Department of Information Engineering, University of Pisa, Italy

Ph.D. Thesis - XVIII Cycle

Routing in multi-hop Ad Hoc networks: an Experimental Approach

Ph.D. Candidate

Eleonora Borgia

Advisors

Giuseppe Anastasi

Enrico Gregori

February 2006

Contents

List of Figures

List of Tables

1 Introduction

Mobile Ad Hoc networks (MANET) consist of groups of self-organizing mobile nodes that communicate over the wireless medium without any form of pre-existent infrastructures. They are no centralized networks where, on the contrary, communication and operations are distributed over all nodes taking part to MANET. Typical applications of such networks are conference events, disaster recovery and military operations. In particular, in the past few years, the imposition of new technologies such as IEEE 802.11 [IEE99] and Blueooth [Blu00] facilitated the development and the growth of many civilian applications, allowing the users to connect to each other sharing information, also guaranteeing their mobility. In fact, nodes belonging to a MANET may move rapidly and unpredictably, resulting in a possibly dynamic network topology. As consequence, the network should be also able to react to the frequent topological changes in a fast manner. To enable multi-hop communication in a distributed manner, all nodes also act as *routers* for each other, guaranteeing that nodes not in direct communication can exchange data.

The characteristics and the highly dynimic nature of mobile Ad Hoc network make routing area the most active research area within the MANET domain. In fact, Internet routing approaches are not appropriate for MANET, while special routing protocols are required. Hence, they should be designed taking into account these peculiar charcteristics of the network but also the coinstrains existing on the node resources as poor devices, limit bandwith and high bit error rate. Thus, the goal for a routing protocol is to minimize the network traffic in favour of the useful data traffic. At the same time it has to be capable to adapt to links failures/additions due to nodes mobility. Moreover, it has to work in a completely distributed way, and be self starting and self organizing.

Several routing protocols have been proposed in the last ten years [CCL03a] [Bel03], but they can be typically divided into two main categories: **proactive** routing protocols and **reactive** (on-demand) routing protocols [BT99]. Proactive routing protocols, derived for legacy Internet distance-vector and link-state protocols, maintain consistent and updated routing information for every pair of network nodes by propagating

route updates at fixed time intervals. Reactive on demand routing protocols, on the other hand, establish the route to a destination only when there is a demand for it and maintain only routes towards nodes with active communications.

In this thesis we investigate the behaviour and the efficiency of routing protocols for MANET adopting an experimental approach. In fact, in current MANET reaserch most of them have been evaluated and compared through simulations, see for example [DPR00] [DCY00]. Simulators allow the performance evaluation of protocols in different scenarios, defined by varying several parameters (e.g. number of nodes, mobility models, data traffic); however they often introduce simplifying assumptions (e.g., radio propagation model) that mask important characteristics of the real protocols behavior, see for example the so-called "communication gray zones" problem [LNT02]. To avoid these modeling approximations, it is necessary to complement simulation with experiments on a real Ad Hoc network. This work provides a contribution in this direction. In fact only few measurements studies on real Ad Hoc test-beds can be found in literature, see e.g., [Dep] [GKN+04a]. To this aim, we set up a MANET prototype implementing a full ad hoc network architecture, and we report our experiences and results obtained by these measurements on a real Ad Hoc network.

This work started from the study of a single-hop Ad Hoc network where we investigated the behaviour of IEEE 802.11 protocol. Understanding the real behaviour of the MAC protocol and its real interaction with the environment represent the first step of a complete analysis of an Ad Hoc network. Reaveling phenomenon usually neglected in simulation studies as those decribed in [ABC+05] [ABCG05] [ABCG04] [ABCG03] allows to explain and solve routing problems. Then we focused on multi-hop Ad Hoc network analysing performance of routing protocols. In particular, we selected two robust available implementations of routing protocols for MANET, specifically OLSR [CJ03] and AODV [PR03], and we compared then in different scenarios and environments, also investigating their reaction to nodes mobility. We started from a small-scale network of 2/4-hop size up to 8 nodes [Bor05] [BCDP05] [BCDG05] and we extended the experimental analysis to a medium-scale network of 7/8-hop size involving up to 23 nodes [BCDP06] [D16]. To the best of our knowledge this represents one of the largest testbed on multi-hop Ad Hoc networks. Our experimental results highlight that, in contrast with MANET community, the use of a proactive protocol does not penalize the system performance. These results encourage to identify a routing protocol suitable for multi-hop networks in terms of scalability, performance and efficiency in the class of proactive protocol. As proved in [SMSR02] [SSR01], the

Hazy Sighted Link State (HSLS) routing protocol [SR01] exhibits good performance in term of scalability. In particular, in HSLS routing information is propagated to the network nodes with a frequency that decreases with the distance using a binary exponential sequence. As a result, each node builds a "self-centered" topology view, which becomes *hazy* as the distance grows. In the framework of this thesis, an enhanced version of the HSLS routing protocol has been designed and developed, adding *i)* a mechanism to guarantee the reliabilty of LSU packets with any introduction of additional control overhead, and *ii)* a module that allows cross-layer interactions, thus resulting in an easy integration with the cross-layer prototype. The basic functionality of HSLS module has been successfully tested in network of 4-5 nodes [D10].

The rest of the thesis is organized as follows. Chapter 2 analyzes the performance of IEEE 802.11 on single-hop Ad Hoc network. A description of the main solutions exiting for MANET routing protocols is introduced in Chapter 3. Chapter 4 focuses on the routing protocols selected for the experimental analysis (OLSR and AODV) describing their main functionality in details. The experimental comparison between OLSR and AODV on small-scale and medium-scale Ad Hoc networks is reported in Chapters 5 and 6, respectively. The further optimized HSLS routing protocol for MANET is presented in Chapter 7. Chapter 8 concludes the thesis with lessons learned and future works.

2 Single-Hop Ad Hoc network

In this chapter we investigate the performance of Ad Hoc network based on IEEE 802.11 technology. Specifically, we focus on single-hop networks of two or four stations and we analyze the characteristics of the wireless medium by means of an experimental study. Understanding the real behaviour of the MAC protocol and its real interaction with the environment represents the first step of a complete analysis of the tecnology, from the MAC layer to the application layer. Besides reaveling phenomenon usually neglected in simulation studies, the experimental bottom-up approach guarantees to construct robust protocol layers since they are based on information from lower layers (e.g. routing problems could be explained and solved with a better knowledge of the MAC layer). This results in a stable and efficient technology, thus offering a usable product to final users.

2.1 Introduction

Self-organizing wireless networks are nowadays one of the hottest topics in the area of pervasive-computing. The research community is devoting lot of effort in designing protocols to support the Mark Weiser's pervasive networking vision in wich 802.11-based devices have emerged as the de-facto standard technology for investigating ad hoc networks.

The vast majority of works on wireless networks rely on simulation models for evaluations, the main reason being the ease of development and reproducibility with respect to real experiments. However, relying just on simulations may be misleading. Specifically, it is well known that accurately modeling the signal propagation on a wireless medium is a hard task. Unfortunately, an accurate model is often required to correctly evaluate the effectiveness of higher-layer protocols. For example, [GKN+04a], [TJB01] show that the performances of routing protocols (e.g., AODV, DSR) highly depend on the physical-layer model used in simulations. In some cases, simulation results are extremely different from experimental measurements. Furthermore, the relative comparison among couples of protocols can be completely swapped by changing the

physical-layer model. These observations remind that simulation models and outcomes should be *validated against experimental measurements*.

These remarks are the main motivation for the work presented. Specifically, we report the main results from a wide measurement study focused on 802.11 networks. The emphasis of the work is on characterizing key networking features such as the maximum communication distance between a couple of nodes, and the interactions between concurrent transmitting nodes. We study the effect on the communication distance of several environmental parameters (e.g., humidity, distance from ground), providing quantitative evidence of their impact. We also account for the effect of technology-dependent parameters, such as the bit rate. We find that communication distance of 802.11 nodes significantly varies with the data rate, and we sketch possible side effects on routing protocols. Then, we study the effect of concurrent transmitters on each other. Since 802.11 technology adopts a CSMA/CA MAC protocol, the Physical Carrier Sensing mechanism determines the interaction between concurrent senders. Our measures show that Physical Carrier Sensing - and, thus, the dependence among couples of transmitters - extends far beyond the maximum communication distance. Roughly, the maximum Physical Carrier Sensing distance is (at least) twice as large as the maximum communication distance. Based on these measurements we provide a channel model for 802.11 devices. Finally, we exploit the channel model definition to elaborate on the well-known hidden and exposed node problems. The formulations currently reported in computer networking handbooks do not take into consideration the effect of Physical Carrier Sensing beyond the communication distance. Due to the large extension of Physical Carrier Sensing, we find that these formulations should be significantly revised. Hence, we provide novel formulations, which comply with the measurement outcomes.

2.2 IEEE 802.11 Architecture and Protocols

In this section we present the IEEE 802.11 architecture and protocols as defined in the original standard [IEE99], with a particular attention to the MAC layer. Later, in Section 2.4, we will emphasize the differences between the 802.11b standard with respect to the original 802.11 standard.

The IEEE 802.11 standard specifies both the MAC layer and the Physical Layer (see Figure 2.1). The MAC layer offers two different types of service: a contention free service provided by the *Distributed Coordination Function* (DCF), and a contention-free service implemented by the *Point Coordination Function* (PCF). These service

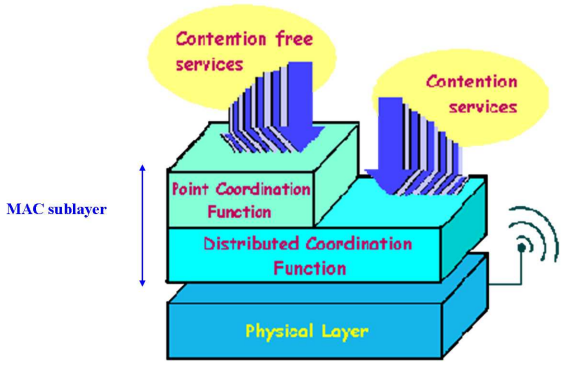


Figure 2.1: IEEE 802.11 Architecture

types are made available on top of a variety of physical layers. Specifically, three different technologies have been specified in the standard: Infrared (IF), Frequency Hopping Spread Spectrum (FHSS) and Direct Sequence Spread Spectrum (DSSS). The DCF provides the basic access method of the 802.11 MAC protocol and is based on a *Carrier Sense Multiple Access with Collision Avoidance* (CSMA/CA) scheme. The PCF is implemented on top of the DCF and is based on a polling scheme. It uses a *Point Coordinator* that cyclically polls stations, giving them the opportunity to transmit. Since the PCF can not be adopted in ad hoc mode, it will not be considered hereafter.

According to the DCF, before transmitting a data frame, a station must sense the channel to determine whether any other station is transmitting. If the medium is found to be idle for an interval longer than the *Distributed InterFrame Space* (DIFS), the station continues with its transmission[1] (see Figure 2.2). On the other hand (i.e., if the medium is busy), the transmission is deferred until the end of the ongoing transmission. A random interval, henceforth referred to as the *backoff time*, is then selected, which is used to initialize the *backoff timer*. The backoff timer is decreased for as long as the channel is sensed as idle, stopped when a transmission is detected on the channel, and reactivated when the channel is sensed as idle again for more than a DIFS (for example, the backoff timer of "Other Station" in Figure 2.2 is disabled while "Source" and "Destination" are transmitting their frame; the timer is reactivated a DIFS after

[1]To garantee fair access to the shared medium, a station that has just transmitted a frame and has another frame ready for transmission must performthe backoff procedure before initiating the second transmission

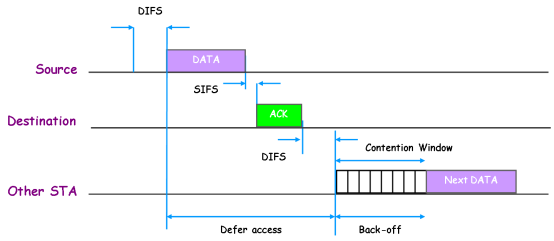


Figure 2.2: 802.11 Basic Access Mechanism

"Destination" has completed its transmission). The station is enabled to transmit its frame when the backoff timer reaches zero. The backoff time is slotted. Specifically, the backoff time is an integer number of slots uniformly chosen in the interval (0, *CW*-1). *CW* is defined as the Backoff Window, also referred to as *Contention Window*. At the first transmission attempt $CW{=}CW_{min}$, and it is doubled at each retransmission up to CW_{max}. In the standard CW_{min} and CW_{max} values depend on the physical layer adopted. For example, for the FHSS Phisical Layer CW_{min} and CW_{max} values are 16 and 1024, respectively [IEE99]. Obviously, it may happen that two or more stations start transmitting simultaneously and a collision occurs. In the CSMA/CA scheme, stations are not able to detect a collision by hearing their own transmission (as in the CSMA/CD protocol used in wired LANs). Therefore, an immediate positive acknowledgement scheme is employed to ascertain the successful reception of a frame. Specifically, upon reception of a data frame, the destination station initiates the transmission of an acknowledgement frame (ACK) after a time interval called *Short InterFrame Space* (SIFS). The SIFS is shorter than the DIFS (see Figure 2.2) in order to give priority to the receiving station over other possible stations waiting for transmission. If the ACK is not received by the source station, the data frame is presumed to have been lost, and a retransmission is scheduled. The ACK is not transmitted if the received frame is corrupted. A *Cyclic Redundancy Check* (CRC) algorithm is used for error detection. After an erroneous frame is detected (due to collisions or transmission errors), a station must remain idle for at least an *Extended InterFrame Space* (EIFS) interval before it reactivates the backoff algorithm. Specifically, the EIFS shall be used by the DCF whenever the physical layer has indicated to the MAC that a frame transmission was begun that did not result in the correct reception of a complete MAC frame with a correct FCS value. Reception of an error-free frame during the EIFS re-synchronizes the station to the actual busy/idle state of the medium, so the EIFS is terminated and normal medium access (using DIFS and, if necessary, backoff)

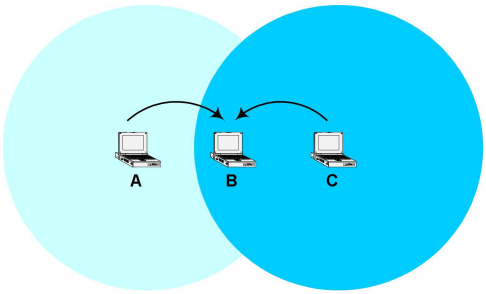


Figure 2.3: The "hidden station" problem

continues following reception of that frame.

2.3 Common Problems In Wireless Networks

In this section we shortly discuss the main problems that arise in wireless ad hoc networks. A detailed discussion can be found in [ACG03]. The characteristics of the wireless medium make wireless networks fundamentally different from wired networks. Specifically, as indicated in [IEE99]:

- the wireless medium has neither absolute nor readily observable boundaries outside of which stations are known to be unable to receive network frames;
- the channel is unprotected from outside signals;
- the wireless medium is significantly less reliable than wired media;
- the channel has time-varying and asymmetric propagation properties.

In wireless (ad hoc) networks that rely upon a carrier-sensing random access protocol, like the IEEE 802.11, the wireless medium characteristics generate complex phenomena such as the **hidden station** and the **exposed station** problems. Figure 2.3 shows a typical "hidden station" scenario. Let us assume that station B is in the transmitting range of both A and C, but A and C can not hear each other. Let us also assume that A is transmitting to B. If C has a frame to be transmitted to B, according to the DFC protocol, it senses the medium and finds it free because it is not able to hear A's transmissions. Therefore, it starts transmitting the frame but this transmission will results in a collision at the destination Station B.

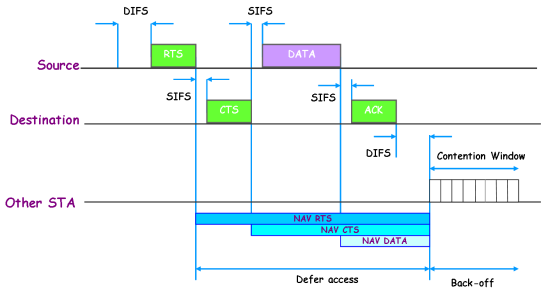


Figure 2.4: Virtual Carrier mechanism

The hidden station problem can be alleviated by extending the basic mechanism by a **virtual carrier sensing** mechanism (also referred to as floor acquisition mechanism) that is based on two control frames: *Request To Send* (RTS) and *Clear To Send* (CTS), respectively. According to this mechanism, before transmitting a data frame, the source station sends a short control frame, named RTS, to the receiving station announcing the upcoming frame transmission (see Figure 2.4). Upon receiving the RTS frame, the destination station replies by a CTS frame to indicate that it is ready to receive the data frame. Both the RTS and CTS frames contain the total duration of the transmission, i.e., the overall time interval needed to transmit the data frame and the related ACK. This information can be read by any station within the transmission range of either the source or the destination station. Such a station uses this information to set up a timer called *Network Allocation Vector* (NAV). While the NAV timer is greater than zero the station must refrain from accessing the wireless medium. By using the RTS/CTS mechanism, stations may become aware of transmissions from hidden station and on how long the channel will be used for these transmissions. Figure 2.5 depicts a typical scenario where the "exposed station" problem may occur. Let us assume that Station A and Station C can hear transmissions from B, but Station A can not hear transmissions from C. Let us also assume that Station B is transmitting to Station A and Station C receives a frame to be transmitted to D. According to the DCF protocol, C senses the medium and finds it busy because of B's transmission. Therefore, it refrain from transmitting to D although this transmission would not cause a collision at A. The "exposed station" problem may thus result in a throughput reduction.

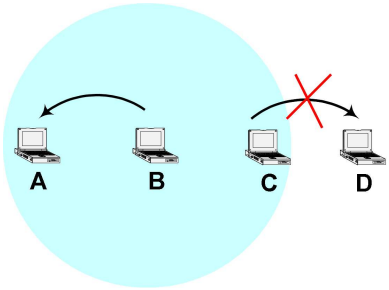


Figure 2.5: The "exposed station" problem

2.3.1 Simulation Analysis of IEEE 802.11 Ad Hoc Networks

The performance provided by the 802.11 MAC protocol in an ad hoc environment have been extensively analyzed via simulation. The studies presented in the literature have been pointed out several performance problems. They can be summarized as follows. In a dynamic environment, mobility may have a severe impact on the performance of the TCP protocol [HV99], [HV02], [CRVP01], [LS01], [LS02], [AAS00], [DB01]. However, even when stations are static, the performance of an ad hoc network may be quite far from ideal. It is highly influenced by the operating conditions, i.e., TCP parameter values (primarily the congestion window size) and network topology [FZX+02], [LBDC+01]. In addition, the interaction of the 802.11 MAC protocol (hidden and exposed station problems, exponential back-off scheme, etc.) with TCP mechanisms (congestion control and time-out) may lead to unexpected phenomena in a multi-hop environment. For example, in the case of simultaneous TCP flows, severe unfairness problems and - in extreme cases - capture of the channel by few flows [TG99], [XS01], [XS02], [XBLG02] may occur. Even in the case of a single TCP connection, the instantaneous throughput may be very unstable [XS01], [XS02]. Such phenomena do not appear, or appear with less intensity, when the UDP protocol is used. All these previous analysis were carried out using simulation tools (GloMosim [Glo], *ns-2* [Ns0], Qualnet [Qua], etc.), and thus the results observed are highly dependent on the physical layer model implemented in the simulation tool. In addition, these results have been obtained by assuming the IEEE 802.11 original standard operating at 2 Mbps. Currently, however, the IEEE 802.11b is the *de facto* reference technology for ad hoc networking. Therefore, in our experimental analysis, we investigate the performance of the IEEE 802.11b when operating in ad hoc mode. Our results point out that several important aspects of IEEE 802.11b are not considered in the previous studies.

2.4 Experimental Analysis of 802.11b Ad Hoc Networks

The 802.11b standard extends the 802.11 standard by introducing a higher-speed Physical Layer in the 2.4 GHz frequency band still guaranteeing the interoperability with 802.11 cards. Specifically, 802.11b enables transmissions at 5.5 Mbps and 11 Mbps, in addition to 1 Mbps and 2 Mbps. 802.11b cards may implement a dynamic rate switching with the objective of improving performance. To ensure coexistence and interoperability among multirate-capable stations, and with 802.11 cards, the standard defines a set of rules that must be followed by all stations in a WLAN. Specifically, for each WLAN is defined a *basic rate* set that contains the data transfer rates that all stations within the WLAN will be capable of using to receive and transmit.

To support the proper operation of a WLAN, all stations must be able to detect control frames. Hence, RTS, CTS, and ACK frames must be transmitted at a rate included in the basic rate set. In addition, frames with multicast or broadcast destination addresses must be transmitted at a rate belonging to the basic rate set. These differences in the rates used for transmitting (unicast) data and control frames has a big impact on the system behavior as clearly pointed out in [Eph02].

Actually, since 802.11b cards transmit at a constant power, lowering the transmission rate permits the packaging of more energy per symbol, and this makes the transmission range increasing. In the next sections we investigate, by a set of experimental measurements,

i) the relationship between the transmission rate of the wireless network interface card (NIC) and the maximum throughput (two-stations experiments);

ii) the communication zone and the carrier sensing zone of a node S and their relationship with the transmission rate (two-stations experiments);

iii) Hidden and/or exposed station situations (four-stations experiments).

To better understand the results presented below, it is useful to provide a model of the relationships existing among stations when they transmit or receive. In particular, it is useful to make a distinction between the transmission range, the interference range and the carrier sensing range. The following definitions can be given.

- The *Transmission Range* (TX_{range}) is the range (with respect to the transmitting station) within which a transmitted frame can be successfully received. The transmission range is mainly determined by the transmission power and the radio propagation properties.

- The *Physical Carrier Sensing Range* (PCS_{range}) is the range (with respect to the transmitting station) within which the other stations detect a transmission. It mainly depends on the sensitivity of the receiver (the receive threshold) and the radio propagation properties.

- The *Interference Range* (IF_{range}) is the range within which stations in receive mode will be "interfered with" by a transmitter, and thus suffer a loss. The interference range is usually larger than the transmission range, and it is function of the distance between the sender and receiver, and of the path loss model.

In the previous simulation studies the following relationship has been generally assumed: $TX_{range} \leq IF_{range} \leq PCS_{range}$. For example, in the ns-2 simulation tool [Ns0] the following values are used to model the characteristics of the physical layer: $TX_{range} = 250$m, $IF_{range} = PCS_{range} = 550$m. In addition, the relationship between TX_{range}, PCS_{range}, and IF_{range} are assumed to be constant throughout a simulation experiment. On the other hand, from our measurements we have observed that the physical channel has time-varying and asymmetric propagation properties and, hence, the value of TX_{range}, PCS_{range}, and IF_{range} may be highly variable in practice.

2.5 Experimental Environment

Before analysing the system performance, the description of the experimental environment is required. The 802.11b measurement test-bed is based on an Ad Hoc network made up of laptops with different capabilities running the Linux operating system and equipped with the D-Link DWL-650 wireless cards compliant to IEEE 802.11b standard. Since the main goal is investigating the main features of the 802.11b standard, we consider only single-hop static scenarios, i.e., network where all the stations do not change position during the experiments and where communicating stations are within their transmission range. Thus, we set up the ad hoc network in an open field of about 350 meters long. There were no phisical obstacles (e.g., buildings, trees) among nodes, thus each pair of adjacent nodes was in line of sight. This guarantees that other phenomenon can interfere with the performed experiments, e.g. link failures. The specific setup of each experiment (e.g., number of used nodes, type of traffic, data rate) is postponed to the beginning of each paragraph.

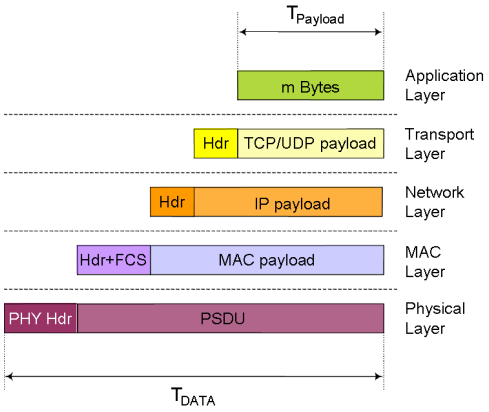


Figure 2.6: Encapsulation overhead

2.6 Available Bandwidth

In this section we analyze the maximum throughput offered by the MAC protocol of 802.11b. In particular we show that only a fraction of the 11 Mpbs nominal bandwidth of the IEEE 802.11b cards can be used for data transmission. This is performed through an analytical model validated against experiments outcomes.

To this end we need to carefully analyze the overheads associated with the transmission of each packet (see Figure 2.6). Specifically, each stream of m bytes generated by a legacy Internet application is encapsulated by the TCP/UDP and IP protocols that add their headers before delivering the resulting IP datagram to the MAC layer for transmission over the wireless medium. Each MAC data frame is made up of: *i)* a *MAC header*, say MAC_{hdr}, containing MAC addresses and control information[2], and *ii)* a variable length data payload, containing the upper layers data information. Finally, to support the physical procedures of transmission (carrier sense and reception) a *physical layer preamble* (PLCP preamble) and a *physical layer header* (PLCP header) have to be added to both data and control frames. Hereafter, we will refer to the sum of PLCP preamble and PLCP header as PHR_{hdr}. Note that these different headers and data fields are transmitted at different data rates to ensure the interoperability between 802.11 and 802.11b cards. Specifically, the standard defines two different formats for the PLCP: Long PLCP and Short PLCP. Hereafter, we assume a Long PLCP that includes a 144-bit preamble and a 48-bit header both transmitted

[2]Without any loss of generality we have considered the *frame error sequence* (FCS), for error detection, as belonging to the MAC header.

at 1 Mbps while the MAC_{hdr} and the $MAC_{payload}$ can be transmitted at one of the NIC data rates: 1, 2, 5.5, and 11 Mbps. In particular, control frames (RTS, CTS and ACK) can be transmitted at 1 or 2 Mbps, while data frame can be transmitted at any of the NIC data rates.

