# MobiMESH: An Experimental Platform for Wireless MESH Networks with Mobility Support

Antonio Capone
Politecnico of Milan Italy
capone@elet.polimi.it

Stefano Napoli
Politecnico of Milan Italy
napoli@elet.polimi.it

Alberto Pollastro
Politecnico of Milan Italy
pollastro@elet.polimi.it

## ABSTRACT

Wireless Mesh Networks represent nowadays' new frontier of wireless networking technology. In fact, WMNs allow the extension of traditional wireless access networks through multihop relaying; this leads to the creation of easily deployable, flexible and reliable networks. WMNs have been largely studied through theoretical models and simulations, but the first experimental testbed studies have shown that simulation results are not always accurate because simulation cannot account for some aspects of a mesh network that are indeed crucial to performance evaluation. Deploying an experimental testbed represents an interesting challenge, because of many problems that have to be solved both on the architectural design side and on the practical implementation side. In this paper we present MobiMESH, a WMN architecture that has been implemented in a real life testbed. The architecture is designed with high mobility support and with integration capabilities. Mobility management is supported with a set of procedures that constitute an intermediate stratum between layer 2 and layer 3. Experimental results and performances are shown, and problems and future works are outlined.

## Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless Communication

## General Terms

Measurement, Experimentation, Performance

## 1. INTRODUCTION

Wireless Mesh Networks (WMNs) have emerged as a new network architecture able to extend the coverage and increase the capacity of wireless access networks [2]. WMNs bear many advantages on other wireless technologies, so they are being intensively studied and evaluated mainly by simulation. However, open technical issues still exist. In fact, recent analysis [9] [1] show that the evaluation of the employed protocols and algorithms conducted by simulation, although essential in understanding the effectiveness of the proposed solutions, may result only partially representative of real mesh scenarios. In fact, simulations cannot always accurately account for many physical layer issues such as ground effect, antenna proximity, and variations in interference and delay. Therefore, a major gap exists between solutions proposed in literature and their experimental assessments; hence, testbed deploying is needed, in order to be able to study mesh networking in a real environment, and to gather useful data and experience, following the pioneering work of [3] and [8]. Deploying an experimental testbed involves dealing with both design and practical issues [6]. Design challenges include connectivity granting for different kind of clients, mobility management and integration of the WMN with heterogeneous networks; the designed architectures must consider those requirements and meet them. Moreover, many practical and physical issues that are not always considered in simulation studies influence performance of a real life testbed. Electromagnetic leakage from the network adapters, unpredictable in-band radio phenomena and link instability are just few of the factors that widen the gap between simulated results and experimental performance. We have designed and implemented MobiMESH, a mesh network architecture with mobility support; the implementation has been carried out with IEEE 802.11 off-the-shelf technology, but it can be easily adapted to other technologies. In this paper we describe the architecture and we report the gathered performance and the practical problems encountered and solved. We have focused our work on obtaining a fully working wireless mesh network, offering connectivity, mobility support and Internet access to different type of clients. The analysis is conducted in order to find out critical issues in deploying a mesh networking testbed from both the design side and the practical implementation side. The rest of the paper is organized as follows: in the next section we present the MobiMESH architecture. In section 3 we outline the implementation of the architecture in a real life testbed. In section 4 we present the experimental results obtained on the testbed, while in Section 5 we describe some problems we have encountered while developing the testbed. The conclusions of the work are discussed in Section 6.

*QShine* '06 Waterloo, Ontario, Canada

## 2. DESIGN ISSUES

The MobiMESH network has been designed to offer the following features in terms of services and performance:

- **connectivity**: the network should grant connectivity to both ad hoc clients and to standard WLAN clients;
- **transparency:** the network should offer the same interface as a standard WLAN network to standard clients;
- **mobility:** clients should be able to roam freely in the coverage area of the network; handover, client tracking and the other mobility management procedures should be handled by the network alone;
- **integration:** it should be possible to integrate MobiMESH with different kind of networks, especially with wired networks (and therefore with the Internet).

