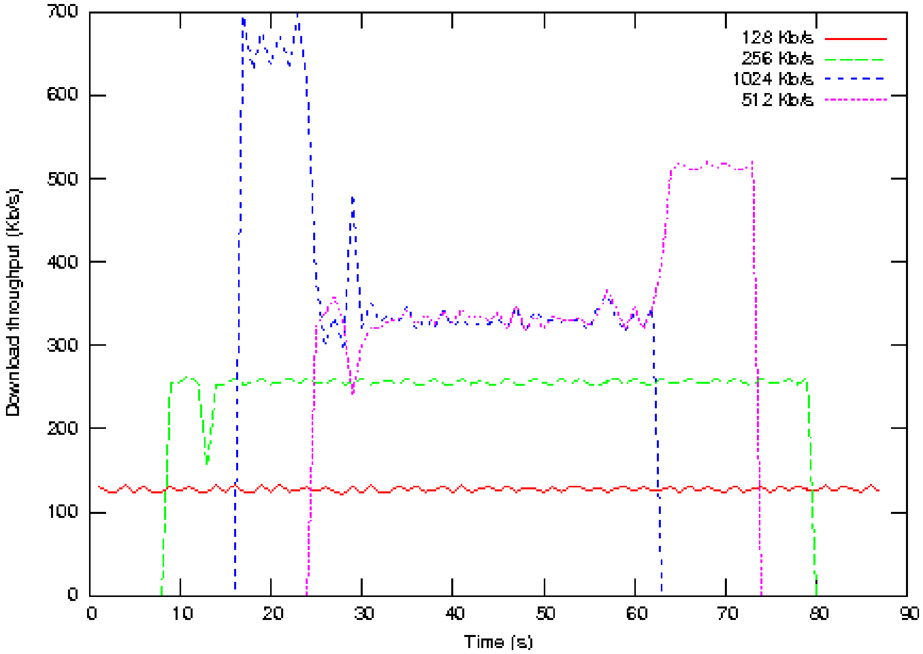


Figure 3.9: Download speed when the bandwidth towards the FTP server is limited to 1024 Kb/s.

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 networks paradigms, is able to reach distant areas at a low cost, while offering a wide range of telecommunication services and applications.

3.2.2 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 telecommunication network is shown in Figure 3.11. 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://www.grc.upv.es/6software/index.html`. RuralNet makes use of the TocToc software architecture to provide a flexible way to capture users request.

We have built a prototype composed by multiple Wi-Fi access points connected to a main server, based on TocToc's architecture. Our architecture is conceptually organized into three levels, which are: the management level (main server), the network connection level (called Backbone Net), 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 Net level. In this level is where the TocToc functionality resides.

- **Backbone Net 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

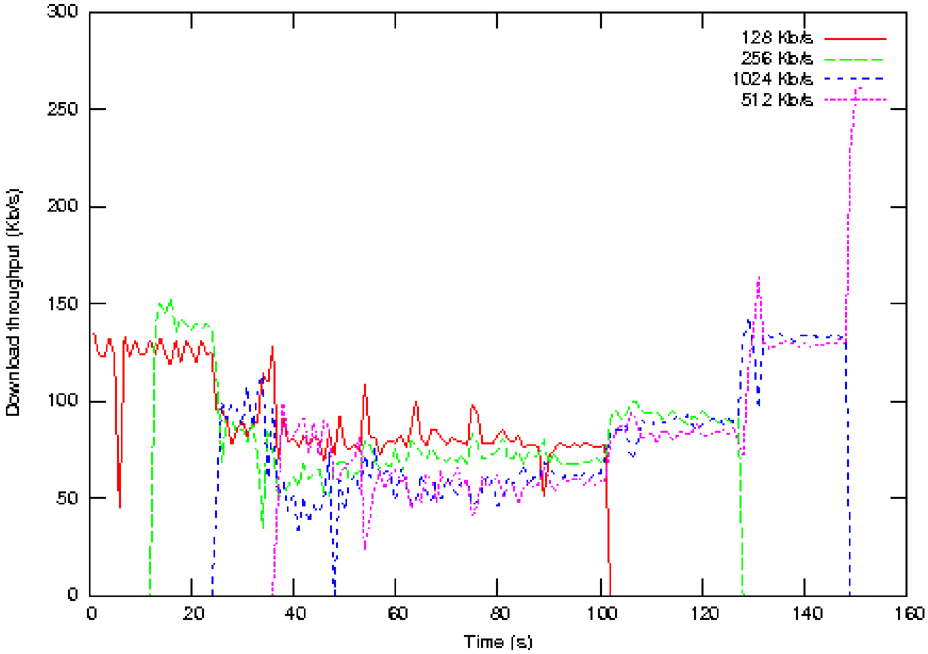


Figure 3.10: Download speed when the bandwidth towards the FTP server is limited 256 Kb/s.

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 [61] firmware, which allow us to implement access control into the node while increasing the hotspot coverage by implementing a multi hop 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 to 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.

Every client within the coverage area of RuralNet can access all the services offered, which does not mean free access or uncontrolled access. Our system is implemented under a captive portal solution (using TocToc, presented in section 3.1) 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 is it necessary to use special software.

3.2.3 Controlling the Access to RuralNet

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. The main server controls client access depending on whether he is a registered user or not. Depending on the client's access level, different services will be provided.

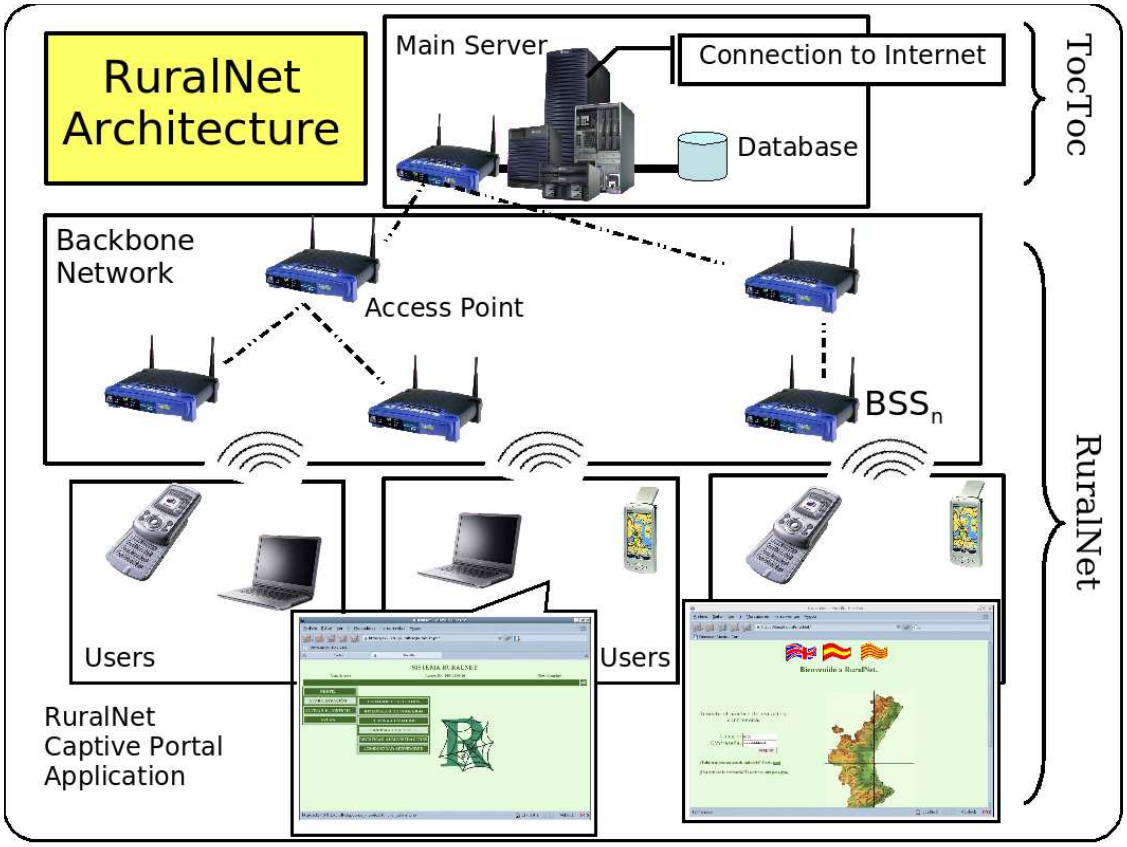


Figure 3.11: The RuralNet system architecture.

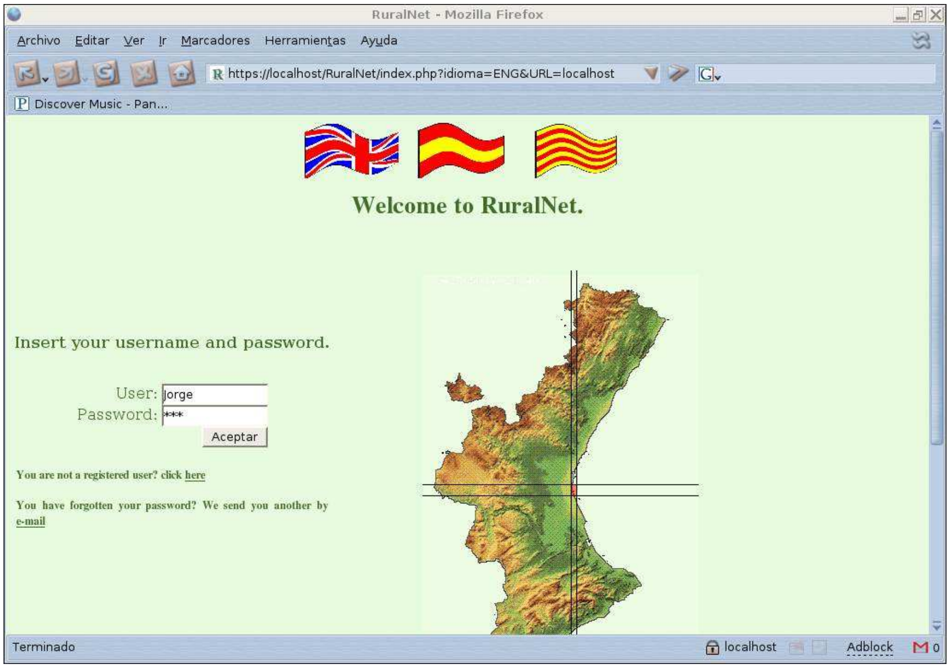
Figure 3.12 shows RuralNet presentation screen. The first time a client accesses the system he is asked to register himself with the captive portal. After a login process the user can use any of the freely available services or purchase others, like Internet access. Concerning the Internet access service, RuralNet allows each client to choose among multiple connection speeds, making the price vary accordingly.

3.2.4 Evaluation

The access control is the same of TocToc and therefore doesn't need extra evaluation. We focus instead on evaluating the behaviour of the distribution network using ping sessions and evaluating the impact on round-trip time. Figure 3.13 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 that 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). 10% of the traffic suffers from poor performance. More specifically, 4% of the traffic experiences too high delays, being considered lost in practice.

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.


RuralNet - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
https://localhost/RuralNet/index.php?idioma=ENG&URL=localhost
Discover Music - Pan...
Aceptar
Terminado
localhost
Adblock
0

Welcome to RuralNet.

Insert your username and password.

User: Jorge
Password: ***

You are not a registered user? click here

You have forgotten your password? We send you another by e-mail

Figure 3.12: RuralNet presentation screen.

3.2.5 Summary

In this section we proposed RuralNet, a captive portal based architecture especially designed to provide Internet access in rural areas.

We first analyzed the architecture of the system, putting into evidence the most important elements that conform RuralNet's infrastructure. We then describe 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.

Overall, this section 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.

3.3 Extension to the Proposed Architecture

In section 3.2 we develop an extension of TocToc for rural areas. In this section now we explain Dulenduè, an extension with extra functionality for urban environments based on the position of the user's device.

Dulenduè [1] was born as a tourist portal with extra functionability like a service searcher. It is based on the position of the user and the services existing around him. This technology is called context-aware computing [62]. In computer science it refers to the idea that computers can both sense and react based on their environment. Devices may have information about the circumstances under which they are able to operate and based on rules, or an intelligent stimulus, react accordingly. The term context-awareness in ubiquitous computing was introduced by Schilit [63]. Context-aware devices

[1] Dulenduè means "Where are you?" in a dialect of the region of Lombardía, close to Milán.