By taking into considerations the above quantities, Equation 2.1 defines the maximum expected throughput for a single active session (i.e., only a sender-receiver couple active) when the basic access scheme (i.e., DCF and no RTS-CTS) is used. Specifically, Equation 2.1 is the ratio between the time required to transmit the user data and the overall time the channel is busy due to this transmission:

$$Th_{noRTS/CTS} = \frac{m}{DIFS + T_{DATA} + +SIFS + T_{ACK} + \frac{CW_{min}}{2} * Slot\_Time} \quad (2.1)$$

where

m is the number of bytes generated by the application.

T_{DATA} is the time required to transmit a MAC data frame using one of the NIC data rate, i.e., 1, 2, 5.5 or 11 Mbps; this includes the PHY_{hdr}, MAC_{hdr}, $MAC_{payload}$ and FCS bits for error detection.

T_{ACK} is the time required to transmit a MAC ACK frame; this includes the PHY_{hdr}, and MAC_{hdr}.

$\frac{CW_{min}}{2} * Slot\_Time$ is the average back off time.

When the RTS/CTS mechanism is used, the overheads associated with the transmission of the RTS and CTS frames must be added to the denominator of (1). Hence, in this case, the maximum throughput $Th_{RTS/CTS}$, is defined as

$$Th_{RTS/CTS} = \frac{m}{DIFS + T_{RTS} + T_{CTS} + T_{DATA} + T_{ACK} + 3 * SIFS + \frac{CW_{min}}{2} * Slot\_Time} \quad (2.2)$$

where T_{RTS} and T_{CTS} indicate the time required to transmit the RTS and CTS frames, respectively.

Equation 2.1 and 2.2 are used to obtain the theoretical throughput for a single session with UDP traffic. Indeed, when using the TCP protocol, overheads due to the TCP_ACK transmission must be considered. More precisely, the technique of cumulative ACK answering to two consecutive TCP_DATA is used. Thus, a TCP handshake is composed by TCP_DATA1, TCP_DATA2 and TCP_ACK. Figure 2.7 shows the TCP handshake on the channel when using the basic acess mechanism and the RTS/CTS mechanism, respectively. In particular, DATA1 and DATA2 packets

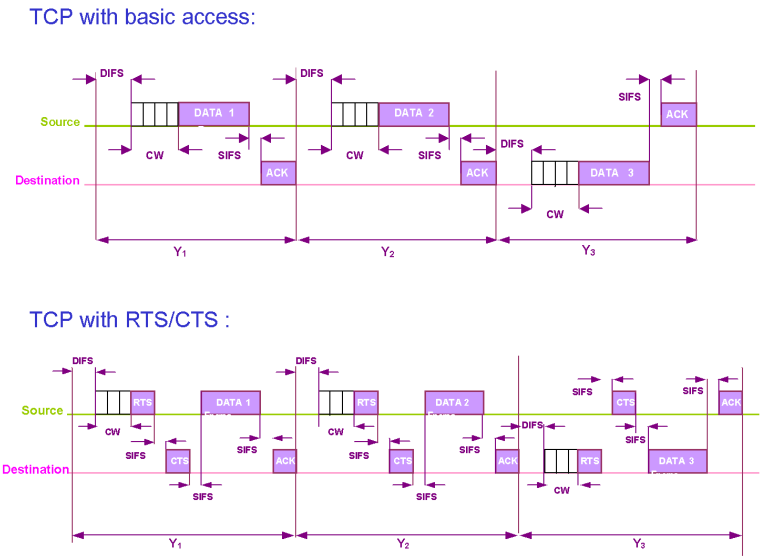


Figure 2.7: TCP handshake with the basic and the RTS/CTS mechanism

$Slot\_Time$	τ	PHY_{hdr}	MAC_{hdr}	$BitRate(Mbps)$
$20\mu sec$	$\leq 1\mu sec$	192 bits ($9.6t_{slot}$)	272 bits	1, 2, 5.5, 11
$DIFS$	$SIFS$	ACK	CW_{min}	CW_{max}
$50\mu sec$	$10\mu sec$	112 bits + PHY_{hdr}	32 t_{slot}	1024 t_{slot}

Table 2.1: IEEE 802.11b parameter values

are obtained by the encapsulation of TCP_DATA1 and TCP_DATA2, instead DATA3 is obtained by the encapsulation of the TCP_ACK. The theoretical throughput is the ratio between the time to transmit two user data frame and the overall time for the complete transmission on the channel:

$$Th = \frac{2 * m}{y_1 + y_2 + y_3} \tag{2.3}$$

where y_i represents the time required to the transmission of the DATAi packet on the channel.

The numerical results presented in the next sections depend on the specific setting of the IEEE 802.11b protocol parameters. Table 2.1 gives the values for the protocol parameters used hereafter.

In Table 2.2 and Table 2.3 we report the expected throughputs (with and without

	m = 512 Bytes		m = 1024 Bytes	
	No RTS/CTS (Mbps)	RTS/CTS (Mbps)	No RTS/CTS (Mbps)	RTS/CTS (Mbps)
11 Mbps	3.337	2.739	5.120	4.386
5,5 Mbps	2.490	2.141	3.428	3.082
2 Mbps	1.319	1.214	1.589	1.511
1 Mbps	0.758	0.738	0.862	0.839

Table 2.2: Maximum throughputs in Mbps at different data rates for a UDP connection

	m = 512 Bytes		m = 1024 Bytes	
	No RTS/CTS (Mbps)	RTS/CTS (Mbps)	No RTS/CTS (Mbps)	RTS/CTS (Mbps)
11 Mbps	2.456	1.979	4.015	3.354
5,5 Mbps	1.931	1.623	2.858	2.507
2 Mbps	1.105	0.997	1.423	1.330
1 Mbps	0.661	0.620	0.796	0.766

Table 2.3: Maximum throughputs in Mbps at different data rates for a TCP connection

the RTS/CTS mechanism), for different data rates, for a UDP and TCP connection, respectively. These results are computed by applying Equations 2.1 and 2.2 and 2.3, and assuming a data packet size at the application level equal to m=512 or m=1024 bytes. As shown in the tables, only a small percentage of the 11 Mbps nominal bandwidth can be really used for data transmission. Obviously, this percentage increases with the payload size. However, even with large packets sizes (e.g., m=1024 bytes) the bandwidth utilization is in the order of 46% for UDP traffic and 36% for TCP traffic.

The above theoretical analysis has been complemented with measurements of the actual throughput for a single connection at the application level in a real environment. Specifically, we have considered two types of applications: `ftp` and CBR. In the former case the TCP protocol is used at the transport layer, while in the latter case the UDP is adopted. In both cases the applications operate in asymptotic conditions (i.e., they always have packets ready for transmission) with constant size packets of 512 bytes. The results obtained from this experimental analysis are reported in Figure 2.8.

The experimental results related to the UDP traffic are very close to the maximum throughput computed analytically. On the other hand, in the presence of the TCP protocol the measured throughput is much lower than the theoretical maximum throughput, as expected. Similar results have been also obtained by comparing the maximum throughput derived according to 2.1 and 2.2, and the real throughputs measured when the NIC data rate is set to 1, 2 or 5.5 Mbps.

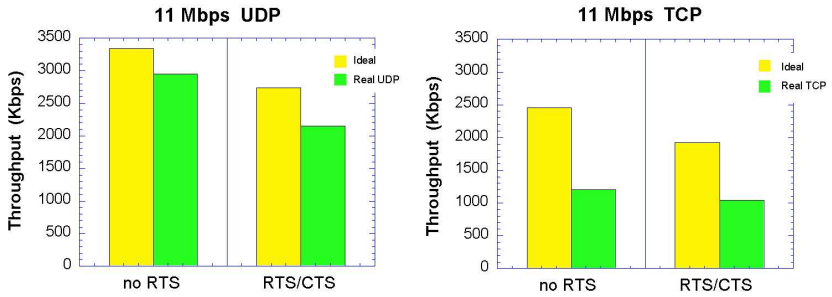


Figure 2.8: Comparison between the teoretical throughput and the actual throughput achieved by TCP/UDP connections.

2.7 Communication Zone

The goal of this section is to characterize the "communication zone" of a sending node S, meaning the zone around S where other nodes can receive S's transmissions. Mostly, we are interested in understanding which is the maximum communication distance (TX_{range}) at which a receiver can correctly receive S transmissions.

Several works in the literature highlight that the shape of the communication zone greatly depends on the environment where nodes are placed [ABB+04], [GKN+04a]. To have a reference point, we firstly try to avoid measurements biasing by environment parameters. To this end, we collect a first set of measures by using a couple of nodes - say S and R -, where S is the sender and R the receiver. S and R communicate in open space to avoid influence of obstacles. Experiments are run in sunny days, as humidity has a great impact on the communication distance (see below). In addition, nodes are placed high enough to avoid signal reflections on ground and antennas are oriented so as to maximize their performance in connecting S and R. We place S and R at variable distance from each other, and we measure the probability of R to correctly receive a packet sent by S.

2.7.1 Ideal Environment

The dependency between the data rate and the transmission range was investigated by measuring the packet loss rate experienced by two communicating stations whose network interfaces transmit at a constant (preset) data rate. Specifically, four sets of measurements were performed corresponding to the different data rates: 1, 2, 5.5, and

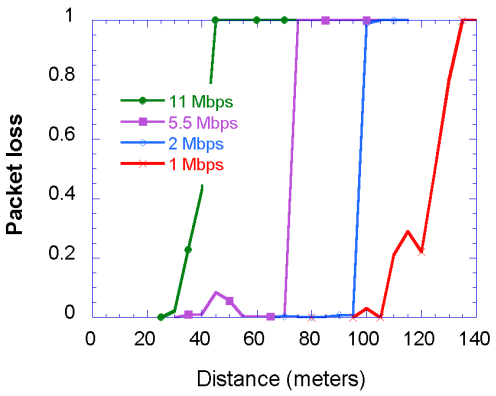


Figure 2.9: Packet loss rate as a function of the distance between two communicating stations for different data rates.

	11 Mbps	5.5 Mbps	2 Mbps	1 Mbps
Data TX_{range}	30m	70m	90-100m	110-130m
Control TX_{range}			$\approx$ 90m	120m

Table 2.4: Estimates of the transmission ranges at different data rates.

11 Mbps. In each set of experiments the packet loss rate was recorded as a function of the distance between the communicating stations. Fig 2.9 reports the behaviour for each data rate defined in the 802.11b standard. Taking into account that we define the maximum communication distance as the point where the packet loss drops above 25%, the Table 2.4 summerizes the transmission estimes for each data rates.

Results plotted in Figure 2.9 are interesting in many respects. Specifically:

i) some gray-zone phenomenon can be observed also in this case, mainly at 1 Mbps data rate;

ii) the maximum communication distance is larger for lower data rates. This is intuitive, since at low data rates more energy is packed with each bit transmitted, and hence transmissions can travel further away;

iii) the maximum communication distance changes significantly with the data rate (see table 2.4).

Point *iii)* above has two very important consequences. First of all, it is interesting to compare the communication distance used in the most popular simulation tools, like ns-

2 and Glomosim/Qualnet, with the outcomes of our experiments. In these simulation tools a communication distance equal to 250 m and 376 m is assumed, respectively. Since the above simulation tools only consider a 2 Mbps bit rate we refer to the communication distance estimated for the 2-Mbps data rate. As it clearly appears, the value used in the simulation tools (and, hence, in the simulation studies based on them) is 2-3 times higher than the values measured in practice. This difference is very important for example when studying the behavior of routing protocols: the shorter is the communication distance, the higher is the frequency of route re-calculation when the network nodes are mobile. Clearly, the maximum communication distance depends on the transmission power. Our results are obtained by setting the transmission power to 15 dBm.

The large difference in communication distances at different data rates has another important side effect. It is worth recalling that, to allow interoperability with legacy 802.11 nodes, different MAC-level frames are transmitted at different rates by 802.11b nodes. For example, control frames such as RTS, CTS and ACK are typically transmitted at 1 or 2 Mbps, irrespective of the data rate used to transmit data frames. Therefore, assuming that the RTS/CTS mechanism is active, if a node transmits a data frame at 11 Mbps to another node within its transmission range (i.e., less then 30 m apart) it reserves the channel for a radius of approximately 90 (120) m around itself. Such kind of behaviors may severely impact routing-protocols performance, as shown in [LNT02].

2.7.2 Non-Ideal Environment

The setup of the experiments presented so far is quite optimistic, since it limits as much as possible factors that may reduce the signal propagation such as obstacles, humidity, etc. In this section we show much the maximum communication distance is affected by two environment parameters, i.e., the humidity, and the nodes' height from ground. In these experiments, the other test-bed parameters are as in the optimal configuration.

Figure 2.10 highlights the influence of humidity on the communication distance. Specifically, it shows the difference between the packet reception probability experienced by 802.11 nodes transmitting at 1 Mbps in a sunny and a cloudy day, respectively. It clearly emerges that humidity plays a substantial role in determining nodes' communication distance. The decrease of the communication distance in humid environments is caused by water particles that interact with electromagnetic waves and absorb part

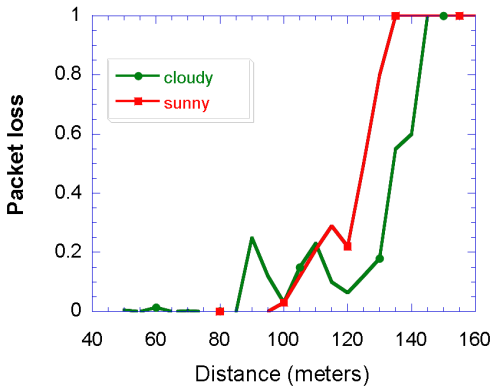


Figure 2.10: Communication distance in humid environment.

of their energy causing signal attenuation.

While running the experiments presented so far, we observed a dependence of the communication distance on the mobile devices' height from the ground. Specifically, in some cases we observed that, while the nodes were not able to communicate when located on stools, they started to exchange packets by lifting them up. Thus, we decided a careful investigation of this phenomenon. Figure 2.11 plots the achieved throughput as a function of ohe nodes'height from ground, in the case of 802.11 nodes. In particular, the measures are collected by placing the nodes, S and R, 30 m apart, and by setting the data rate to 11 Mbps and 2 Mbps. In general, when S and R are within the corresponding maximum communication distance in both cases, the packet reception probability should be close to 100%. On the contrary, Figure 2.11 shows that the communication distance depends on the nodes' height from ground, and is reduced when nodes' height is low. The work presented in [GO02] provides a theoretical framework to explain this phenomenon. The analytical model predicts that - in our test-bed configuration - effects related to ground reflections disappear if the distance from ground of 802.11 nodes is greater than 0.97 m. Results from this analytical framework are thus aligned with our measurements.

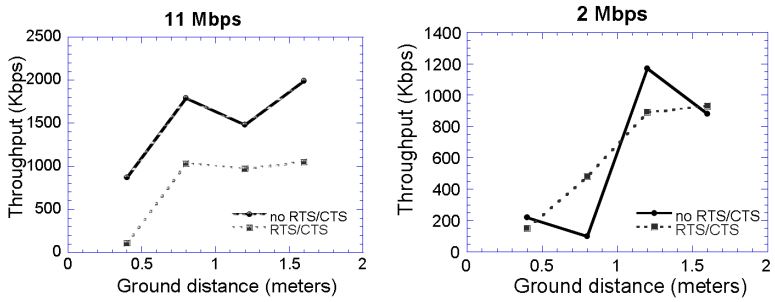


Figure 2.11: Relationship between throughput and devices's height.

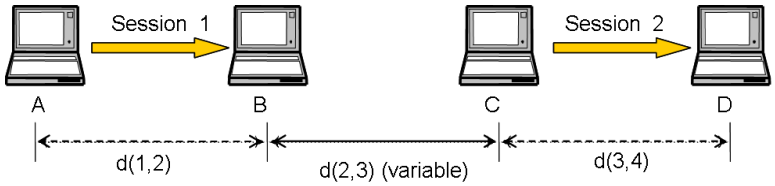


Figure 2.12: Reference network scenario.

2.8 Phisical Carrier sensing Zone

In the previous section we have analyzed the networking features of 802.11 devices in terms of communication distance. This analysis is not sufficient to derive the channel models for the reference technology. The wireless medium has neither absolute nor readily observable boundaries outside of which nodes are known to be unable to sense signal. Therefore, due to the carrier sensing nature of the MAC protocols used by 802.11, couples of nodes may interact also at a distance far greater than the maximum communication distance. In this section we investigate the extent of the Physical Carrier Sensing zone, i.e., we measure the maximum distance at which a node A senses the channel busy due to an ongoing transmission of a node B. A direct measure of this quantity seems difficult to achieve because it is not possible to have information about the channel carrier sensing. Therefore, we define an indirect way to perform these measurements. We utilize the scenario shown in Figure 2.12. Nodes A and C are the senders, while node B and D are the receivers. The distance between each sender-receiver couple is fixed ($d(A,B) = d(C,D) = 10m$), while the distance between the two couples (i.e., $d(B,C)$) is variable. All the other test-bed parameters are as in the optimal configuration. We increase optimal $d(B,C)$ until no correlation is

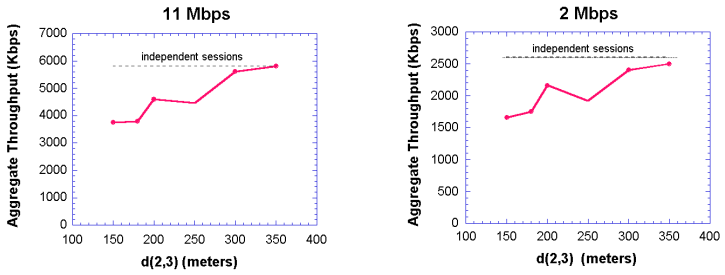


Figure 2.13: Sessions' throughput as function of distance.

measured between the couples of nodes. To quantify the correlation degree between the two sessions we measure (at the application-level) the throughput of each session in isolation, i.e., when the other session is not active. Then we measure the throughput achieved by each session when both sessions are active. Obviously, no correlation exists when the aggregate throughput is equal to the sum of the throughput of the two sessions in isolation. Figure 2.13 shows the results from our measurements. Specifically, the aggregated throughput experienced by the two sessions in isolation, and while concurrently running, is plotted. To show that, as expected, the Carrier Sensing is independent of the data rate, we replicate the 802.11 experiments by setting the data rate at 11 Mbps and 2 Mbps (see Figure 2.13). As it clearly appears from Figure 2.13, there are two steps in the 802.11 aggregate throughput: one after 180 m and the other after 250 m. This behavior can be explained as follows. Taken a session as a reference, the presence of the other session may have two possible effects on the performance of the reference session: 1) if the two sessions are within the same Physical Carrier Sensing zone, they share the same physical channel; 2) if they are outside of the Physical Carrier Sensing zone of each other, the radiated energy from one session may still affect the quality of the channel observed by the other session. As the radiated energy may extend over unlimited distances, we can expect that the second effect completely disappears only for very large distances among the sessions [Eph02]. Hence, we can assume that the first step in both graphs of Figure 2.13 coincides with the end of the Physical Carrier Sensing zone, while the second one occurs when even the second effect becomes almost negligible. Note that the extent of the Physical Carrier Sensing zone is almost the same for the two different transmission rates. The Physical Carrier Sensing mainly depends on two parameters: the nodes' transmitting power and the distance between transmitting nodes. The rate at which data are transmitted has no effect on these parameters. Based on the above results,

a very interesting outcomes can be drawn: the Physical Carrier Sensing zone extends for at least twice as much as the communication zone.

2.9 Channel Model for 802.11b Network

The results shown in Sections 2.7 and 2.8 allow us to derive a very interesting channel structure observed in 802.11. This technology presents a zone around the sender where the packet reception probability is high and pretty stable. Beyond the maximum communication distance, a gray zone exists, where the packet reception probability drops towards 0 in a somehow random way. Finally, a pretty large zone exists where the packet reception probability is 0, but carrier sensing is active. Based on these remarks it is possible to define a channel model as shown in Figure **??**. Specifically, given a node S trasmitting with a rate x ($x \in \{1, 2, 5.5, 11\}$), nodes around it can be partiioned into four classes depending on their distance, d, from S:

1. stations within the communication zone ($d < TX_{range}(x)$) are able to correctly receive transmissions from S;

2. stations beyond the communication zone but within the gray zone may correctly receive transmissions from S; nodes close to each other in this region may experience completely different qualities of the link with S;

3. station beyond the gray zone but within the Physical Carrier Sensing zone ($TX_{range}(x) < d < PCS_{range}$) are not able to correctly receive transmissions from S; however, when S is transmitting they observe the channel busy and thus they defer their transmissions;

4. stations beyond the Physical Carrier Sensing zone ($PCS_{range} > d$) do not measure any significant energy on the channel when S is transmitting, therefore they can start transmitting contemporarily to S; however, the quality of the channel they observe may be affected by the energy radiated by S.

In addition, in case 4, if $d < PCS_{range} + TX_{range}(x)$ some interference phenomena may occur (see below). This interference depends on the IF_{range} value. This value is difficult to model and evaluate since it depends on several factors (mainly the power at the receiving site). Adopting our experiment-based channel model leads to very interesting remarks. For example, once this channel model is assumed, the traditional formulations of the hidden and exposed node problems do not hold anymore. The

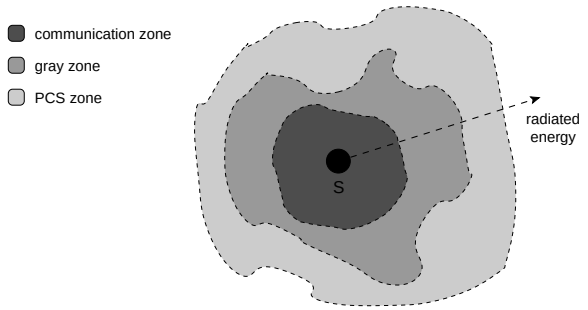


Figure 2.14: 802.11 Channel model.

hidden station phenomenon, as it is usually defined in the literature (see Section 2.3), is almost impossible with the ranges measured in our experiments. Indeed, the PCS_{range} is more than twice $TX_{range}(1)$, i.e., the larger transmission range. Furthermore, two stations, say S1 and S2, that can start transmitting towards the same receiver, R, must be at a distance $\leq 2 * TX_{range}(1)$, and thus they are inside the physical carrier sensing range of each other. Hence, if S1 has an ongoing transmission with R, S2 will observe a busy channel and thus will defer its own transmission. This means that, in this scenario, virtual carrier sensing is not necessary and the RTS/CTS mechanism only introduces additional overhead.

While the hidden station phenomenon, as defined in the literature, seems not relevant for this environment point iiii above highlights that packets cannot be correctly received due to the interference caused by a station that is "hidden" to the sending station. An example of this type of *hidden station phenomenon* is presented in Figure 2.15. In this figure we have two transmitting stations, S and S1 that are outside their respectively PCS_{range} and hence they are hidden to each other. In addition we assume that the receiver of station S (denoted by R in the figure) is inside the interference range (IF_{range}) of station S1. In this scenario S and S1 can be simultaneously transmitting and, if this occurs, station R cannot receive data from S correctly. Also in this case the RTS/CTS mechanism does not provide any help and new coordination mechanisms need to be designed to extend the coordination in the channel access beyond the PCS_{rRange}. Note that, in our channel model, the exposed station definition (see Figure 2.16) must be modified too. In this scenario, exposed stations are those stations at a distance $PCS_{Range} - TX_{Range}(1) < d < PCS_{Range}$. Indeed, these stations are exposed to station S transmissions, while they are in the transmission range of stations with $d > PCS_{Range}$. The following example outline problems

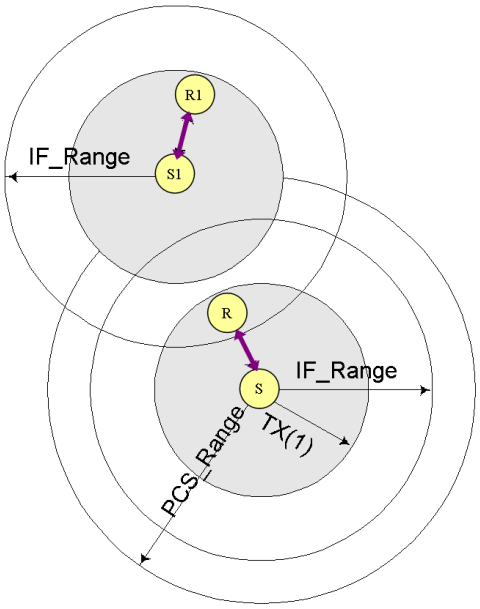


Figure 2.15: Interference-based hidden station phenomenon.

that may occur in this case. Let us denote with S1 a station at a distance d from S: $PCS_{Range} < d < PCS_{Range} + TX_{Range(x)}$. Station S1 can start transmitting, with a rate x, towards a station E that is inside the physical carrier sensing of S; station E cannot reply because it observes a busy channel due to the ongoing station S transmissions, i.e., E is exposed to station S. Since station S1 does not receive any reply (802.11 ACK) from E, it assumes an error condition (collision or CRC error condition), hence it backoffs and then tries again. If this situation repeats for several times (up to 7), S1 assumes that E is not anymore in its transmission range, gives up the transmission attempt and (wrongly) signals to the higher layer a link breakage condition, thus forcing higher layers to attempt a recovery action (e.g., new route discovery, etc. - see Section 2.3.1).

To summarize, results obtained in the configuration we analyzed indicate that the hidden station and exposed station definitions must be extended. These new hidden-station and exposed-station phenomena may produce undesirable effects that may degrade the performance of an ad hoc network, mainly if the TCP protocol is used. Extending the coordination in the channel access beyond the PCS_{Range} seems to be the correct direction for solving the above problems. This may be achieved by cross-layer interactions among a link-state routing protocol and the MAC layer. For example,

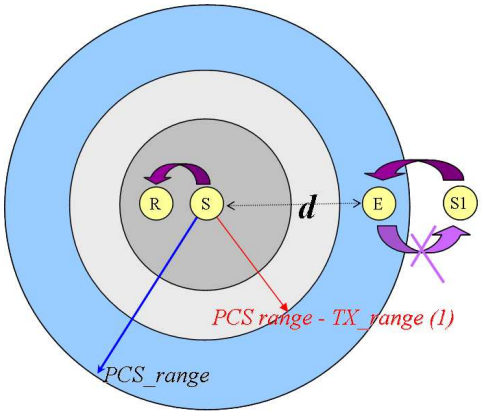


Figure 2.16: Interference-based exposed station phenomenon.

periodic link-state advertisements sent by C might be exploited to spread information about the channel load C is experiencing. The MAC layer of B and D may use this information to tune the CSMA algorithm. The interested reader is referred to [D10] for details.

