MOBILEMAN

IST-2001-38113

Mobile Metropolitan Ad hoc Networks

MOBILEMAN

MobileMAN first phase

Deliverable D8

Contractual Report Preparation Date: July 2004
Actual Date of Delivery: 6 August 2004
Estimated Person Months: 11.5
Number of pages: 58

Contributing Partners: Consiglio Nazionale delle Ricerche (Italy), University of Cambridge (UK), Institut Eurecom (France), Helsinki University of Technology (Finland), NETikos (Italy), Scuola Universitaria Professionale della Svizzera Italiana (Switzerland)

Authors: Patrizia Andronico, Eleonora Borgia, Marco Conti, Franca Delmastro, Luciana Pelusi (CNR), Claudia Brazzola, Jennifer Duyne Barenstein, (SUPSI)

Abstract: The aim of this deliverable is to document a preliminary evaluation of the MobileMAN solutions both from a technical and social standpoint. From the *technical standpoint*, we present the results obtained from an extensive experimental study performed by setting up a prototype of a multi-hop ad hoc network, and testing its performance on a small-scale ad hoc network (up to 12 nodes). From the *social standpoint,* we present the methods and tools to introduce the MobileMAN solutions to the users and to collect preliminary users' feedbacks about expectations and acceptance of this new networking paradigm.

Project funded by the European Community under the "Information Society Technologies" Programme (1998-2002)

CONTENTS LIST

SUMMARY

The aim of this deliverable is to document the first-phase evaluation of the MobileMAN paradigm. This evaluation addresses both technical and social issues. The aim is to test the current status of the MobileMAN solutions, and to identify possible risks in achieving the project goals (the development of a fully functioning testbed on which we can perform direct experimentation with users).

The technical evaluation is based on constructing and testing, in a realistic scenario, a simplified MobileMAN prototype with the aim to: i) fix software problems/errors, ii) identify integration problems and errors, iii) test existing software to decide if it is suitable for MobileMAN, or new solutions need to be developed; and iv) obtain preliminary estimates of a MobileMAN performance.

Although MANET research has been ongoing for more than ten years [BCGS04], there are relatively few experiences with real ad hoc networks. Instead, a large portion of protocols' development is done in (often unrealistic) simulation settings, only. In this deliverable we discuss lessons learned from our experimental work. Specifically, we present results from our prototype integrating a p2p middleware (*FreePastry*) on top of a multi-hop ad hoc network based on 802.11b technology. Recently, for this technology, it has pointed out the existence of an *ad hoc horizon* (2-3 hops and 10-20 nodes) after which the benefit of multi-hop ad hoc networking vanishes [GLNT04]. All the experiments we performed were inside this ad hoc horizon. The aim is to compare and contrast solutions for this realistic setting, and to quantify the Quality of Service (QoS) the system is able to provide to the users. Our measurements pointed out that also in this limited setting, several problems still exist to construct efficient multi-hop ad hoc networks. Cross-layering seems to be an effective approach to fix some of the problems identified in our analysis.

At this stage of the project, as direct testing with the users of the MobileMAN technology is not possible (the MobileMAN technology is not yet mature enough to be fruitful by users), for the social evaluation of the MobileMAN paradigm we identified a set of tools and methods to involve users and to collect their feedbacks. These include: an interactive users web site, users mailing list, direct presentations to the users, etc. All these "social" activities are presented in this deliverable together with very preliminary results of our interactions with potential MobileMAN users.

1. INTRODUCTION

One of the MobileMAN objectives is the development of a fully functioning testbed and experimenting with this testbed to gain insight on the real behaviour and potentialities of mobile ad hoc networks. This includes involving real users in the experimentation of a MobileMAN. The reaction of the users to this new philosophy, which impacts the application level as well, is an important measurement tool for tuning and modifying the ad hoc networking paradigm. At the same time, this activity is essential for discovering the ad hoc networking potentiality in terms of users' needs and requirements' satisfaction, as well as its social and economic value. As pointed out in the Annex 1, several risks are associated with these objectives:

- **Technical risks**: due to the interdependency of so many new technical elements, ranging from media access to applications. Software development and integration need to be carefully addressed to avoid problems at later stages.

- **Social risks**: users can be involved in the direct experimentation only when the MobileMAN technology is mature enough to be fruitful by not-expert people. This means that users' feedbacks can be obtained only in the final stages of the project.

To cope with technical and social risks, an **incremental phases** approach has been identified. From the technical standpoint this means to create incremental testbeds, while from the social standpoint different tools for the users' involvement in the first phases of the are required. In this way, even if our efforts to create a fully functional MobileMAN are not all successful, with the incremental approach, many piecewise contributions will still prove very useful.

This deliverable summarizes the first phase in this process both from the technical and social standpoint. Specifically,

- from the *technical standpoint* it presents the results obtained from a small area MobileMAN solution obtained by integrating part of the software we developed with existing software. This enabled us, by verifying that the system operates correctly, to: i) fix software problems/errors, ii) identify integration errors, iii) obtained preliminary estimates of the MobileMAN performance, and iv) to compare and contrast alternative solutions;

- from the *social standpoint* it presents the methods and tools to introduce the MobileMAN solutions to the users and to collect preliminary users' feedbacks about expectations and acceptance of this new networking paradigm.

2. EVALUATION OF MOBILEMAN TECHNICAL SOLUTIONS

According to Annex 1, the aim of this phase is to investigate the developed solutions in a quasi-static configurations (users move slowly) on a relatively small campus-wide MobileMAN.