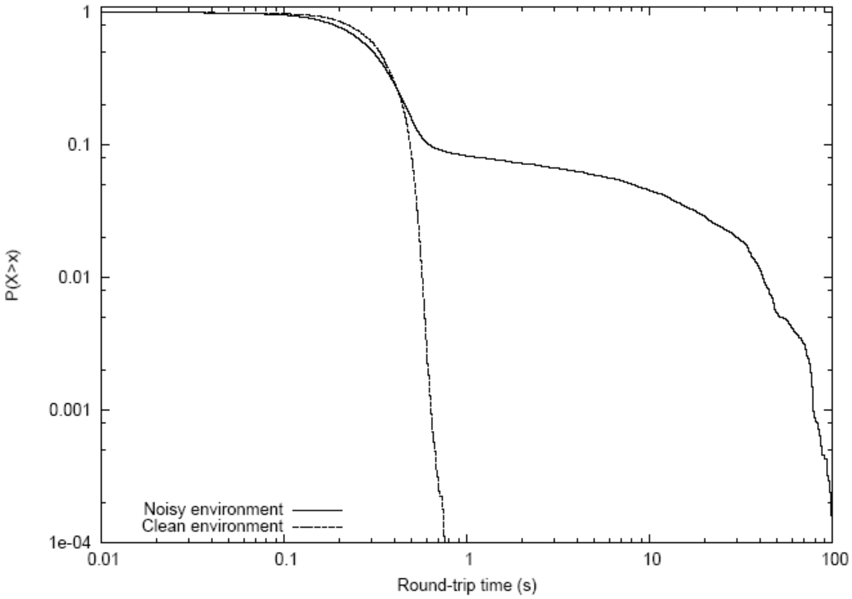


Figure 3.13: Evaluation of the distribution network.

may also try to make assumptions about the user's current situation. Dey [64] defines context as "any information that can be used to characterize the situation of entities."

3.3.1 Objectives and Architecture of Dulenduè

The general purpose of Dulenduè is to generate a context-aware application to strengthen the tourism in the city of Valencia.

The application can work like a tourist information point as well as a shopping consulting center where tourists can search for information about the city anytime that he needs.

This architecture is designed to cover an urban area, using a public network as telephony. Figure 3.14 shows the overall system architecture for Dulenduè. This architecture is similar to RuralNet's architecture, except that it does not use a mesh network, relying on a wired connection instead.

Our architecture is conceptually organized into four levels, which are: the management level (main server), the network connection level (divided into the connection network and the access network), and the user level. The system has two different networks, the connection network used to connect the main server with the access point, and the access network, used to connect the clients with the access point.

- **Main server.** A server that controls user authentication. It also has a database to store information about all users and information about services offered. It is connected to the wired network.
- **Wired network.** This network connects each access point of the access network to the main server. Can be an ethernet network or use another public network like the telephony network.
- **Access network.** A wireless network deployed in the urban context. Formed by access points installed into the city streets, covering the entire urban area. Each node can cover an area

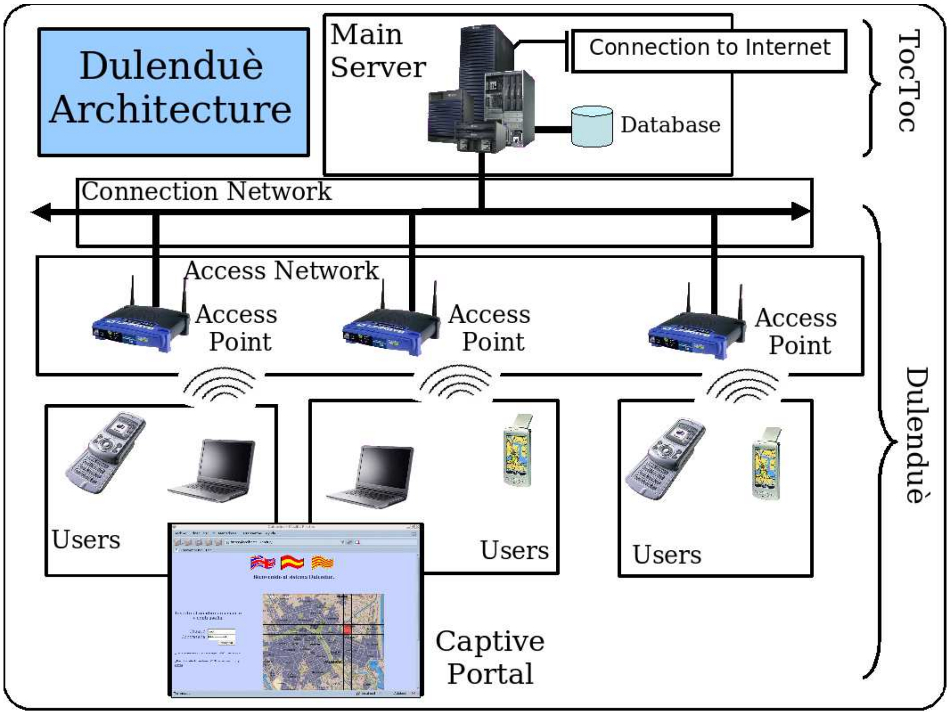


Figure 3.14: Architecture of Dulenduè

around 100 meters, the size of a city block. This network is composed by different access points connected to the wired-network.

- **User level.** Similarly to TocToc and RuralNet, the user only needs a device with a Wi-Fi card and a web browser. The Wi-Fi card connects the device to the access network and the web browser allows interaction with Dulenduè.

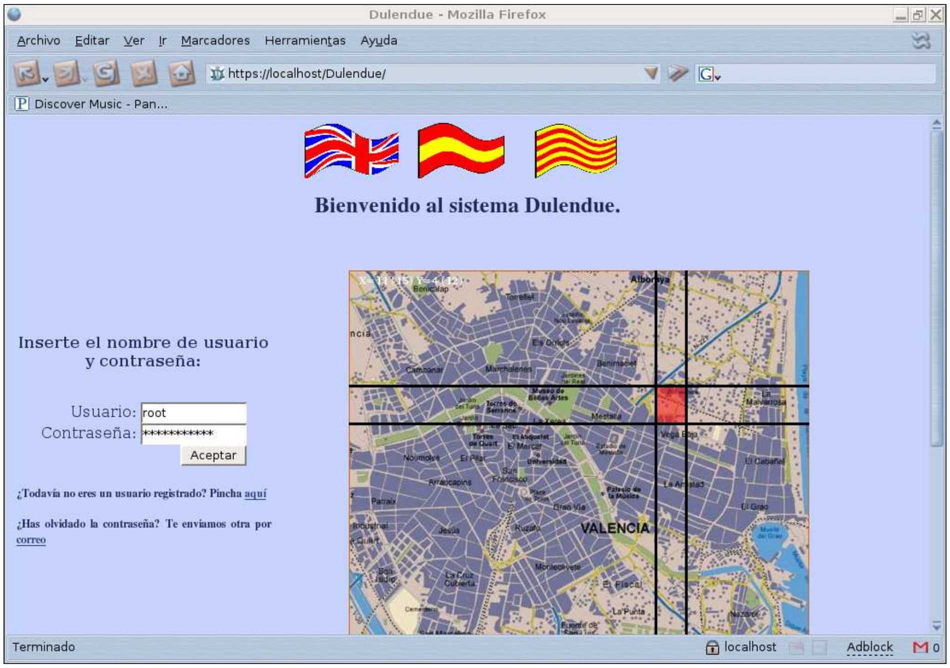
3.3.2 The Dulenduè Localization System.

The position of the user is obtained from the access point used to access to the Internet. Each access point is stored into the database with its localization. When a user connects to Dulenduè, the MAC address of packets arriving to the main server becoming to the access point. The system searches for this MAC into the database and obtain the position of the user with an error of less than 100 meters. Enough precision for this context aware application. Figure 3.15 shows the main screen of Dulenduè with the position of the user.

The services are stored into a database in the main server. When a new offered service is created, like a restaurant or a pharmacy, it also stores its position. If a user connects to Dulenduè and searches for a service, the system obtains the position of the user and, afterwards , searches the database for all services near this position. Finally, it returns the results to the user in a web page.

With this information, the user can search for services, shops or any date stored in the database. The system answers all the queries depending on the position of the device, and prioritizes the results according to the user position.

Figure 3.16 shows the result when a user searches for a restaurant. In this example the user selects a restaurant with an average price of less than 40 euros. The user can also select the proximity of the restaurant. Then the application shows all the restaurants that match the specifications of the user. If needed, Dulenduè can show a map with the position of the restaurant or a map with the closest



Bienvenido al sistema Dulendue.

Inserte el nombre de usuario
y contraseña:

Usuario: root
Contraseña: ************
Aceptar

¿Todavía no eres un usuario registrado? Pincha aquí

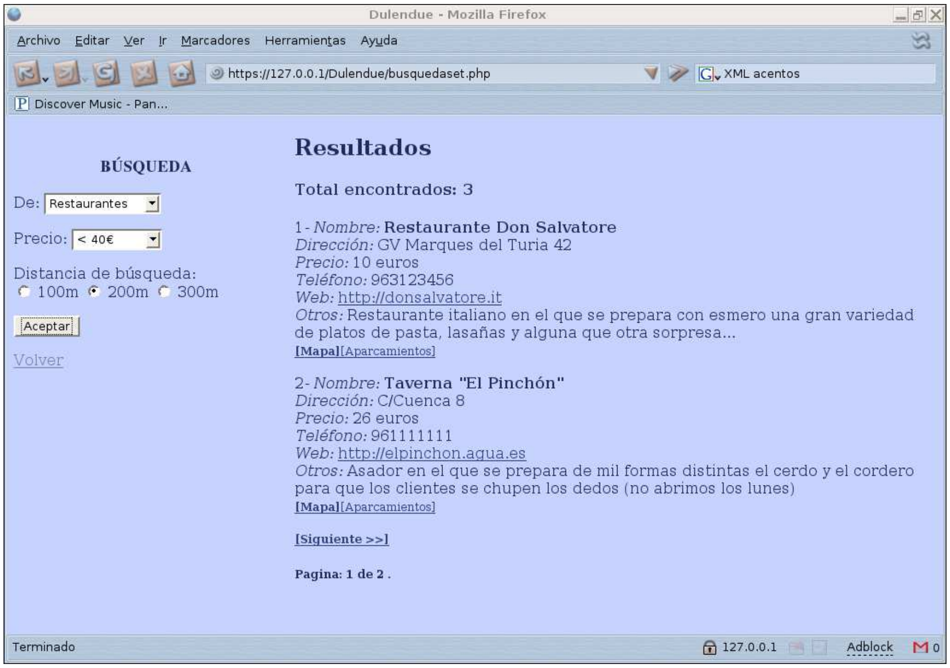
¿Has olvidado la contraseña? Te enviamos otra por correo

Figure 3.15: Main screen of Dulenduè with the position of the client.

parkings. Figure 3.17 shows the map that appears to an user when he selects the option to see the location of a restaurant.

3.3.3 Summary

In this section we present Dulenduè, a possible extension of RuralNet with extra functionality based on the concept of context-aware computing. Dulenduè adds a service searcher based on the position of the user in the city. The results obtained by Dulenduè vary accordingly with the user localization.


Dulendue - Mozilla Firefox
Archivo Editar Ver Ir Marcadores Herramientas Ayuda
https://127.0.0.1/Dulendue/busquedaset.php
XML acentos
Discover Music - Pan...
Terminado
127.0.0.1
Adblock
0

BÚSQUEDA

De: Restaurantes

Precio: < 40€

Distancia de búsqueda:
○ 100m ● 200m ○ 300m

Aceptar

Volver

Resultados

Total encontrados: 3

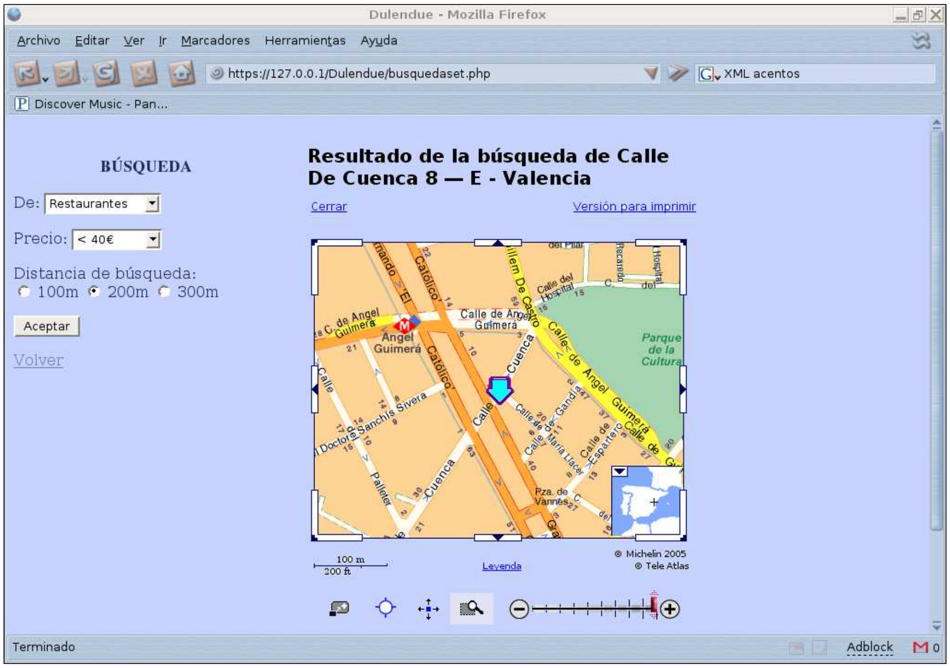
1 - *Nombre:* Restaurante Don Salvatore
Dirección: GV Marques del Turia 42
Precio: 10 euros
Teléfono: 963123456
Web: http://donsalvatore.it
Otros: Restaurante italiano en el que se prepara con esmero una gran variedad de platos de pasta, lasañas y alguna que otra sorpresa...
[Mapa][Aparcamientos]

2 - *Nombre:* Taverna "El Pinchón"
Dirección: C/Cuenca 8
Precio: 26 euros
Teléfono: 961111111
Web: http://elpinchon.agua.es
Otros: Asador en el que se prepara de mil formas distintas el cerdo y el cordero para que los clientes se chupen los dedos (no abrimos los lunes)
[Mapa][Aparcamientos]

[Siguiente >>]

Pagina: 1 de 2 .