2.10 Conclusions

In this chapter we have characterized several key networking features of Ad Hoc networks based on 802.11 technology. We have adopted an experimental approach, since real measurements are strongly required to understand the actual behavior of wireless networks. The experimental results presented have confirmed that basing wireless network models on experiments, and validating simulation outcomes against experimental results, is necessary to derive reliable conclusions about wireless network behavior.

First of all, we have analyzed the communication zone ($TX_{range}(x)$), i.e., the maximum distance at which two nodes are able to correctly detect transmissions of each other. This part of the analysis has shown that several assumptions that are commonly used in simulation and analytical models should be carefully revised. For example, the dependence of the communication distance on the physical data rate is typically not modeled in 802.11 simulations. Furthermore, common simulation models assume communication distances far greater than what we measured in reality. Among other effects, this may lead to unreliable evaluations of routing protocol performances.

A second set of experiments has been devoted to analyze the Physical Carrier Sensing

zone (PCS_{range}), i.e., the zone around a sending node within which another node senses the channel busy. Interestingly, we have found that this zone is at least twice as large as the communication zone. Based on these measurements, we have defined an innovative wireless link model for 802.11 devices.

The model we have derived from experimental results is quite different from traditional wireless network models. Specifically, no sharp boundary exists between the region (around a sending node) where packets can be correctly received, and the region where packets are not received at all. Instead, a pretty large "gray zone" exists, where the packet reception probability is almost unpredictable. Finally, a large Carrier-Sensing zone extends outside the gray zone. Experiments have also shown that the shape of these zones (i.e., the communication zone, the gray zone, and the Carrier-Sensing zone) is not a perfect sphere around the sender, but is quite irregular, and depends on several environment and node-configuration parameters. We believe that using such a realistic channel model in simulation and analytical evaluations is key to clearly understand wireless network performances. For example, the traditional hidden and exposed node formulations has to be revised once our model is assumed.

3 Routing

The highly dynamic nature of a mobile ad hoc network results in frequent and unpredictable changes of network topology, adding difficulty and complexity to routing among the mobile nodes. The challenges and complexities, coupled with the critical importance of routing protocol in establishing communications among mobile nodes, make routing area the most active research area within the MANET domain. In particular, to be efficient a routing protocol for MANET has to meet some requirements:

- minimize the control overhead due to the routes creation and maintenance;
- adapt dynimically to the network changes;
- minimize the limited available resources such bandwith and power devices;
- minimize the processing overhead in term of complexity of algorithm and allocated resources.

Taking into account these targets, the MANET Community has proposed numerous routing protocols and algorithms, studing and comparing their performance under various network environments and traffic conditions. Several surveys and comparative analysis of MANET routing protocols have been published [BT99], [Bel03]. [Per00] provides a comprehensive overview of routing solutions for ad hoc network, while an updated and in depth analysis of routing protocols for mobile ad hoc network is presented in [Bel03].

A preliminary classification of the routing protocols can be done via the type of cast property, i.e. whether they use a **Unicast**, **Geocast**, **Multicast**, or **Broadcast** forwarding [PK]. Broadcast is the basic mode of operation over a wireless channel; each message transmitted on a wireless channel is generally received by all neighbors at one-hop from the sender. The simplest implementation of the broadcast operation to all network nodes is by naive flooding, but this may cause the broadcast storm problem due to redundant re-broadcast [NTCS99]. Schemes have been proposed to alleviate this problem by reducing redundant broadcasting. [SW03] surveys existing methods

for flooding a wireless network intelligently. Unicast forwarding means a one-to-one communication, i.e., one source transmits data packets to a single destination. This is the largest class of routing protocols found in ad hoc networks. Multicast routing protocols come into play when a node needs to send the same message, or stream of data, to multiple destinations. Geocast forwarding is a special case of multicast that is used to deliver data packets to a group of nodes situated inside a specified geographical area. Nodes may join or leave a multicast group as desired; on the other hand, nodes can only join or leave a geocast group only by entering or leaving the corresponding geographical region. From an implementation standpoint, geocasting is a form of restricted broadcasting: messages are delivered to all the nodes that are inside a given region. This can be achieved by routing the packets from the source to a node inside the geocasting region, and then applying a broadcast transmission inside the region. Position-based (or location-aware) routing algorithms, by providing an efficient solution for forwarding packets towards a geographical position, constitute the basis for constructing geocasting delivery services. Hereafter, we surveyed the characteristics of unicast routing protocols, while a comprehensive analysis of MANET routing protocols can be found in [Bel03] [CCL03b].

3.1 Unicast Routing Protocols for MANET

A primary goal of unicast routing protocols is the correct and efficient route establishment and maintenance between a pair of nodes, so that messages may be delivered reliably and in a timely manner. MANET routing protocols are typically subdivided into two main categories: **proactive** routing protocols and **reactive** on-demand routing protocols [BT99]. Proactive routing protocols are derived for legacy Internet distance-vector and link-state protocols. They attempt to maintain consistent and updated routing information for every pair of network nodes by propagating, proactively, route updates at fixed time intervals. As the routing information is usually maintained in tables, these protocols are sometimes referred to as Table-Driven protocols. Reactive on demand routing protocols, on the other hand, establish the route to a destination only when there is a demand for it. The source node through the route discovery process usually initiates the route requested. Once a route has been established, it is maintained until either the destination becomes inaccessible (along every path from the source), or until the route is no longer used, or expired [BT99] [Bel03]. Most work on routing protocols is being performed in the framework of the IETF MANET working group, where four routing protocols are currently under active development.

These include two reactive routing protocols, AODV and DSR, and two proactive routing protocols, OLSR and TBRPF. There has been good progress in studying the protocols' behavior (almost exclusively by simulation), as can be seen in the large conference literature in this area, but the absence of performance data in non-trivial network configurations continues to be a major problem. The perception is that of a large number of competing routing protocols, a lack of WGwide consensus, and few signs of convergence [MAN]. To overcome this situation, a discussion is currently ongoing to focus the activities of the MANET WG towards the design of IETF MANET standard protocol(s), and to split off related long-term research work from IETF. The long term research work may potentially move to the IETF's sister organization, the IRTF (Internet Research Task Force) that has recently established a group on "Ad hoc Network Scaling Research".

3.2 Proactive Routing Protocols

As previously said, proactive routing protocols are derived from the traditional distance vector [MS95] and link state [Moy95] protocols designed and used in the wired Internet. Their main characteristic is the constant maintaining of a route by each node to all other network nodes in the routing table. The route creation and maintenance are performed through both periodic and event-driven messages. Periodic update means that nodes exchange routing information every fixed interval, indepentently of the nodes mobility. On the contrary, event-driven update means that each time an event such as links breakages occurs an update message is sent in the network. Obviously, in the last case, the higher is the mobility nodes the higher is the frequency of event-driven updates. When a node receives a message, it first updates its internal data structures and then computes the shortest path towards each destination applying the Dijkstra algorithm. Using this approach there is no initial delay in communication since routes are always avalaible, hence when a node application has to send data it checks its routing table and start communicating immediately. However, the control overhead introduced due the periodic information exchange can increase the overall network load, especially in quick mobility scenarios.

The main MANET IETF proactive protocols are: Destination Sequenced Distance Vector (DSVD) [PB94], Optimized Link State Routing (OLSR) [CJ03], Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [BOT01], Fisheye State Routing (FSR) [PGH00].

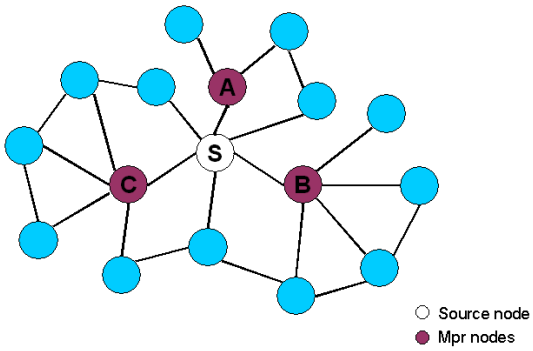


Figure 3.1: Multipoint relays selection

3.2.1 Destination Sequenced Distance Vector (DSDV)

DSDV protocol [PB94] is a distance vector protocol that adopts many optimizations to be more suitable for Ad Hoc networks. In particular, it uses a per-node sequence number to avoid the *couting to infinity* problem. Each node increments its sequence number whenever there is a change in the nearby, thus a node can use the most recent information to select the route to a destination when more choises are available. To propagate topology information, DSDV encapsulates the node's routing table in messages and sends them in the network using both periodic and event-triggered updates. In order to reduce the bandwith consumption, the full updates cointaining the entire routing table are unfrequently, while the incremental updates storing only the routing entries changed during the last interval.

3.2.2 Optimized Link State Routing (OLSR)

OLSR protocol [CJ03] is an optimization for MANET of legacy link-state protocols. The key point of the optimization is the *multipoint relay* (MPR). Each node identifies (among its neighbors) its MPRs, as shown in Fig. 4.1 . By flooding a message to its MPRs, a node is guaranteed that the message, when retransmitted by the MPRs, will be received by all its two-hop neighbors. Furthermore, when exchanging link-state routing information, a node lists only the connections to those neighbors that have selected it as MPR, i.e., its Multipoint Relay Selector set. The protocol selects bi-directional links for routing, hence avoiding packet transfer over unidirectional links. A more detail overview of OLSR is in Chapter 4.

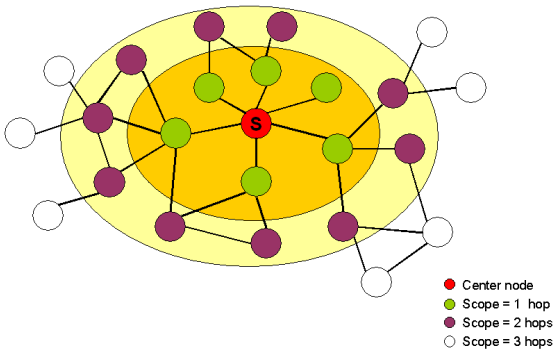


Figure 3.2: FSR scopes

3.2.3 Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)

Like OLSR, TBRPF [BOT01] is a link-state routing protocol that employs a different overhead reduction technique. Each node computes a shortest-path source tree to all other nodes, but to optimize bandwidth only part of the tree is propagated to neighbors. More precisely, the *reportable subtree* (RT) is sent in the network in two different messages: periodic updates piggyback the entire RT, while differential updates (more frequent) piggyback only those changes of RT between two consecutive periodic update. The first type of updates allows new nodes to be aware of RT; the second one guarantees a fast propagation of network information due to the smaller dimension of messages. In addition to the routing module for topology discovery and route computation, TBRPF performs the neighbor discovery using differential HELLO messages which report only changes in the status of neighbors. This results in HELLO messages that are much smaller than those of other link-state routing protocols such as OSPF.

3.2.4 Fisheye State Routing (FSR)

FSR [PGH00] is an optimization of the link state routing protocols since it uses the partial dissemination of routing information. Adopting the fisheye tecnique [KS71], FSR nodes exchange link state information with a frequency which depends on distance (in terms of hop-number) to the destination, see Fig. 3.2. More precisely, entries of routing tables related to nodes in the nearby are propagated with the highest frequency to neighbors, while the rest of the entries are sent out at a lower frequency.

Consequently, each node has its own view of the topology and precisly it is more precise toward closer nodes but hazy toward distant nodes. This inaccurancy of path does not affect the deliver of data traffic towards destinations since it is compensated by two factors: it is normalized with respect of distance and when a data packet is flooded to the destination the forwarding nodes have a more accurate knowledge of the current position of the destination, and hence they forward the data packet through the optimal path.

3.3 Reactive Routing Protocols

The reactive protocols, also called *on demand*, depart from the legacy Internet approach. In contrast with the wired network in which connectivity between nodes is stable, in the Ad Hoc networks links vary quickly, thus maintaining the complete topology graph is quite expensive. To reduce the control overhead, reactive protocols adopt a different strategy: instead of compute all paths towards all nodes, routes are discovered only when it is needed. When a node wants to initiate a communication with a destination, it first checks its route cache looking for a valid route to that node. If any route is available, it starts a *route discovery* procedure in the network to discover a valid path. In addition, the source applies a *route maintainance* procedure in order to maintain the found paths.

As said previously, the main advantage is the reduction of the introduced control overhead. On the other hand, since routes are not yet available in the route cache there is an initial delay at the beginning of data session.

Representative reactive routing protocols include: Dynamic Source Routing (DSR) [JM03], Ad hoc On Demand Distance Vector (AODV) [PR03].

3.3.1 Dynamic Source Routing (DSR)

DSR is a loop-free, source based, on demand routing protocol [JM03], where each node maintains a route cache storing the source routes learned by the node. Since it is a source routing protocol, each data packet sent in the network containes the complete sequence of nodes that the packet will cross through to reach the destination. The route discovery process is initiated when a source node do not already have a valid route to the destination in its route cache. The source broadcasts in the network a Route Request message (RREQ) cointaining the destination IP address and a route record to store other nodes IP addresses, see Fig. 3.3. When a neighbor receives the message, it

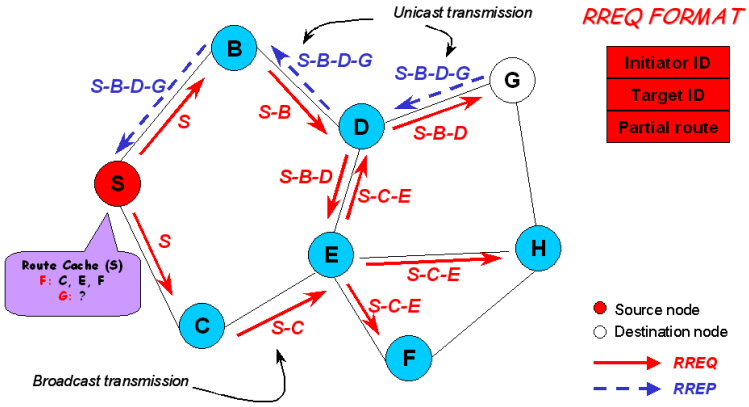


Figure 3.3: DSR route discovery

first updates its cache, then appends its address in the RREQ route record and finally broadcasts the RREQ message again. Thus, the complete path to destination is built in the message. When the destination or a node with a valid route to the destination receives a RREQ, it creates a Route Replies message (RREP) storing the complete source path and sends back to the source using the reverse discovered path. Note that entries in the route cache are continually updated through the maintainance procedure: when a link between couple of nodes is broken it is removed from the cache and a Route Error message (RERR) is sent back to the sender. Once arrived at the sender, the source removes the broken link from its cache, also deleting all other paths that contain that link. Afterwords, since nodes in DSR maintain in their cache multiple routes to a destination, if the source has another route to destination it can use it immediately, otherwise it should perform another Route Discovery for that target. In addition, DSR nodes can use the *promiscuos listening* to gratuitously learn new routes for other destinations.

3.3.2 Ad hoc On Demand Distance Vector (AODV)

AODV is a reactive improvement of the DSDV protocol. AODV minimizes the number of route broadcasts by creating routes on-demand [PR03], as opposed to maintaining a complete list of routes as in the DSDV algorithm. Similar to DSR, route discovery is initiated on-demand, the route request is then forward by the source to the neighbors, and so on, until either the destination or an intermediate node with a fresh route

to the destination, are located. DSR has a potentially larger control overhead and memory requirements than AODV since each DSR packet must carry full routing path information, whereas in AODV packets only contain the destination address. A deep AODV description can be found in Chapter 4.

3.4 Hybrid Routing Protocols

Hybrid routing protocols integrate the characteristics of proactive and reactive routing protocols reducing the protocol overhead and the latency necessary to recover a new route. They exhibit proactive behavior given a certain set of circumstances, while exhibiting reactive behavior given a different set of circumstances. These protocols allow for flexibility based on the characteristics of the network. Hybrid approachs include the Zone Routing Protocol (ZRP) [Hsi01].

3.4.1 Zone Routing Protocol (ZRP)

The Zone Routing Protocol (ZRP) [Hsi01] integrates both proactive and reactive routing components into a single protocol. It is based on the concept of *routing zone*. Around each node, ZRP defines a routing zone whose radius is measured in terms of hops, for example in fig. 3.4 the routing zone radius is 2 hops. The nodes of a zone are divided into *peripheral* and *interior* nodes. The peripherial nodes are nodes whose distance to central node is exactly equal to the zone radius, as shown in fig. 3.4. Each node utilizes proactive routing within its zone (Intrazone Routing Protocol - IARP) and reactive routing outside of its zone (Interzone Routing protocol - IERP). Hence, a given node knows the identity of and a route to all nodes within its zone. When the node has data packets for a particular destination, it checks its routing table for a route. If the destination lies within the zone, a route will exist in the route table. Otherwise, if the destination is not within the zone, a search to find a route to that destination is needed. Thus, the source sends a query message in the network using the peripheral nodes to cross the adjacent zones and reach the destination. Once a node discovers the destination, it unicasts a reply message to the source node.

3.5 Other Approaches

All routing protocols discussed in the previous sections are *flat* protocols and, as said before, their main advantage is the control overhead to create the paths toward nodes.

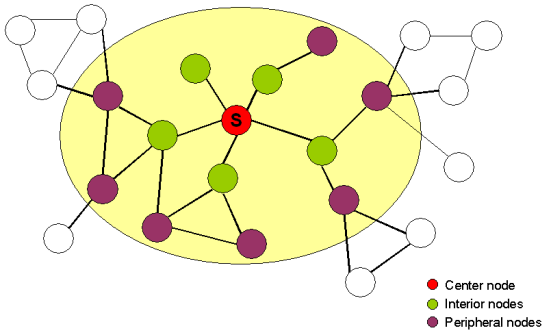


Figure 3.4: ZRP Zone Radius

Another approach to increase the scalability of Ad Hoc network is the use of *hierchical* [RS98] or *clustering* protocols. The MANET is divided into groups of nodes *clusters* basing on specific criteria (i.e. position, functionality). Each cluster has a *cluster leader*, initialzed through distributed algorithm, that generally processes control packets on behalf of their member nodes. Moreover, cluster can be group forming a multi-level hierarchies. Advantages of this approach is the use of hierarchical routing: routes are stored at the cluster level, specifying only the clusters leader and not the intermediate nodes. Thus, the routing is more flexible and robust since routes can be repaired more easily. The main disadvantages are creating and maintaining cluster leaders and the centralization of routes through their use.

Other routing protocols have been designed with the specific goal of minimization of power consumption. These protocols decrease the energy adopting several techniques. For example, the Geographical Adaptive Fidelity protocol (GAF) [XHE01] un-utilized nodes are powered down. In Battery Energy Efficient protocol (BEE) [CR00] a cost function based on energy cost and battery lifetime is assigned to each route, and the best route in term of minimum cost function is selected each time.

Other works focus the security problem for Ad Hoc networks. Examples of routing protocols designed with this primary goal are Authenticated Routing for Ad Hoc Networks (ARAN) [SDL+02] in which all network nodes can obtain a certificate from a trusted certificate server before joining the network, or the Secure Routing Protocol (SRP) [PH02] based on a secure association between the source and the destination obtained with an initial negotiation of a shared secret key.

4 OLSR & AODV

Despite the large volume of research activities and rapid progress made in the MANET routing protocols, only four routing protocols are currently under active development. These include two reactive, AODV and DSR, and two proactive routing protocols, OLSR and TBRPF. Specifically, AODV and OLSR are the most mature from the implementation standpoint. For this reason, we selected these two routing protocols as reference protocols for our experimental evaluation of a real MANET. This chapter focuses on the description of the general behavior, summarizing the main functionality of both protocols.

4.1 Optimize Link State Routing Protocol: OLSR

The OLSR routing protocol [CJ03] is a proactive routing protocol belonging to the link state family. Due its proactive nature it has the advantage of having routes immediately available. To minimize the overhead introduced in the network due the flooding of control traffic, each nodes selects its **Multipoint Relay (MPR)** to retransmit control messages. This strategy allows to reduce the number of retransmissions. OLSR consists of a set of "core" functionality (always requiered to provide routing in a MANET, e.g. link sensing, topology dissemination, route calculation) and a set of auxiliary functionality that can be used in other scenarios (e.g., in order to connect the MANET with other networks).

4.1.1 Multipoint Relays

To reduce the overhead of flooding messages in the network, each node selects a set of nodes among its neighbors, named **Multipoint Relays (MPR)**, with the task to retransmit its packets. More precisely, each node identifies the set of MPRs among its symmetric neighbors so that it can reach all its two-hops neighbors through the MPR nodes. In Figure 4.1, node S elects its MPR set, node A, B and C in figure. They broadcast packets received by the source S, while all the other nodes not in the MPR

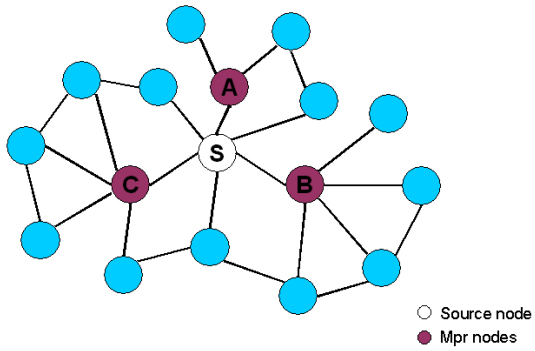


Figure 4.1: Multipoint relays selection

set of S receive and process packets coming from S but do not retransmit them. Each node maintains also information about which nodes have elected it as MPR, collecting their addresses in the *MPR Selector Set*. As a consequence, each node must retransmit only packets coming from nodes stored in its MPR Selector Set. This strategy limits the number of retransmissions in the network and, In addition, to reduce the overhead again, each node declares only a subset of its neighbors.

4.1.2 Link Sensing and Neighbor Discovery

To obtain a complete knowledge of the network topology, a node first has to detect which are its neighbors and which is the state of each link, on each wireless interface. To this aim, in the Link Sensing & Neighbor Discovery phase, it periodically sends **Hello** messages cointaining the list of status links on that interface together with the list of the entire 1-hop neighbood and the associated neighbor type. The `Hello` messages are sent using broadcast transmissions and, once received, they are not transmitted again. Figure 4.2 a. shows the format for an `Hello` message. In particular:

- **Htime**: it specifies the emission interval of an `Hello` message
- **Willingness**: it represents the willigness of a node to carry and forward traffic to other nodes
- **Link Code**: it specifies information about the status link between the interface of the sender and the interfaces of its neighbors list in the message
- **Link Message Size**: it represents the size of the message

⋆ **Neighbor Address**: it is the address of the neighbor

In the `Hello` messages a node lists either bidirectional links and unidirectional links towards its 1-hop neighbors. Adopting this mechanism, a node can know all its 2-hop neighbors and hence is able to compute its MPR set. The node announces its selected MPRs in the following `Hello` messages setting the Link Code field. Upon receiving an `Hello` message, nodes that are announced as MPRs, update its *MPR selector set*.

4.1.3 Topology Dissemination

Basically, the Link Sensing and Neighbor Discovery phase gives to nodes: *i)* the list of neighbors with which they could directly communicate, and *ii)* an optimized mechanism to flood information in the network based on MPRs. In order to compute and build routes to nodes, some of this information has to be disseminate to the entire network. To this aim, in the Topology Dissemination phase each node periodically sends **Topology Control (TC)** messages cointaining the list of nodes that has elected itself as MPR, i.e. nodes stored into its *MPR selector set*. This information carried into `TC` messages reduces the size of messages and is enough to build the routing table. The `TC` messages are sent using broadcast transmissions and, once received, they are transmitted again by MPRs in order to diffuse topology information in the network. Figure 4.2 b. shows the format for a `TC` message. In particular:

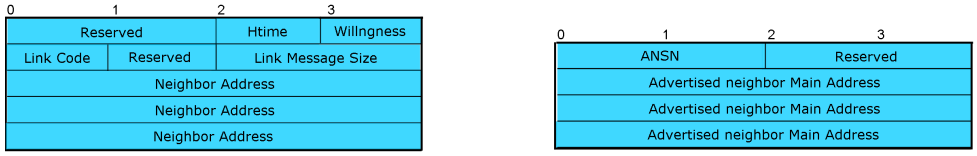
⋆ **Advertised Neighbor Sequence Number (ANSN)**: it is the sequence number associated to the set of advertised nodes. Each time a node detects a change in its advertised set, it increases ANSN to take note about fresh information

⋆ **Advertised Neighbor Main Address**: it contains the main address of the advertised node

Each node stores information coming from `TC` messages in a repository named *Topology Set*.

4.1.4 Route Calculation

Due to the proactive nature of OLSR, each node maintains in the *Routing Table* the routes towards all nodes in the network. Using information of Link Set and Topology Set, routes are built and stored using a shortest path algorithm as the Dijkstra's algorithm. For each path the associated entry maintains the next hop (gateway) and the metric (measured in number of hops) to reach a known destination. The *Routing*



(a) Hello message.

(b) TC message.

Figure 4.2: OLSR messages used in the Neighbor Discovery and Topology Dissemination.

Table is updated each time a change is detected in Link Set or in Topology Set (e.g., a neighbor appears or is lost), not implying any transmission of specific message, neither in the neighborhood nor in the entire network.

4.1.5 Auxiliarity Functionality

As mentioned previously, the functionality of OLSR is divided into *core* and *auxiliary*. This last group provides to OLSR nodes additional functionality that can be used in specific scenarios. In the following a brief description of the auxiliary funtionality is given; for additional details see [CJ03].

A node may have multiple interfaces, each of them with a distinct IP address, that may partecipate or not in the OLSR routing domain. These additional features allow nodes to announce their multiple interfaces, offering them also the external connectivity towards non-OLSR domain. In such situations a node with multiple interfaces acts as a node with a single interface, i.e. performing the link sensing, neighbor detection, topology dissemination and route calculation. However, it is identify uniquely with a *Main Address*, i.e. the address of a wireless interface of the node operating in the OLSR domain, and it always uses its *Main Address* as Originator Address of its packets.