During the second year of the project each project partner started to construct small testbeds to experiment and validate the developed solutions. Some problems emerged. Each partner was able to involve in the testbed construction only a limited number of laptops/PDAs and few researchers. The latter being the most critical constraint. As a consequence of these resources' limitations only very small tesbeds (i.e., 3-4 nodes) can be set up, in which only a limited set of software functionalities can be investigated. In addition, also software integration and experimentation cannot be done in an efficient way having each partner working in isolation. To cope with these problems, in the MobileMAN meeting in Helsinki (7-8 June, 2004), we decided to set up a group of junior researchers that should meet on a regular basis to integrate the developed software, and set up a "reasonable" size MobileMAN to validate the developed solutions. This group of people coordinated by CNR research assistants, Eleonora Borgia and Franca Delmastro, met in Pisa at the end of June for a 7-day (24 June - 2 July) MobileMAN first testing phase. Specifically, the following junior researchers:

- CNR: Eleonora Borgia, Franca Delmastro, Luciana Pelusi;
- HUT: Nicklas Beijar, Jarrod Creado, Olmo León;
- Cambridge: Ziran Sun;
- Eurecom: Claudio Lavecchia;
- Netikos: Veronica Vanni;

with the help of two CNR technicians (Riccardo Bettarini and Leonardo Mariani) set up in the CNR campus in Pisa a small area MobileMAN involving up to 12 nodes. These numbers may appear not meaningful respect to simulations scenarios using hundreds of mobile nodes. However, recent results pointed out the existence, with the current technology, of an *ad hoc horizon* of two-three hops and 10 to 20 nodes. Beyond these limits the benefit from wireless multi-hop ad hoc networking virtually vanishes [GLNT04].

It is also worth noting that, in the current literature, only Uppsala University developed a much larger testbed (about 30 nodes) [APE02]. However, the Uppsala testbed addressed only the routing functionalities of ad hoc networks. On the other hand, the MobileMAN testbed developed in Pisa integrated a peer-to-peer middleware (Pastry) on top of a multi-hop ad hoc network based on AODV and OLSR routing solutions. In addition, simple test applications have been used on top of Pastry to measure the application-level quality of service.

The main outputs of this testing has been:

- To identify, and whenever possible, to immediately fix the identified software problems;
- To verify that routing software was properly working, and that Pastry correctly operates on top of our small-area ad hoc network.
- A detailed presentation of the MobileMAN tests performed in Pisa is reported in the next section (see Section 2.1).

Starting from this experience, the CNR group (integrated with three master students[1] performing their training stage at CNR), in the second half of July, set up a new testbed with the aim to integrate previous qualitative results with a quantitative evaluation of a small area MobileMAN. The aim of this second set of experiments was to compare and contrast alternative solutions (e.g., proactive vs. reactive routing, etc.). A detailed summary of this experimentation is presented in the Section 2.2.

[1] Master in Internet Technology jointly organized and managed by IIT-CNR and department of Information Engineering of the University of Pisa.

2.1 MobileMAN Experimentation

An extensive experimentation on mobile ad hoc solutions was carried out in Pisa at CNR (Italian National Research Council) from 24th June to 2nd July 2004. The participants came from the CNR itself, the Technical University of Helsinki (Finland), the Institut Eurecom (France), the NETikos company (Italy), and the University of Cambridge (UK). The experiments concerned different layers:

- Routing: testing different implementations of proactive OLSR and reactive AODV routing protocols [BCGS04] to check their implementations' status, validate their functionalities and conduct a comparative analysis on them all;
- Middleware: testing an implementation of Pastry [DR01] working with different kind of routing protocols and evaluating the heaviness of the resulting solutions.
- Application: evaluating the impact of running routing protocols over some application performance parameters (e.g., transmission delay).

The considered routing protocol implementations were the UNIK-OLSR [OLSR] by the University of Oslo (Norway), the UU-AODV [UU] by the Uppsala University (Sweden) and HUT-OLSR [HUT] integrated in the Ad Hoc Framework environment proposed by the HUT University (Finland).

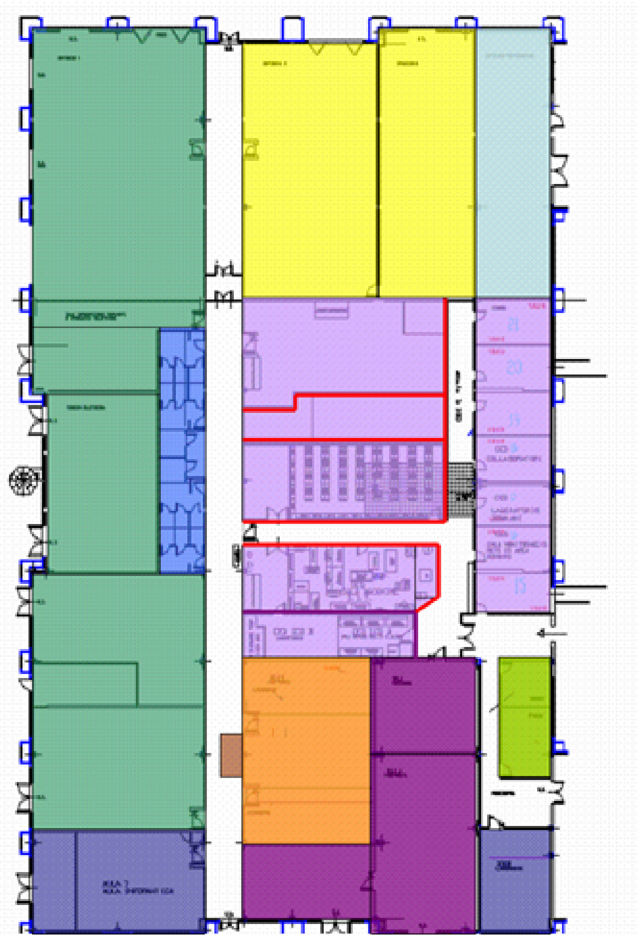


Figure 1: CNR ground floor

All the tests were performed at the CNR ground floor (Figure 1) where the CNR computing centre (CED) is located 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 panels" of plastic materials which don't reach the height of the ceiling (in red in Figure 1) or metal panels till the ceiling (in violet in Figure 1): 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.