Figure 3.16: Looking for a restaurant with Dulenduè



BÚSQUEDA

De: Restaurantes

Precio: < 40€

Distancia de búsqueda:
○ 100m ● 200m ○ 300m

Aceptar

Volver

Resultado de la búsqueda de Calle De Cuenca 8 — E - Valencia

Cerrar

Versión para imprimir

Figure 3.17: Map to localize the address of a restaurant in Dulenduè.

Chapter 4

Castadiva: a MANET Emulator

In Chapter 3 we present an architecture to deploy a wireless network to offer TCP/IP based services. In this chapter we are going to present a test-bed designed to evaluate this architecture. Castadiva is a test-bed solution that allows to emulate wireless scenarios using low cost commercial-of-the-shelf devices combined with the power of Linux tools. Castadiva allows to generate a network topology, export it to real devices and obtain test results. It can also generate different traffic between nodes. A test-bed approaches the real behaviour of a network, and this is the point where Castadiva can help in network research efforts. The test-bed runs a real implementation of routing protocols in real devices, allowing to test the implementations before deploying them in a real context.

The rest of this chapter is structured as follows. Section 4.1 explains the objectives of Castadiva. Section 4.2 describes the proposed architecture and Section 4.3 describes its implementation. Section 4.4 describes the evaluation made to validate Castadiva. Finally, Section 4.5 draws the conclusions of this chapter.

4.1 Objectives of Castadiva

The main objective of Castadiva is to generate a test-bed to emulate MANETs where we can test the behaviour of the network with different routing protocols.

This test-bed must 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 simulator ns-2 used in our group for comparison purposes and have an user-friendly interface that allows the user to generate the entire simulation easily.

4.2 Architectural Overview

Castadiva is a test-bed developed to evaluate and analyze protocols and applications for MANETs. The test-bed relies on an actual wireless network between nodes for testing purposes. Castadiva is composed by a server that runs the main application, several wireless nodes, two different networks and an application to coordinate all devices.

Castadiva's server executes the application and configures the network devices. Our prototype is a Pentium IV with 1 GB of RAM memory, and has a Linux Debian distribution installed.

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. 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 [61] kernel is a good 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/control purposes.

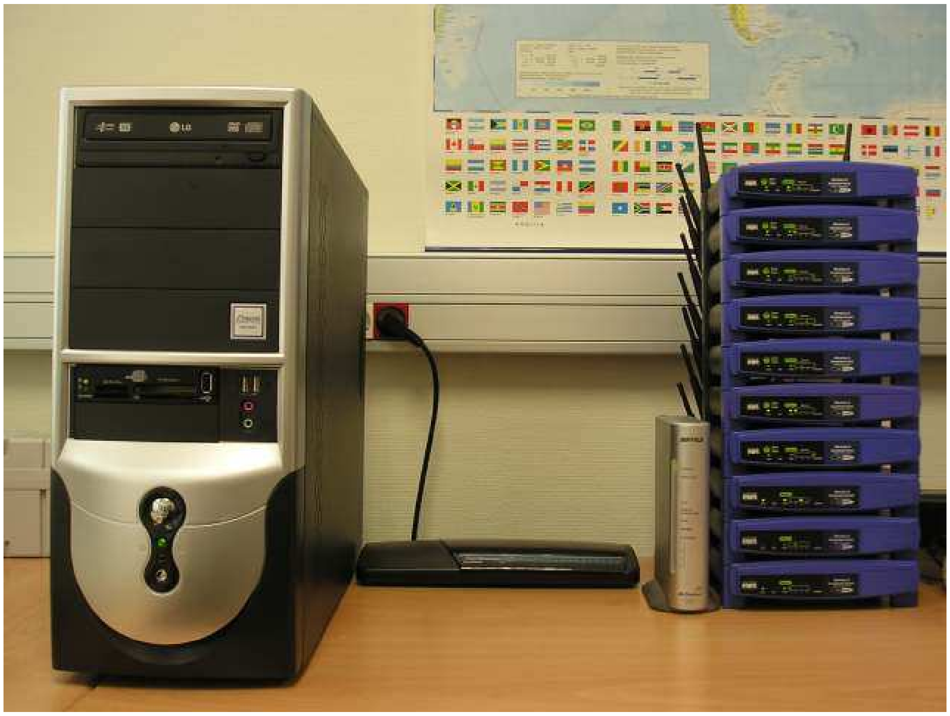


Figure 4.1: Castadiva's physical network.

Figure 4.1 shows our Castadiva test-bed. One switch connects Castadiva's server with all the wireless nodes for coordination purposes. On the right hand of the picture the group of wireless nodes being used is shown. It consists of ten Linksys routers (models WRT56G and WRT56GL) and a Buffalo router (model WZR-RS-G54). The wireless ad hoc network conformed by these nodes is the one used in Castadiva's test-bed experiments.

Since the controlling application also requires communicating with nodes to send control packets, Castadiva combines two different networks: the coordination network (wired), that connects Castadiva's core with the wireless nodes, and the wireless network, where actual tests are run. Figure 4.2 shows a schema of Castadiva's components.

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, avoiding large latency. Basically, this 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 so as to create the desired network topology, and also to run small traffic-generating applications available on each wireless node. For communication purposes, we rely on the SSH protocol to send instructions through this network. Using a fast network means that all nodes will start an experiment at about the same time, avoiding significant latency and maximizing 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 IEEE 802.11g technology.

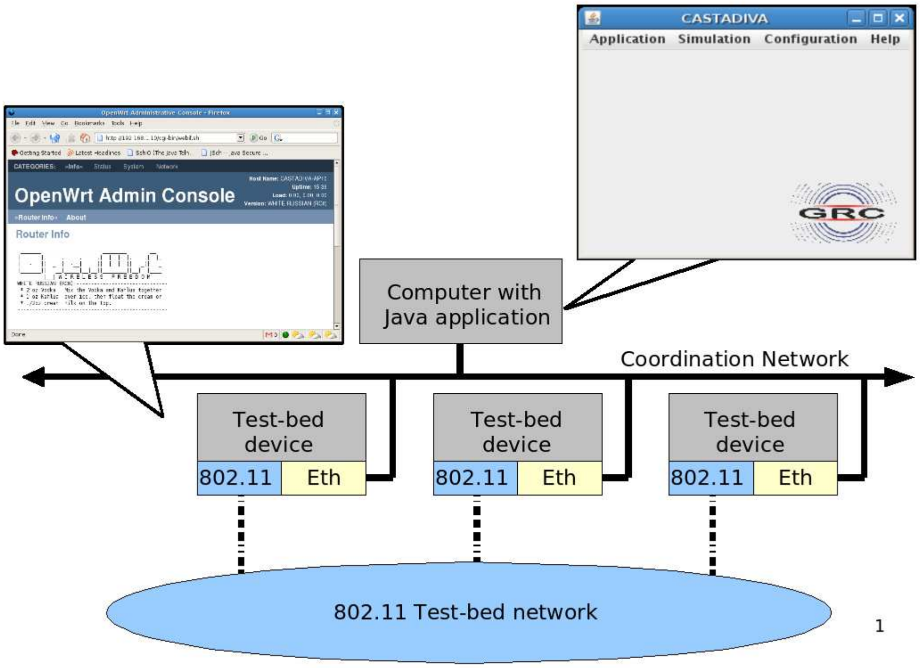


Figure 4.2: Schema of Castadiva's components.

The main application, called Castadiva's core, is developed in Java. It has two main functions: (a) to allow a user to interact with the system so as to define all the test parameters required (GUI) and (b) to coordinate the wireless nodes during an experiment and manage traffic generation between pairs of nodes. By using Castadiva's GUI a user can control 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.

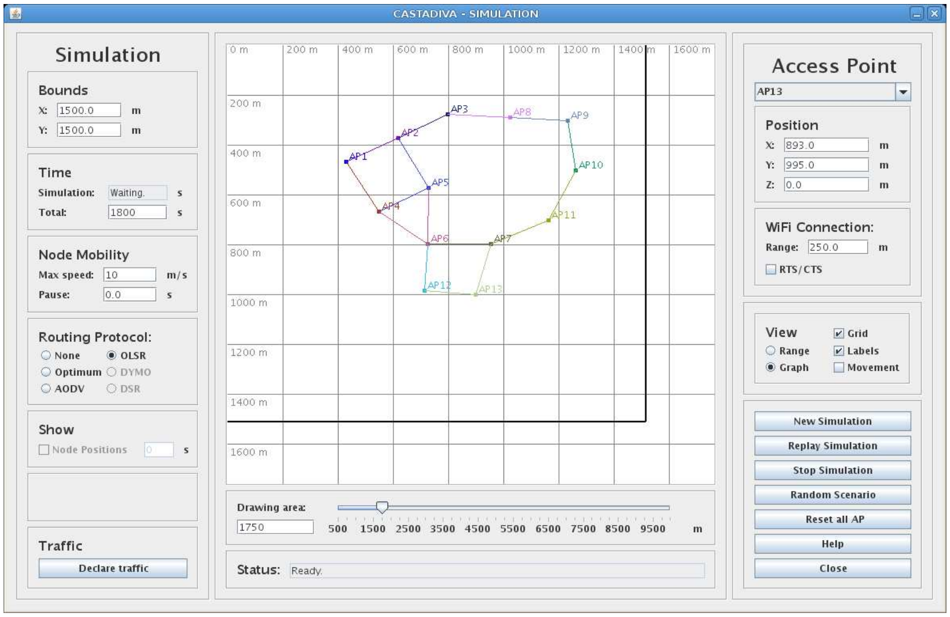
Figure 4.3 shows how Castadiva allows a user to interact with the network. By filling all scenario parameters and deploying all nodes are emulate a network where the user can study the behaviour of applications and routing protocols. When all nodes are deployed the user can press the button Simulate, and each physical node will be re-programmed so as to enforce the chosen network topology.

4.3 Castadiva's Implementation Details

In this section we detail the requirements of Castadiva. We describe the software tools we have developed to connect all the wireless nodes with the server, and how Castadiva allows making 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. Each node must be a Linux based system with an SSH server and an NFS client; besides, it must include the libgcc library and have the IPtables toolset installed (see figure 4.4).

The connection between Castadiva's core element (server) and each node is made using both SSH



Bounds
X: 1500.0 m
Y: 1500.0 m

Time
Simulation: Waiting. s
Total: 1800 s

Node Mobility
Max speed: 10 m/s
Pause: 0.0 s

Routing Protocol:
○ None ● OLSR
○ Optimum ○ DYMO
○ AODV ○ DSR

Show
□ Node Positions 0 s

Traffic

Drawing area: 1750

Status: Ready.

AP13

Position
X: 893.0 m
Y: 995.0 m
Z: 0.0 m

WiFi Connection:
Range: 250.0 m
□ RTS/CTS

View
○ Range
● Graph
☑ Grid
☑ Labels
□ Movement

Figure 4.3: A view of Castadiva's GUI.

and NFS connections. On Castadiva's server, the user interacts with the 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 into a server folder shared through NFS. This is a easy way to spread instructions to all nodes, and it solves storage limitation problems on nodes. When tests start, a group of files with the results are created and stored into Castadiva's server by again 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, obtaining the different test-bed statistics. Finally, the application displays results to the user.