When all interfaces of a multiple interface node are in the OLSR domain, periodically the node sends information describing its interfaces configuration. Thus, a **Multiple Interface Declaration (MID)** message is broadcasted in the network and, consequently, retransmitted by MPRs in order to diffuse it in the entire network.

When a node is connected to an extra-OLSR domain (e.g. the Internet), the node has to inform the network that exists a possibility to reach other domains. In particular,

it periodically sends an **Host and Network Association** message in the network announcing itself as gateway to specific networks. This information is broadcasted and, through MPRs, transmitted in the network.

4.2 Ad hoc On-Demand Distance Vector Routing Protocol: AODV

The AODV routing protocol [PR03] is a reactive routing protocol that offers quick adaptation to dynamic link conditions, low processing and memory use, low network utilization, and determines unicast routes to destinations in an on-demand manner. AODV nodes have to maintain routing information only of active communications. They are also able to respond to link breakages and changes in network topology in a timely manner. AODV associates a sequence number to each route, avoiding the "counting to infinity" problem and guaranteeing loop-free operations.

4.2.1 Route Discovery

As previously mentioned, in AODV the route discovery is an on-demand procedure. When a node wants to communicate with an unknown node (i.e. a node without any routing information available), it starts a route discovery procedure broadcasting a **Route Request (RREQ)** message in the network, as shown in Figure 4.3 a. In particular:

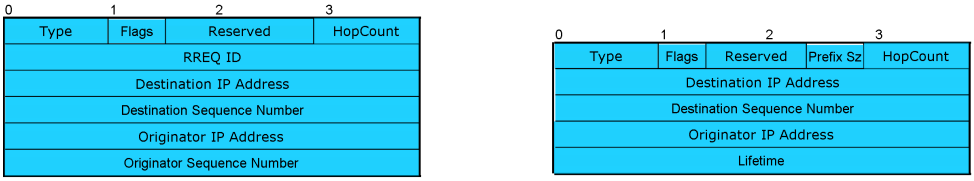
⋆ **Hop Count**: it is the number of hops from the Originator IP Address to the Destination IP Address

⋆ **RREQ ID**: it is the sequence number that identies uniquely, together with the source IP address, the particular `RREQ`

⋆ **Destination IP Address**: it is the IP Address of the destination node for which a route is required

⋆ **Destination Sequence Number**: it represents the last known destination sequence number for this destination

⋆ **Originator IP Address**: it is the IP Address of the node that has generated the `RREQ` message



(a) RREQ packet.

(b) RREP packet.

Figure 4.3: AODV packets used in the Route Discovery phase.

⋆ **Originator Sequence Number**: it represents the sequence number to be used in the routing table to identify the originator of `RREQ`

Before broadcasting the `RREQ`, the originator buffers the RREQ ID and the Originator IP Address with an associated timeout. This allows the node to identify and discard copies of its `RREQs` forwarded by its neighbors.

Each node receiving the `RREQ` message checks in its cache if it has a "fresh enough" route to the destination, i.e. a valid route entry for the destination whose associated sequence number is at least as great as the one cointaned in the `RREQ` message. If any route to the destination is available, the intermediate node caches a route back to the originator of the request. This reverse route can be used to send back a `RREP` message (i.e., the answer to the `RREQ` as explained below) or, in general, application traffic. In addition, if any route is available, the intermediate node first increases the *Hop Count* field and then broadcasts the `RREQ` in the network.

A route can be determined if the `RREQ` reaches the destination itself or an intermediate node that has a valid route stored in its cache. In such situations, a **Route Reply (RREP)** message is generated and sent back to the destination with a unicast transmission. The format of a `RREP` message is shown in Figure 4.3 b. When generating a `RREP` message, a node copies the *Destination IP Address* and the *Originator Sequence Number* from the `RREQ` message into the corresponding fields of the `RREP` message. In addition:

- if the generating message is the destination, it copies its sequence number into the *Destination Sequence Number* and puts zero into the *Hop Count* field
- if the generating message is an intermediate node, it copies its known sequence number for that destination into the *Destination Sequence Number* field and

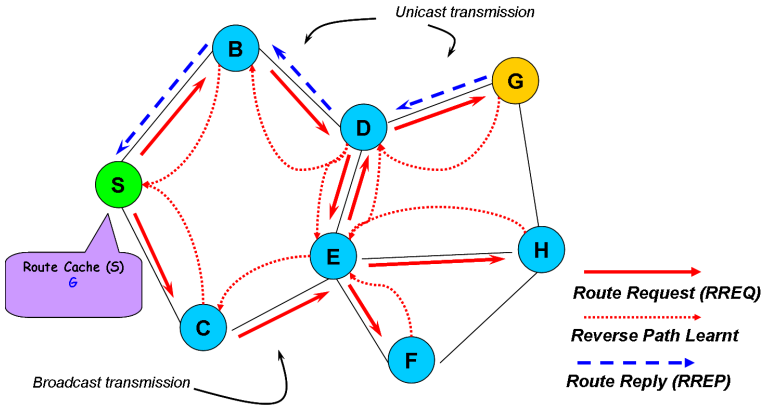


Figure 4.4: Route Discovery procedure.

stores its distance (in hops) from the destination into the *Hop Count* field of `RREP`. It also updates the forward cache and the reverse cache with the new next hop.

Figure 4.4 shows an example of a route discovery procedure started at node S to find a valid route towards a node G.

To guarantee the use of bidirectional links in the route discovery procedure, each node maintains a *Blacklist* set. When a node detects that a `RREP` transmission is failed (e.g., due to the presence of unidirectional link), it stores in the *Blacklist* set the next hop of the failed `RREP`. Thus, a node discards `RREQs` from nodes of its *Blacklist* set avoiding the creation of routes composed of unidirectional links.

An originator node has several attempts to discover a valid route to the destination. After broadcasting a `RREQ`, a node waits for a `RREP`. If any reply is received in a specific time window, the node tries to discover the route again broadcasting another `RREQ`, up to maximum number of attemps. If any route is found, all data packets to that destination are dropped from the buffer and a message of *Destination Unreachable* is delivered to the application.

4.2.2 Local Connectivity

In addition to the Route Discovery process, AODV nodes maintain the local connectivity with its neighbors broadcasting also local **Hello** messages. More precisely, each node periodically xchecks if it has sent a broadcast message (e.g., a `RREQ` message) in the last time interval. If not, it broadcasts an `Hello` message that is a `RREP` message

with TTL field equal to 1. Thus, a node can determine its local connectivity even though there's no explicit route request by listening for packets from its set of neighbors. If a node receives an `Hello` message from a neighbor in a specific time window it deduces that the link is active, otherwise it assumes that the link is lost.

4.2.3 Route Maintainance

Each node detects continuously the status of its active neighbors. This operation is possible through: *i)* the reception of `Hello` messages as explained previously or any other packet; *ii)* a `RREQ` unicast to the next hop, asking for a route to the next hop; *iii)* a link-layer notification if available. If a node deduces that an active link towards a node is lost, a Route Maintainance procedure starts. More precisely, the node emittes a **Route Error (RERR)** message, also invalidating active routes in the cache that use the unreachable neighbor as the next hop. The format of `RERR` packet is shown in Figure 4.5 and the main fields are:

- ⋆ **Dest Count**: it is the number of unreachble destinations included in the packet
- ⋆ **Unreachable Destination IP Address**: it is the IP Address of the unreachable destination node
- ⋆ **Unreachable Destination Sequence Number**: it represents the sequence number in the route table for the unreachable destination declared in the previous Unreachable Destination IP Address field

When a node receives a `RREP`, it updates its data structures and forwards the message towards the originator. Once arrived at the originator node, it removes the route using that unreachable link and, if it wants to communicate towards the same destination, it starts a new Route Discovery. Figure 4.5 shows the explained procedure.

4.2.4 AODV optimizations

The AODV protocol adopts several strategies in order to reduce the overhead introduced in the network.

To prevent unnecessary dissemination of `RREQs`, especially in large scale networks, an *expanding ring search* technique can be used. As explained previously, the discovery procedure is executed performing several attempts. Using an *expanding ring search* technique, the searching area is increased each time setting the TTL field. More precisely, in the first attempt the TTL field is set to 1 and the node waits for the

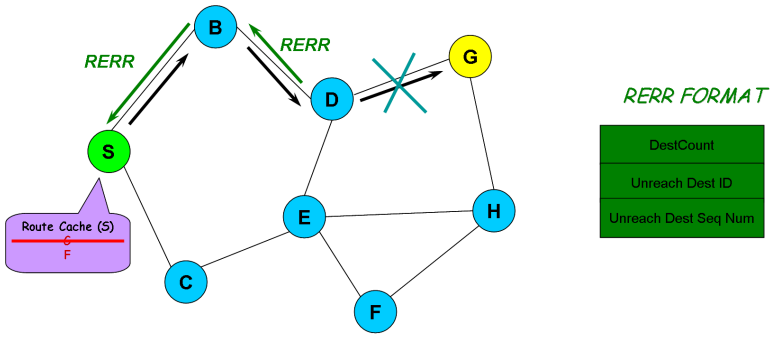


Figure 4.5: Route Maintainance procedure.

corresponding `RREP`. If the *RREQ timeout* expires without any answer, the originator node broadcasts the `RREQ` again increasing the TTL field to 2, etc.. This procedure continues until the TTL field reaches a threshold.

As further optimization, AODV prevents the loop's creation in the route discovery phase. Each request is uniquely identified by the values (RREQ-ID, Originator IP Address), thus each node can detect if it has already received it. Consequently, a node replies only to the first received request, discarding the others. This mechanism reduces the overhead and limits the choices for the reverse paths.

5 Small scale multi-hop Ad Hoc network

In this chapter we report an experimental comparison between two Ad Hoc routing protocols on real networks. Specifically, we evaluate performance of OLSR and AODV either in indoor and outdoor environments on networks of 2-4 hops size with up to 8 nodes, representing realistic scenarios of few people exploiting the Ad Hoc network to share documents. In fact, as pointed out in [GLNT05], with current technology, benefits of Ad Hoc network will vanish beyond the Ad Hoc horizon of 2-3 hops and 10-20 nodes. Our analysis shows that with semi-static topology the proactive approach performs much better than the reactive from the efficiency and QoS standpoint, and it introduces a limited overhead. On the other hand, even in these simple scenarios, AODV performances are often poor introducing delays of seconds in order to ping a node few hops away.

5.1 Introduction

Ad Hoc wireless networks consist of groups of mobile nodes that may communicate without any form of pre-existent infrastructures. Nodes belonging to MANET may move dynamically and unpredictably, so the network should be able to react to the frequent topological changes. For this reason conventional Internet routing approaches are not appropriate, while special routing protocols adapting to the peculiar characteristics of these networks are required. Several routing protocols have been proposed in the last ten years [CCL03a] [Bel03] and most of them have been evaluated and compared through simulations, see for example [DPR00] [DCY00]. Simulators allow the performance evaluation of protocols in different scenarios, defined by varying several parameters (e.g. number of nodes, mobility models, data traffic); however they often introduce simplifying assumptions that mask important characteristics of the real protocols behavior. Chapter 2 gives some examples in this direction [ABCG04]. Another example is the so-called "communication gray zones" problem [LNT02]. This problem

was revealed by a group of researchers at the Uppsala University, while measuring the performance of their own implementation of the AODV routing protocol in an IEEE 802.11b ad hoc network. Observing an unexpected large amount of packets' losses, mainly during route changes, it was found that an increase in packet loss occurred in some specific geographic areas called "communication gray zones"[1]. Note that the communication-gray-zone problem was not revealed by commonly used simulation tools (e.g., NS-2, Glomosim) as in their 802.11 models both unicast and broadcast transmissions are performed at 2 Mbps, hence having the same transmission range. In order to obtain more realistic performance results, and to evaluate the actual inaccuracy of simulation's models, protocols evaluation via simulation has to be complemented by experiments with real prototypes, even though experimental testbed are not so easy to implement and only small-medium size testbeds can be generally set up. The availability of prototypes can also increase the creation of user communities that, experimenting this technology, can provide feedbacks on usability and possible relevant applications for the contemporary society.

Currently, only few measurements studies on real ad hoc test-beds can be found in literature, see e.g., [Dep] [GKN+04a]. The Uppsala University APE test-bed [Dep] is one of the largest, having run tests with more than 30 nodes. The results from this test-bed are very important [GLNT05] and point out that more research in this direction is required to consolidate the ad hoc networking research field.

This work provides a contribution in this direction. Hereafter, we report our experiences and results obtained by measurements on a real ad hoc network, implementing a full ad hoc network architecture. In particular, we set up a MANET prototype on which we performed several sets of experiments; specifically we focused the study on different solutions for routing protocols and middleware platforms. The novelty of this work is twofold:

1. we investigate a full protocol stack with particular attention to routing and middleware layers;

2. we evaluate through experimental results the advantages of a cross-layer architecture, presented in [Del05], mainly focusing on routing and middleware interactions.

In this chapter we focus the discussion on the network layer. The interested read-

[1] This phenomenon is due to the different ranges between unicast (data) and broadcast frames (i.e., routing information) in 802.11 networks. A station inside a *gray zone* is considered using the routing information reachable by a neighboring station, while actual data communication between the station is not possible.

ers referred to [Del05][BCDG05] to go into middleware results and cross-layer results. In this experimental testbed we evaluate the performances of two routing protocols (i.e., the reactive AODV [PR03] and the proactive OLSR [CJ03]). Having in our testbed one proactive and one reactive routing protocol enables us to compare these two approaches in a realistic scenario. In the literature, it is a common use to consider that on-demand reactive protocols are more efficient than proactive ones. As deeply explaned in Chapter 3, on-demand protocols minimize control overhead and power consumption since routes are only established when required. By contrast, proactive protocols require periodic route updates to keep information current and consistent; in addition, maintaining multiple routes, which might never be needed, causes unnecessary routing overheads. On the other hand, proactive routing protocols provide better quality of service than on-demand protocols. As routing information is constantly updated in the proactive protocols, routes to every destination are always available and up-to-date, and hence end-to-end delay can be minimized. For on-demand protocols, the source node has to wait for the route to be discovered before communication can happen. This latency in route discovery might be intolerable for real-time communications. Thus we compare and contrast them in different environments, i.e. indoor and outdoor, and different topology, i.e. static and mobile, evaluating their performance from the efficiency and QoS standpoint on network on 2-4 hops size with up to 8 nodes.

5.2 Experimental Environment

Before analyzing system performances, a detailed description of the testbed architecture and the experimental environment is needed. The measurement test-bed is based on an Ad Hoc network made up of laptops with different capabilities running Linux and equipped with two different wireless cards compliant to IEEE 802.11b standard working at a constant data rate (11 Mbps). Moreover, we considered a static network where all stations do not change their position during the experiments, introducing also scenarios with topology changes due to events of nodes' connection/disconnection. In the performed experiments we used a limited number of nodes. This scenario could seem not meaningful if compared to those simulations scenarios using hundreds of mobile nodes. However, recent results pointed out the existence, with the current technology, of an *ad hoc horizon* of 2-3 hops and 10-20 nodes. Beyond these limits the benefit from wireless multi-hop ad hoc networking virtually vanishes [GLNT05]. Indeed all the experiments presented hereafter fall inside this *ad hoc horizon*. This may represent a scenario consisting of few people forming an ad hoc network to share

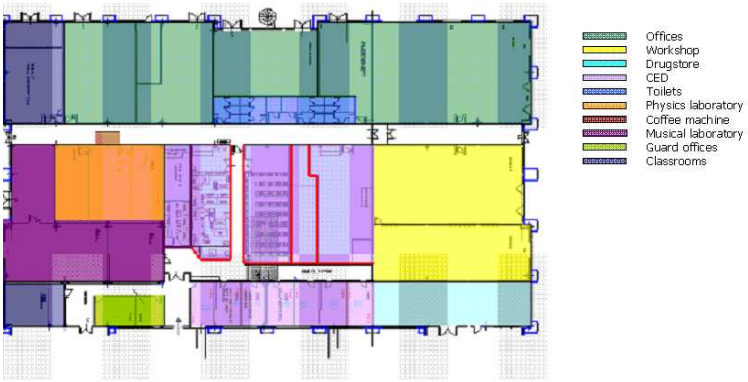


Figure 5.1: Experimental Area

documents.

5.2.1 Software

As mentioned in the previous Chapter, most work on routing protocols is being performed in the framework of the IETF MANET working group, but only four routing protocols are currently under active development. These include two reactive routing protocols, AODV and DSR, and two proactive routing protocols, OLSR and TBRPF. In particular, AODV and OLSR are the most mature routing protocols from the implementation standpoint; for the other MANET protocols either updated implementations are not available (DSR) or there is no freely available implementation (TBRPF). For this reason, we integrated in our test-bed the implementation of one reactive (AODV) and one proactive (OLSR) routing protocol. Furthermore, some DSR experimentations are carried out. However, due to some problems with existing implementations (existing implementations are CPU intensive, thus causing poor performance), we decided not to include, at this stage of our study, DSR in our testbed. The considered routing protocol implementations were the UNIK-OLSR [OLSR] by the University of Oslo (Norway), and the UU-AODV [UU] by the Uppsala University (Sweden).

5.2.2 The network topology

Indoor

All the indoor tests were conducted at the ground floor in the CNR campus in Pisa (see Figure 5.1). At this level there is the computing center (CED) together with some companies offices and measurement laboratories with several kinds of instrumentations. The structural characteristics of the building, and particularly of this floor, strictly determine the transmission capabilities for the nodes of a wireless network situated within. Rooms (offices, laboratories, etc.) are generally delimited by masonry padding walls situated between reinforced concrete pillars; in the CED area, instead, locations are separated by either "sandwich panel" of plastic materials which dont reach the height of the ceiling) or metal panels till the ceiling: these generally cause minor impediments to the waves compared to masonry walls or reinforced concrete pillars. Wireless links are also influenced by the presence nearby of Access Points and measurement instrumentation which introduce quite a lot of noise. Moreover, about 30-40 people work in this floor every day and get around from office to office or towards service areas with coffee machines, toilets, etc. This makes the transmission coverage characteristics of the floor and the stability of the links modify continuously and in an unpredictable manner. As a result the whole place can be considered quite a realistic environment for testing an ad hoc network. Figure 5.2 presents the detailed map of the place together with the static transmission coverage characteristics of the area: nodes are situated where devices were placed during the experiments and straight lines are used to point out the presence of wireless links (two nodes see each other at one hop distance if a single straight line joins them).

Outdoor

In order to perform experiments on string topology we also set up the network in an open field of about 300 meters long. There were no physical obstacles (e.g., buildings, trees) among nodes, thus each couple of adjacent stations was in line-of-sight and in their respective transmission ranges. In order to obtain an open environment aligned with the indoor scenario (i.e., a string topology where only adjacent nodes are in the transmission range of each other) we had to increase the distances between stations up to 70 meters, while in indoor, due to walls, doors obstacles, they were at a distance of about 15m. The characteristics of open spaces are quite different from indoor spaces. In both environments, wireless links can vary frequently and rapidly in time and space due to several factors but in the open space the node-distance increase makes

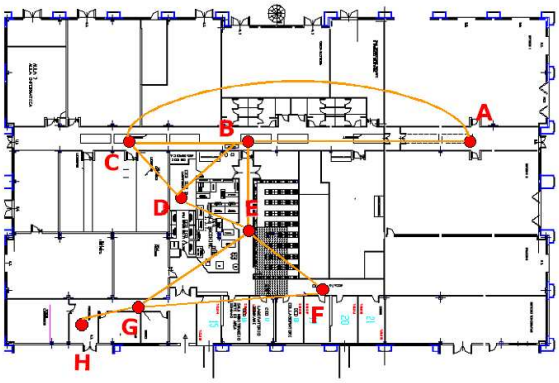


Figure 5.2: Network Topology

the wireless links more unstable. For example, [GKN+04b] shows with an extensive test-bed that wave's propagation in a real environment is very complex depending on phenomena such as background noise, obstacles' presence and orientation between sender and receiver antennas. In the outdoor systems, the longer distances cause higher links variability. In addition, it is possible that not all nodes are in the same carrier sense range as illustrated in Chapter 2, thus the coordination may result very complex.

5.3 Routing Experiments Warm-up: A Qualitative Analysis

As the first step, we conduct a qualitative analysis of the routing protocols to check if the selected software behaves correctly. Thus, we test the selected implementations of proactive OLSR and reactive AODV routing protocols with a real experimentation. We check their state of implementation, validating their functionality and conducting a comparative analysis on them all. Hereafter, we report the main results of this phase. A detailed presentation about all experimentation can be found in [D8].

5.3.1 UNIK-OLSR Testing

Due to the proactive nature of the protocol the test was based on observing the status of the nodes routing tables while nodes were added/removed to/from the Ad Hoc network. This testing was subdivided in two steps. In the first step we used a 5-

node network. In this case the kernel routing tables were small and could be read in real-time, hence it was possible to follow configuration changes while in progress. Upon the beginning of the experiments, node insertions and removals were provided so to check that configuration updates effectively took place. Moreover, by changing the time lag duration between successive node insertions and/or deletions, it was also possible, to some extent, measure the configuration-update delays after the appearance/disappearance events. In all the experiments the protocol showed a correct behavior. The routing tables quickly updated upon node insertion and removal. We then considered a 12-node network. The increased number of nodes led to the increase of the number of protocol packets exchanged. This allowed the validation of UNIK-OLSR behavior in a more congested context. Also in this set of experiments, all the routing-forwarding operations were correctly performed.

After this analysis, we investigated the ability of an OLSR-based network to transfer data between nodes at a distance of few hops. To this end, the UNIK-OLSR protocol was started on all the nodes at the same time, then after a little delay to let the routes stabilize, an FTP transfer was started between two nodes. The destination was at three hops distance from the source. The aim was to transfer a 34 megabytes (MB) file. Several problems were experienced in this case. Intermediate nodes along the sender-destination path stopped working correctly after a while. This was due to the wireless card do not properly working. It seemed that the excessive traffic they have to manage caused problems to their cards' drivers. In these experiments the routing protocol still behaved correctly by selecting alternative routes to avoid the out-of-service nodes. The file continued until a network partition occurred. At this time the destination host had received only the first 15MB of the file. The throughput during the transmission had been just about 180Kbps. We repeated the experiment and similar problems were notised. Specifically, we observed that the file-transfer started correctly but while the transfer proceeded the throughput of the connection reduced. This type of behavior can be explained with problems produced by the interaction between TCP and the 802.11 MAC extensively investigated in the literature, see Chapter 3 in [BCGI04] for a summary.

5.3.2 UU-AODV Testing

As this protocol is reactive, some application-level traffic was introduced in order to observe the route creation process. Specifically, each node sent periodically a set of pings to different destinations and this forced the routing protocol to set up, for each

ping operation, a route toward the ping-destination node. In this case, each sender was always able to discover the correct paths towards chosen destinations, however the discovery of the paths was very time-expensive. In the next section, some estimates of these delays will be provided.

After this, we tested the UU-AODV ability to support user-data transfer. Once again we used a file transfer application. The file transfer was started, from a couple of nodes at a 3-hop distance, with the aim to transfer a 5MB file. The transfer was definitely too slow and after 16 minutes only 140KB had reached the destination; the experiment was then interrupted. Again, the problem seemed related to interaction of MAC and TCP mechanisms. Packet losses caused a TCP congestion-reaction that slowed down the connection throughput. In addition, in this case, the reactive nature of the routing protocol made the things worse.

5.4 Static Scenario

In this section we present the performance of OLSR and AODV routing protocols for MANET in *static* scenarios. The performance comparison is based on the following performance indices:

- the *network overhead* introduced by routing messages
- the *delay* introduced in data transfer
- the *packet loss* suffered at the application level

To have a meaningful comparison, we introduced some traffic at the application layer using the `ping` utility. This guarantees that AODV runs in a complete manner; otherwise, without any application-level traffic, its routing information is reduced only to `Hello` packets exchanges.

5.4.1 8-Nodes Experiments

The experiments reported in this subsection were made in the *indoor* environment shown in Figure 5.2. For ease of reading, in Figure 5.3 we report the derived network graph in which we label the MANET nodes in order to identify them in the following discussion. A line among a couple of nodes indicates that a link exists among them. In this scenario we focused on the *overhead* introduced and the *delay* introduced in the network due to routing protocols. In our scenario, a selected node generates `ping`

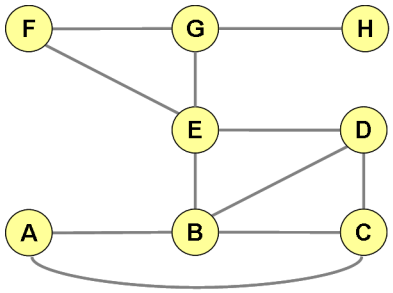


Figure 5.3: Network Topology Graph.

traffic towards the remaining nodes of the network according to a random selected sequence, and precisely it pings a node for 1 minute, and then starts pinging the next node in the sequence. We performed the following two sets of experiments changing the "pinger", e.g. the node selected as the source of `ping` traffic, in order to evaluate the impact of node position on the routing load.

- Experiment 1: all nodes started running the routing protocol together. After 30 sec, the central node E starts pinging all the other nodes with the following sequence: A, H, D, F, G, B, C. Each ping operation lasts for 1 minute.
- Experiment 2: all nodes started running the routing protocol together. After 30 sec, the external node H starts pinging continuously (for 400 sec) the same destination A following the shortest path available in the network (H-G-E-B-A). After x seconds from the beginning of the experiment (x equals 250 and 180 sec in OLSR and AODV experiments, respectively), node B disconnects itself from the network. This topology change forces the network to react, searching for a new route in order to deliver packets to node A. After B disconnection, packets start to follow the unique available path through nodes D and C.

We repeated the same set of experiments several times producing similar results, so we present just one of them.

Experiment 1 Analysis

The resulting behavior of the network in terms of the overhead introduced by OLSR and AODV is presented in Figure 5.4 and 5.5, respectively. The curves show the amount of control traffic observed by each node of the network as the sum of routing traffic generated locally by the node and the one received from other nodes and forwarded by it.