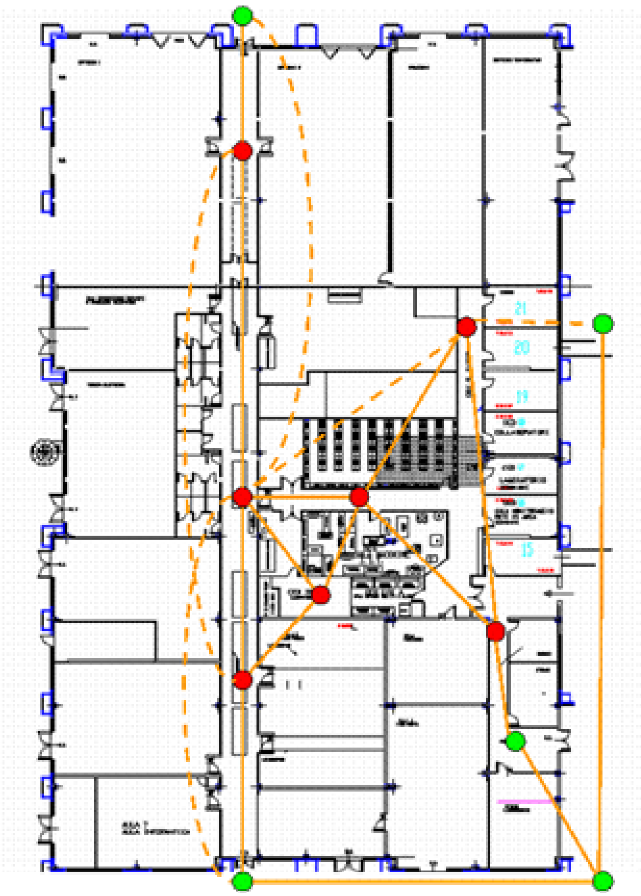


Figure 2

Figure 2 illustrates the detailed map of the place together with the transmission coverage characteristics of the area: nodes are situated where devices were placed during the experiments[2] 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); dashed lines are used instead to point out weaker wireless links wherever a couple of nodes see just sometimes each other and their communications are affected by a considerable packet loss.

[2] Terminals mobility was simulated by laptops/PDAs connecting/disconnecting from the network.

2.1.1 Devices

The devices used for the experiments were both laptops and PDAs and had wireless cards with different capabilities: this caused some links appear/disappear in different experiments depending on the power of wireless cards used by the nodes at each side of the link. In the following the devices will be referred to as numbers that correspond to the last byte of the IP addresses devices were assigned during the experimentation.

PDAs

- .2 .3 .4 .5 .6:
  Model: Compaq iPAQ 3950
  Processor: Intel® XScale-PXA250 – 400MHz revision 4
  Wireless Card: PCMCIA D-Link DCF-660W

Laptops

- .1, .20, .30:
  Model: IBM ThinkPad R40 Series - Centrino® Mobile Technology
  Processor: Intel® Pentium® M - 1300MHz
  Wireless LAN PC Card: D-Link DWL 650 – 15 dBm
- .10:
  Model: IBM ThinkPad R40 Series - Centrino® Mobile Technology
  Processor: Intel® Pentium® M - 1300MHz
  Wireless LAN PC Card: 3COM 3CRWE62092A – 14 dBm
- .50:
  Model: IBM ThinkPad R40 Series
  Processor: Mobile Intel® Pentium® 4 – 2.000GHz
  Wireless LAN PC Card: 3COM 3CRWE62092A – 14 dBm
- .60:
  Model: Acer Aspire 1200XV
  Processor: Celeron - 1GHz
  Wireless LAN PC Card: D-Link DWL 650 – 15 dBm
- .90:
  Model: Compaq Evo N110
  Processor: Celeron - 550MHz
  Wireless LAN PC Card: D-Link DWL 650 – 15 dBm

2.1.2 Journal of the experiments

Testing UNIK-OLSR

Experiments using the OLSR implementation by the University of Oslo (UNIK-OLSR) were carried out on 24th June on networks which consisted of both 4 and 5 nodes (experiments 1-5) and on 29th June on a bigger network of 12 nodes (experiments 6-9).

The first set of experiments over small networks provided a first study of UNIK-OLSR: with a few nodes the kernel routing tables were small and could be read very quickly, as a result, it was possible to follow configuration changes while in progress, also thanks to the helpful UNIK-OLSR Graphical User Interface (GUI). 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 delay that configuration updates experimented after the appearance/disappearance events. Finally, it was checked whether the established routes were the shortest ones or not.