In order to meet such requirements, MobiMESH is designed according to the hybrid mesh network paradigm. It is therefore composed by a mesh backbone core section, which is responsible for routing and mobility management, and by an access network, that hosts WLAN clients. The backbone network is the core of MobiMESH, since it provides routing functions, mobility management and integration with heterogeneous networks. This section of MobiMESH is based on the multi hop network paradigm, where all nodes are mesh routers and therefore collaborate to routing duties; routing is granted through a proactive ad hoc routing protocol, OLSR [4],[5]. The mesh backbone can be considered a wireless distribution system that connects all the access points, granting access and connectivity to WLAN clients. It significantly differs from the IEEE 802.11 Wireless Distribution System (WDS) since WDS only forwards frames, and has no routing functions, while the mesh backbone has routing capabilities. The core network hosts also some network elements that are crucial for the correct functioning of the network, such as the DHCP server, a mobility management database which will be discussed later and one or more gateways that connect MobiMESH to other networks. The other component of the mesh network is the access network, that is an infrastructure based wireless network in which clients connect to the access point which provides connectivity to the backbone. The access network is designed so that clients perceive the network as a standard WLAN and behave accordingly; this way MobiMESH can be accessed by standard WLAN client with no specific software installed. The device that connects the backbone and the access network is called Access Router (AR); it acts both as mesh router and as access point, and it is the fundamental block of our network. ARs are equipped with at least two radio interfaces, one of which belongs to the backbone network, and the other one which serves as access point for the access network.

## 2.1 IP Organization

Designing the IP organization of a mesh network is very different from doing it for a standard WLAN, since it involves the creation of different IP subnetworks in order to distinguish the access network from the backbone. MobiMESH IP layer organization is depicted in Figure 1. The mesh network is divided into two IP subnetworks; the core backbone is the first IP subnet, while the access network is a second subnet, including all the BSS areas of the access routers. This way WLAN clients do not change IP subnet while roaming, while they still perform layer 2 mobility procedures when passing from an AR to another. Therefore, from the client point of view, the network is perceived exactly like as a standard WLAN. Obviously, an intermediate layer is needed in order to keep IP routing on the backbone consistent with the state of the layer 2 network. This intermediate layer consists in a set of mobility management procedures, totally transparent to the clients, which are implemented in the Access Routers.

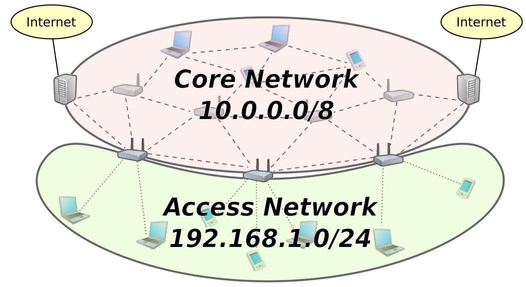

**Figure 1: IP organization of the network**

## 2.2 Mobility management

Client mobility management mechanisms are natively supported by the architecture. Advanced and standard WLAN clients can roam freely in the coverage area of the network without losing connection; tracking of such terminals is always possible. Mobility is handled in a different way whether the client is a mesh client or a standard client. In fact, mesh clients are advanced terminals that run the ad hoc routing protocol, which is responsible for managing mobility. When a mesh client moves, routing mechanisms automatically update its position and the terminal does not loose connectivity. Mobility management for standard IEEE 802.11 clients is more complex. In fact, the client must perceive the mesh network as a standard WLAN, where mobility is managed through Inter Access Point Protocol (IAPP) or similar proprietary protocols. WLAN's mobility, yet, is level 2 mobility, while mesh mobility involves layer 3 and therefore it influences and it is influenced by routing mechanisms. When a client moves from a BSS area to another, layer 2 mobility is managed by the MAC layer, which provides the reassociation to the new access point. Moreover, since WLANs are layer 2 networks, no change occurs in the IP addressing and routing mechanisms. In mesh scenarios, moving into a new BSS area requires changes in the routing of the backbone, because the client's position must be known to the multi hop routing protocol in order to correctly deliver messages. In the MobiMESH architecture, since the access network is a single IP subnet, no IP layer mobility is adopted, because access clients, when associating to a different access point, do not change IP subnet. In order to track mobile clients, MobiMESH features an intermediate layer that informs layer 3 of layer 2 changes. When a client moves from a BSS to another, the access point uses the intermediate layer to inform the routing mechanisms that the client has moved and it must be reached through a different AR. The key element of this procedures is the MAC-IP association table, a table that joins the information of the two layers involved and that can be used in order to associate layer 2 mobility to layer 3 routing changes. Such MAC-IP table is created with static or dynamic information, and resides on a database that can be distributed or centralized, depending on the size of the network and on other design factors.