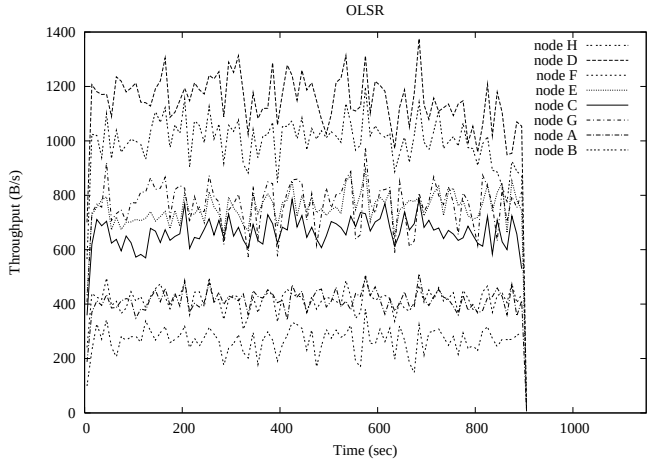


Figure 5.4: Pinger E: OLSR Overhead

As it clearly appears in the graphs, the position of the node and how it is connected to the other nodes strictly determine the control traffic observed by it. If we look at Figure 5.4 for example we can notice that curves seem to form four clusters. Specifically, node B and D observe the highest traffic of about 1.1 KBps, nodes C, E and G have an intermediate load around 800 Bps, nodes A and F observe traffic of about 400 Bps and at last node H obtains the lowest load (300 Bps) that represents 1/4 of the traffic load performed by B and D. Thus, we can conclude that there is a connection between the obtained load and the role in the network graph and, more precisely, the traffic load scales with the node's degree. Since node H is a leaf and it is connected to the network with one link it observes the lowest load; while increasing the number of neighbors the introduced overhead is higher. For AODV protocol (see Figure 5.5) we do not observe the same regular relationship as pointed out previously for OLSR. For example, nodes with the highest degree (B and E) experience an intermediate load. An explanation of this behavior is the reactive nature of AODV protocol that makes routing overhead dependent on the traffic flows at the application level. From the quantitative standpoint, obviously the overhead introduced by OLSR is significantly higher than the one produced by AODV due to the different policy to create and maintain routes. Specifically, OLSR overhead falls in a range of [200-1200] Bps, while using AODV it is around [200-400] Bps. However, it is important to highlight that these values reduce the available 802.11 bandwidth only of a negligible percentage, in the worst case of a quantity of 1.2 KBps.

To evaluate the delay introduced by the selected routing protocols we measured the

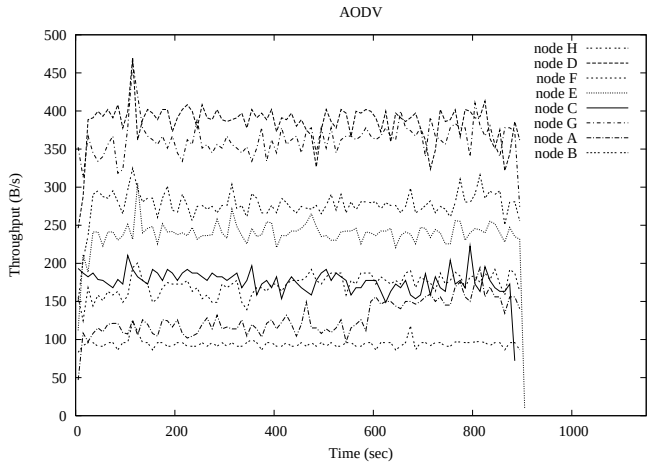


Figure 5.5: Pinger E: AODV Overhead

end-to-end latency for completing a simple `ping` operation between couples of nodes. In the following analysis, we refer to results for nodes at 2 hops distance.[2] At the start-up, when both protocols are not yet stabilized and all data structures are empty, AODV suffers a delay of 19-20 seconds to find the path toward the destination A, while OLSR requires about 8 seconds to complete the same operation. The subsequent `ping` operations take about 200 msec (or less) when using OLSR (because of its frequent updates of routing tables) and about 1 sec in case of AODV. In the last case, those performances happen when the route has just been stored in some neighbor's cache, and hence the `RREQ` doesn't need to reach the destination. Observing that the first path discovery requires many seconds, we decided to investigate AODV's performance when routes' entries expire in the cache, thus we introduced a sleep time of 20 seconds between two consecutive `ping` operations. The measured delay is about 2 sec in almost all cases except the node A case where we still measure 20 sec. This difference can be explained taking into account that when H pings nodes in the network (excluding A) AODV protocol has already achieved a steady state since each node knows at least 1-hop neighbors due to the `Hello`'s exchange. To summarize, in this experiment delays introduced by AODV are significantly longer compared to those obtained using OLSR. Furthermore numerical results indicate that QoS problems may occur when using the AODV protocol; applications with time constrains may suffer a long latency to discover paths in a reactive way.

[2] As expected there is no difference among protocols when pinging 1-hop neighbors

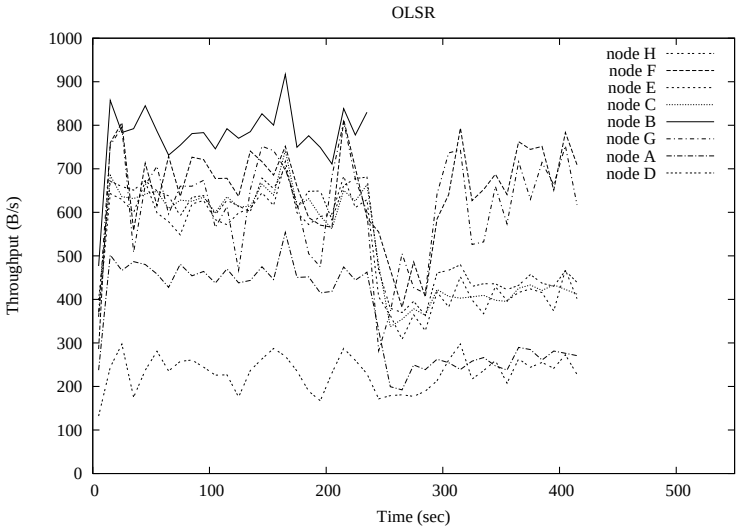


Figure 5.6: Disconnection's event: OLSR Overhead

Experiment 2 Analysis

The results for OLSR overhead and AODV overhead are summarized in Figure 5.6 and 5.7, respectively. Referring to OLSR (see Figure 5.6), we can note a load distribution similar to the one observed in the first set of experiments, e.g., node B and node H experience the highest and lowest load, respectively. After node B disconnection, there is a transient phase in which the nodes' traffic decreases (due to some missing routes); after this period, a new steady state is achieved. In this new state, we can observe a significant decrease of the traffic in the nodes that are connected with node B (A, E, D, C), while nodes far from the "dead" node perform almost the previous overhead. Once again, the position of a node in the network determines its load. Looking at AODV results (see Figure 5.7) we observe less marked differences in the entire duration of the experiment. After the transient state following the B shut down, the active nodes almost observe the same load: the traffic has a range variability of about 100 Bps. This confirms that in this case protocol overhead is correlated to the application flow.

As far as the delay is concerned, we got results similar to those observed during the start-up phase of the first set of experiments (i.e, larger delays with AODV). Moreover, referring to the disconnection's event, the `ping` operation performed while OLSR is

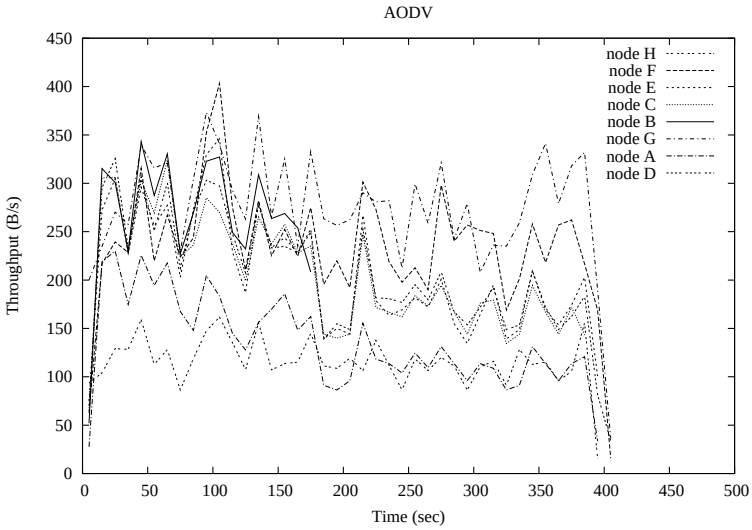


Figure 5.7: Disconnection's event: AODV Overhead

updating its routing tables experience a delay of about 6 sec to be completed; while AODV introduces a delay of [9-14] sec to discover a new route to the same destination A.

5.4.2 4-Nodes String Experiments

In a third set of experiments, we compared OLSR and AODV performances in a string topology (see Figure 5.8), both in *indoor* and *outdoor* environments. The main performance index used in this case is the *Packet Delivery Ratio*. The PDR index is calculated as the total number of packets received at the intended destinations and divided by the total number of generated packets. We also check briefly the introduced delay comparing them with the previous result.

The string testing methodology is similar to the one adopted in the previous scenario: the sender A pings continuously each node in the network with the sequence B, C, D. However, in this case the duration of each `ping` operation is variable and in particularly it scales with the distance to the intended destination (i.e., 1 minute for 1-hop node, 2 min for 2-hops node and so on).

Looking at the introduced delay, the outdoor results add no new qualitative information respect of the previous discussion. However, in outdoor, times required to complete a `ping` operation may further increase due to the links variability; for exam-

Figure 5.8: String Topology

	INDOOR			OUTDOOR		
	B	C	D	B	C	D
OLSR	1	0.97	0.98	0.98	0.65	0.47
AODV	0.85	0.9	0.49	0.95	0.01	0

Table 5.1: Overall Packet Delivery Ratio

ple, OLSR introduces a 1 sec delay in a 3-hops connection.
Table 5.1 shows the PDR for the two protocols averaged over several repeated test-runs. Looking at the indoor results, we noticed that OLSR delivers packets with high probability to all nodes in the string topology; AODV works properly with node in 2-hops neighborhood, but its performance decreases up to 50% of packets delivery when the distance sender-receiver grows up to 3 hops. We investigated which phenomena caused the enormous packet loss on AODV examining the log files. We discovered that sometimes unidirectional links between not adjacent nodes may appear in the network. Since AODV exploits also unidirectional links, it is possible that `ICMP` packets follow different paths in the end-to-end communication. Furthermore, since these links vary with high frequency, every time they disappear a `RERR` packet is generated and a new path discovery starts (we observed that several times). If no route is found before a timeout expiration all buffered application packets are lost. OLSR doesn't suffer this problem because only symmetrical links are considered resulting in a more stable network. The outdoor results show a good behavior of the two protocols only in the nearby; in fact when the sender-receiver distance increases, both algorithms suffer significant packets' losses. The packet delivery ratio of OLSR decreases up to 50% when it pings the farthest node D. Performances of AODV drastically degenerate when running in outdoor environment: almost all `ping` operations to nodes distant more than 1 hop failed. In addition, we run the same set of experiments varying the data rate to 2 Mbps. In this case AODV reaches a better performance increasing its PDR up to 0.8 when pinging node C (OLSR result is aligned with AODV), but no improvement is obtained towards node D. It seems that in open space the reactive nature of AODV is more penalized than the OLSR proactive one. Due to walls etc. one-hop distances involved in indoor environment are much shorter than those used in open space, thus a better coordination at the MAC layer guarantees a higher packets

delivery. As previously explained, in outdoor it is possible that not all nodes are in the same carrier sense range, affecting the overall performance. Having in advance redundant routing information, as with proactive protocols, guarantees that each node is able to create and maintain its own view of the entire network, even in bad conditions, and consequently delivering at least a percentage of packets successfully. Hence OLSR results more robust than AODV in outdoor environments.

5.5 Mobile Scenario

In this section we present the performance evaluation of the routing protocols on a string topology. In the previous section we considered only static networks and we focused on the overhead introduced by them; here we introduced mobile nodes in order to evaluate the impact of the mobility on routing protocols. To this end, we considered a string topology network of four nodes and we performed three sets of experiments increasing the number of mobile nodes (all the scenarios are shown in fig.5.9). In this scenarios the connectivity between the sender and the receiver changes from 1 hop to 3 hops and viceversa during the experiments. To have a comparison between OLSR and AODV, we studied the same parameters used in the static scenarios:

- the *Packet Delivery Ratio* (total number of packets received at the intended destinations divided by the total number of generated packets)
- the *delay* needed for the network's reconfiguration due to the movements of nodes.

In particular, in our scenarios, all the nodes start running the routing protocol and, after an initial period necessary for the network topology stabilization, node A pings continuously node D until the end of the experiment. We repeated the same set of experiments several times; obtained results were similar, so we present an average of them. One may argue that similar set of experiments were already available in literature. On the other hand we think that there are several main reasons to perform these experiments in our environment:

1. Our cross layer architecture assumes an underlying proactive routing protocol. Comparing AODV and OLSR performance enable us to better understand if and how the proactive assumption impacts on the overall system performance. Results presented in the previous section and those presented here indicate that in small-medium scale networks and low mobility scenarios OLSR does not penalize the system performance;

2. Measurements related to the topology management provide a reference to understand the behavior of the p2p protocols [BCDP05] and, in the specific case of CrossROAD (see [Del05]), also give a direct measurement of the expected delays in the overlay construction and reconfiguration. Therefore, a better understanding of the routing protocol performance will be useful when analyzing the behavior of the p2p platforms.

The configuration and the methodology used for the experiments follow those published in [Lun], and can be taken as a reference for our performance evaluation.

- Experiment 1: In the first set of experiments, called **Roaming node**, there are 3 static nodes (B, C, D) and the "roaming" node A. The experiment lasts 2 minutes: from the initial position W, node A starts moving and every $20sec$ it reaches the next position in the line (X, Y, Z); once it has reached the last position Z, it immediately moves in the opposite direction following the reverse path and reaches the starting position near node D after another minute.

- Experiment 2: The second set of experiments is referred as **End node swap** due to the movement of the two communicating nodes (A and D), while the rest of the network remains in the same configuration. More specifically, the two end nodes maintain their initial position for the first $20sec$ of the ping operation, then they start moving reaching the next position in the line every $20sec$. The experiment lasts other $20sec$ after the end nodes have swapped their positions.

- Experiment 3: The last set of experiments, named **Relay swap**, is similar to the previous one: there are 2 mobile nodes in the network that change positions during the test. In this case after $20sec$ from the beginning of the ping operation, central nodes start moving and swap their positions after $20sec$, then they remain in this new configuration until the end of the experiment (it lasts $60sec$).

In all the performed experiments each mobile node moves along the line with a speed of about $1m/s$, since we are interested in investigating low mobility scenarios.

Looking at the *Packet Delivery Ratio* index, as shown in Table 5.2, we notice that increasing the complexity of the proposed scenarios, the performance of the two routing protocols decreases up to about 60% of packets delivery in case of Relay swap scenario. Specifically, in the Roaming node scenario we can note that both protocols have similar behaviors: there is a packet loss of about 25%. Examining the log files, we observe that, for both protocols, packet losses mainly occur when node A goes beyond position Y and reaches the string's end; specifically this represents the time in which the connection

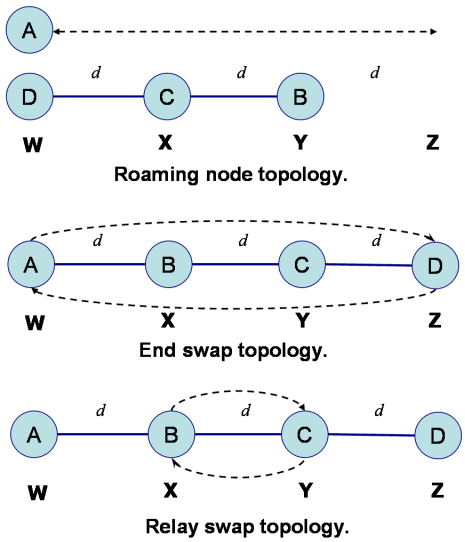


Figure 5.9: Mobility Scenario.

A-D changes from 2-hop to 3-hop connection, due to the loss of the direct link A-C. In the End swap scenario, the proactive protocol performs better than the reactive protocol: delivered packets increase of 10%. OLSR introduces the high percentage of its packet loss in the last $40sec$ of the test-run when the connection becomes again a 3-hop connection; on the other hand at the beginning of the experiment all packets were correctly received since the network was already stabilized when data transfer started. In contrast AODV distributes uniformly its packet loss during the entire test-run. As previously said, in the third set of the experiments the packet delivery ratio of OLSR and AODV decreases up to 66% and 60%, respectively. In particular, from the log files we notice that packet losses occur during the relay swap phase (i.e., from 20 to $40sec$), in which only half of the number of packets generated by node A reaches the destination successfully.

To evaluate the *delay* introduced by the two routing protocols due to nodes' movements, we measured the time needed to update the routing table for OLSR and to discover new paths to the destination for AODV. In the first scenario, when node A moves toward position Z, OLSR requires $5sec$ to discover a 2-hop path to D after the direct link A-D is lost; while it needs $10sec$ when the path in the connection increases from 2 to 3 hops. AODV introduces a delay of $2sec$ for the first topology change, and $7sec$ for the second one. Both protocols do not introduce any additional delay in the reverse path (from Z to W position). In the End swap scenario, OLSR introduces a

PDR	Roaming node	End node swap	Relay swap
AODV	0.87	0.67	0.60
OLSR	0.83	0.77	0.66

Table 5.2: Overall Packet Delivery Ratio.

delay of $15sec$ when the topology changes from a fully connected (each node see all the others) to a topology of three hops. In the same topology change, AODV experiences a delay of $10sec$ but it also introduces a similar delay to move from the starting configuration to a fully connected topology. In the last scenario, during the relay movement, OLSR introduces a delay of $15sec$ for the routing table reconfiguration, while AODV requires $11sec$ to discover a new route to the destination.

5.6 Conclusions

The aim of this experimental testbed was investigating the performances of two routing protocols for small scale ad hoc networks, i.e. network of 2-4 hops size up to 8 nodes falling inside this *ad hoc horizon*. Thus, we selected two robust implementation of the proactive OLSR and the reactive AODV. Firstly, we conducted a qualitative analysis on them, checking their state of implementation and validating their functionality. Then we analyzed their performance with a quantitative analysis, setting up a real Ad Hoc network and comparing them in different scenarios and environments. Our results in this small scale Ad Hoc network point out severe QoS problems, mainly when using AODV due to the reactive nature of the protocol, and indicate that, with a proactive protocol: *i)* the response times are much better (200 ms vs. 2 sec when pinging 2-hops neighbors), *ii)* the protocol overheads, at least inside our small network, are not heavy (i.e., in the worst case 1.2 KBps), and *iii*) the success of packets delivery is higher. Note that the conclusions of the work represent only the measurements of our selected testing environment. A different environment would produce different results. Nevertheless, our results indicate that OLSR performs better than AODV protocol. Furthermore, when considering higher level protocols on top of Ad Hoc test-bed, e.g. FreePastry [D8][D16][BCDP05], benefits in using a proactive approach are more evident. Moreover, the advantages increase when considering a cross-layer architecture exploiting interactions between proactive protocols and enhanced p2p platforms (CrossROAD) (see [Del05]). For details of these results refer to [Del05] [BCDG05].

6 Medium scale multi-hop Ad Hoc network

In this chapter we present experimental results of an innovative testbed on a 23 nodes MANET with particular attention to routing and middleware performance, representing an extension of the work in small scale network. Specifically, a proactive and a reactive routing protocol have been analysed before experimenting a new optimized p2p system based on cross-layer interactions with a proactive routing protocol. The experimental analysis on this medium-scale MANET has been carried out in the framework of the FET-IST MobileMAN project. Main results show that the proactive approach does not negatively influence system performance, even better it supports upper-layer protocols sharing complete network topology information.

6.1 Introduction

In this chapter we report the experimental activity on medium scale Ad Hoc network. This work represents an extension of the work discussed in the previous chapter on networks of up 8 nodes. This extensive experimentation was carried out in Pisa, setting up in the CNR campus a multi-hop network involving up to 23 nodes. The novelty of this work is represented by testing different protocol solutions in networks of such dimension. Even though we mainly consider static scenarios, this testbed, together with the APE testbed [Dep], represents one of the largest tesbed on multi-hop network.

The experimentation focused on the analysis of different layers of the protocol stack in order to compare results of a legacy-layer architecture with those of a cross-layer architecture. Specifically, we mainly focused on:

- a comparative analysis of two different routing protocols (OLSR and AODV) and the evaluation of their performance on static and mobile scenarios;
- a comparative analysis of two different middleware platforms (Pastry for the legacy architecture and CrossROAD for the cross-layer architecture).

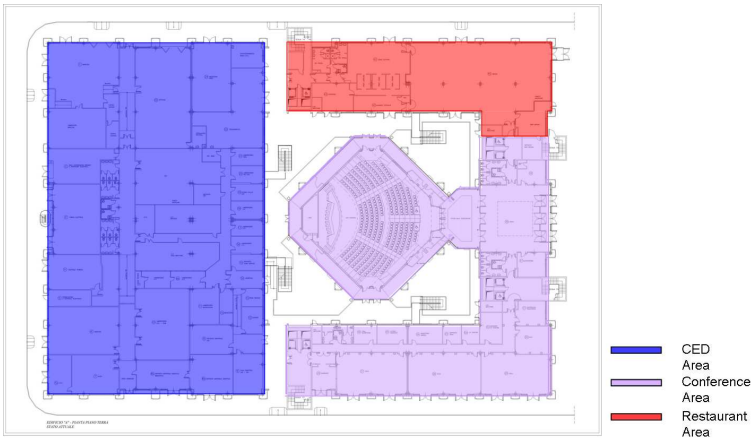


Figure 6.1: Experimental Area.

In the following sections we refer only to experiments at the network layer. The interested readers refer to [D16] for details about middleware performance. Referring to the routing experiments, we study the performance of OLSR and AODV considering the overhead introduced by them, the packet loss suffer at the application layer and the delay introduced in data transfer as major indices.

6.2 Experimental Environment

All the experiments took place at the ground floor in CNR campus in Pisa. Since more than 20 nodes were involved in this experimentation, a wide area has been used for testing a medium scale network. Hence, in addition to the CED Area used in previous testbed (see Chapter 5), the Conference Area located in the adjoining building was also used (see Figure 6.1). The structural characteristics of these buildings strictly determine the transmission capabilities for nodes of a wireless network located within. Rooms are generally delimited by masonry padding walls situated between reinforced concrete pillars; in addition, in the CED area some locations are separated by either "sandwich panels" of plastic materials which don't reach the height of the ceiling or metal panels till the ceiling. Wireless links are also influenced by the presence nearby of Access Points and measurement instrumentations which introduce quite a lot of noise. Moreover, about 30-40 people work in this floor every day and get around from office to office or towards service areas with coffee machines, toilets, etc. This makes the transmission coverage characteristics of the floor and the stability of the links modify

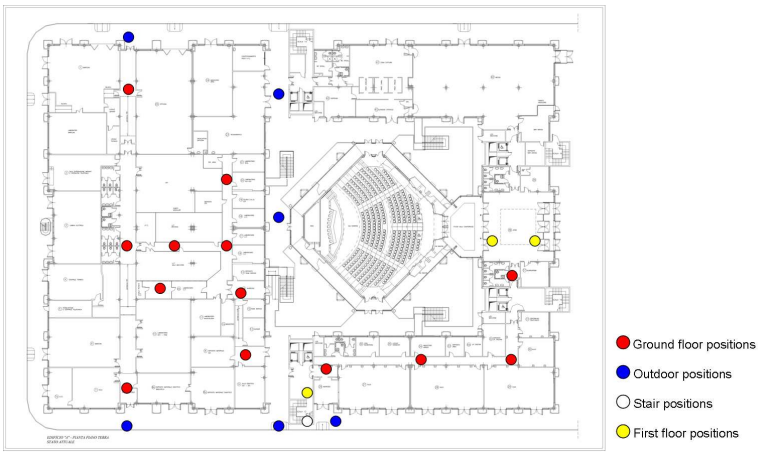


Figure 6.2: Physical position of nodes.

continuously and in an unpredictable manner. For this reason all the experiments were executed during Saturday or non-working days to reduce human interferences maintaining a realistic environment to test an ad hoc network.

6.2.1 Devices and Software

Devices used for these experiments were laptops running Linux with different hardware capabilities. They were equipped with wireless cards compliant to IEEE 802.11b standard working at the constant data rate of 11Mbps. The most sort of laptops was equipped with an integrated wireless card, while for the others PCMCIA cards were used. The variety of devices caused appearing/disappearing of some links in different experiments, depending on the power of wireless cards. As in the experimentation on small-scale ad hoc networks, the main goal of this work was to test different implementations of protocols. In particular more recent software versions of the two selected routing protocols were considered. In this phase we used UU-AODV v.0.8. 2, developed by Uppsala University (Sweden), as reactive protocol. On the other hand we used UNIK-OLSR v.0.4.8 2, developed by University of Oslo (Norway) as proactive protocol.

During the experiments we used the simple `ping` utility to evaluate delays and packet loss, while we used a distributed application on top of a p2p system to evaluate the overhead introduced by routing protocols in case of a more realistic scenario involving a complete MANET architecture. Note that, due to the reactive nature of AODV, we

need any sort of application traffic to establish paths between distant nodes, otherwise only Hello packets are exchanged.