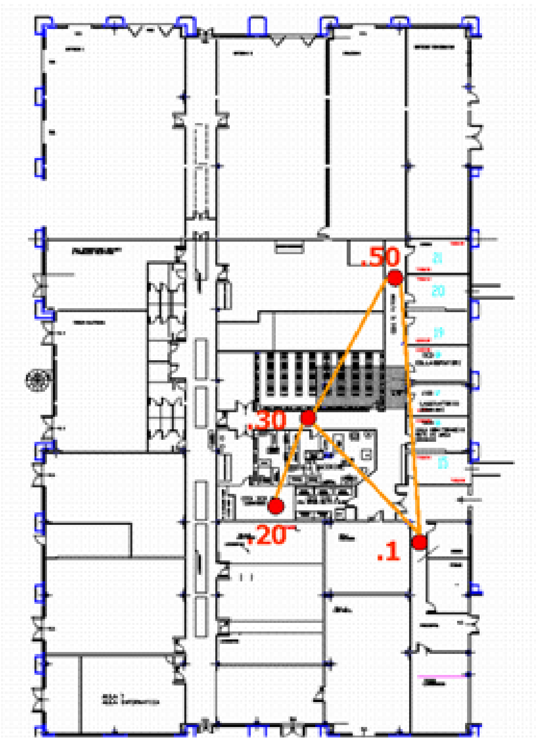


Figure 3

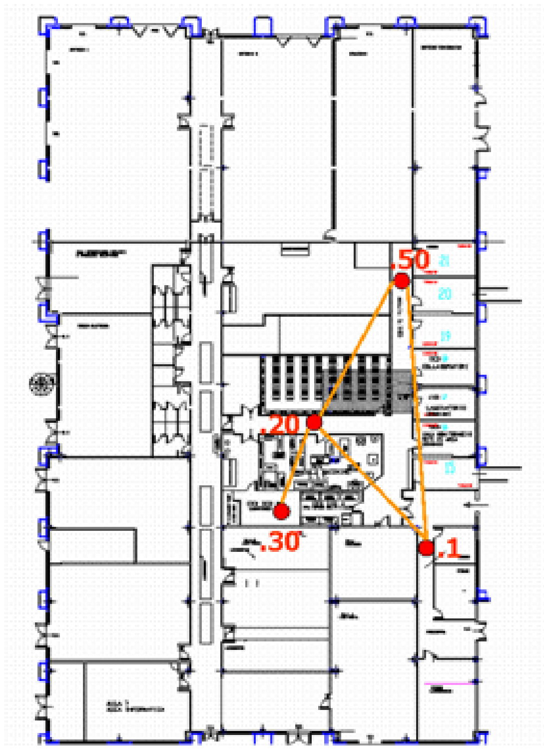


Figure 4

- Experiment 1: nodes to be used were .1, .20, .30, .50, all laptops (Figure 3). Three of them -.1, .30, .50- were started together (t = 0s) while .20 was started 60s later (t = 60s). All the laptops worked together for the next 60s, then (t = 120s) .30, which was the central node, was stopped first and all the other laptops stopped 30s later (t = 150s). The protocol behaviour was as expected: upon .20's insertion the configuration was soon updated and upon .30's disappearance the network split into two networks one with the standalone .20 and the other with both .1 and .50.

- Experiment 2: nodes to be used were the same as in the experiment 1 (Figure 3). Nodes .1, .30 and .50 were started together (t = 0s) and .20 was started 60s later (t = 60s). All the laptops worked together for the next 30s, after that (t = 90s) .50 was stopped first and 30s later (t = 120s) .30 was stopped too; finally after other 30s all the other laptops were stopped (t = 150s). Compared to the previous experiment, this time a shorter time lag (30s vs. 60s) was waited after that the last node had joined the network, before starting nodes' removal; the same time lag was waited between consecutive nodes' removals nevertheless its duration was sufficient to allow the establishment of both the first configuration and the subsequent updates.

- Experiment 3: nodes .1, .20, .30, .50 were used as in previous experiments 1 and 2 but nodes .20 and .30 were inverted (Figure 4): .30 was placed inside the laboratory because its battery was flat and needed power supply which wasn't instead provided where it had been previously placed. This experiment provided two node insertions upon the beginning, and one single node disappearance before the end: nodes .1 and .50 were started together (t = 0s), .30 was started 30s later (t = 30s) and .20 after other 30s (t = 60s); all laptops worked together for the subsequent 30s and then (t = 90s) .50 was first stopped while all the other laptops after the next 90s (t = 180s). This time, the initial configuration setting was given a shorter time (30s instead of 60s) before the network started to change because of node insertions and deletions anyway the protocol could always timely realize the correct configuration.