### 2.3 Integration with heterogeneous networks

Integration with heterogeneous networks is a crucial feature for a WMN, especially because the mesh network can be employed as an access network to provide broadband connectivity, and therefore it must be attached to wired backbones; moreover, it can be integrated with networks based on different technologies (802.16, sensor networks...). The connection with other networks requires the creation of mesh devices equipped with an interface on the other network. MobiMESH is provided with special ARs, called gateways, that offer such functionalities. When multiple gateways are present on the network, the ad hoc routing protocol guarantees that every host employs the closest gateway; however, if gateways are connected to the same LAN, it is required for the ad hoc routing protocol to communicate with the wired routing protocol in order to share routing information. This is accomplished by MobiMESH gateways, that translate ad hoc routing information into wired routing information, thus informing wired routers of the presence of the MobiMESH network and allowing direct Internet connectivity.

## 3. IMPLEMENTATION OF MOBIMESH

We have chosen to implement the MobiMESH architecture with IEEE 802.11 technology, even though the architecture can be easily implemented, in general, with different technologies. Since the backbone network is based on the ad hoc paradigm, it exploits the Independent BSS (IBSS) IEEE 802.11 operating mode. The employed ad hoc routing protocol is the Optimized Link State Protocol (OLSR), in particular the UniK implementation[1]. We have chosen a proactive protocol because every node has the complete knowledge of the topology of the network. This bears two major advantages:

- mobility management procedures are easier to design;
- the network has faster reaction times, since no Route Request procedure must be started before sending a packet.