6.2.2 The network topology

The first step to set up the Ad Hoc network and start investigating software features was configuring the network topology. We had 23 nodes to be distributed inside the CNR campus to carry out a multi-hop ad hoc network as much large as possible. For this reason we used an heterogeneous environment consisting of indoor and outdoor spaces since not all buildings are strictly connected between them. We started from the same configuration used in the experimental session with 12 nodes, explained in Chapter 5. Since we used a greater number of laptops with different capabilities (also for the transmission range of wireless cards), a new measurement of the link connectivity had to be done. In this case the interested area was extended from the CED area to the nighborhood of the conference area as shown in Figure 6.2. Most part of nodes (17) was located inside buildings. In particular 13 at the ground floor (red circles), three at the first floor (yellow circles), and one on the stairs (the white circle). The last six nodes were located outside the buildings (blue circles) along the street or the corridor between the involved buildings. In order to verify the coverage area of every device, each node started running UNIK-OLSR for five minutes storing the kernel routing table in a log file every second. Then, we analysed the set of 1-hop neighbors of each node to define the final network topology. Considering a large multi-hop ad hoc network we could test and evaluate features and performance of a complete MANET architecture. For this reason, since many devices had a wireless card with a high trasmission power, we had to reduce it on single nodes (if allowed by the driver of the wireless card) to remove some redundant links. We repeated this procedure many times to check if the obtained configuration was stable. Fig. 6.3 shows the final network topology, where straight lines point out the presence of stable links (two nodes directly see each other), dashed lines show the presence of weaker links (the communication between two nodes is affected by a considerable packet loss). We thus obtained a multi-hop MANET of 23 nodes with the maximum extension of 8 hops. To simplify the explanation of single experiments, we referred to the network topology through the graph illustrated in Fig. 6.4.

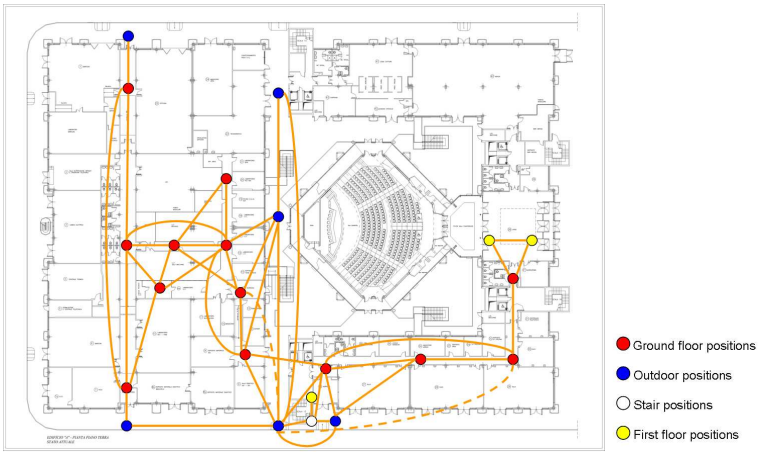


Figure 6.3: Network Topology.

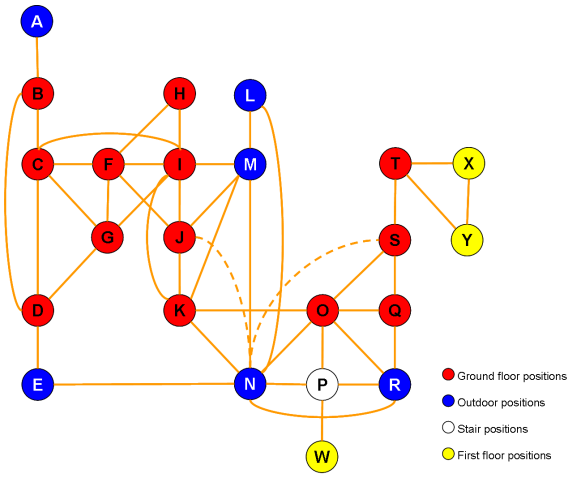


Figure 6.4: Topology graph.

6.3 Routing Experiments

The second step was investigating the performance of OLSR and AODV routing protocols for MANET in *static* and *mobile* scenarios. We analysed several parameters to make a comparison between them. We focused on:

- the network overhead introduced by routing messages
- the packet loss suffered at the application level
- the delay introduced in data transfer

In case of mobility, we reduce the analysis to packet loss and average delays, since it is interesting to evaluate the impact of network reconfigurations due to topology changes on the system performance.

The description and the analysis of the performed experiments divided into static and mobile scenarios are detailed in the following sections.

a. STATIC SCENARIO:

- Experiment 1: all nodes started running the OLSR protocol at the same time. After 30 sec the external nodes A and Y started pinging all the other nodes in the network using a random sequence. Each ping operation lasts for 1 minute. The two sequences used for the ping operation were different and precisely:
  - Pinging sequence for node A: R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D.
  - Pinging sequence for node Y: E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q.

  At the end of the ping operation, each node kept running OLSR for other 30 sec and then stopped. The whole experiment lasted 23 minutes.
- Experiment 2: all nodes started running the OLSR protocol at the same time. After 30 sec used to stabilize the network topology, all the nodes started pinging all the other nodes in the network using a selected sequence. First of all, we ordered the nodes in a random sequence independently of their physical positions.

  The reference sequence is *A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D.*

PINGER	PINGING SEQUENCE
A	R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D.
B	L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K.
C	T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S.
D	A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W.
E	F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y.
F	M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E.
G	P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J.
H	K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M.
I	J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O.
J	G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I.
K	B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H.
L	O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B.
M	X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F.
N	Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T.
O	I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L.
P	W, D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G.
Q	Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N.
R	S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A.
S	C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R.
T	N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C.
X	H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q, Y, E, F, M.
Y	E, F, M, X, H, K, B, L, O, I, J, G, P, W, D, A, R, S, C, T, N, Q.
W	D, A, R, S, C, T, N, Q, Y, E, F, M, X, H, K, B, L, O, I, J, G, P.

Table 6.1: Sequence for the Ping operation used by each node.

As first destination each node chooses its next node in the sequence. It pings continuously that destination for 1 minute and then moves to the next one in the sequence. It executes the same operation for each node in the sequence. For example, node X starts from node H and ends with node M. Table 6.1 shows the complete sequences used by each node during the ping operation. Nodes ran the routing protocol for other 30 sec before stopping. The whole experiment took 23 minutes.

- Experiment 3: in this case we used ADOV as routing protocol; the methodology and the duration of the testrun are equal to the previous experiment. The sequences used in the ping operations are the same of Table 6.1 in order to have a direct comparison between the two routing protocols.
- Experiment 4: all the nodes are synchronized and started running the routing protocol at the same time. After 30 seconds used to stabilize the network topology, each node ran the distributed application on top of the Pastry middleware and, more precisely, joined the overlay using a random sequence and maintaining the Pastry overlay for 4 minutes.

b. MOBILE SCENARIO: We performed three different types of experiments, changing the number of mobile nodes. In the first experiment referred as **Roaming node**, a node moves along the network. In the second experiment two central nodes exchange their position, we refer to it as **2-Central node swap**. Finally, in the third experiment 4 central nodes rotate their positions (**4-Central node swap**). A detailed description of the performed experiments follows.

- **Roaming node:** all the nodes are static except the *"roaming node"* Y that moves along a fixed path, crossing the entire network. The reference scenario is shown in Figure 6.6. All the nodes start running the routing protocol at the same time and, after 30 seconds, node Y starts pinging node A (see Figure 6.6) continuously for 380 seconds. After 1 minute from the beginning of the ping operation, the pinger Y starts moving along the corridor with a speed of about 1 m/sec reaching the position of other nodes in the order A-C-I-K-O-S-X. Each step requires a different time interval due to the physical distance of nodes as explained in Table 6.2. Once it has reached the last position near node X, it immediately moves in the opposite direction following the reverse path and maintaning the same speed as in the forward path. After having reached the starting position, it keeps pinging

Distance to cover (nodes)	Required time interval (x sec)
A-C	30
C-I	15
I-K	15
K-O	15
O-S	30
S-X	30

Table 6.2: Required time (sec) needed to cover a distance between nodes in the Roaming node scenario.

node A for another minute and then it stops. The whole experiment lasts 410 sec.

- **2-Central node swap:** in this experiment the central nodes J and N change their position during a continuous ping operation from two external nodes. More precisely, all the nodes start running the routing protocol at the same time. After 30 seconds the external node Y starts pinging node A continuously for 210 sec. At t=90sec nodes J and N (two central nodes) start moving and swap their positions after 30 seconds (t=120sec). Then they remain in the new configuration until the end of the experiment. The experiment ends at t=240sec.

- **4-Central node swap:** in this case the four central nodes J, M, O and N change their positions in clockwise manner during a continuous ping operation between the two external nodes Y and A. More precisely, all the nodes start running the routing protocol at the same instant and, after 30 seconds, the external node Y starts pinging node A continuously for 300 seconds. After 1 minute from the beginning of the ping operation, the four nodes started moving in turn reaching a new position in the network in 30 seconds and remaining in this new location until the end of the experiment. Since the mobile nodes are four, we identified four different events taking place in sequence, one after the other. More precisely, in Event 1 (see Figure 6.5) N is the mobile node and it reaches the position of node J; in Event 2 node J moves towards the location of node M; in Event 3 node M reaches the position of node O; finally in Event 4 node O moves to the initial location of node N. At the end of the four events nodes keep running the routing protocol until the end of the experiment.

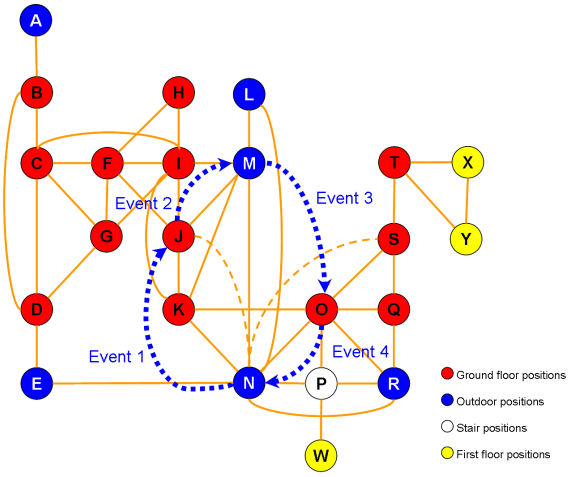


Figure 6.5: 4-Central nodes Swap scenario.

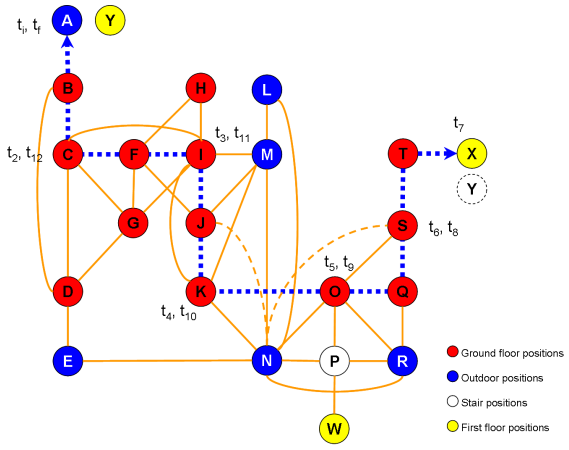


Figure 6.6: Roaming node scenario.

6.4 Static Scenario

In this section we compare OLSR and AODV referring to results obtained in experiment 2, 3 and 4. The reference scenario is shown in Fig. 6.4. We evaluate their performance basing on all the three indices as explained in the previous section. Specifically, in order to evaluate the overhead introduced by the routing protocols on the network we consider an experiment that involves not only the routing layer but also the p2p2 system. In this way, the traffic generation on top of the routing protocols is more complete than a single `ping` utility due to the presence of TCP and UDP connections. This results in a more realistic evaluation of the bandwidth utilization. Referring then to performance evaluation of the routing protocols in terms of overall packet loss and delay suffered in the network, we simplify the proposed scenario using a lighter data traffic as the ICMP traffic. In fact, the `ping` utility is sufficient to measure the RTT of a small application packet and it points out how a simple application can suffer such low performance in a medium scale MANET.

6.4.1 Overhead analysis

In order to evaluate the overhead introduced by the routing protocols on the network we referred to the experiment 4. Figure 6.7 presents the total overhead introduced by OLSR and AODV as a function of time. The curves are obtained averaging the total control traffic generated and forwarded by each single node over the number of nodes taking part to the experiment. As it clearly appears from the picture, OLSR and AODV have different behavior. Specifically, the proactive protocol introduces an overhead of about 600 Bps in the starting phase (first 40 sec), then its load decreases to 400 Bps for the next 50 sec, finally a new steady state is achieved till the end around 250 Bps. On the contrary, AODV reaches a steady phase with a load of 400 Bps between 40 and 80 sec, then its load doubles with a peak of about 750 Bps around 90 sec, finally it stabilizes again varying from 300 and 500 Bps till the end of the test-run. OLSR introduces a higher overhead during the starting phase, then after a second phase in which its performed throughput coincides with AODV throughput, OLSR performs better for the rest of the experiment. In fact, AODV peaks of traffic are mainly due to several discovery procedures to maintain the overlay network. However, in an overall view the overhead of both protocols falls in a range of [200, 700]Bps. These results confirm that also in medium scale network the overhead introduced by routing protocols, either using proactive and reactive approaches, doesn't affect negatively the system performance, indeed it reduces the available 802.11 bandwidth only of a small

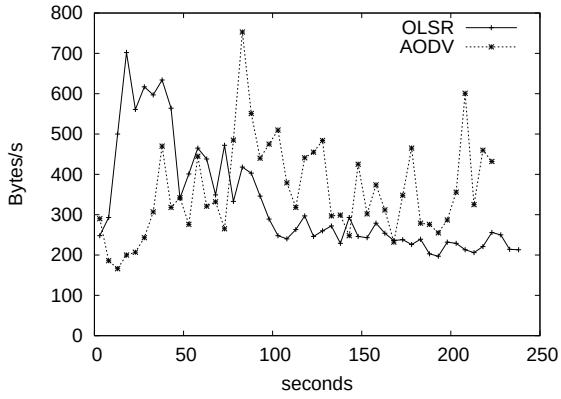


Figure 6.7: Average overhead introduced by OLSR and AODV.

quantity.

Figure 6.8 and Figure 6.9 show the average overhead introduced by OLSR and AODV, respectively, for different nodes depending on their position in the network. Since there are 23 nodes in the network, in order to obtain a sharper graph only some of them are plotted. Referring to OLSR results (see Figure 6.8), note that node C, J, and O observe the highest load since they are better connected with the rest of the network with 5 or more neighbors each, instead nodes E, H, and T have an intermediate load since they have less neighbors. At last nodes A and W obtain the lowest traffic around 100 Bps because they are located in marginal position since they are leaves for the network (see figure 6.4). For AODV (see Figure 6.9) node F and O have the highest throughput, instead an intermediate load is performed by nodes L and B. In this case the lowest load is experienced by node E that, even though it is not a leaf, has a marginal location with only two neighbors. This is mainly due to the reactive nature of AODV that makes the network load also dependent on the application traffic.

6.4.2 Packet Loss analysis

In the following paragraphs we analyse network's performance taking experiment 2 and 3 as reference scenarios. To evaluate the overall packet loss suffered at the application level, we averaged all Ping operations between couples of nodes at x-hop distance. Table 6.3 shows the percentages obtained for different number of hops. Looking at the obtained results we can notice that OLSR performs better than AODV. In particular, OLSR delivers almost all packets at 1-hop distance, suffering a packet loss of [15%,

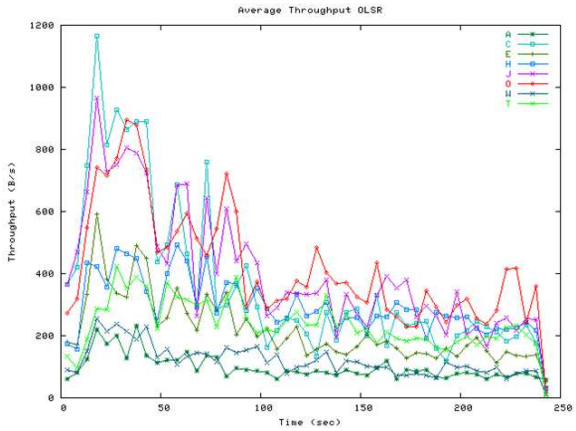


Figure 6.8: Overhead introduced by OLSR for different nodes.

HOPS	1	2	3	4	5	6	7
OLSR (Packet Loss)	5%	15%	28%	35%	45%	52%	67%
AODV (Packet Loss)	20%	15%	51%	61%	67%	86%	89%

Table 6.3: Overall Packet Loss for different number of hops

45%] for nodes distant [2, 5] hops. Finally it delivers less than 50% of the application traffic with connections of 6-7 hops. On the contrary, problems with the reactive protocol are more evident. AODV does not properly work even nearby, achieving 20% packet loss even at 1 hop. Its performance further decreases to 50% at a distance of 2-3 hops, drastically degenerating (more than 85%) beyond 5 hops.

Another observation can be derived taking into account results from the indoor string topology as explained in Chapter 5. To summarize, in that scenario the OLSR performance was acceptable in all Ping operations towards each node in the string, instead AODV loses 50% of ICMP packets while communicating with nodes at 3-hop distance (see Table 5.1.). On the contrary, in this medium scale environment we observe greater percentages of undelivered packets also with few hops. Possible explanations of these results are the different network size (small vs medium) and the complexity of the experiment (1 Ping operation vs 23 simultaneously Ping operations). In particular with concurrent connections each node can act as destination for a Ping operation and also as router for another one. Thus the probability of collision at MAC layer is increased considerably causing also several route failures.

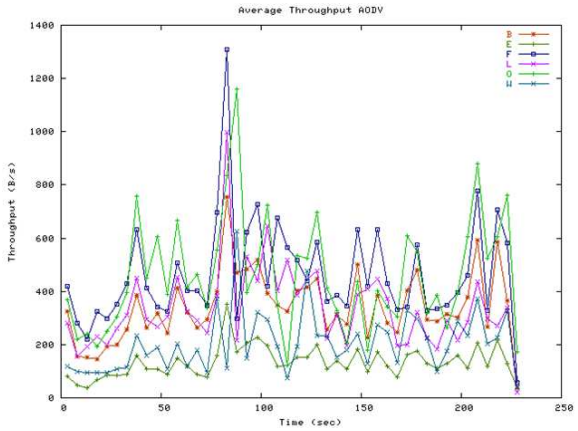


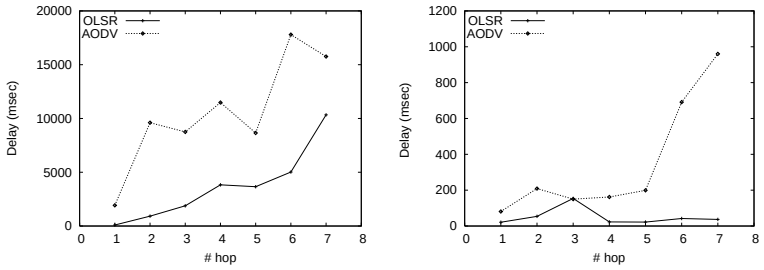
Figure 6.9: Overhead introduced by AODV for different nodes.

6.4.3 Delay analysis

To evaluate the delay introduced by the routing protocols, we measured the end-to-end latency for completing a Ping operation between couples of nodes. In particular, we consider two different delays in the network:

- the Average delay to deliver the first successful ICMP packet to a selected destination
- the Average delay to deliver all the other packets of the ping operation

Each value is avereged over couples of nodes distant x nodes. Figures 6.10 (a) and (b) present the obtained results (expressed in msec) for different number of hops. Figure 6.10(a) shows the average delay needed to complete successfully the first ping operation. As it clearly appears, OLSR curve is obviously lower than AODV curve due to the different nature of the routing protocols. In particular OLSR increases almost linearly up to 6 hops, and then it doubles at 7 hops. This is mainly due to the network instability that implies some network reconfiguration and the consequent increase of the delays. On the contrary, AODV curve is a step function. It needs about 2 seconds to discover routes to 1-hop neighbors, about 10 seconds for nodes in the range of 2-5 hops distance, and finally [15, 17] seconds to discover valid paths towards nodes distant 6 hops and more. These high delays are due to several attempts performed in the route discovery process. In fact, we have seen that each node makes about 5-6 attempts in order to discover a valid route to its destination. Looking at Figure 6.10(b), note that



(a) Average Delay on the first successfull ICMP packet.

(b) Average Delay on delivered packets.

Figure 6.10: Average Delay suffered by OLSR and AODV for different number of hops.

OLSR requires delays in the range of [20msec, 60msec] independently from the number of crossed hops, while AODV introduces higher delays. More precisely, AODV Ping connections perform the following delays: about 200 msec when they are shorter than 6 hops, about 700 msec towards nodes distant 6 hops and about 1 sec toward nodes at 7-hop distant. From the log files of the experiment we noticed that AODV is not able to maintain the first discovery path to the same destination for the entire connection, but it requires 1 or 2 attempts in order to re-establish a valid route to the destination. This is the main reason of low performance of AODV in the static scenario.

6.5 Mobile Scenario

In this section we compare routing protocols considering mobile scenarios with nodes that change positions during the entire last of the experiment. Specifically, we performed three different types of experiments, changing the number of mobile nodes, as explained previously. In all the performed experiments each mobile node moves in the network with a speed of about 1m/s, thus we are interested in investigating low mobility scenarios. We analyse their results with particular attention to the packet loss and the introduced delays for network reconfiguration.

6.5.1 Packet Loss analysis

Analyze the packet loss in the Roaming node experiment, OLSR performs a packet loss of 25% while AODV delivers only 50% of packets. Examining the log files, we

observe that for OLSR the packet loss mainly occurs in the way back between node X and Q (see Fig. 1.2.3). The gap in the packet delivery corresponds to the time interval in which Y's routing table get emptied. Node Y recovers the route to the destination only in the proximity of node Q, losing all the packets sent in that time interval. AODV instead loses all ICMP packets when node Y goes beyond node D. In this case, when the connection becomes longer than 4 hops none of ICMP packets reach the destination.

Increasing the complexity of the proposed scenarios the performance of the routing protocols worsen: only a little percentage of Ping operation is successfully completed. Two are the main causes:

1. the ping operation between node Y and A is a 7-hop connection. In the previous section we have shown how the network's performance decreases with long connection in static network.

2. the complexity of the network increases adding nodes mobility. To better understand, let us consider the similar scenario analysed in the small scale network (see Chapter 5). Also in that case more than 40% of packets were lost with both routing protocols.

Hence, these two factors cause the network breakdown.

6.5.2 Delay analysis

To evaluate the delay introduced in the network in the three scenarios, we investigate the time needed to update the routing tables with a valid route to the destination after topology changes, independently from the correct delivery of packets. All the values are measured as the RTT of ICMP packets sent by node Y.

Starting from the Roaming node scenario, OLSR performs a delay in a range of [4-9] sec to update routing table each time the connection becomes 1 hop longer than the previous one. AODV suffers delays between [4-10] sec to discover routes to node A from 1 to 4 hops long. The route discovery process takes more than 10 seconds when Y goes beyond node O (in this case routes are 5 or 6 hops long), but since valid paths are maintained in the routing table only for few seconds no ICMP packets are successfully delivered. Both protocols do not introduce any additional delay in the reverse path.

In the 2-Central node Swap they lose the path to the destination as soon as the mobile nodes start moving. The central nodes' exchange needs 30 seconds. OLSR is

able to reconfigure a 8-hop route only after 5 seconds from the end of the exchange, instead AODV requires other 90 seconds to discover a valid route to node A.

In the 4-Central nodes scenario OLSR and AODV suffer higher delays for each event. Specifically, OLSR loses the routing table's entry to node A from 1 to 4 seconds after the start of each event and it needs delays between 5 and 50 seconds to reconfigure properly the routing table. AODV becomes aware of the new event after 3-8 seconds from the beginning and needs from 5 to 10 seconds to re-establish a valid route with also a peak of 60 seconds when all the network changes are completed. Note that, in case of AODV, most of the discovered routes are stored in routing tables only for few seconds, thus the discovery process is repeated frequently. This is due to the nature of AODV that stores also unstable paths in its routing tables. Consequently, ICMP packets are not correctly received by the destination decreasing the overall system performance. On the contrary, even though OLSR performs on average higher delays for network reconfigurations due to a slow propagation of topology changes, its new paths are maintained in the routing tables till the beginning of the new event. In fact, the proactive protocol looks for more stable routes and this allows the source to send and receive application data successfully.

6.6 Conclusions

We really examined system features and performance in real conditions, setting up a wireless network of 7-8 hops size with up to 23 nodes. We performed an extensive set of experiments comparing AODV and OLSR routing protoocols in static and mobile scenarios. The overhead analysis confirms that also in medium scale networks the use of a proactive protocols doesn't reduce the system performance since it introduces an overhead of the same order of AODV. In addition with a OLSR a higher amount of data are delivered successfully even through long connections. Referring to delay introduced in the network, OLSR time responses are much better than AODV. Finally considering the mobile scenario, even though OLSR is slower than AODV to propagate the network changes, it performs better than the reactive protocols discovering more stable routes and hence delivering more application data.

7 Towards a further optimized scalable proactive routing protocol: Hazy Sighted Link State

Our experimental results on real Ad hoc networks highlight that, in contrast with MANET community, the use of a proactive protocol does not penalize the system performance. These results encourage to identify a routing protocol suitable for multi-hop networks in terms of scalability, performance and efficiency in the class of proactive protocol. As proved in [SMSR02] [SSR01], the **Hazy Sighted Link State (HSLS)** routing protocol [SR01] exhibits good performance in term of scalability. In the framework of this thesis, an enhanced version of the HSLS routing protocol has been designed and developed. This chapter will provide a description of the protocol together with the added features.

7.1 Introduction

The experimental evaluation presented in previous chapters shows that, in contrast with MANET community, the use of a proactive protocol as OLSR does not penalize the system performance. Our results on small and medium scale environments point out that having in advance a knowledge of the network topology seems more suitable in the Ad Hoc architecture, either in legacy architecture and in the cross-layer architecture. In addition, in a cross-layer architecture, the richer amount of information collected by proactive protocols can be exploited, for purposes other than routing, at other layers. Overheads cannot be evaluated in isolation focusing on a certain level, but new cross-layer metrics must be applied. An example of this concept could be the service location for middleware: once the routing protocol has discovered the topology of the network, the middleware can use it to identify the node that provides a certain service without performing a new route discovery. We can identify many other examples that clearly indicate the advantages for a node to have knowledge of the network

topology. The result is that proactive approaches may better satisfy the self-organizing requirement of general-purpose Ad Hoc networks.