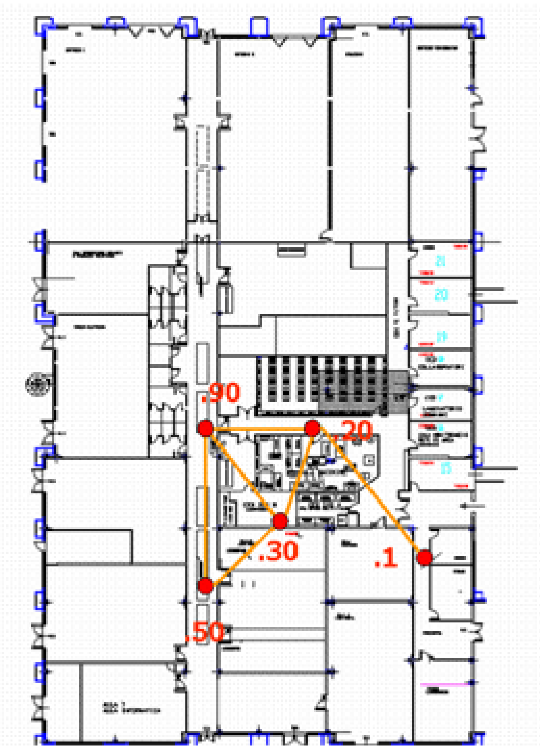


Figure 5

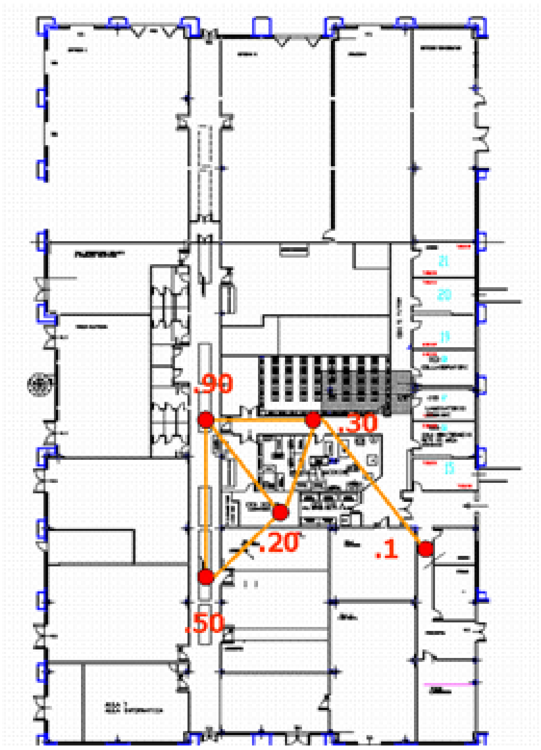


Figure 6

- Experiment 4: this experiment, as the next one, was performed with one more node than the previous ones so more links were present and consequently more alternative paths between couples of nodes too; this was helpful to check protocol choices about paths. Nodes .1, .20, .30, .50, .90 were used (Figure 5); nodes .20, .30, .90 were started together (t = 0s), .1 was started 30s later (t = 30s) and .50 30s later again (t = 60s). All laptops worked for the subsequent 120s and then stopped all together (t = 180s). The protocol always chose the cheapest path (the one with the minimum number of hops) between any couple of nodes. The complete configuration, at the end of the experiment, had been kept unmodified for 120s however, some configuration changes were noticed so the same experiment was repeated and a longer final time lag applied: its description follows.

- Experiment 5: the same nodes as in the previous experiment were used however nodes .20 and .30 were inverted (Figure 6) because of battery problems, as before. Nodes .20, .30, .90 started first (t = 0s); .1 and .50 started 30s later one at a time distance of 30s from the other (t = 30s, t = 60s); all the nodes stopped together after the subsequent 180s (t = 240s). Upon the last modification to the network, some changes in the configuration could be noticed as in the previous experiment despite having increased the time lag duration before the final stop. In fact those configuration changes weren't sort of mistakes but the effect of continuous topology refreshes the protocol had been providing. It seemed that the protocol, in order to refresh a kernel routing table entry, provided to completely remove it first and then rewrite it from the beginning. Due to these refreshes the kernel routing tables had been changing continuously and sometimes they even looked completely blanked out (when all their entries needed refreshing). Another problem was the instability of the links which was sometimes responsible for the temporary disappearance of a/some node/s and consequently of some links too: these were timely captured by the running protocol and immediately caused the update of the network topology. Continuous updates were however very quick and, as a result, the global network configuration could be considered quite stable.

The next set of experiments was performed on 29th June over a bigger network of 12 nodes. As the nodes consisted of both laptops and PDAs, this mixed environment could be considered quite a realistic ad hoc network. Moreover, the increased number of nodes led to the increase of the number of protocol-packet exchanged, which allowed the validation of the previously observed behaviour of UNIK-OLSR in a more critic context. The experiments were generally organized as follows: a group of internal nodes had to start first at the beginning; the external nodes had to start later (the delay was different in different experiments to check the protocol reaction time). In some experiments node deletions were provided too and the number and the position of the nodes to disappear also varied. Log files were created by picking up the kernel routing tables every 10s. Often the same experiment was repeated more than once to yield more consolidated results.

- Experiment 6: involved nodes were .2, .3, .4, .5, .6, .1, .10, .20, .30, .50, .60, .90 (Figure 7). Nodes .1, .10, .20, .30, .50, .60, which were all central nodes, started first at the same time (t = 0s) while .2, .3, .4, .5, .6, .90 started with a delay of 120s (t = 120s): this delay was introduced to let the network reach quite a stable configuration. All devices stopped together after a time lag of 120s (t = 240s). This experiment was repeated three times. The first time there were problems with nodes .90, .6 and .5: they all couldn't be seen. .90 had been situated behind the closed firebreak doors and .6 behind the closed external door so to be sure that .6 could be seen by .1 only through .90 and not directly; however the link between .1 and .90 resulted too weak so .90 could be seen by .1 just sometimes while .6 never. In subsequent trials the firebreak doors were kept open and the external door closed so both .90 and .6 could be seen by .1, however, the presence of the quite instable link between .6 and .1 was responsible for lots of configuration changes in kernel routing tables because in fact the protocol always chose the cheapest path towards .6 and it was, from .1, either direct or through .90 depending on the link .1-.6 app/disappearance. On the other hand, to cope with the visibility of .5 (which had been previously kept behind the closed external door) this door was open. Subsequent repetitions of the experiment resulted in a more correct behaviour except for the lack of the link .4-.5: .5 could only be seen either through .60 or directly through .1. This problem was solved in subsequent experiments by slightly changing the position of the node .5: a little farer from the external door. The rest of the network configured as expected. Just an ending note about the absence of the link .10-.30 as shown in Figure 7; this link was present in previous 1-5 experiments however either the laptop .20 or .30 were used in those experiments instead of .10 and they had more powerful wireless cards (D-Link DWL 650 – 15dBm) than the laptop .10 (3Com 3CRWE62092A – 14dBm).