The application was developed using the Java programming language and the BASH scripting language. To make SSH connections through Java we use JSch [65], by JCraft. It is required to coordinate all the nodes during experiments. Castadiva's implementation can be divided in two parts: the main application and the light-weight applications running on wireless nodes.

4.3.1 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 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, Ttcp, and TrafficFlow. The first two are open source and exist in most Linux distributions, while the third was developed by us.

Network topology configuration is made through the IPtables [58] 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 4.1 we show an

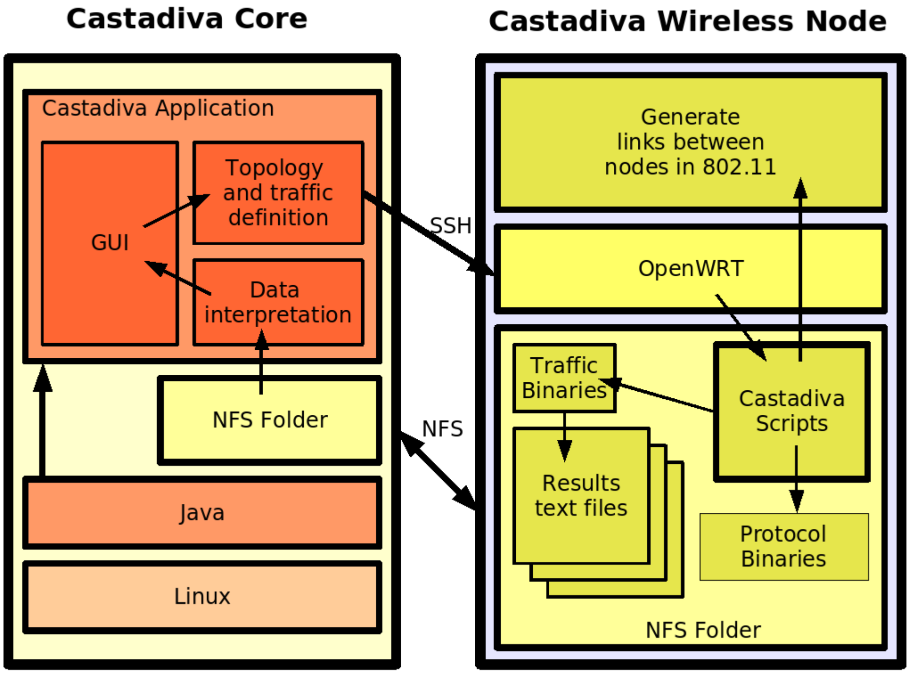


Figure 4.4: Software components for Castadiva.

example of IPtables rules generated from Castadiva (using the scenario of Figure 4.7 as reference).

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
2 (00:0F:66:D9:BE:01)	(none)
3 (00:13:10:83:99:CA)	/usr/sbin/iptables -I INPUT -m mac –mac-source 00:18:39:BC:B5:8E -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 4.1: Iptables rules: example of usage in Castadiva's framework.

To generate traffic we use both Ttcp and TrafficFlow tools. Ttcp is an application with network testing purposes. We use this tool to generate all the UDP traffic between nodes. Ttcp allows us to set the number of packets per second sent to a node, along with the size of these packets. The TrafficFlow tool is designed to create a TCP flow between two nodes. 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.

Castadiva also includes routing agents for well-known routing protocols, such as AODV [66] and OLSR [67]. We made different tests using both AODV and OLSR, and finally we decided to use the OLSR since OpenWRT includes a stable implementation.

4.3.2 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 performance analysis. A user can define the characteristics of wireless nodes, i. e., each node is deployed at a specific position in a simulated area as chosen by the user, conforming the network topology. 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 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. In its development we have used the Model - View - Controller [68] design as reference.

Scenario Generation Tool

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 routing protocol used. By pressing the start button tests begin, and Castadiva returns the test results automatically. We now offer more details about Castadiva's GUI.

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 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.

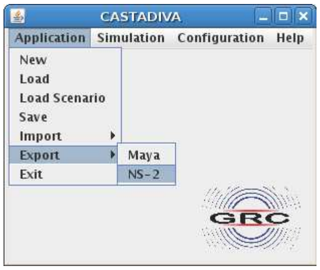


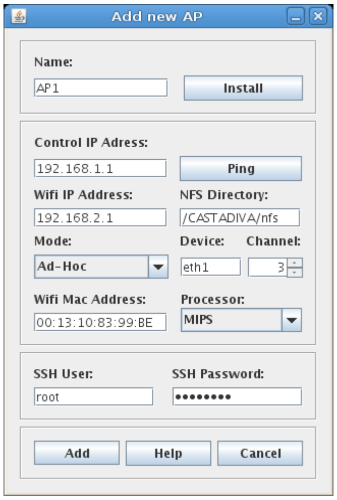
Figure 4.5: Application control menu.

Figure 4.5 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.

Adding Nodes to the Test-bed

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 4.6 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. 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



Name: AP1

Control IP Adress: 192.168.1.1

Wifi IP Address: 192.168.2.1

NFS Directory: /CASTADIVA/nfs

Mode: Ad-Hoc

Device: eth1

Channel: 3

Wifi Mac Address: 00:13:10:83:99:BE

Processor: MIPS

SSH User: root

SSH Password: •••••••••

Figure 4.6: Node configuration interface.

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 the 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 are run on each node, and must be compiled for all types of CPU 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 to each individual router to send the required commands. Also, a Ping button was included to allow testing the connectivity between the server and routers.

Ad hoc Network Scenario Generation

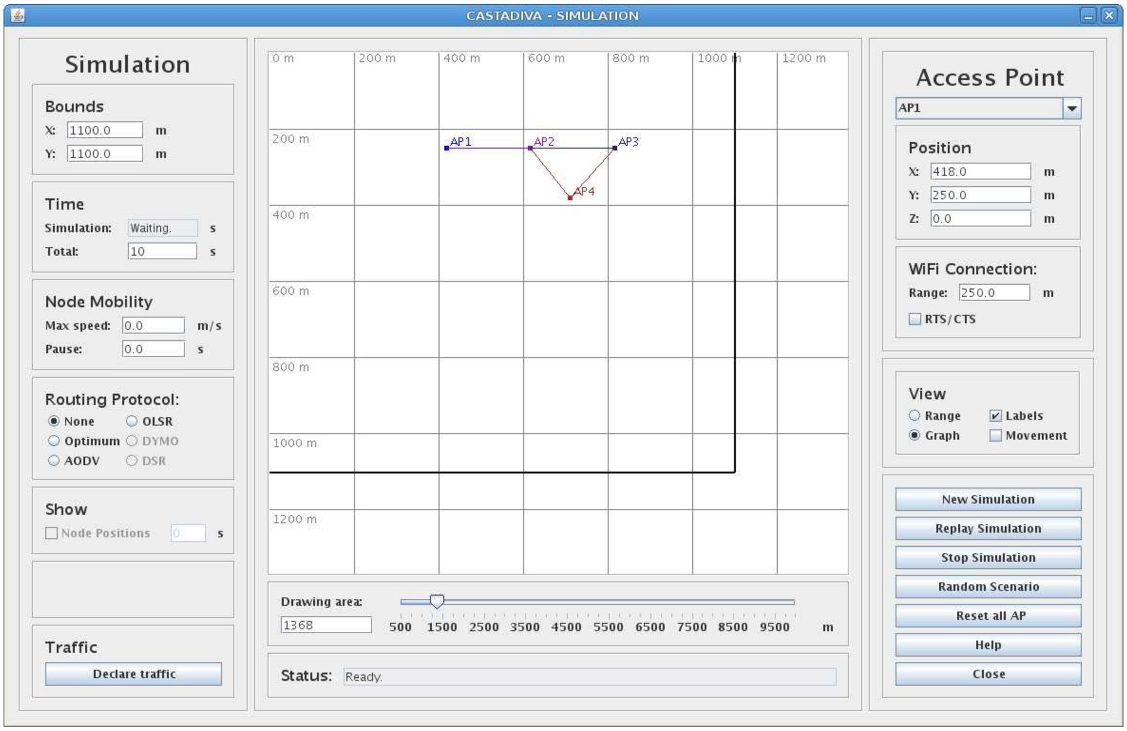
Once all the nodes are defined, they can be distributed to conform a scenario. Castadiva supports both manual and random topology generation, and scenarios are set through Castadiva's blackboard. The blackboard is a representation of a virtual environment where nodes are located. Nodes are differentiated through different colors and labels. The radio communication range is also shown through a circle of the same color.

Figure 4.7 shows the topology generation environment. We can see four nodes located in a scenario of 1000 x 800 meters (scenarios bounds are marked with a darker line).

At the right hand we may edit node properties, such as position and signal range. 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 selected routing protocol. The "Declare Traffic" button allows setting traffic, as shown below, and the stop button halts it.

A status bar provides general information to inform the user about what is being done, and the horizontal scrolls allow zooming in and out. Finally, the user may alternate between two different views: radio ranges or a graph. Every edge of the graph represents an IEEE 802.11 link connection, which is a more intuitive view.



Simulation

Bounds
X: 1100.0 m
Y: 1100.0 m

Time
Simulation: Waiting. s
Total: 10 s

Node Mobility
Max speed: 0.0 m/s
Pause: 0.0 s

Routing Protocol:
● None ○ OLSR
○ Optimum ○ DYMO
○ AODV ○ DSR

Show
□ Node Positions 0 s

Traffic

Drawing area: 1368

Status: Ready.

Access Point
AP1

Position
X: 418.0 m
Y: 250.0 m
Z: 0.0 m

WiFi Connection:
Range: 250.0 m
□ RTS/CTS

View
○ Range ☑ Labels
● Graph □ Movement

Figure 4.7: Scenario definition with Castadiva.

Mobility in Castadiva.

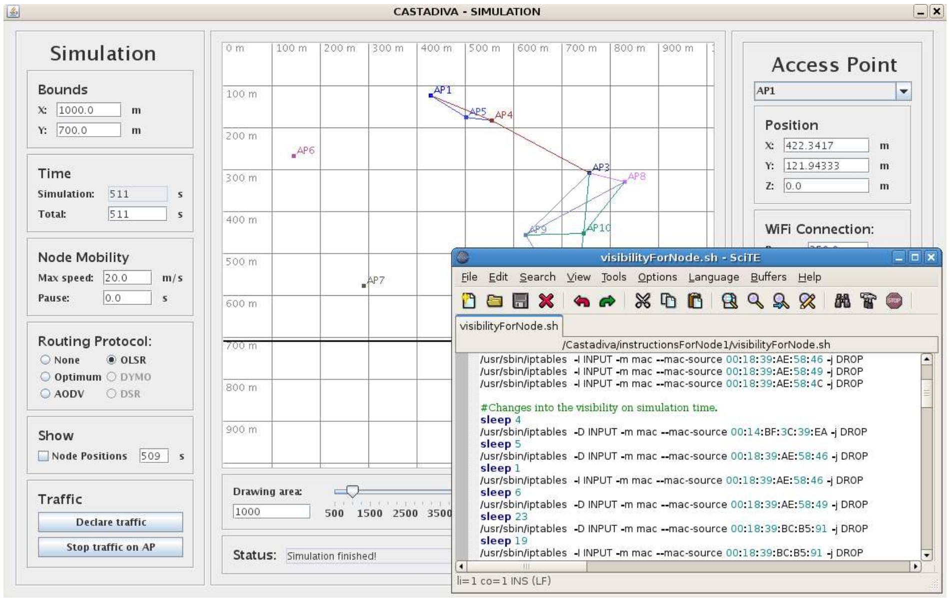
It is important to point out the speed and pause option of the Scenario Window. If a user picks a value greater than zero in the speed option, Castadiva uses a mobility model similar to the random waypoint model used in ns-2, where each node acquires a random movement with a speed between zero and the inserted value. When a node arrives to the destination point, it waits for a selected pause time and then selects a new random destination point to move to.

For the implementation of mobility, Castadiva generates all node movements required for the emulation before it starts. Then it calculates the visibility range for each node every second. The obtained visibility is translated to Iptables rules and written to one file for each node. This files will be loaded at each access point through NFS when the simulation starts. Figure 4.8 shows the file loaded by the access point called "AP1". This figure shows different Iptables rules inserted between sleep rules. The sleep time distributes the Iptables rules throughout the simulation time. So, each rule is loaded only when the emulation represents a node that changes the link connection with respect to other node, Algorithm 3 shows the behaviour of Castadiva when a node (with MAC 00:14:BF:3C:39:EA) is outside its range at second 10, and then returns to its range at second 25.