**Figure 2: MobiMESH Access Router implementation**

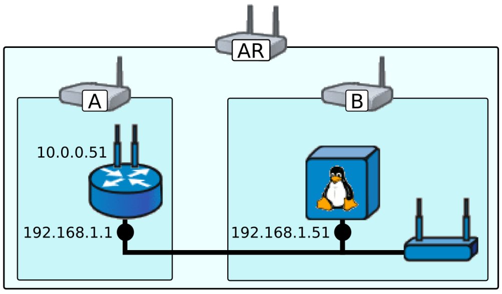

**Figure 3: Access Router logical configuration**

The OLSR routing protocol's Hosts and Network Association (HNA) feature is exploited by the MobiMESH architecture, since it allows an ad hoc node to communicate to the other nodes that it can reach networks not directly participating to the MANET; therefore, ARs can advertise on the backbone which hosts belonging to the access network are associated to it. This can be used in the mobility management procedures, because HNA messages can be used to advertise to the backbone that a client is reachable through the AR it is associated to. It is also useful in the creation of Gateways, since it can announce on the backbone the networks it can reach. The access network is a standard infrastructure wireless network, therefore it employs the IEEE 802.11 Infrastructure operating mode, so that ARs act as access points and standard WLAN clients can associate to them. The access network works with the same technology as the backbone network, and this could be lead to high interference; yet the two networks are set on different (orthogonal) channels, i.e. 1 and 11, that are the most separate channels in the IEEE 802.11b/g spectrum, therefore minimizing channel interference. The whole testbed is based on off-the-shelf devices, in order to have more customizable and open environments. Access clients and mesh clients have been deployed with 5 laptops equipped with Cardbus Linksys DWL-G650 cards, that feature Atheros chipsets driven by customized MADWifi drivers. Moreover, we have employed 2 desktop computers equipped with PCI Linksys DWL-G520 cards, featuring the same Atheros chipsets and driven by the same drivers. Mobile and fixed computers were running Debian GNU/Linux distribution, testing (*Etch*) release, with kernel 2.6.16. MobiMESH network has been connected to the Internet through a Cisco 2621 (2600 family) router running IOS 12.0. Access Routers have been more of a challenge; in fact, since few commercial devices are natively equipped with two radio interfaces, we have created the AR with two wireless routers connected with an Ethernet link, as shown in Figure 2. The employed routers are Linksys Broadband Wireless Router WRT54G, with the Linux based open firmware OpenWRT; the Broadcom chipset is driven by a proprietary driver (*wl*), that allow a reasonable degree of control over the wireless properties. The architecture of such routers is based on the MIPSEL architecture, a Broadcom 4712 processor overclocked to 216MHz. Every router features a single radio interface with two antennas for spacial diversity purposes, and five wired interfaces. We have employed 10 wireless routers, realizing 5 ARs. Figure 3 shows the internal organization of the Access Router. The left wireless router, named A, acts as mesh router on the mesh backbone; it has two interfaces: the radio interface on the core backbone, and the wired interface

---

[1] Available: http://www.olsr.org

on the access network. On the contrary, the second wireless router, named B, acts as an access point; its only IP interface is for management purposes and for MobiMESH procedures. The shown configuration is the same for every AR; the IP address on the backbone is obviously different for every AR, while the wired address remains always the same, since it is advertised as the default gateway for access clients. Moreover, the AR must perform DHCP Relay and Proxy ARP functions, in order to grant transparency of mobility management procedures to the access client; in fact it must masquerade to the client that a single IP network is mapped on different WLANs, so it must handle ARP queries and forward frames for the correct hosts. MobiMESH procedures have been implemented with user space daemons that gather information about layer 2 and layer 3 through software hooks in the GNU/Linux network stack and process them. In fact our tools obtain information from the driver, perform the MAC-IP mapping and then deliver the information to the routing daemon through an OLSR plugin that accesses to the internal OLSR information database. Such procedures are implemented in the AR; yet, since our ARs are composed of two separate wireless routers that manage respectively layer 2 and layer 3 of the network stack, our procedures communicate over the wired link between the two routers. In particular, the core functions are hosted on the B wireless router (on the IP entity of the router, drawn in Figure 3), while router A hosts the OLSR plugin and spreads the information on the backbone network.

# 4. TESTBED PERFORMANCE

The MobiMESH testbed has been extensively tested in order to gather performance measurements of throughput and of join and mobility delays. Tests have been conducted in indoor environment to gather real life usage data. Many topologies have been tested, we report the most significative: a full mesh topology, where the five ARs and the two desktop computers were in direct radio visibility, and a string topology, where we have have created a ”string” of 4 ARs in which every AR was only connected to the previous and the subsequent AR.

The creation of topologies has been accomplished only by means of radio inspection of the environment and correct placement of the nodes; this has led us to put the nodes far apart from each other and to lower the transmission power, with negative impact on the performance. Our tests, therefore, represent a lower bound to mesh networking performance in real life environment. The full mesh topology has been used to establish a baseline for measuring throughput, while the string topology shows the variation of throughput when multi hop routing is employed in the backbone.