In the framework of this thesis, we worked to identify, if possible, a routing protocol suitable for multi-hop networks in terms of scalability, performance and efficiency, but also able to provide a rich set of information about the network that can be exploited to improve the other protocols of the cross-layer architecture. A very promising indication has been recently provided by the theoretical analysis presented in [SMSR02] [SSR01]. Here, the authors develop an analytical framework to evaluate the protocol scalability taking into consideration, in addition to the proactive and reactive overheads, also the effect introduced by the *sub-optimality* of routes, accounted for as the additional bandwidth required for using a sub-optimal path. From this perspective, the authors show that a simple **Link State protocol** with **Limited dissemination** scales better than more complex hierarchical protocols and hence this class of protocols can be an efficient routing alternative for large-scale ad hoc networks. In such protocols, the link-state updates (LSU) are sent through the network controlling the *scope* and the *frequency* of floods. In particular routing information is propagated to the network nodes with a frequency that decreases with the distance. As a result, each node builds a "self-centered" topology view, which becomes *hazy* as the distance grows. This analytical results are very important since they indicates that link-state routing strategies based on limited dissemination of state information not only provide several qualitative advantages when used in a cross-layer architecture, but also provide effective solutions from a quantitative standpoint. Even though this result cannot be apparently intuitive, it can be explained by observing that nodes that are far away do not need to have precise topological information to make a good next hop decision. As pointed out in [BCSW98], the inaccuracy in the topological information is balanced by the distance effect: "the greater the distance separating two nodes, the slower they appear to be moving with respect to each other". Hence, the required accuracy of the location information decreases with the distance from the node. Examples of this approach are hierarchically link state [RS98], FSR [PGH00], GSR [CG98]. In particular, as proved in [SMSR02], the best among them is the **Hazy Sighted Link State (HSLS)** [SSR01] [SR01] in which routing updates are flooded in the network with a binary exponential sequence. Hence, taking into account these analytical studies and our experimental results, a prototype of the HSLSL protocol has been developed from scratch. This decision is twofold:

- any HSLS implementation is not available

- an implementation from scratch allows us to add new *features* to the prototype, increasing the value of the work

In addition to the basic functionality, the HSLS protocol has been enhanced with:

1. a mechanism to guarantee the reliabilty of LSU packets with any introduction of additional control overhead

2. a module that allows cross-layer interactions, thus resulting in an easy integration with the cross-layer prototype

In the following sections an overview of how HSLS works and a complete view of the system architecture with tecnical details are described.

7.2 Hazy Sighted Link State Routing Protocol (HSLS)

In Hazy Sighted Link State protocol [SSR01] [SR01], similarly to others link-state proactive protocols, each node sends periodic route updates (LSU packets) containing its one-hop neighborhood, allowing other nodes to have a complete view of the network; but, as explained previously, to reduce the overall control overhead, and have good scalability properties, there is a restriction of the scope of routing updates in time and/or space. Specifically, periodically each node broadcasts the list of its 1-hop neighbors over the network with a frequency that decreases with distance. Thus each node has a partial knowledge of the topology (i.e. not real-time uploaded); its topology view is more precise in the nearby and more *hazy* far from a node. This strategy, if coupled with a forwarding strategy that in each node independently selects the next hop towards a destination, is expected not causing any major impact on the selection of the path towards the destination.

In HSLS periodic updates occur at discrete time interval. A node collects one or more link status changes in a single packet which is transmitted only at particular time instants that are multiple of te seconds. Furthermore, the dissemination of this information is controlled by specifying the area of the network in which the Link State Update (LSU) will be distributed. This control is implemented by setting the TTL (*Time To Live*) field of the LSU packets thus limiting the number of hops the packet will perform in the network. More precisely, let us indicate with 0 the time instant at which a node sends a *global LSU* (packet that travels over the entire network), providing a complete knowledge of link changes to all nodes in the network, then a node wakes up:

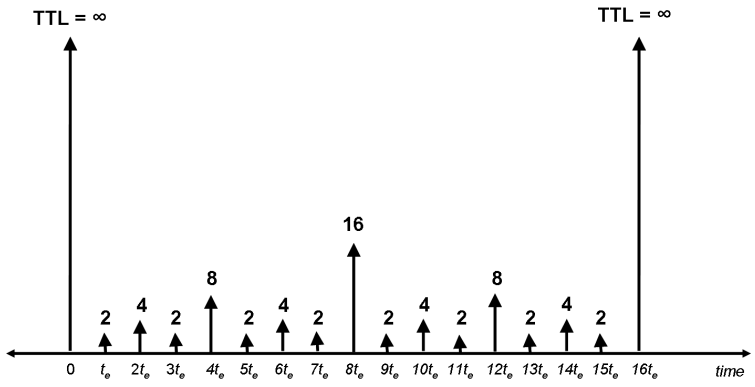


Figure 7.1: LSUs generation process in high mobility scenario.

- every t_e seconds, and transmits an LSU with the TTL field equal to 2 if there has been a link status change in the last t_e seconds;
- every $2 * t_e$ seconds, and transmits an LSU with TTL field equal to 4 if there has been a link status change in the last $2 * t_e$ seconds;
- ...

In general, a node wakes up every $2^i * t_e$ seconds (with i = 0, 1, 2, 3) and sends an LSU with $TTL = 2^{i+1}$ if there has been a link status change in the last $2^i * t_e$ seconds. If the value 2^{i+1} is greater than the distance from this node to any other node in the network, the TTL field is set to infinity (i.e., a global LSU) and all counters and timers are reset.

Figure 7.1 and 7.2 show some examples of HSLS's LSUs generation process. In Figure 7.1 we assume a high mobility scenario in which a link change occurs every te seconds, and hence LSU packets, represented by vertical arrows, are sent in the network every t_e seconds; The height of the arrow represents the TTL value. On the other hand, the Figure 7.2 represents a lower mobility scenario in which there is not a link change every t_e seconds. Specifically, changes are marked with an 'x' on time axis and, as it appears in the figure, that LSU packets are less frequent, and it may happen that some updating points are skipped if in the last interval no change occurred. The above approach guarantees that 2^{i+1} hops neighbors from a tagged node will realized topology changes at most after $2^i * t_e$ seconds. Figure 7.3 shows the latency in propagation of link state information performed by HSLS protocol.

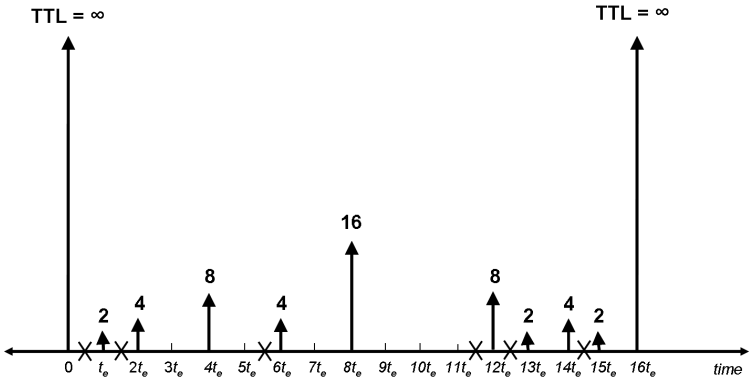


Figure 7.2: LSUs generation process in low mobility scenario.

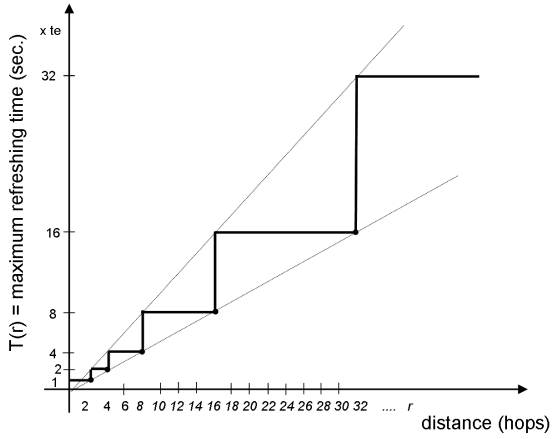


Figure 7.3: Maximum refresh time as a function of distance from link event.

7.3 HSLS enhancements

7.3.1 Reliable HSLS

Generally, link state routing protocols don't provide any form of acknowledgement for the control packets because link state information is spread in the network using a broadcast process. Moreover, the 802.11 MAC protocol delivers broadcast packets in an unreliable fashion, i.e. without an explicit acknowledgement. Therefore if a node sends an LSU and it is lost due to collisions or channel interference, that packet is never retransmitted neither at network, nor at link layer. To guarantee a reliable delivery of LSU packets, a reliability mechanism should be added to HSLS. Hereafter, we present our approach to improve HSLS reliability in an efficient way.

Instead of introducing additional control packets, broadcasted LSUs are used as acknowledgements of the LSU previously sent. In order to record the history about LSUs sent and received from the network, each node stores them in two caches: *sentLSUcache* maintains information about LSUs generated by the node itself; instead *receivedLSUcache* stores LSUs coming from other nodes.

Referring to Figure 7.4, let's suppose that a node X is a originator of an LSU packet; after its reception, a generic node A forwards it with $TTL = i$. Node A will consider an ACK for this LSU packet, `ACK_LSU`, any LSU packet it will receive from its 1-hop neighbors with originator node X and $TTL = i - 1$. More precisely, the following procedure is executed on each node to guarantee reliability of the LSU dissemination process:

1. Node A sends an `LSU` with $TTL = i$;

2. Node A counts the number of `ACK_LSU` packets received from its 1-hop neighbors during a fixed time window $T \ll t_e$; in particular, it stores the number of received `ACK_LSU` for each `LSU` into *sentLSUcache* if it is the originator of this `LSU`, or into *receivedLSUcache* otherwise;

3. If the number of `ACK_LSU` $\geq$ ACK_threshold, it can be assumed that the `LSU` sent by A has been correctly received from most of its neighbors; on the other hand, if the number of `ACK_LSU` $<$ ACK_threshold node A has to retransmit the same `LSU` again;

4. An explicit ACK_LSU is sent in the last hop. The explicit ACK_LSU, named **ACK_exp**, is a copy of the received LSU without message body.

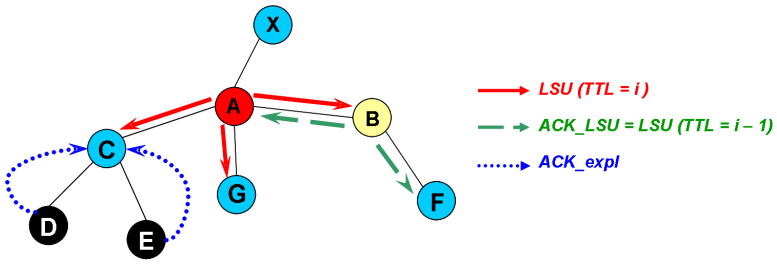


Figure 7.4: ACK_LSUs generation process.

The fourth point assures the uniformity in the reliability process also in the last hop. Suppose that node C (in Figure 7.4) forwards an `LSU` packet with $TTL = 0$; nodes D and E receive and process it, but they will not forward it anymore because of the TTL value. Consequently, the timeout T at node C would expire without having received any acknowledgment. Thus, node C would make a wrong decision to retransmit that `LSU` packet. To avoid this, node D and E will send an explicit ACK_LSU (`ACK_exp`) to the sender C as confirmation of their previously correct reception. In this way only negligible additional control traffic is added to the original protocol.

The ACK_threshold should be a value between 1 and the number of 1-hop neighbors. For instance, using the lowest value 1 we guarantee that at least one neighbor has received the original LSU propagating route updates in one direction. We are currently investigating how the ACK_threshold value affects the HSLS's behavior in terms of both overhead and reliability.

7.3.2 Cross-layer interactions with HSLS

One of the major challenges in the research on mobile ad hoc networks is to form a functional network with good performance and, at the same time, able to communicate with the rest of the Internet. The IETF MANET WG proposes a view of mobile ad hoc networks as an evolution of the Internet. It consists of a layered architecture with an IP-centric view of the network. The use of the IP protocol simplifies MANET interconnection to the Internet, also guaranteing the independence from wireless technologies [MC04]. However, current results show that the layered approach is not equally valid in terms of performance [GW02]. The layered approach leads the research efforts mainly to target isolated components of the overall network design (e.g., routing, MAC, power control). Each layer in the protocol stack is designed and operated separately, with interfaces between layers that are static and independent of the individual network

constraints and applications. However, in a MANET some functions cannot be assigned to a single layer. For example, energy management, security and cooperation, quality of service cannot be completely implemented in a single layer but they are developed by combining and exploiting mechanisms implemented in all layers.

Two solutions allow to export informations between not-adjacent layers. At one end, solutions based on layer triggers are still compatible with the principle of separation among layers. Layer triggers are pre-defined signals to notify some events to the higher layers, e.g., failure in data delivery, thus increasing the cooperation among layers. On the other end, solutions based on a full cross-layering-design represent the other extreme in order to exploit, in the protocols design, layers' interdependencies to optimize the overall network performance. These approaches allow the possibility of protocols belonging to different layers to cooperate by sharing network-status information still maintaining layers' separation for protocols design. In this case, control information is continuously flowing top down and bottom up through the protocols' stack and a protocol behavior adapts both to higher and lower protocols' status. For example, the physical layer can adapt rate, power, and coding to meet the requirements of the application given current channel and network conditions; the MAC layer can adapt based on underlying link and interference conditions as well as delay constraints and bit priorities. Adaptive routing protocols can be developed based on current link, network, and traffic conditions. Finally, the application layer can utilize a notion of soft QoS that adapts to the underlying network conditions to deliver the highest possible application quality [GW02].

In the framework of the IST-FET MOBILEMAN project we have defined a reference architecture for MANET able to exploit the advantages of a balanced cross-layer design (see [D10]). Figure 7.5 shows the MOBILEMAN cross-layer reference architecture. Briefly, in this architecture, cross layering is limited to parameters and implemented through data sharing. As shown in the figure, the **Network Status** module is a shared memory that stores all the network status information collected by the network protocols. All protocols can access this memory to write the information to share with the other protocols, and to read information produced/collected from the other protocols. This avoids duplicating the layers' efforts for collecting network-status information, thus leading to a more efficient system design. In addition, inter-layer co-operations can be easily implemented by variables sharing. However, protocols are still implemented inside each layer, as in the traditional layered reference architecture. This guarantees several advantages:

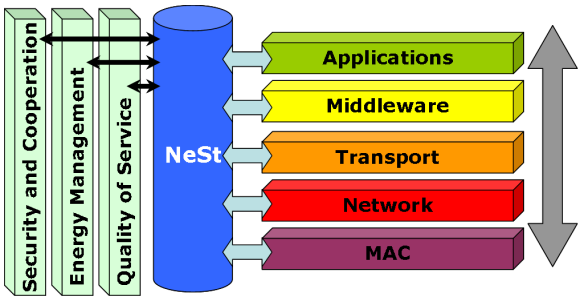


Figure 7.5: Cross-layer reference architecture.

- full compatibility with standards since it is not necessary to modify the core functions of each layer.
- robust to upgrading, and protocols belonging to different layers can be added/removed from the protocol stack without modifying the operations at the other layers.
- all the advantages of a modular architecture are maintained.

To summarize the MOBILEMAN reference architecture tries to achieve the advantages of a full cross layer design (i.e., joint optimization of protocols belonging to different layers) still satisfying the layer separation principle. Information regarding the network topology, energy level, local position, etc., is made available by the NeSt to all layers, in order to achieve optimizations, and offer performance gains from an overhead point of view. Although this awareness is restricted to the node's local view, protocols can be designed so as to adapt the system to highly variable network conditions (the typical ad hoc characteristic).

In particular, in the framework of the MOBILEMAN project, we focus on cross-layer interactions between routing and the middleware layers, as shown in Figure 7.6. We investigate how middleware level can benefit of information collected at network layer to build its overlay network. In this case the network layer could contribuite exporting its network topology helping the middleware layer to maintain a corrispondence between phisical and logical space address. In addition the middleware layer could exploit the presence of a proactive routing protocol to run a Service Discovery process spreading middleware information encapsulated into routing packets.

To this aim the prototype of HSLS routing protocol is able to be integrated with the cross-layer prototype allowing cross-layer interactions between routing and middleware layers.

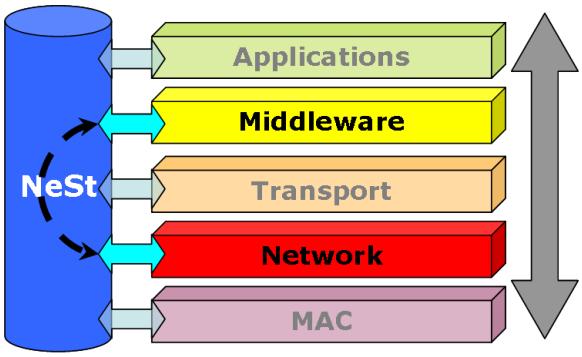


Figure 7.6: Cross-layer interactions between routing and middleware protocols.

7.4 HSLS Implementation

In the following subsections an overwiew of the prototype implementing the enhanced HSLS routing protocols is presented. The complete HSLS architecture, the used information repositories and the interactions between packages are shown, also specifying implementation decisions and tecnical details. The HSLS module is implemented for the Linux platform due to its open-source nature that allows accessing to OS kernel freely. Furthermore, since the Linux kernel, together with most other parts of the OS, is written in C, we have decided to use the C language; in this way, direct communications such as recovering of network information or frequently interactions with the kernel routing table (e.g. routes' addition and removal) become easier and faster.

7.4.1 Neighbor Discovery

Obviously, HSLS needs some mechanism to discover its 1-hop neighbors and detect the status of link communication with them. To this aim **Hello** packets are sent in the network periodically. Figure 7.7 shows the simple procedure used by a node to discover its 1-hop neighbors. Node A sends an empty `Hello` message (event 1 in figure). Node B receives this message and stores node A as its *asymmetric* 1-hop neighbor. B generates an `Hello` message declaring node A as its asymmetric neighbor (event 2). A receives B's `Hello` and stores B as its *symmetric* neighbor since it has found its address in B's `Hello`. In the next `Hello` message generated by node A, it declares node B as *symmetric* (event 3). This time node B updates the status of node A in *symmetric*. The next `Hello` message emitted by node B cointains node A declared as *symmetric* (event 4).

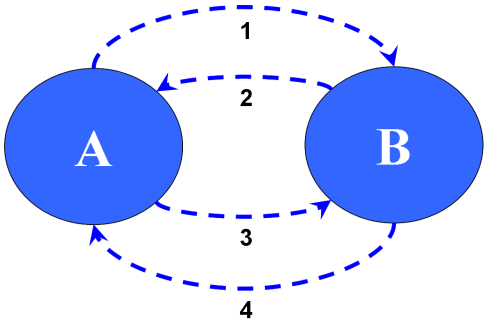


Figure 7.7: Neighbor discovery procedure using `Hello` packets.

In `Hello` messages each node transmits information about its 1-hop neighbors related to the wireless interface on which the message is sent, declaring also the type of the link. 1-hop neighbors are stored in `Hello` messages groupping by the link status. The Section 7.4.7 explains the format of `Hello` packet.

7.4.2 Topology Dissemination

Link-state routing protocols are based on a flooding process of topology information. Each node trasmits its local topology information encapsulated into **LSU** packets, thus the other nodes in the network are able to build their own view of the topology. In particular, in order to reduce the network load, the HSLS protocol implements an efficient dissemination procedure as deeply discussed in Section 7.2. The Figure 7.8 shows the procedure for `LSU` generation.

In `LSU` messages each node transmits only information about its 1-hop neighbors that are declared as symmetric, omitting those stored as asymmetric since they are not required in the routing calculation process. Thus, the size of `LSU` packets is reduced. The Section 7.4.7 explains the format of `LSU` packet.

7.4.3 Processing & Route Calculation phase

In this phase all incoming packets are processed and the corrispondent information repositories are updated. In addition, the routing table is computed. Basically, once a packet comes from the network, the following actions can be executed:

- Discard the packet if it is found invalid (e.g. if the packet type is not valid or if it is a duplicate packet)

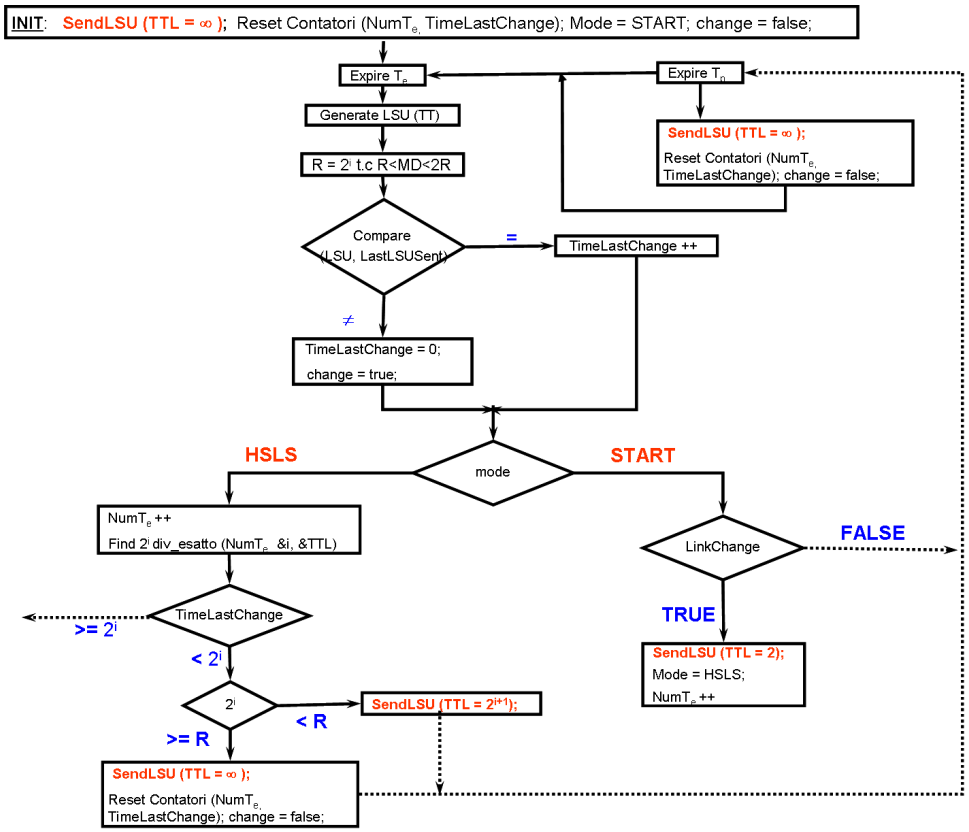


Figure 7.8: Diagram for LSU generation.

- Process the packet according to specific instructions related to the packet type
- Update the related data structures
- Compute the routes towards all nodes in the network using the Dijkstra's algorithm
- Forward the packet if it is an `LSU` and its TTL field is ≥ 1

7.4.4 Garbage Collector

The heart of a table driven routing protocol is the repositories in which the current state of the network is stored. All these tables have an associated timeout to maintain fresh all stored information. This means that when the timeout is expired the relative

entry has to be deleted. Thus, all table's entries are periodically checked and deleted if the associated timeout is expired. As consequence, a new recalculation of all routes is needed and the Dijkstra's algorithm is used to this aim.

7.4.5 HSLS Software Architecture

The Figure 7.9 shows the software architecture for HSLS. It consists of six main packages with different functionalities and able to manage related data structures. Specifically, we have:

♢ **Initialization**: it initializes data structures, manages wireless interface and sets socket options

♢ **Socket Manager**: it manages the socket used to send and receive packets from the network

♢ **Packet Manager**: it is the core of the HSLS deamon. It defines, generates and process HSLS messages and updates data structures

♢ **Information Repositories**: it contains all data structures used to store local node information and usd by the HSLS protocol to collect routing information

♢ **Garbage Collector**: it always maintains fresh and valid routing information deleting old entries in the information repositories

♢ **NeSt Communicator**: it is used in the cross-layer architecture in order to interface the routing protocol with the NeSt functionalities

Specifically, the *Packet Manager* package implements the most number of HSLS functionalities. In particular it can be divided into the following four sub-packages:

⋆ **Hello**: it defines and generates the Hello messages used for the neighbor discovery procedure

⋆ **LSU**: it defines and generates the LSU messages according to LSU generation procedure as explained in Section 7.2

⋆ **Processing**: it processes all the messages defined by HSLS, updates the data structures and computes the routing table

⋆ **Reliability**: it implements the Reliability process as previously explained (see Section 7.3.1)

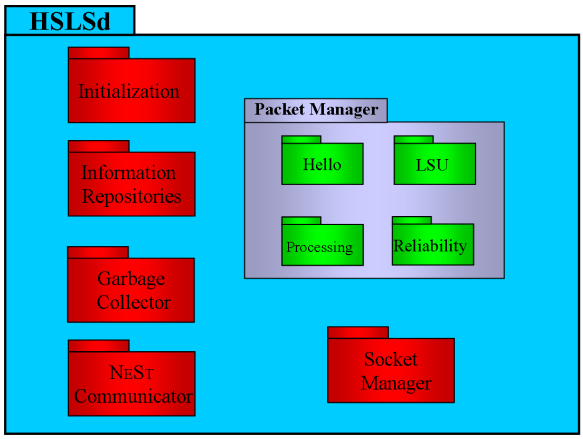


Figure 7.9: Package scheme of the HSLS implementation.

7.4.6 HSLS Data Structures

The HSLS deamon maintains running state into several information repositories. These data structures are inizialized during the start-up of the HSLS protocol and updated dynamically during the processing phase; the stored information is used to generate messages. Here follows a brief look at the different information repositories used in HSLS.

- **Interface**: it contains local information of the node (e.g. socket descriptor, name interface, wireless interface, network information,).
- **Topology Table (TT)**: it stores information of all known nodes of the network. For each connected couple of nodes, the associated link status is also maintened (i.e. ASYM/SYM).
- **Minimum Tree (MT)**: it used to apply the Dijkstra's algorithm in order to find the shortest path towards each node in the network.
- **Routing Table (RT)**: it contains the result of the Dijkstra's algorithm. Each entry registers the next hop (gateway), the corrispondent mask and the associated cost to reach each node of the network. Moreover, the RT cointains general information of the network, such as number of nodes, number of stored routes and the maximum distance in term of hops to the farthest node.
- **SentLSUcache (LSC)**: it maintains information about LSUs generated by the node itself. In particular, for each emitted message, a copy of the message,

together with the number of ACK_LSU received in the fixed time window, are stored in the cache.