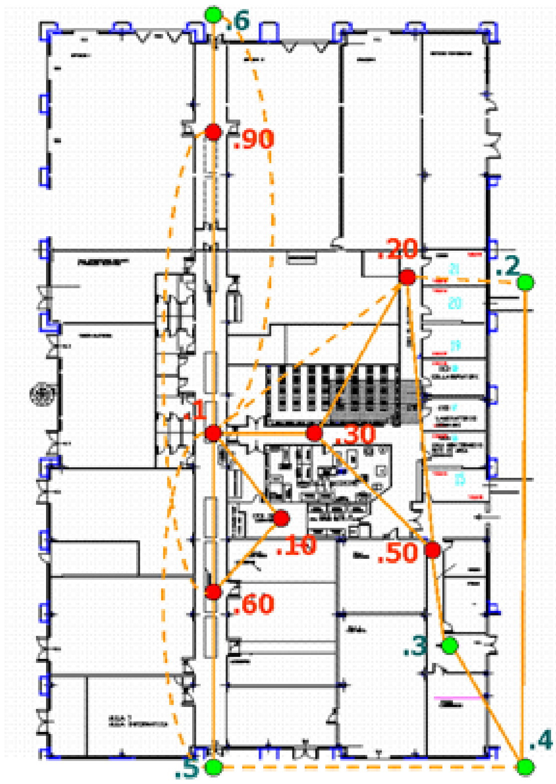


Figure 7

- Experiment 7: the network topology was the same as in the previous experiment (Figure 7); this time the network was given less time to configure, then a node disappearance was provided to verify that the network could reconfigure itself. At first nodes named .1, .10, .20, .30, .50, .60 were started (t = 0s). After a time lag of 90s (t = 90s) nodes .2, .3, .4, .5, .6, .90 were started too and 90s later (t = 180s) the device .30 was stopped. The remaining devices continued running the protocol for other 120s, then (t = 300s), they all stopped together. Upon the stop of the node .30, the network split into two distinct parts, one with .1, .10, .60, .5, .90, and .6, the other with .2, .20, .50, .3 and .4: .4 couldn't see .5 and .1 couldn't see .20 so the two networks remained distinct till the end of the experiment. The experiment was carried out another time over the same network as in the first trial except for the position of the device .20 which was moved (Figure 8) to improve its visibility to the rest of the network and have one single network reconfigured upon the disappearance of .30. Nevertheless the link between .1 and .20 was too weak, so when .30 stopped running the protocol and the network started to reconfigure, nodes could observe in their kernel routing tables all the other nodes only 2-3 times out of 11. The resulting configuration was very unstable till the end. Moreover, in both the repetitions of the experiment it was possible to observe the instability of links connecting .6 and .5 to the rest of the network. .6 was generally seen directly through .1, sometimes through .90; .5 instead was generally seen through .60 sometimes it was seen directly through .1 sometimes neither .6 nor .5 could be seen at all. Finally almost always no link was present between .20 and .2 so there was no alternative path towards .2 but through .4.

- Experiment 8: this experiment was carried out over the same network configuration as in the experiment 7 (Figure 8). Nodes .1, .10, .20, .30, .50, .60 started all together at first (t = 0s). 90s later (t = 90s) .2, .3, .4, .5, .6, .90 started too. The entire network ran the UNIK-OLSR protocol for the next 90s, then (t = 180s) the node .10 stopped and after a time lag of 120s (t = 300s) all the other devices stopped too. The experiment was the same as the previous one but the choice of the node to stop first was different so to point out the difference in the reconfiguration of the network when one of the central nodes disappeared. The experiment was repeated twice. During the first time the node .3 had problems at the beginning and stopped immediately. The second time the node .2 was slightly changed in