Algorithm 3 Iptables rules for a simulation between two nodes.

```
sleep 10
iptables -I INPUT -m mac –mac-source 00:14:BF:3C:39:EA -j DROP
sleep 15
iptables -D INPUT -m mac –mac-source 00:14:BF:3C:39:EA -j DROP
```

Castadiva also allows a user to see the position of every node at a given instant. When a simulation finishes, the user can activate the Show option and pick a certain instant. Immediately Castadiva shows the network topology in that instant of time. This option is useful to do an off-line evaluation of the



Simulation

Bounds
X: 1000.0 m
Y: 700.0 m

Time
Simulation: 511 s
Total: 511 s

Node Mobility
Max speed: 20.0 m/s
Pause: 0.0 s

Routing Protocol:
○ None ● OLSR
○ Optimum ○ DYMO
○ AODV ○ DSR

Show
□ Node Positions 509 s

Traffic

Drawing area: 1000

Status: Simulation finished!

Access Point
AP1

Position
X: 422.3417 m
Y: 121.94333 m
Z: 0.0 m

WiFi Connection:

visibilityForNode.sh - SciTE

File Edit Search View Tools Options Language Buffers Help

/Castadiva/instructionsForNode1/visibilityForNode.sh

```
/usr/sbin/iptables  -I INPUT -m mac --mac-source 00:18:39:AE:58:46 -j DROP
/usr/sbin/iptables  -I INPUT -m mac --mac-source 00:18:39:AE:58:49 -j DROP
/usr/sbin/iptables  -I INPUT -m mac --mac-source 00:18:39:AE:58:4C -j DROP

#Changes into the visibility on simulation time.
sleep 4
/usr/sbin/iptables  -D INPUT -m mac --mac-source 00:14:BF:3C:39:EA -j DROP
sleep 5
/usr/sbin/iptables  -D INPUT -m mac --mac-source 00:18:39:AE:58:46 -j DROP
sleep 1
/usr/sbin/iptables  -I INPUT -m mac --mac-source 00:18:39:AE:58:46 -j DROP
sleep 6
/usr/sbin/iptables  -D INPUT -m mac --mac-source 00:18:39:AE:58:49 -j DROP
sleep 23
/usr/sbin/iptables  -D INPUT -m mac --mac-source 00:18:39:BC:B5:91 -j DROP
sleep 19
/usr/sbin/iptables  -I INPUT -m mac --mac-source 00:18:39:BC:B5:91 -j DROP
```

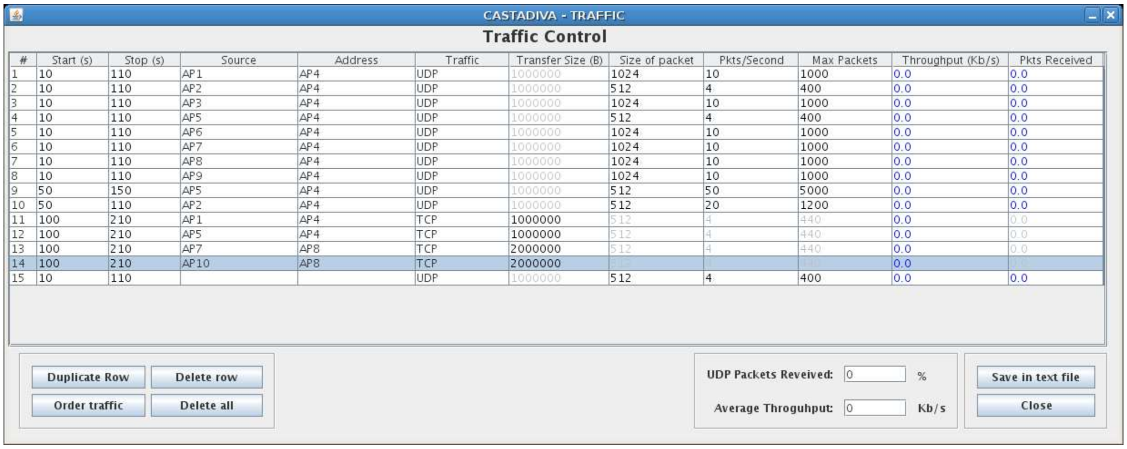
li=1 co=1 INS (LF)

Figure 4.8: Mobility implementation.

network topology along the simulation time.

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 i. e. TCP or UDP, and invalid values are marked with red. Examples of parameters are: packet size, packets per second, start time, end time and maximum number of packets sent. Figure 4.9 shows a usage example of this tool, where rows define seven traffic connections. It contains some helpful buttons that allow making row operations (delete, order by starting time, or copy). Traffic settings are exportable to ns-2 format also.



Traffic Control

#	Start (s)	Stop (s)	Source	Address	Traffic	Transfer Size (B)	Size of packet	Pkts/Second	Max Packets	Throughput (Kb/s)	Pkts Received
1	10	110	AP1	AP4	UDP	1000000	1024	10	1000	0.0	0.0
2	10	110	AP2	AP4	UDP	1000000	512	4	400	0.0	0.0
3	10	110	AP3	AP4	UDP	1000000	1024	10	1000	0.0	0.0
4	10	110	AP5	AP4	UDP	1000000	512	4	400	0.0	0.0
5	10	110	AP6	AP4	UDP	1000000	1024	10	1000	0.0	0.0
6	10	110	AP7	AP4	UDP	1000000	1024	10	1000	0.0	0.0
7	10	110	AP8	AP4	UDP	1000000	1024	10	1000	0.0	0.0
8	10	110	AP9	AP4	UDP	1000000	1024	10	1000	0.0	0.0
9	50	150	AP5	AP4	UDP	1000000	512	50	5000	0.0	0.0
10	50	110	AP2	AP4	UDP	1000000	512	20	1200	0.0	0.0
11	100	210	AP1	AP4	TCP	1000000	512	4	440	0.0	0.0
12	100	210	AP5	AP4	TCP	1000000	512	4	440	0.0	0.0
13	100	210	AP7	AP8	TCP	2000000	512	4	440	0.0	0.0
14	100	210	AP10	AP8	TCP	2000000	512	4	440	0.0	0.0
15	10	110			UDP	1000000	512	4	400	0.0	0.0

UDP Packets Reveived: 0 %

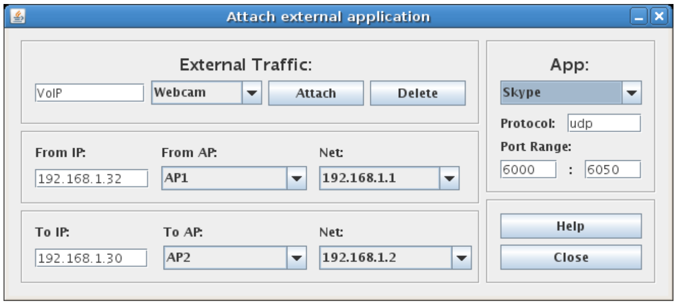
Average Throguhput: 0 Kb/s

Figure 4.9: Traffic declaration tool.

When an experiment finishes, Castadiva fills in this table with results, including throughput and, if traffic is UDP based, the percentage of packets received.

Adding External Traffic

This functionality is specially interesting in those scenarios where the user wants to study other traffic flows different that those generated by Castadiva. We know the limitations of the synthetic traffic tools that we develop for Castadiva. For this reason Castadiva also incorporates an extra functionality that allows attaching a laptop or a computer to a node. Figure 4.10 shows an example of this functionality.



External Traffic: VoIP Webcam

App: Skype

Protocol: udp

Port Range: 6000 : 6050

From IP: 192.168.1.32 From AP: AP1 Net: 192.168.1.1

To IP: 192.168.1.30 To AP: AP2 Net: 192.168.1.2

Figure 4.10: External traffic declaration.

Such functionality allows the use of laptops to generate any flow of traffic and redirect it to specific nodes of Castadiva. You can use real applications like Ekiga [69] or Skype [70] to generate a video-conference and study the behaviour of H.323 and SIP protocols in MANETs.

Figure 4.11 shows a laptop running the Ekiga application to generate a video-conference. Near to this laptop is another one used to receive the call. The webcam creates the video traffic and Castadiva redirects all traffic related to this video-conference through the emulated MANET.

Random Test 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 a different number of nodes and randomly-generated network topologies. This is achieved through the Random test window shown in Figure 4.12.

The user must 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 increase on granularity. (i. e., 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). At the top left the current scenario generated is displayed; again, all the tests can be stored in either Castadiva's or ns-2's format.

4.4 Performance Evaluation and Validation

To verify that the proposed tool behaves correctly, and to test its functionality, we have chosen a representative scenario where nodes are located so that the maximum number of hops between nodes is of five. The topology used in our evaluation is shown in Figure 4.13.

Since Castadiva is completely compatible with the ns-2 file format for scenarios, we used it to perform the comparison in a simple and straightforward manner. We selected a range of both static and dynamic MANET scenarios and compared the results obtained using Castadiva against those obtained using ns-2. We confirmed that the ns-2 simulator is a reliable tool that provides results which are comparable, although not identical, to those obtained with a more realistic MANET environment.

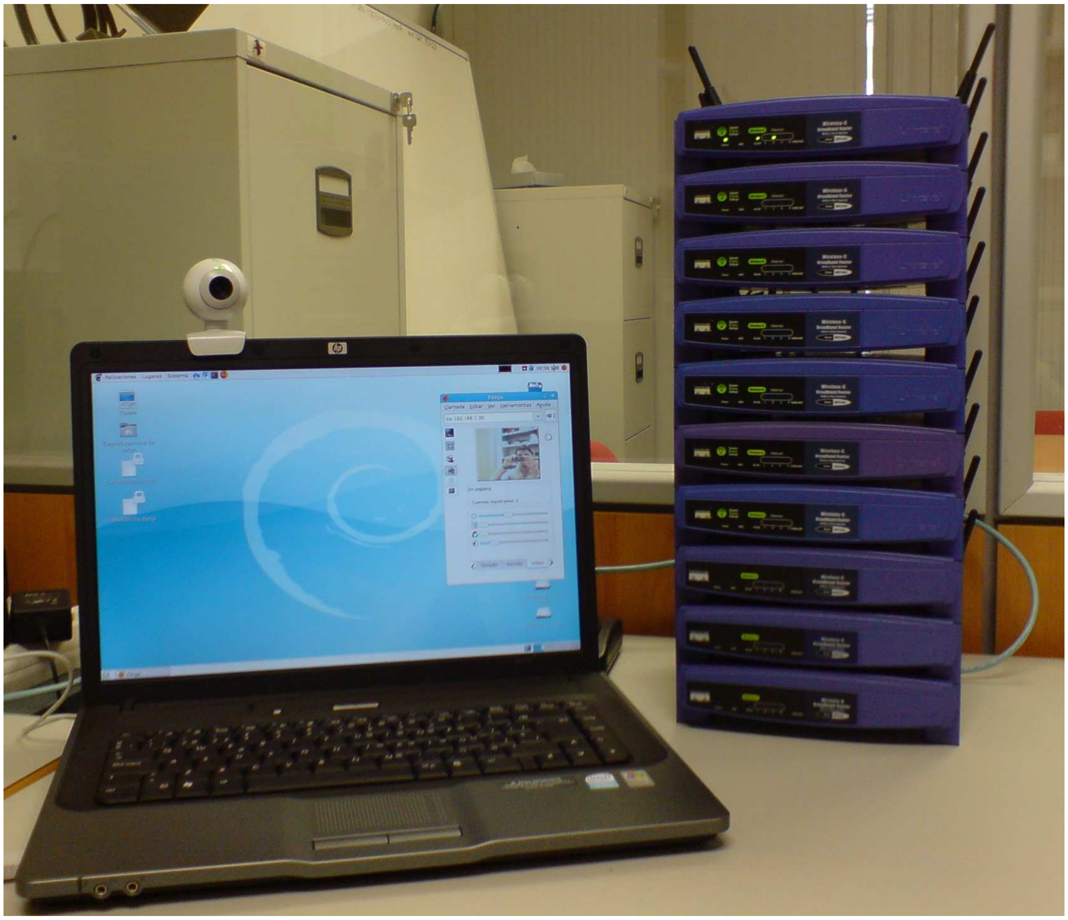


Figure 4.11: Using a laptop to add real video traffic.