## 4.1 Throughput results

Figure 4 shows a section of the full mesh topology employed in the first test; we have measured the throughput of two connections with different characteristics; results are reported in Figure 5. Case 1 shows a connection between an access client and a server residing on the backbone network; such connection involves two hops, the first on the access network and the second on the backbone: since channels for the two networks are separate, interference is minimized. The test has been conducted sending increasing traffic and measuring the received traffic; we have verified that the maximum received traffic is around 17Mbit/s, which is the same as a standard IEEE 802.11g WLAN (54Mbit/s is the data link traffic, while user traffic is around 17Mbit/s). Case 2 represents a data transfer between two access clients; three links are involved, two of which are on the same subnetwork. The received traffic is much lower than Case 1 because the first and the last hop share the same channel, causing severe interference that halves throughput. The string topology, instead, has been employed to study the influence of multi hop routing on throughput. Figure 6 shows the performed data transfers, that involve from one to three hops on the backbone network; transfers have been made with both TCP and UDP streams. Results are shown in Figure 7 show that every hop reduces throughput of about 25%, and the same behavior is detected for TCP and UDP traffic; UDP obviously experiences a higher throughput since TCP retransmits lost packets, measuring a lower goodput. Absolute performance is lower than the previous case since in order to obtain the string configuration we have significantly reduced the transmission power, so that links were at low rates and throughput was severely reduced.

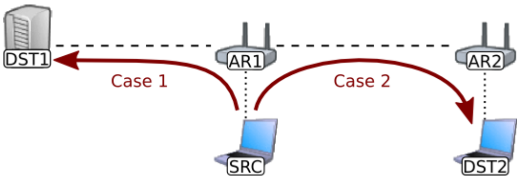

**Figure 4: Scenario for throughput measures: two wireless hops VS three wireless hops**

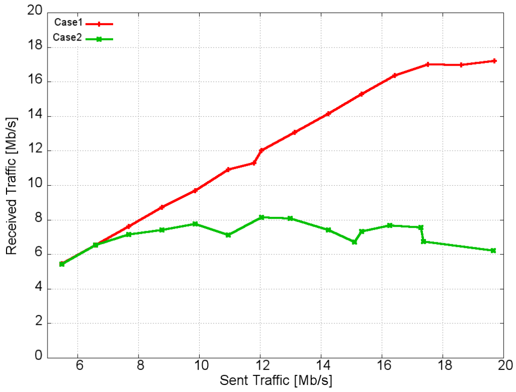

**Figure 5: Throughput performance. *Case 1*: two hops on different channels. *Case 2*: three hops, two on the same channel**