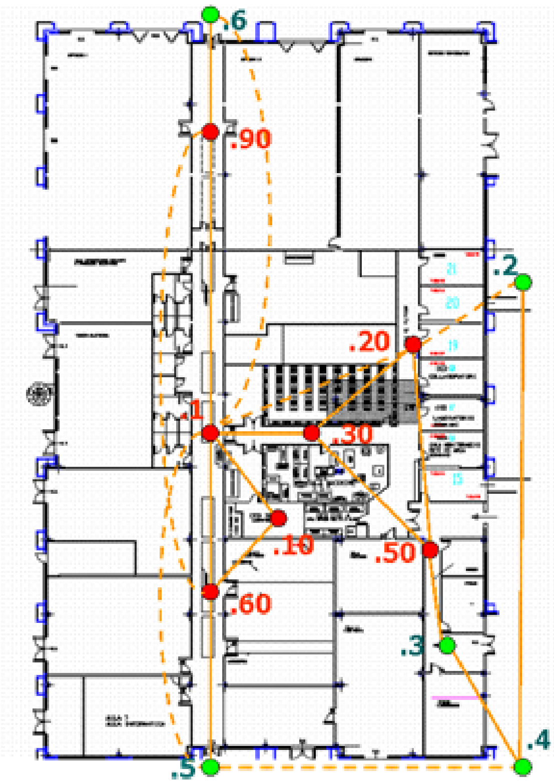


Figure 8

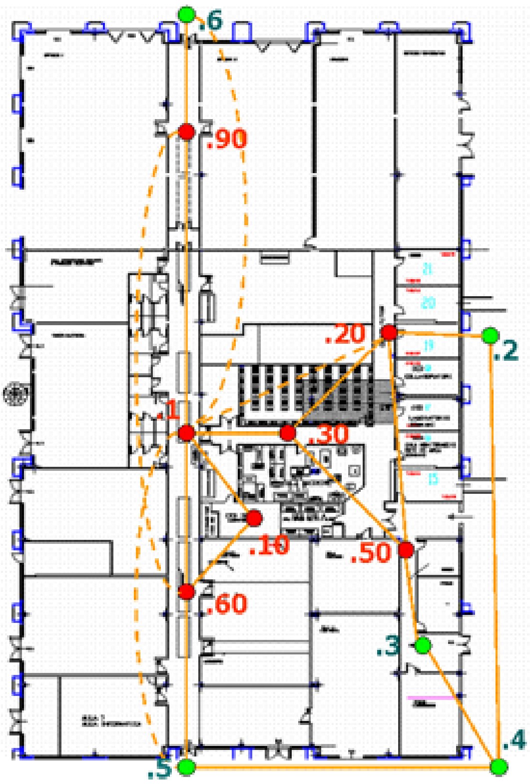


Figure 9

position for a better visibility with the .20 node (Figure 9). In both repetitions upon the disappearance of the .10 node the remaining nodes succeeded in reconfiguring into one single network: in the first trial nodes .3, .4, .2 weren't visible due to the crash of .3 together with the weakness of the links .20-.2 and .5-.4. The second trial was carried out with both .2 and .5 changed in position: .2 was put in line with .20 and .5 was moved a little farer from the external door -which was also closed- more next to .4; the network configuration became finally complete.

- Experiment 9: the network used was the same as in the previous experiment (Figure 9). Nodes .1, .10, .20, .30, .50, .60 started all together at first (t = 0s); 90s later the nodes .2, .3, .4, .5, .6, .90 started too (t = 90s). Subsequently the .20 node stopped after 90s (t = 180s) and after 30s again .60 stopped too (t = 210s). All the remaining nodes continued to work for a time lag of 90s, then stopped together (t = 300s). This experiment was repeated twice too. The network reconfigured well upon both the node disappearances: paths towards .2 stopped being through .20 and moved through .50; .5 could also be seen through .4.

Testing HUT Ad Hoc Framework

The Ad Hoc Framework by the University of Helsinki could not be completely tested because the implementation of the new release of AODV from Uppsala University [UU] was released recently and its integration within the HUT framework was not yet completed when the experimentation started. So only the OLSR and CCRS modules were tested in an integrated way, while AODV was tested in isolation (see next section). As far as OLSR, HUT researchers evaluated that the complexity to integrating existing OLSR implementations in the developed routing framework was higher than developing their own OLSR implementation (HUT-OLSR) integrated in the Ad Hoc Framework. In order to compare the two implementations of OLSR (HUT-OLSR and UNIK-OLSR), the same experiments as for UNIK-OLSR were performed but, as further explained in the following, smaller networks had to be used; moreover a pure comparison wasn't possible as well because to date the HUT-OLSR didn't exactly follow the OLSR specs as described in RFC 3626 and, particularly, didn't include yet the Dijkstra algorithm for the shortest path calculation.

- Experiment 1: in this experiment some nodes had to start running together at the beginning (t

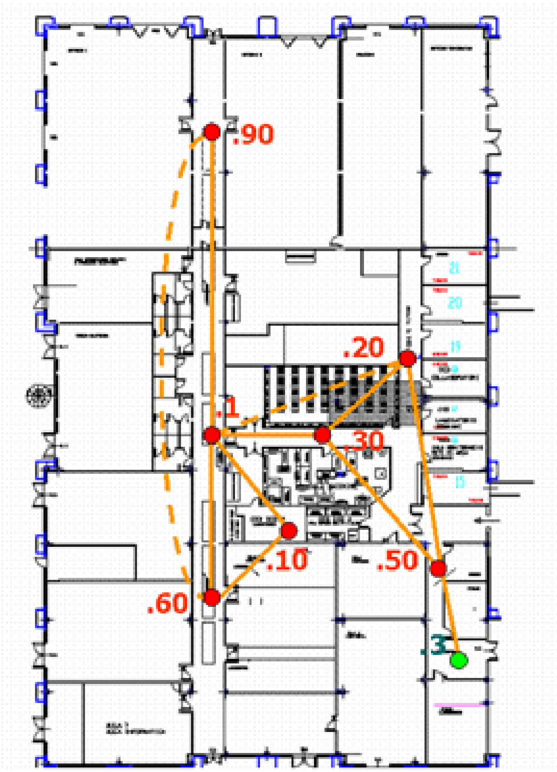


Figure 10

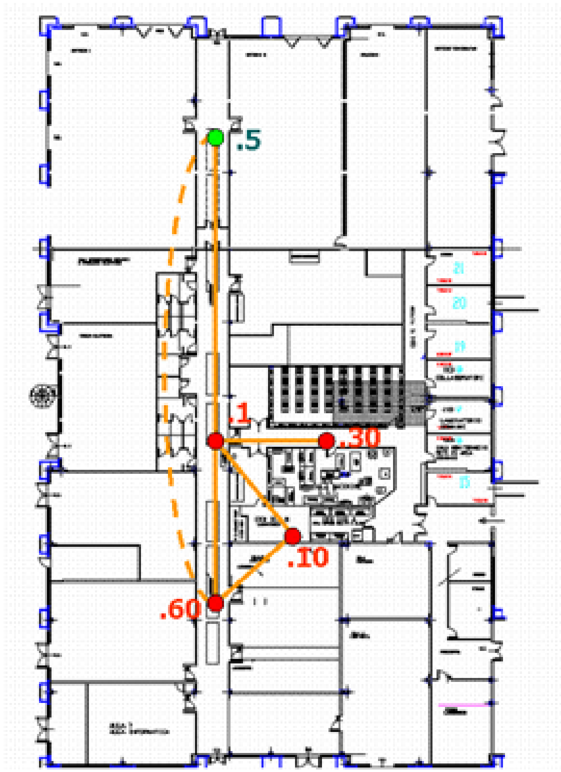


Figure 11

= 0s) while all the remaining nodes 120s later (t = 120s). They all had to work for the next 120s and then stop together (t = 240s). The experiment had to be repeated many times because there were some problems with the code: firstly when the protocol was stopped on a node the kernel routing table wasn't blanked out and this resulted in permanent routes remaining in the kernel until they were manually deleted, as a result subsequent experiments were corrupted;