4.4.1 Evaluation of Castadiva with a Static Scenario

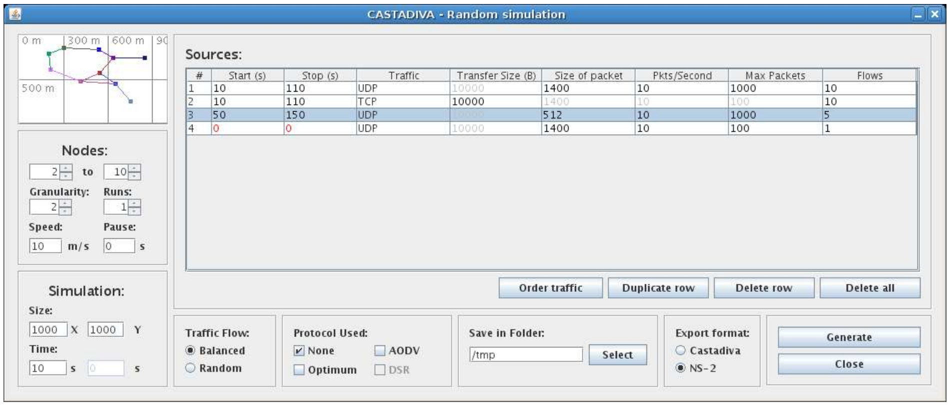
The scenario is defined in a 1000m x 700m area, and the test time is of 100 seconds. We set the wireless nodes' range equal to 250 meters. In terms of traffic, we define both UDP and TCP connections between each participating node and node 6. For TCP connections, the maximum transfer size is of 100 MB. UDP flows generate 55 packets per second, and packet size is fixed at 512 bytes.

In our first evaluation we have not selected any routing protocol. Figure 4.14 and Figure 4.15 show the results obtained in this test. 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.

On both tests we observe that Castadiva has a lower throughput than ns-2. We must take into account that the shared wireless media is prone to both transmission errors and contention among stations. In the case of ns-2, only contention effects are simulated, which explains the observed discrepancy.

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

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. Figures 4.22 and 4.17



Sources:

#	Start (s)	Stop (s)	Traffic	Transfer Size (B)	Size of packet	Pkts/Second	Max Packets	Flows
1	10	110	UDP	10000	1400	10	1000	10
2	10	110	TCP	10000	1400	10	100	10
3	50	150	UDP	10000	512	10	1000	5
4	0	0	UDP	10000	1400	10	100	1

Nodes: 2 to 10

Granularity: 2 Runs: 1

Speed: 10 m/s Pause: 0 s

Simulation:

Size: 1000 X 1000 Y

Time: 10 s 0 s

Traffic Flow: ● Balanced ○ Random

Protocol Used: ☑ None ☐ AODV ☐ Optimum ☐ DSR

Save in Folder: /tmp

Export format: ○ Castadiva ● NS-2

Figure 4.12: Random test window.

show 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 4.15), which is due to competing traffic from AP7, AP8 and AP9. For these nodes throughput decreases with increasing number of hops, as expected. With ns-2 we observe that both AP8 and AP9 suffer from starvation; one of the reasons that could explain this behaviour is that, with Castadiva, all nodes share a same medium and packet collisions between out-of-range nodes does occur since we are using emulation. Nevertheless, we consider that this issue deserves further scrutiny.

4.4.2 Evaluation of Castadiva with a Mobile Scenario.

We now define a 1000m x 700m area scenario, using simulation time of 510 seconds. 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 seconds, and packet size is fixed at 512 bytes. The traffic starts at the simulation time of 10 seconds, allowing some previous node movement to take place. In the first test, each node has a maximum speed of 5 m/s.

In an initial evaluation we have not selected any routing protocol. Figure 4.18 and Figure 4.19 show a comparison between Castadiva and ns-2 node by node. The selected scenario was generated by ns-2 and imported to Castadiva to grasp exactly the same node behaviour.

The figure shows that the obtained results are quite similar, which validates Castadiva's implementation. Since we have not selected any routing protocol, only those connections which go through directly connected nodes are successful. We also observe that Castadiva has a lower throughput than ns-2. When Castadiva is selected, 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.

We now evaluate the impact of node speed. To do that we vary node speed in different tests. Maximum node speeds of 0, 5, 10, 15 and 20 m/s are tested on Castadiva and ns-2. As for the previous test, each scenario was generated by ns-2 and imported to Castadiva to have exactly the same node movements. Figure 4.20 and Figure 4.21 show the results obtained in this test.

On 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 exaggerated that for UDP. To discover the

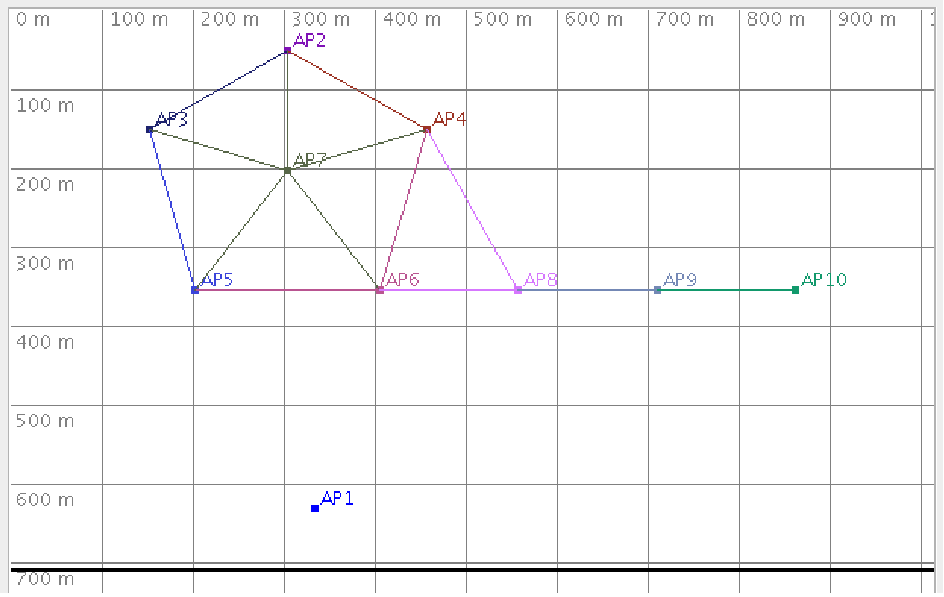


Figure 4.13: Scenario used for evaluation purposes.

reason of this, we study a controlled scenario with only two nodes and do a measure of the arriving time of each packet. With this experiment we obtain the mean delay to do a retransmission when a node recovers from a previous lost link. In ns-2 it is of five seconds. However, our routers have a mean delay of almost eight seconds. This three second difference causes ns-2 to achieve a higher throughput.

We again repeat the previous experiment with routing and forwarding enabled. We pick the OLSR option, so that Castadiva activates the OLSR protocol on each node. OLSR requires configuring five parameters: HELLO_INTERVAL, REFRESH_INTERVAL, TC_INTERVAL, MID_INTERVAL and also HNA_INTERVAL. These values are explained in the official RFC [40]. The first one represents the period of time between consecutive hello packets sent to neighbours. The purpose of such packets is populating the local link information base and the neighborhood information base. REFRESH_INTERVAL is a period of time where each neighbor node must be cited at least once. The TC_INTERVAL value represents the time at which a TC packet is sent. This packet is sent by a node in the network to declare a set of links which must include at least the links to all nodes of its MPR Selector set. MID_INTERVAL is the interval to send a MID packet, used by a device to declare its multiple interfaces. HNA_INTERVAL represents the time to send an HNA packet that provides connectivity from the OLSR MANET to non OLSR interfaces.

We use the values proposed in the RFC, shown in Table 4.2.

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

Table 4.2: Default OLSR parameters.

Figures 4.22 and 4.23 show comparison results obtained node-by-node, where each node has a maximum speed of 5 m/s.

Tests show that the average percentage of UDP packets received is increased when we use the

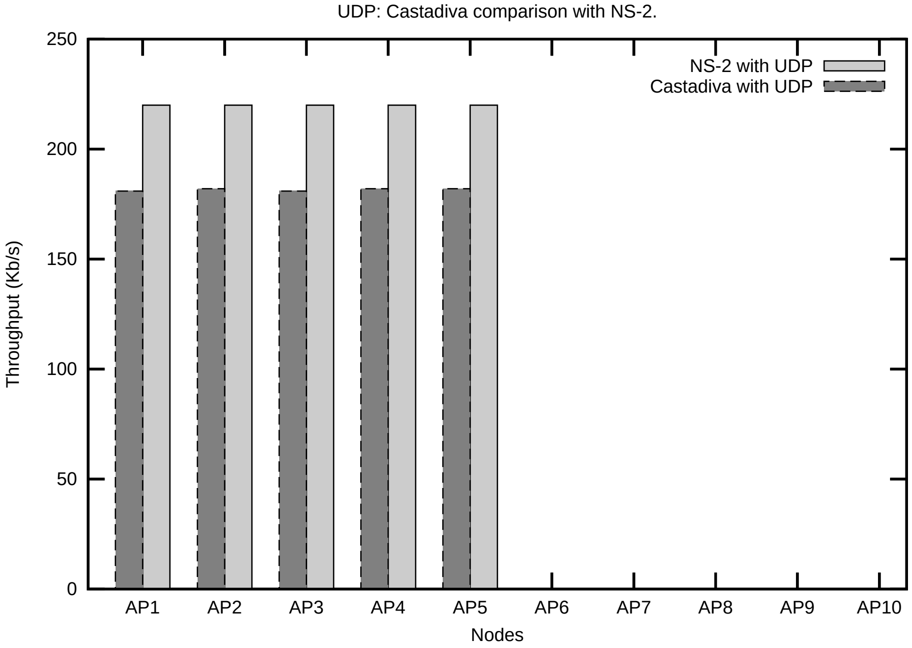


Figure 4.14: Comparison of Castadiva with ns-2 without routing using UDP traffic

routing protocol, since it allows nodes AP9 and AP10 to reach their destinations. When we study the behaviour of the network using TCP traffic, we can observe that the average throughput its not increased because in both simulations the network is saturated. However, throughput is shared between more nodes since OLSR allows nodes located more than one hop away to also send traffic to their destination, that has its interface saturated.

Figures 4.22 and 4.23 show the average percentage of packets received for maximum node speeds of 0, 5, 10, 15 and 20 m/s using the OLSR protocol.

We can observe important similarities between Castadiva and ns-2. For both protocols the behaviour of these platforms is quite similar. With Castadiva we achieve slightly less throughput than with ns-2 for the reasons explained in the test without routing.

4.5 Summary

In this chapter we have presented Castadiva, a novel architecture to improve research in the MANETs field by allowing to make real test-bed experiments.

Castadiva is a test-bed architecture based on low-cost off-the-shelf devices, which is used to test protocols developed for MANETs. Castadiva is completely compatible with the file format used by the ns-2 simulator, thereby simplifying and allowing a fair comparison between the two systems.

By using both TCP and UDP data traffic under a variety of static and dynamic MANETs scenarios we show that Castadiva is able to obtain reliable results using real wireless off-the-shelf devices. We also demonstrate that the ns2 simulator is a reliable tool for the evaluation of MANET protocols.

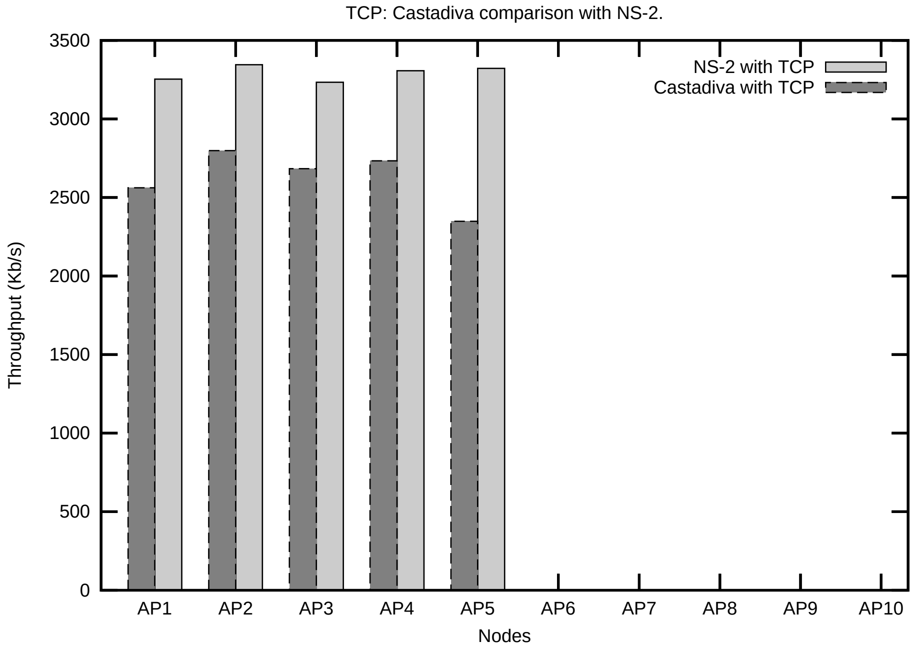


Figure 4.15: Comparison of Castadiva with ns-2 without routing using TCP traffic.