## 4.2 Delay Analysis

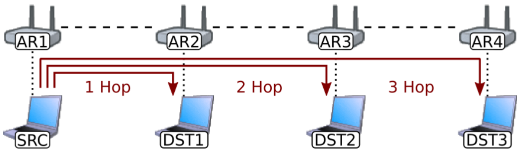

**Figure 6: String topology with different traffic paths**

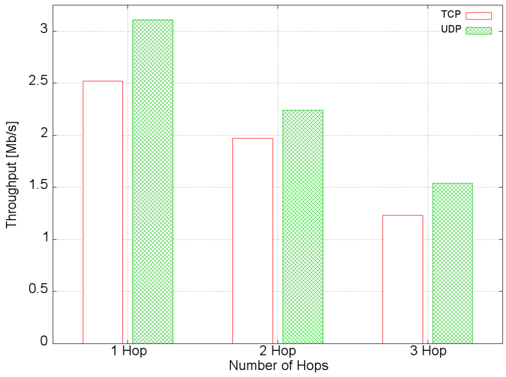

**Figure 7: UDP and TCP performance with multi hop backbone**

Delay analysis has been conducted both monitoring events on the network and measuring interruptions in traffic flows. We have measured the time it takes to a WLAN client to join the network and the time the network needs to hand a connection over from an access point to another. When a client comes in proximity of the access point, it associates to the access point at MAC level; this association triggers MobiMESH procedures that try to map the associated MAC onto an IP address. Every client has to obtain its IP address from the DHCP server, relayed by the DHCP Relay, so that when the mapping is available, the AR can advertise on the backbone network the existence of the new associated client. *Join time* is therefore composed by different factors: MAC layer switch, MobiMESH procedures' time and routing information propagation. On the other hand, we have measured the time it takes to the network to react to the movement of an already associated client, and to reroute connections. We have measured such time by sending UDP traffic with the D-ITG traffic generator from a server on the backbone to the moving client, and by measuring the connection's idle time. This is an upper bound to the connection delay, since routing has to be updated in order to reach the client; upstream traffic (from client to backbone) experiences smaller delays since routing remains the same. As shown in Table 1, the total mobility management time is 2777ms. We can split this time in different contributes, in order to better understand the network mechanisms. The first contribution is due to the layer 2 mobility time, that is the time it takes to the network adapter from when it takes the handover decision to when it is associated to the new AR; this depends on the implementation of the device driver and on the hardware itself. The network takes some time to detect such movement; in fact, since the wireless router's driver is binary and closed source, we couldn't manipulate it in order to signal a new association, so our tools continuously check the associated MAC table trying to detect changes; the table lookup is done every 100ms, introducing an average 50ms delay in association detection. MobiMESH procedures, including MAC-IP mapping procedures and local routing updates, introduce further delay. After 277 ms, in fact, the local OLSR daemon is informed that a new client has associated to the AR, and it is ready to spread such information on the mesh backbone by sending HNA messages. Routing update time is the most consistent contribute to the overall mobility management time, introducing almost 2 seconds delay. This is due to the fact that since HNA is not a core OLSR feature, HNA messages are sent and processed with lower priority and frequency than normal route updates, and HNA message management is less reactive and precise. Figure 8 shows the received bitrate during three movements of the client; the mobility management time is measured from the application point of view. Such delay is not a problem for data traffic, while it is more of a concern for real time traffic. However results show that improving OLSR implementation could lead to high delay reductions.

| | Time (ms) | Time (%) |
|---|---|---|
| Layer 2 Switch | 650 | 23% |
| New Association Detection | 50 | 2% |
| MobiMESH protocols | 277 | 10% |
| Routing update | 1800 | 65% |
| Total | 2777 | 100% |

**Table 1: MobiMESH Mobility Management Time split into its components**

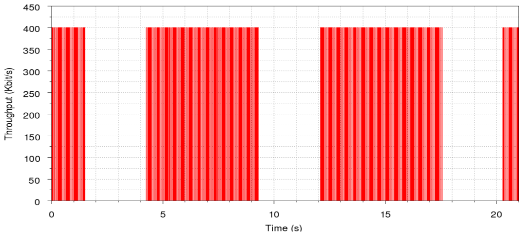

**Figure 8: Received downlink bitrate during three client movements**

## 5. PROBLEMS AND ISSUES

During testbed deployment we have encountered problems that have influenced performance measurements and architecture design. Interference has proven to be a major issue, both because of hardware problems and because of IEEE 802.11 design problems. In fact commercial network adapters usually experience radiation leakages and interferences that highly influence performance. Moreover, the distance between the two interfaces on a AR severely affects performance [9]: even though the two interfaces work on different and theoretically orthogonal channels (i.e. channel 1 and 11), they experience mutual radio interference. This can be reduced by taking the two interfaces apart from each other. This is possible in our testbed since MobiMESH's ARs are composed by two separate wireless routers, that can be placed at a arbitrary distance by simply extending the Ethernet cable between them. However, in order to realize the topologies employed in our tests, we had to keep an AR small and compact; therefore, tests have been performed with the two ARs placed one over the other, as shown in Figure 2, so that the wireless interfaces were few centimeters far from each other. Since our future work involves the creation of an integrated AR based on a single board, and the interfaces can't be put too far away from each other, we propose the use of different technologies (i.e. IEEE 802.11a or IEEE802.16 and IEEE802.11g) for the two