⋆ **ReceivedLSUcache (RLC)**: it stores information about LSUs coming from other nodes. In particular, for each received message, a copy of the message, together with the number of received ACK_LSU received in the fixed time window, are stored in the cache.

⋆ **Monitor Topology**: it is responsible for detecting changes in the Topology Table. Whenever changes are being made to entries in TT (e.g. insertion/removal due to topology changes or timeout expiration of some entries), the *change topology* field in this data structure is setted causing the recalculation of all routes performing the Dijkstra's algorithm.

⋆ **Service**: it is used in the cross-layer architecture to exchange information between routing and middleware layer. The optional information to be encapsulated or extracted by `LSU` packets are stored here.

As previously said, SentLSUcache (LSC) and ReceivedLSUcache (RLC) are used as repositories of LSU information. In particular, they are used for a twofold reason: to implement the reliability process (as explained in Section 7.3.1), but also to detect duplicate packets in order to avoid their processing.

The correct behavior of the HSLS behavior is strictly correlated to these structures; stored information must be always fresh and valid to assure good decisions in routes' calculation and packets' delivery. Hence some data structures' entries have an associated timeout, i.e. the Topology Table and the two caches. More precisely:

i) in TT this value indicates how long the stored information can be considered valid and it is set according to a validity time contained in the packets, as explained in the following section. The timeout is set to the sum of the current time and the validity time. As a result, when the current time is higher than the stored time, the tuple is invalided and its content is not used

ii) in the other two caches (i.e. LSC and RSC) the timeout is set to the time window T used for the reliability process; after its expiration there will be a retransmission of the same LSU

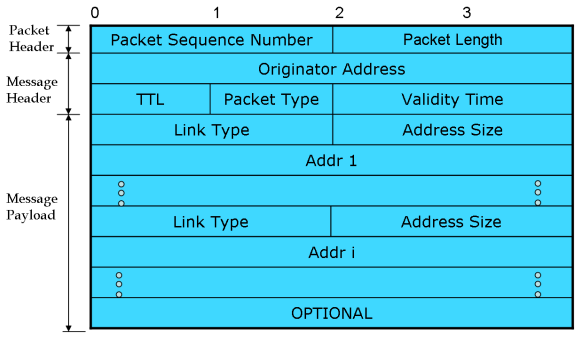


Figure 7.10: HSLS packet format.

7.4.7 HSLS Packets

HSLS utilizes several control packets. It sends routing information over the network using `Hello` packets during the 1-hop neighborhood's discovery phase, and `LSU` packets during the topology dissemination phase. Furthermore, in our HSLS implementation, special packets for the reliability process (`ACK_exp`) and packets for the cross-layer interaction (`LSU_opt`) must be generated, as well.

As in the other link-state routing protocols, all packets are trasmitted with a broadcast transmission. They are further encapsulated in UDP datagrams and then sent through the network using UDP connections.

Our HSLS module uses a unified packet format to flood information in the network. As shown in Figure 7.10, the HSLS packet is made of a Packet Header, a Message Header and a Message Payload of a variable length. More precisely:

♢ **Packet Sequence Number (PSN)**: (2 byte) it is incremented each time a new HSLS packet is generated and transmitted in the network.

♢ **Packet Length (PL)**: (2 byte) the field stores the total length (in byte) of the packet.

♢ **Originator Address (OA)**: (4 byte) since each node in the network is uniquely identified with an IP address, this field represents the IP address of the node that has generated the packet. This value does not change during the flooding process.

♢ **Time to Live (TTL)**: (1 byte) it contains the maximum number of propagation hops for a packet; each time a node receives a packet it decrements the TTL field

before broadcasting it to the network; if its value is equal to 0 the forwarding process is stopped.

♢ **Packet Type (PT)**: (1 byte) it indicates which type of packet is encapsulated. Possible values are `Hello`, `LSU`, `LSU_opt` and `ACK_expl`.

♢ **Validity Time (VT)**: (2 byte) this value indicates how long a node can consider valid the packet information after its reception.

♢ **Link Type (LT)**: (2 byte) it indicates the type of link between the originator node and the advertised neighbors listed after this field. Possible values are symmetric link (`SYM`) and asymmetric link (`ASYM`).

♢ **Address Size (AS)**: (2 byte) this field stores the length (in byte) of the list of advertised neighbors that follow a LT field.

♢ **Neighbor Address (NA)**: (4 byte) it represents the main IP address of the advertised neighbor node.

♢ **Optional**: this field contains information coming from other levels that are not strictly correlated with the routing protocol (i.e. services for middleware).

As explained in the previous sections, the `Hello` packet contains the list of neighbors considering both symmetric and asymmetric links, while only 1-hop neighbors connected through symmetric links are stored in `LSU` packets. The `LSU_opt` packet is an ordinary `LSU` which also encapsulates in the Optional field extradata coming from the NeSt. The `ACK_exp` packet, used in the reliability process, is made only of Packet and Message Header without any Message Payload.

7.4.8 HSLS Modules Interactions

To optimize the system performance, the software architecture of HSLS protocol is represented by a *multi-thread* system consiting of several threads running concurrently, using the POSIX thread library `pthread`. After an initialization phase in which all repositories are initialized, six main threads are created: *Hello* thread, *LSU* thread, *Reliability* thread, *Processing* thread, *Route Calculation* thread, the *Garbage Collector* thread.

As previously explained, the *Hello* and *LSU* threads generate their own messages according to the relative procedures recovering information from the TT and then update the SLC cache. The *Reliability* thread checks the relative caches (SLC and

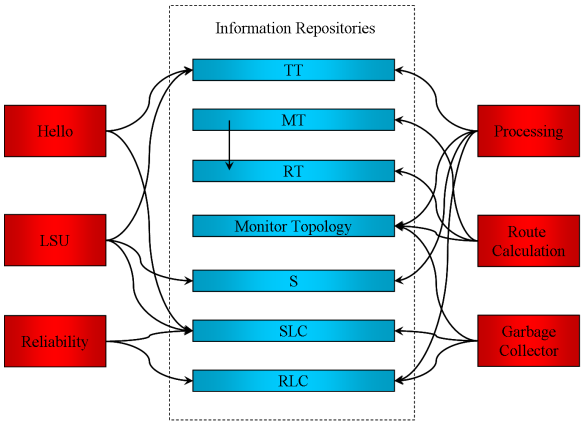


Figure 7.11: HSLS information repositories relation overview.

RLC) and generate its message according to the Reliability procedure. The *Processing* thread receives all incoming packets, checks for duplicate packets looking into the RLC cache, processes them, updates the TT and in case sets the Monitor Topology if it has modified the TT. The *Garbage Collector* thread periodically checks data repositories to delete expired entries, setting the Monitor Topology if the TT has been modified. The *Route Calculation* thread checks the Monitor Topology and when its value is changes, it wakes up computing the routing table according the Dijkstra's algorithm using information in TT; as consequence it updates the RT and the Kernel Routing Table using *ioctl* system call.

Since HSLS system runs in thread, it is possible that there can be simultaneously multiple accesses to the same data structures. The Figure 7.11 displays an overview of the information repositories and their relationship with the different threads. For example when a `Hello` packet is generated by reading information stored in TT, and at the same time an `LSU` packet is processed causing the update of the same repository. In order to guarantee data integrity, (part of the code of) threads must run in mutual exclusion locking and unlocking shared resources when they are needed; `pthread mutex` are used with this aim.

7.5 Conclusions

In the framework of this thesis, we worked to identify, if possible, a routing protocol suitable for multi-hop networks in terms of scalability, performance and efficiency, but

also able to provide a rich set of information about the network that can be exploited in a cross-layer architecture. The experimental evaluation presented in previous chapters shows that, in contrast with MANET community, the use of a proactive protocol as OLSR does not penalize the system performance. Moreover, recent studies [SMSR02] [SSR01] have analytically proved that the **Hazy Sighted Link State (HSLS)** routing protocol [SR01] exhibits good performance in term of scalability. HSLS is a *Link State protocol* with *Limited dissemination* in which routing updates are flooded in the network with a binary exponential sequence. In the framework of this thesis, a software module that implements the HSLS routing protocol has been designed and developed. In addition to the basic functionality, the HSLS protocol has been enhanced with: *i)* a mechanism to guarantee the reliabilty of LSU packets with any introduction of additional control overhead; *ii)* a module that allows cross-layer interactions, thus resulting in an easy integration with the cross-layer prototype. Its basic functionality has been successfully tested in network of 4-5 nodes. As next step a new experimental phase is planned in order to verify HSLS advantages promised by theoretical analysis and to compare it with the other MANET routing protocols.

8 Conclusions

In this thesis we have investigated the behaviour and the efficiency of routing protocols for MANET adopting an experimental approach. In fact, in current MANET reaserch most of them have been evaluated and compared through simulations, but they introduce simplifying assumptions (e.g., radio propagation model) that mask important characteristics of the real protocols behavior.

This work started from the study of a single-hop Ad Hoc network where we have evaluated the behaviour of IEEE 802.11 protocol, with particular attention to analyze the communication zone ($TX_{range}(x)$), (i.e., the maximum distance at which two nodes are able to correctly detect transmissions of each other) and the Physical Carrier Sensing zone (PCS_{range}), (i.e., the zone around a sending node within which another node senses the channel busy). Based on these measurements, we have defined an innovative wireless link model for 802.11 devices with a consequent redifinition of the traditional hidden and exposed node formulations.

Then we focused on multi-hop Ad Hoc network analysing performance of routing protocols. In particular, we selected two robust available implementations of routing protocols for MANET, specifically OLSR [CJ03] and AODV [PR03], and we compared them in different environments, i.e. indoor and outdoor, and different topology, i.e. static and mobile, starting from small-scale network (2-4 hops size with 8 nodes) up to medium-scale network (7-8 hops size with up to 23 nodes). We evaluated their performance from the efficiency and QoS standpoint. Our results highlight severe QoS problems when using AODV due to the reactive nature of the protocol, and indicate that, with a proactive protocol the network perform better. Furthermore, when considering higher level protocols on top of Ad Hoc test-bed, e.g. FreePastry or an optimized p2p platform named CrossROAD [Del05], benefits in using a proactive approach are more evident [D8][D16][BCDP05].

These results encourage to identify a routing protocol suitable for multi-hop networks in terms of scalability, performance and efficiency in the class of proactive protocol. As proved in [SMSR02] [SSR01], the **Hazy Sighted Link State (HSLS)** routing protocol [SR01] exhibits good performance in term of scalability. In particu-

lar, in HSLS routing information are propagated to the network nodes with a frequency that decreases with the distance using a binary exponential sequence. As a result, each node builds a "self-centered" topology view, which becomes *hazy* as the distance grows. In the framework of this thesis, an enhanced version of the HSLS routing protocol has been designed and developed, adding *i)* a mechanism to guarantee the reliabilty of LSU packets with any introduction of additional control overhead, and *ii)* a module that allows cross-layer interactions, thus resulting in an easy integration with the cross-layer prototype. The basic functionality of HSLS module has been successfully tested in network of 4-5 nodes [D10]. As next step a new experimental phase is planned in order to verify HSLS advantages promised by theoretical analysis and to compare it with the other MANET routing protocols.

Bibliography

[AAS00] A. Ahuja, S. Agarwal, and R. Shorey. "Performance of TCP over different routing protocols in mobile ad-hoc networks". In *Proc. of IEEE VTC 2000*, pages 298–303, Tokyo, Japan, May 2000.

[ABB+04] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. "Link-level Measurements from an 802.11b Mesh Network". In *Proc. of SIGCOMM 2004*, Portland, OR, September 2004.

[ABC+05] G. Anastasi, E. Borgia, M. Conti, E. Gregori, and A. Passarella. "Understanding the Real Behavior of Mote and 802.11 Ad Hoc Networks: an Experimental Approach". *Pervasive and Mobile Computing*, 1(2):237–256, July 2005.

[ABCG03] G. Anastasi, E. Borgia, M. Conti, and E. Gregori. "IEEE 802.11 Ad Hoc Networks: Performance Measurements". In *Proc. of MWN 2003, in conj. with ICDCS 2003*, Providence, Rhode Island, May 2003.

[ABCG04] G. Anastasi, E. Borgia, M. Conti, and E. Gregori. "Wi-Fi in Ad Hoc Mode: A Measurement Study". In *Proc. of PerCom 2004*, Orlando, Florida, March 2004.

[ABCG05] G. Anastasi, E. Borgia, M. Conti, and E. Gregori. "IEEE 802.11 Ad Hoc Networks: Performance Measurements". *Cluster Computing Journal*, 8:135–145, 2005.

[ACG03] G. Anastasi, M. Conti, and E. Gregori. *"IEEE 802.11 Ad Hoc Networks: Protocols, Performance and Open Issues"*. in *Mobile Ad hoc networking*, S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (Editors), IEEE Press and John Wiley and Sons, Inc., New York, 2003.

[BCDG05] E. Borgia, M. Conti, F. Delmastro, and E. Gregori. "Experimental comparison of Routing and Middleware solutions for Mobile Ad Hoc Networks: Legagy vsCross-Layer approach". In *Proc. of E-WIND Workshop, in conj. with SigCom 2005 Conference*, Philadelphia, PA, August 2005.

[BCDP05] E. Borgia, M. Conti, F. Delmastro, and L. Pelusi. "Lessons from an Ad-Hoc Network Test-Bed: Middleware and Routing Issues". *"Ad Hoc & Sensor Wireless Networks, An International Journal"*, 1(1-2), 2005.

[BCDP06] E. Borgia, M. Conti, F. Delmastro, and A. Passarella. "MANET perspective: current and forthcoming technologies". In *Proc. of 15th IST Mobile Summit*, Myconos, Greece, June 2006.

[BCGI04] S. Basagni, M. Conti, S. Giordano, and Stojmenovic I. in *Mobile Ad hoc networking*, IEEE Press and John Wiley and Sons, Inc., New York, 2004.

[BCSW98] S. Basagni, I. Chlamtac, V. Syrotiuk, and B. Woodward. "A Distance Routing Effect Algorithm for Mobility (DREAM)". In *Proc. of ACM MOBICOM'98*, Dallas, Texas, October 1998.

[Bel03] E.M. Belding. *"Routing Approaches in Mobile Ad Hoc Networks"*. In *Mobile Ad hoc Networking*, Basagni, Conti, Giordano and Stojmenovic, (Editors), IEEE Press and John Wiley and Sons, New York, 2003.

[Blu00] Bluetooth Special Interest Group, "Specification of Bluetooth System V.1.1", Dicember 2000.

[Bor05] E. Borgia. "Experimental Evaluation of Ad Hoc Routing Protocols". In *Proc. of PWN 2005, in conj. with Percom 2005 Conference*, Kauai Island, Hawaii, March 2005.

[BOT01] B. Bellur, R.G. Ogier, and F.L. Templin. "Topology Broadcast based on Reverse-Path Forwarding (TBRPF)", March 2001. RFC 3684.

[BT99] E.M. Belding and C.K. Toh. *"A Review of Current Routing Protocolos for Ad Hoc Mobile Wireless Networks"*. IEEE Personal Communication Magazine, April 1999.

[CCL03a] I. Chlamtac, M. Conti, and J. Liu. "Mobile Ad Hoc Networking: Imperative and Challenges". *Ad Hoc Networks Journal*, 1(1), Jan-Febr-Mar 2003.

[CCL03b] I. Chlamtac, M. Conti, and J. Liu. *"Mobile Ad Hoc Networking: Imperatives and Challenges"*. Ad Hoc Networks, No. 1(1), 2003.

[CG98] T. W. Chen and M. Gerla. "Global State Routing: A new Routing Schemes for Ad-Hoc Wireless Networks". In *Proceedings of IEEE ICC'98*, June 1998.

[CJ03] T. Clausen and P. Jacquet. "Optimized Link State Routing Protocol (OLSR)", October 2003. RFC 3626.

[CR00] C. F. Chiasserini and R. R. Rao. "Routing protocol to maximize battery efficiency". In *Proc. of IEEE MILCOM 2000*, Los Angeles, CA, October 2000.

[CRVP01] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash. "A Feedback Based Scheme for Improving TCP Performance in Ad Hoc Wireless Networks". *IEEE Personal Communication Magazine,Special Issue on Ad Hoc Networks*, 8(1):34–39, February 2001.

[D10] MobileMAN Deliverable - D10. http://cnd.iit.cnr.it/mobileMAN.

[D16] MobileMAN Deliverable - D16. http://cnd.iit.cnr.it/mobileMAN.

[D8] MobileMAN Deliverable - D8. http://cnd.iit.cnr.it/mobileMAN.

[DB01] T.D. Dyer and R.V. Boppana. "A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks". In *Proc. of ACM MobiHoc 2001*, Long Beach, CA, October 2001.

[DCY00] S. R. Das, R. Castaneda, and J. Yan. "Simulation Based Peformance Evaluation of Mobile, Ad Hoc Network Routing Protocols". *ACM/Baltzer Mobile Networks and Applications (MONET) Journal*, July 2000.

[Del05] F. Delmastro. "From Pastry to CrossROAD: Cross-layer Ring Overlay for AD hoc networks". In *Proc. of MP2P Workshop in PerCom 2005*, Kauai Island, Hawaii, March 2005.

[Dep] Dep. of Computer Systems, Uppsala University (Sweden). "APE: Ad Hoc Protocol Evaluation Testbed". http://apetestbed.sourceforge.net.

[DPR00] S. R. Das, C. E. Perkins, and E. M. Royer. "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks". In *Proc. of INFOCOM 2000*, Tel Aviv, Israel, March 2000.

[Eph02] T. Ephremides. "A Wireless Link Perspective in Mobile Networking". In *ACM MobiCom 2002, keynote speech*, Atlanta, GA, September 2002.

[FZX+02] Z. Fu, P. Zerfos, K. Xu, H. Luo, S. Lu, L. Zhang, and M. Gerla. "On TCP Performance in Multihop Wireless Networks". *Technical Report, UCLA, Computer Science Department*, 2002.

[GKN+04a] R. S. Gray, D. Kotz, C. Newport, N. Dubrovsky, A. Fiske, J. Liu, C. Masone, S. McGrath, and Y. Yuan. "Outdoor Experimental Comparison of Four Ad Hoc Routing Algorithms". In *Proc. of MSWiM 2004*, Venice, Italy, October 2004.

[GKN+04b] R. S. Gray, D. Kotz, C. Newport, J. Liu, Y. Yuan, and C. Elliott. "Experimental Evaluation of Wireless Simulation Assumptions". In *Proc. of MSWiM 2004*, Venice, Italy, October 2004.

[GLNT05] P. Gunningberg, H. Lundgren, E. Nordstrom, and C. Tschudin. "Lessons from Experimental MANET Research". *Ad Hoc Networks Journal, special issue on "Ad Hoc Networking for Pervasive Systems"*, 3(2), March 2005. M.Conti, E.Gregori (Editors).

[Glo] GloMoSim, Global Mobile Information Systems Simulation Library. http://pcl.cs.ucla.edu/projects/glomosim/.

[GO02] D.B. Green and M.S. Obaidat. "An Accurate Line of Sight Propagation Performance Model for Ad-Hoc 802.11 Wireless LAN (WLAN) Devices". In *Proc. of IEEE ICC 2002*, New York, April 2002.

[GW02] A.J. Goldsmith and S.B. Wicker. "Design Challenges for Energy-Constrained Ad Hoc Wireless Networks". *IEEE Wireless Communication*, 9(4):8–27, August 2002.

[Hsi01] P.H. Hsiao. "Geographical region summury service for geographical routing". *Mobile Computing and Communication Review*, 5(4), October 2001.

[HV99] G. Holland and N. Vaidya. "Analysis of the TCP Performance over Mobile Ad Hoc Networks". In *Proc. of the ACM MobiCom'99*, pages 207–218, Seattle, WA, August 1999.

[HV02] G. Holland and N. Vaidya. "Analysis of the TCP Performance over Mobile Ad Hoc Networks". *ACM/Kluwer Journal of Wireless Networks*, 8(2-3):275–288, 2002.

[IEE99] IEEE standard 802.11, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", August 1999.

[JM03] D.B. Johnson and D.A. Maltz. "Dynimic Source Routing Protocol (DSR)", October 2003. RFC 3626.

[KS71] L. Kleinrock and K. Stevens. "Fisheye: A Lenslike Computer Display Transformation". *Technical Report, UCLA, Computer Science Department*, 1971.

[LBDC+01] J. Li, C. Blake, D. De Couto, H. Lee, and R. Morris. "Capacity of Wireless Ad Hoc Wireless Networks". In *Proc. of ACM MobiCom 2001*, Rome, Italy, July 2001.

[LNT02] H. Lundgren, E. Nordstrm, and C. Tschudin. "The Gray Zone Problem in IEEE 802.11b based Ad hoc Networks". *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3), July 2002.

[LS01] J. Liu and S. Singh. "ATCP: TCP for mobile ad hoc networks". *IEEE Journal on Selected Areas in Communications*, 19(7):1300–1315, July 2001.

[LS02] J. Liu and S. Singh. "How Bad TCP Can Perform in Mobile Ad Hoc Networks". In *Proc. of the IEEE Symposium on Computers and Communications*, pages 298–303, Taormina-Giardini Naxos, Italy, July 2002.

[Lun] H. Lundgren. *Implementation and Experimental evaluation of Wireless Ad hoc Routing protocols.* PhD thesis, http://publications.uu.se/theses/abstract.xsql?dbid=4806.

[MAN] MANET Meeting Report at 55th IETF Meeting in Atlanta, Georgia, USA. http://www.ietf.org/proceedings/02nov/177.htm.

[MC04] J.P. Macker and S. Corson. *"Mobile Ad Hoc Networks (MANET): Routing technology for dynamic, wireless networking"*. in *Mobile Ad hoc networking*, Basagni, Conti, Giordano, Stojmenovic (Editors), IEEE Press and John Wiley and Sons, Inc., New York, 2004.

[Moy95] J. Moy. *"Link-State Routing"*. In M.Steenstrup, editor, *Routing in Communications Networks*, Prentice Hall, 1995.

[MS95] G.S. Malkin and M.E. Steenstrup. *"Distance-Vector Routing"*. In M.Steenstrup, editor, *Routing in Communications Networks*, Prentice Hall, 1995.

[Ns0] The Network Simulator - ns-2. http://www.isi.edu/nsnam/ns/index.html.

[NTCS99] S.Y. Ni, Y.C. Tseng, Y.S. Chen, and J.P. Sheu. "The broadcast storm problem in a mobile ad hoc network". In *Proc. of MOBICOM 1999*, pages 151–162, Seattle, Washington, USA, August 1999.

[PB94] C.E. Perkins and P. Bhagwat. *"Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers"*. SIGCOMM '94: Computer Communications Review, October 1994.

[Per00] C.E. Perkins. *"Ad Hoc Networking"*. Addison-Wesley, Reading, MA, 2000.

[PGH00] G. Pei, M. Gerla, and X. Hong. "Fisheye Routing Protocol in Mobile Ad Hoc Networks". In *Proc. of ICDCS 2000*, Taipei, Taiwan, April 2000.

[PH02] P Papadimitratos and Z. Haas. "Secure Routing for Mobile Ad Hoc Networks". In *Proc. of SCS CNDS 2002*, San Antonio, TX, January 2002.

[PR03] C. Perkins and E. Royer. "Ad-Hoc On-Demand Distance Vector Routing (AODV)", July 2003. RFC 3561.

[Qua] Qualnet Simulator. http://www.qualnet.com/.

[RS98] S. Ramanathan and M. Streenstrup. Hierchically-organized, Multihop Mobile networks for Multimedia Support. *ACM/Baltzer Mobile Networks and Applications*, 3(1):101–119, june 1998.

[SDL+02] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer. "A Secure Routing Protocol for Ad Hoc Networks". In *Proc. of IEEE ICNP 2002*, Paris, France, November 2002.

[SMSR02] C.A. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan. "On the Scalability of Ad Hoc Routing Protocols". In *Proc. of INFOCOM 2002*, New York, June 2002.

[SR01] C.A. Santivanez and R. Ramanathan. "Hazy Sighted Link State (HSLS) Routing: A Scalable Link State Algotithm". In *"BBN technical memo BBN-TM-130", BBN Technologies*, Cambridge, MA, August 2001. Available at http://www.ir.bbn.com/documents/techmemos/index.html.

[SSR01] C.A. Santivanez, I. Stavrakakis, and R. Ramanathan. "Making Link State Routing Scale for Ad Hoc Networks". In *Proc. of MobHoc 2001*, Long Beach, CA, October 2001.

[SW03] I Stojmenovic and J. Wu. *"Broadcasting and Activity-Scheduling in Ad Hoc Networks"*. In *Mobile Ad hoc Networking*, Basagni, Conti, Giordano and Stojmenovic, (Editors), IEEE Press and John Wiley and Sons, New York, 2003.

[TG99] K. Tang and M. Gerla. "Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks". In *Proc. of IEEE MMT'99*, Venice, Italy, October 1999.

[TJB01] M. Takai, Martin J., and R. Bagrodia. "Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks?". In *Proc. of ACM MobiHoc 2001*, pages 87–94, Long Beach, CA, October 2001.

[XBLG02] K. Xu, S. Bae, S. Lee, and M. Gerla. "TCP Behavior across Multihop Wireless Networks and the Wired Networks". In *Proc. of ACM WoWMoM 2002*, Atlanta, GA, September 2002.

[XHE01] Y. Xu, J. Heidemann, and D. Estrin. "Geography-informed Energy Conservation for Ad Hoc Routing". In *Proc. of ACM MobiCom 2001*, Rome, Italy, July 2001.

[XS01] S. Xu and T. Saadawi. "Does the IEEE 802.11 MAC protocol Work Well in Multihop Wireless Ad Hoc Networks?". *IEEE Communication Magazine*, 39(6):130–137, June 2001.

[XS02] S. Xu and T. Saadawi. "Revealing the Problems with 802.11 MAC Protocol in Multi-hop Wireless Networks". *Computer Networks*, 38(4), March 2002.