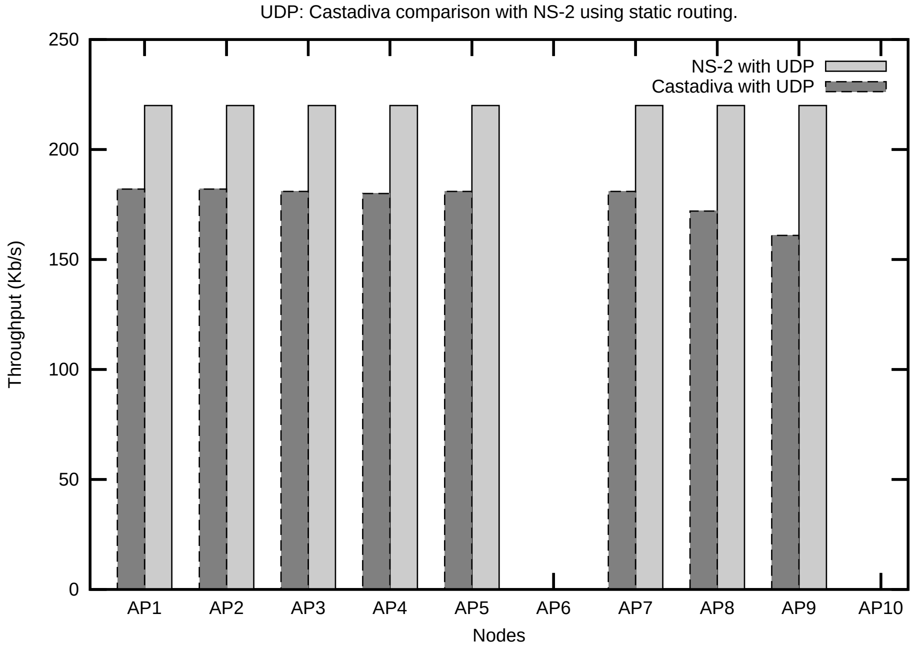


Figure 4.16: Comparison of Castadiva with ns-2 using static routing and UDP traffic.

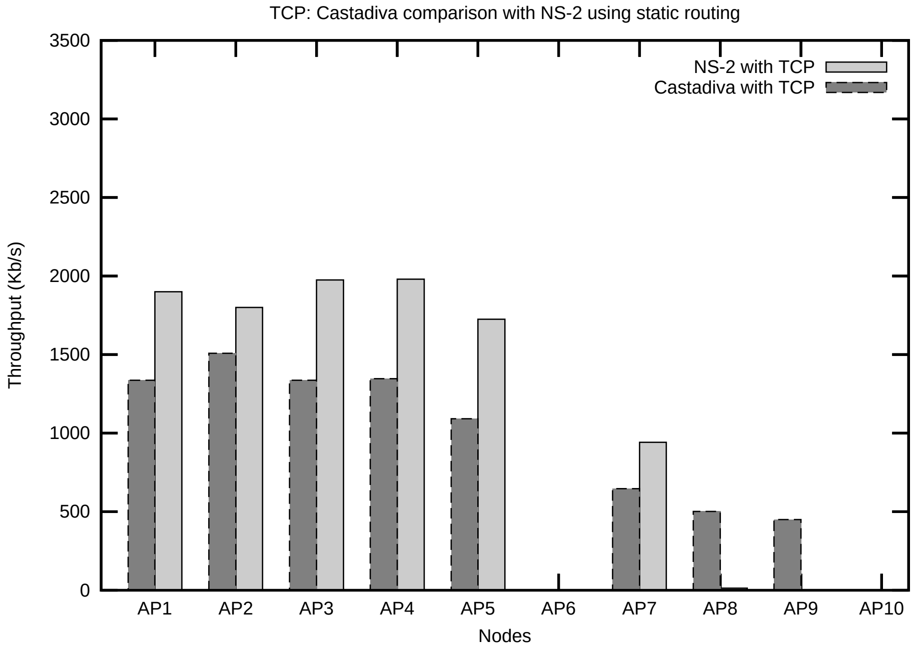


Figure 4.17: Comparison of Castadiva with ns-2 using static routing and TCP traffic.

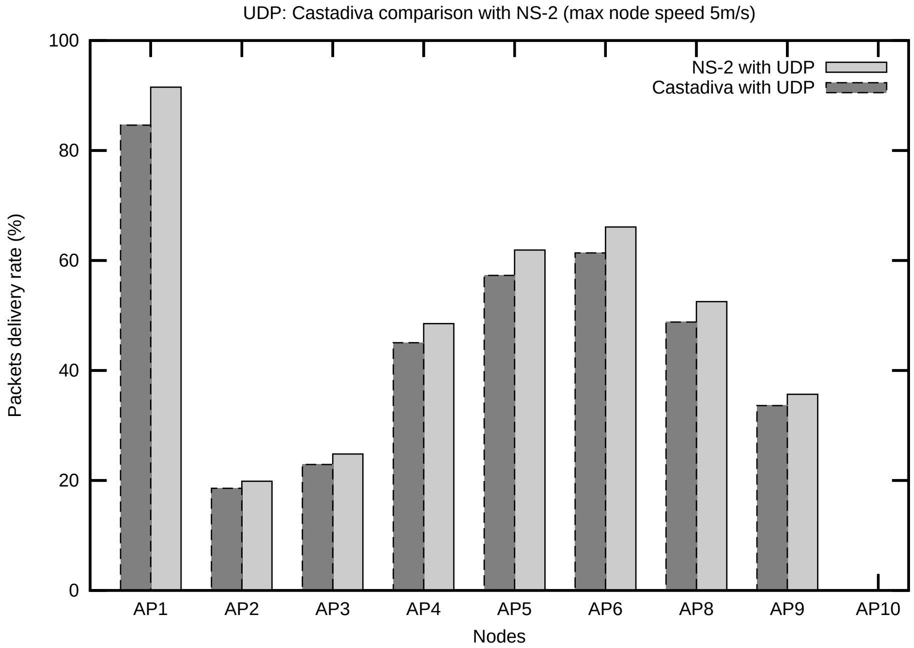


Figure 4.18: Node comparison of Castadiva with ns-2 without routing and using UDP traffic

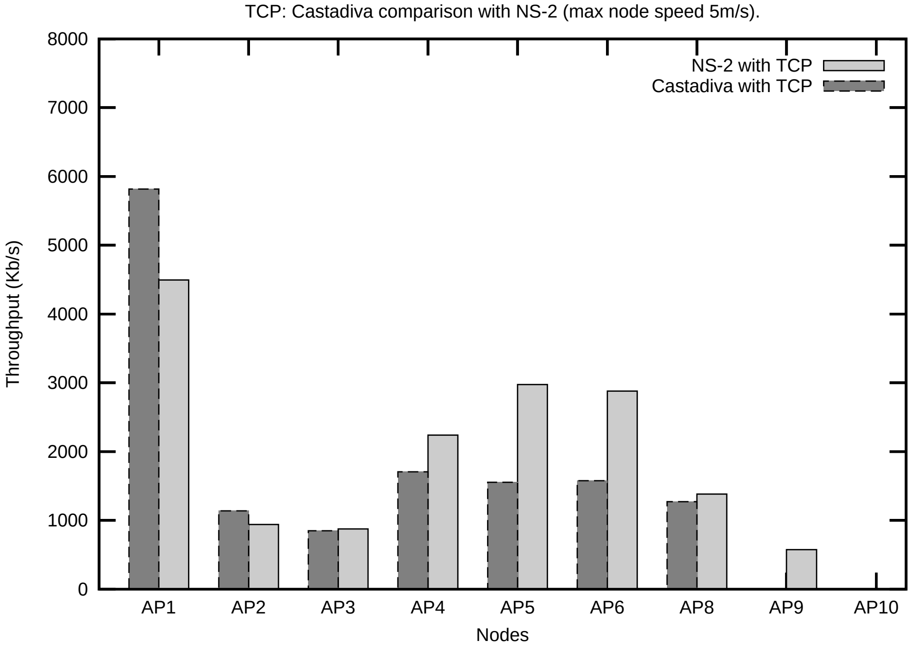


Figure 4.19: Node comparison of Castadiva with ns-2 without routing and using TCP traffic.

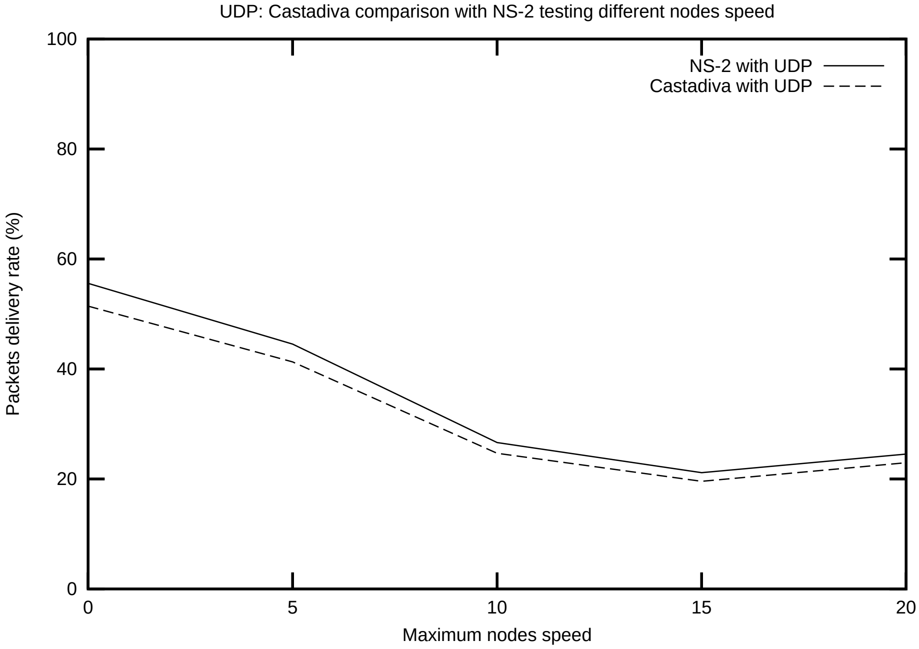


Figure 4.20: Average comparison of Castadiva with ns-2 without routing. Using UDP traffic

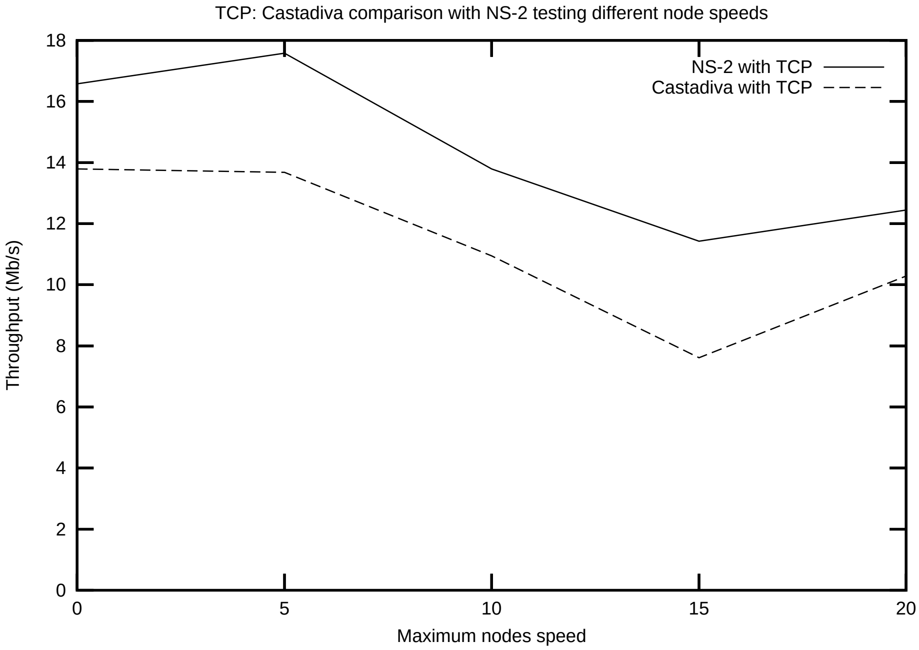


Figure 4.21: Average comparison of Castadiva with ns-2 without routing. Using TCP traffic.