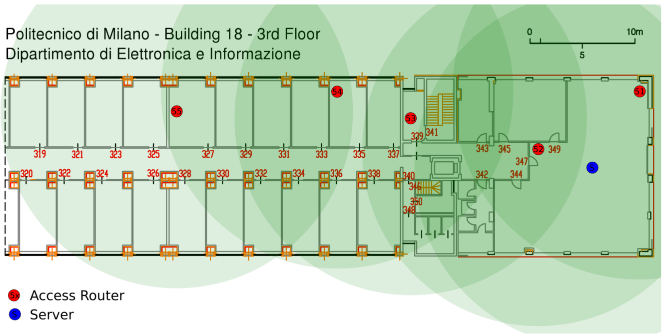

**Figure 9: Access Routers placement in the labs**

network sections. Setting up the topology has been another big issue: in order to test MobiMESH in a real life scenario, we wanted to create the topologies without performing MAC filtering or static routing. To do so, we lowered to 1mW the transmission power of the ARs, and we did perform careful radio inspection of the location of the conducted tests. In order to create the string topology, for example, we have put the ARs far away from each other, as shown in Figure 9. However, low transmission power and great distances make the rate adaptation mechanism unstable: this leads to low throughput and to rapid topology changes (especially in partially meshed backbone topologies), that have great impact on performance. The rate adaptation mechanism can be enhanced and rendered more stable, with great benefit for throughput and overall performance of the network. Furthermore, IEEE 802.11 is prone to structural problems such as the creation of gray zones [7] and IBSS partitioning, that can be partially overcome with careful network planning and cautious testbed deployment. Finally, the aforementioned problems with the OLSR implementation indicate that a better HNA managing mechanism can lead to great performance improvements. Alternatively, a separate protocol can be designed in order to spread mobility information on the backbone network and to directly interact with the OLSR core functionalities. The testbed experiments show that IEEE 802.11 MAC has some design limits that cause interference to heavily impact on the performance. The architecture, yet, is not strictly dependent on the employed MAC layer and on the OLSR ad hoc routing protocol implementation, so that further development on these topics will significantly increase MobiMESH's performance.

# 6. CONCLUSIONS

This paper describes MobiMESH, our implementation of a wireless mesh network architecture with mobility support. We have designed the network aiming at offering access and mobility management to both advanced and standard clients. The network can be integrated with heterogeneous networks in order, for example, to grant Internet access to clients. MobiMESH network features an IP organization designed to hide mobility management from standard access clients; in fact there is no layer 3 handover when a client moves from an AR to another. Since mobility in mesh networks involves routing updating, an intermediate layer acts between layer 2 and layer 3 in order to manage handovers and spread mobility information to the ad hoc backbone routing. We presented an implementation of MobiMESH network based on IEEE 802.11 technology. Numerical results show that throughput can reach the IEEE 802.11 maximum value only if radio interference is avoided. Connectivity between access clients can be provided, although with smaller rates. We also analyzed the effect on throughput of multihop routing of the backbone network, outlining that every additional hop highly impacts on throughput performance. A delay study has been also carried out, measuring join time and handover delays. We demonstrated that seamless mobility is granted to access clients, and the experienced delays don't affect too much data traffic. An accurate analysis of the mobility delay shows that the OLSR implementation is responsible for most of the delay, so performance can be significantly enhanced with modification to such implementation.

# 7. REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level measurements from an 802.11 mesh network. In *Proceedings of the ACM SIGCOMM'04*, August 2004.

[2] I. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks*, 47(4):445–487, March 2005.

[3] B. A. Chambers. The grid roofnet: a rooftop ad hoc wireless network. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2002.

[4] T. Clausen and P. Jacquet. Optimized link state routing protocol (OLSR). RFC 3626, October 2003.

[5] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of International Multi Topic Conference (INMIC'01)*, December 2001.

[6] S. Liese, D. Wu, and P. Mohapatra. Experimental characterization of an 802.11b wireless mesh network. Technical report, University of California, Davis, CA, 2005.

[7] H. Lundgren, E. Nordstorm, and C. Tschudin. Coping with communication gray zones in ieee 802.11b based ad hoc networks. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'02)*, September 2002.

[8] D. A. Maltz, J. Broch, and D. B. Johnson. Quantitative lessons from a full-scale multi-hopwireless ad hoc network testbed. In *Proceedings of the IEEE Wireless Communications and Networking Conference*, September 2000.

[9] J. Robinson, K. Papagiannaki, C. Diot, X. Guo, and L. Krishnamurthy. Experimenting with a multi-radio mesh networking testbed. In *Proceedings of the 1st workshop on Wireless Network Measurements (WiNMee'05)*, April 2005.