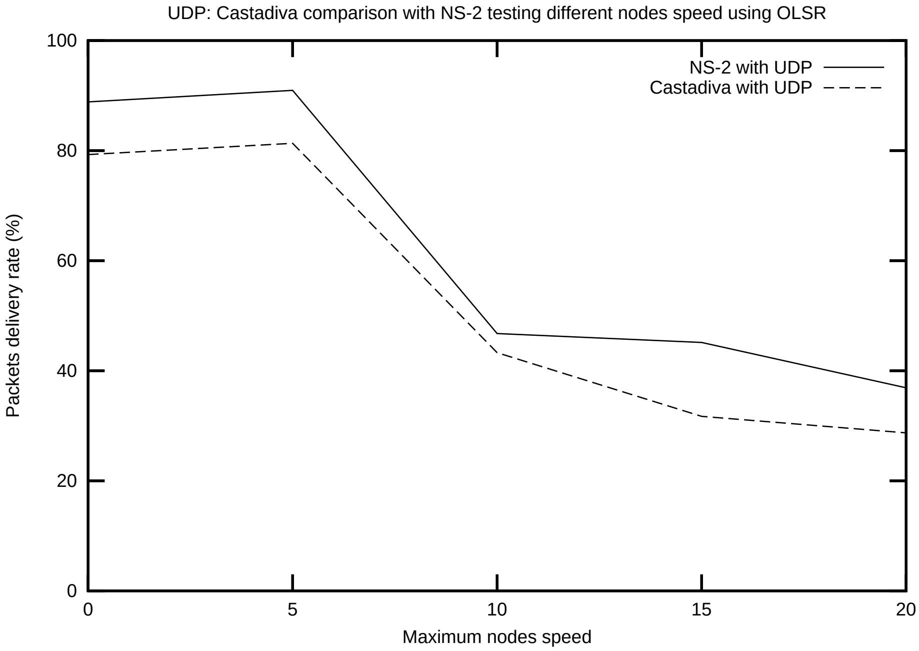


Figure 4.22: Comparison of Castadiva with ns-2 using OLSR and UDP traffic.

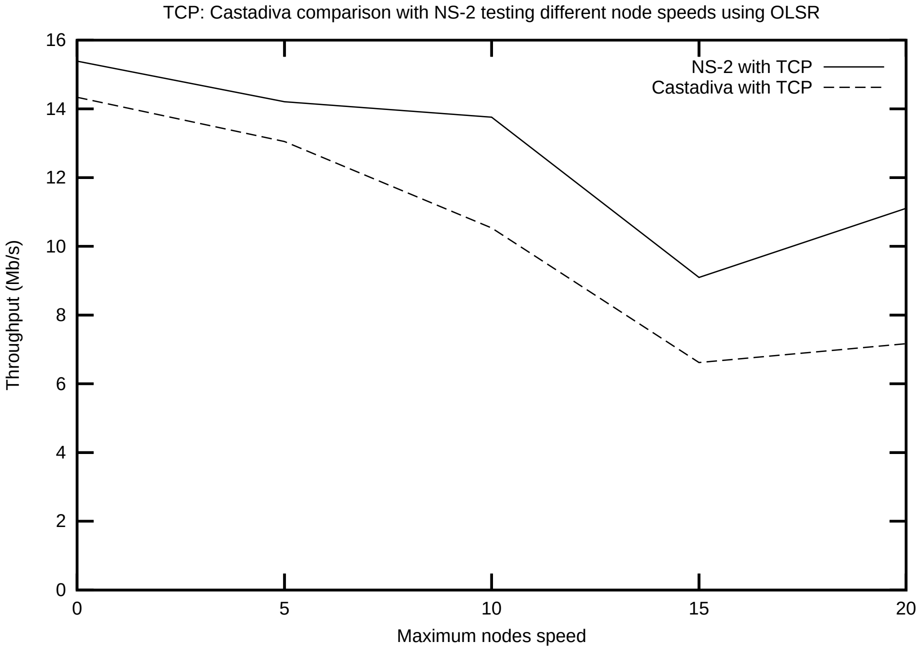


Figure 4.23: Comparison of Castadiva with ns-2 using OLSR and TCP traffic.

Chapter 5

Conclusions, Publications and Future Work

This master thesis focuses on an architecture to deploy a mesh network in rural areas - RuralNet -, an application to control the subscribers that connect to this mesh network - TocToc - and a test-bed to validate this architecture - Castadiva.

RuralNet is an experimental wireless architecture that provides Internet access to rural subscribers, offering a wide range of telecommunications services and applications. TocToc is based on captive portal technology and allows a subscriber to connect to RuralNet merely by making use of a web browser. Castadiva is a test-bed that allows validating software solutions for ad hoc networks such as RuralNet, and offers the possibility to define and test different scenarios and traffic patterns.

The main conclusions drawn from the conduced work and the issues to be addressed in future work are presented in this chapter. Section 5.1 describes the most relevant conclusions of the Thesis. Section 5.2 summarizes the contributions of this work and Section 5.3 contains some issues that should be addressed in the future.

5.1 Conclusions

In this thesis 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. After that, we deployed RuralNet in a small-scale project over a rural area of the Comunidad Valenciana in Spain. We assessed the viability of our system by providing a set of subscribers with Internet connectivity. A performance evaluation was made focusing on the throughput achieved and on the overhead imposed on the Linksys routers 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.

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. Nevertheless, further work must be done to make an in-depth study of deployment and management of mesh networks to find the most appropriate strategy to scale the RuralNet infrastructure in a transparent manner.

In this work we also presented Castadiva, 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 RuralNet and see the behaviour of this 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 result obtained in both platforms strength the affirmation of the validity of Castadiva.

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

5.2 Publications Related with the Thesis

The research work related to this master thesis has resulted in some conference papers. We now proceed by presenting a brief description of each of them. We have organized the different publications based on the chapter were the contents have been discussed.

TocToc, RuralNet and Dulenduê

[71] 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.

[72] 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 if there is not enough. We do several test with differents users and present the validity of the entire system.

[73] 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.

[74] 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 functionability of RuralNet to a developing country. In particular on the utility of this architecture to rural areas of Africa. We focus on the low cost of the chosen devices and the reduced cost of the total deployment. Also, 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.

[75] 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 last paper about RuralNet discusses the possibility to deploy 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 analyzed.

Castadiva

[76] 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 test with synthetic traffic.

[77] 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 test to compare our test-bed with the most spreaded simulator are performed, explaining the little differences between booth tools.

[78] 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 utility of a tool that can use real devices to generate simulation. For example here we use web cams to generate a demonstration of this extra functionality.

5.3 Future Work

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

- Add extra functionality to RuralNet, allowing an easy mechanism to configure all the mesh nodes of the network. For this purpose we plan to merge RuralNet with other application developed by us called MAYA. MAYA is a tool designed to configure automatically all the network parameters of each node.
- RuralNet also needs to be evaluated in real environments, we plan to make a long term study in a real scenario, to improve the prototype to become a commercial application.
- We are planned to deploy RuralNet in Mozambique, to connect three distant rural areas: Nacutxa (that already has Internet connection), Nacala, and Matibane.
- To redesign Castadiva using a modular philosophy. This option will allow the addition of extra functionality without changing the entire source code.

- Generate real traffic in Castadiva using real devices such as webcams and make an evaluation of this traffic.

- To add others technologies like Wi-Max to the test-bed. Allowing the possibility to emulate a mesh network with Castadiva.

- To implement extra nodes mobility policies. Castadiva uses the random waypoint model to emulate nodes movement. We want to add others new strategies like Manhattan model.

This work has been supported by the Spanish *Ministerio de Educación y Ciencia* under Grant TIN2005-07705-C02-01 and by the *Ajudes complementàries per a projectes de I+D+I D.Gral d'Investigació i Transferència Tecnològica*, under Grant ACOMP07/237. Starting next year, Castadiva is going to be a part of the RED project, which belongs to the European EUREKA/Celtic programme. The RED Project (Reaction after Detection) will define and design solutions in order to enhance the detection/reaction process by providing means to enhance the detection diagnosis, to allow a fast reaction through automated and/or computer aided counter measures, and to support policy based reaction. The RED consortium is formed by EADS DS SA (France), Thales Communication (France), France Telecom R&D (France), Exaprotect Technology (France), GET ENST Bretagne (France), Telindus (Luxemburg), Centre Recherche Public Henri Tudor (Luxemburg), Soluciones Globales Internet SA (Spain), Universidad Politécnica de Madrid (Spain), Universidad Politécnia de Valencia (Spain) and Innovae (Spain). It has a total budget of 9.2 millions of euros and a average effort of 83 persons per year.

Castadiva will be used to test attacks to a wireless network and their counter measures in a real network. For this reason, we must add extra functionality to simulate these attacks. Therefore, another research line to be followed is to adapt Castadiva to allow the study of security in networks, adding emulation of attacks to wireless networks and their counter-measures, comparing different secure protocols proposals as well as proposing new proposal.

Bibliography

[1] Ron Olexa. *Implementing 802.11, 802.16 and 802.20 wireless network*. Elsevier, 2005.

[2] 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.

[3] International telecommunication union. http://itu.org.

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

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

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

[7] Richard S. Wolff Mingliu Zang. Crossing the digital divide: Cost-effective broadband wireless access for rural and remote area.

[8] Jing-Hong Liew, Alvin W. Yeo, Khairuddin Ab Hamid, and Al-Khalid Othman. Implementation of wireless networks in rural areas. *Damai Sciences Sdn Bhd*, 2004. Universiti Malaysia Sarawak.

[9] C. Kenny. Information and communication technologies for direct poverty alleviation: Costs and benefits. *Development Policy Review*, 20(2):141–157, 2002.

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

[11] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic. *Mobile ad hoc networking*. IEEE Press, 2004.

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

[13] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol. *International Multi Topic Conference, Pakistan*, 2001.

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

[15] 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.

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

[17] P. Jacquet, A. Laouiti, P. Minet, and L. Viennot. Performance of multipoint relaying in ad hoc mobile routing protocols. *Networking 2002, Pise (Italy)*, 2002.

[18] 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.

[19] 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.

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

[21] J. Moy. OSPF Version 2. IETF RFC 2328, April 1998.

[22] 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.

[23] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[24] 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.

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

[26] 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.

[27] 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.

[28] G. Pei, M. Gerla, and X. Hong. Lanmar: Landmark routing for large scale wireless ad hoc networks with group mobility, 2000.

[29] 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.

[30] 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.

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

[32] 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.

[33] 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.

[34] 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.

[35] 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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] Nocat. http://nocat.net.

[42] Wifidog. http://dev.wifidog.org/.

[43] PatronSoft. Firstspot. http://www.patronsoft.com/firstspot/.

[44] USC/ISI UC Berkeley, LBL and Xerox PARC researchers. Network Simulator - ns (Version 2). Available at: http://www.isi.edu/nsnam/ns/, 1998.

[45] OPNET Technologies Inc. OPNET making networks and applications performs. Available at: http://www.opnet.com/.

[46] 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.

[47] 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.

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

[49] R. Barr, Z. J. Haas, and R. Van Renesse. JiST: embedding simulation time into a virtual machine. In *Proceedings of EuroSim 2004*, September 2004.

[50] R. Barr, Z. J. Haas, and R. Van Renesse. Scalable wireless ad hoc network simulation. *Handbook on Theoretical and Algorithmic Aspects of Sensor*, pages 291–311.

[51] R. Barr and Z. J. Haas. JiST/SWANS. Available at: http://www.cs.cornell.edu/barr/repository/jist/, 2004.

[52] 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.

[53] 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.

[54] 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).*

[55] 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.

[56] A. Karygiannis and E. Antonakakis. mLab: a mobile ad hoc network test bed. National Institute of Standards and Technology.

[57] Y. Zhang and W. Li. An integrated environment for testing mobile ad hoc networks. Available at: http://www.wins.hrl.com/projects/adhoc.

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

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

[60] S. McCanne, V. Leres, and V. Jacobson. The tcpdump manual page. Lawrence Berkeley Laboratory, 6 89.

[61] OpenWRT, wireless freedom. Available at: http://openwrt.org.

[62] Paul Dourish Thomas P. Moran. Introduccion to this special issue on context aware computing. http://hci-journal.com/editorial/si-context-aware-intro.pdf.

[63] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94)*, 1994.

[64] Dey and Anind K. Understanding and using context. *Personal Ubiquitous Computing*, 5, 2001.

[65] A. Yamanaka and JCraft Inc. JSch, the Java secure channel. Available at: http://www.jcraft.com/jsch/.

[66] Uppsala Universitet. AODV-UU. Available at: http://core.it.uu.se/core.

[67] Thomas Clausen, Philippe Jacquet, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized link state routing protocol. Internet Draft, MANET Working Group, draft-ietf-manet-olsr-07.txt, July 2002. Work in progress.

[68] T. M. H. Reenskaug. The model-view-controller (mvc) its past and present. In *Java Zone*, Oslo, September 2003.

[69] Damien Sandras. Ekiga (GnomeMeeting. Available at: http://ekiga.org/.

[70] Skype Limited. Skype. Available at: http://skype.com.

[71] 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.

[72] 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.

[73] 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.

[74] 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.