secondly when the protocol was irregularly interrupted (e.g., when an error in the starting script occurred and one/some node/s didn't start and the whole experiment was first interrupted then restarted) some UNIX local sockets remained dangling and avoided running a subsequent protocol execution (an exception raised when the protocol tried to create a new socket because it already existed): this was a very subtle problem because UNIX local sockets aren't deleted even if a reboot is forced. After having fixed the code the network was reduced to a smaller one for debugging reasons then, when a correct behaviour was finally observed, new trials were performed over bigger networks. The network used for the first trial consisted of 8 nodes: .1, .10, .20, .30, .50, .60, .90 and .3 which was the only PDA to be used (Figure 10). Nodes which started first were .1, .10, .20, .30, .50 and .60 while the last nodes to start were .3 and .90. The node .3 had problems at the start up and didn't work at all. The second time the network was smaller and consisted of only 5 nodes: .1, .10, .30, .60 and .5 (Figure 11); .1, .10, .30 and .60 started at first (t = 0s), .5 started 120s later (t = 120s) and they all stopped together after a time lag of 120s (t = 240s). The third time the network was 7 node big (Figure 12). The devices which participated were all laptops except for 2 PDAs: .1, .10, .30, .50, .60, .3, .5; .1, .10, .30, .50 and .60 started first, .3 and .5 later. The fourth time, in the end, the network was even bigger (8 nodes), composed by the nodes .1, .10, .30, .50, .60, .90, .3 and .5 (Figure 13). Nodes which started first were .1, .10, .30, .50, .60, and .90 while .3 and .5 the ones which started later. The protocol had finally a correct behaviour except for the choice of the routes. Network coped with the first configuration at the beginning and with the insertion of external nodes later. Continuous kernel routing table refreshes were noticed as for UNIK-OLSR.

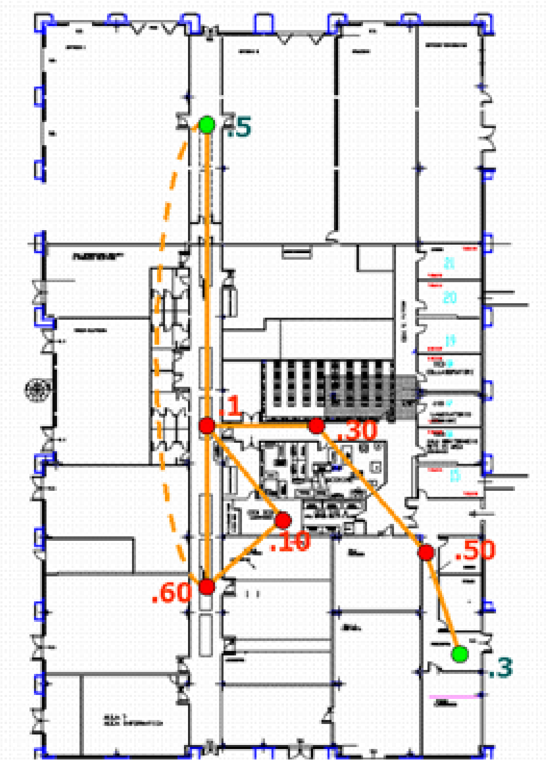


Figure 12

- Experiment 2: 8 nodes were used as in the last previous experiment (Figure 13). Devices .1, .10, .30, .50, .60 and .90 started first (t = 0s), .3, and .5 started 90s later (t = 90s). After the next 90s .30 was stopped (t = 180s) while the other nodes continued working for the subsequent 120s then stopped all together (t = 300s). This experiment was repeated twice: the first time the PDA .3 had problems and started later than 90s from the beginning. Network coped with the deletion of the central node, even if routes towards .30 completely disappeared from the kernel routing tables of the other nodes only after a long delay.

- Experiment 3: the experiment was performed over the network showed in Figure 14. Devices .1, .10, .20, .30, .50 and .60 started first (t = 0s); .3, and .90 started 90s later (t = 90s). After the next 90s .10 was stopped (t = 180s) while the other nodes continued working for the subsequent 120s then stopped all together (t = 300s). The experiment was repeated twice and the protocol worked as expected, even if upon the removal of .10 long delays were experimented by the other nodes before they realized its disappearance and as a result the network reached the new correct configuration.

- Experiment 4: this experiment used the topology illustrated in Figure 14 too. The nodes .1, .10, .20, .30, .50 and .60 started first (t = 0s); .3, and .90 started 90s later (t = 90s). After the subsequent 90s (t = 180s) two nodes had to stop: .20 first and .60 30s later (t = 210s). All the other nodes continued to work for a time lag of 90s and then stopped (t = 300s). This experiment was repeated twice and both times the protocol behaviour was as expected and suffered the usual delays before configuration updates took place.

In conclusion, during the experimentation for the HUT-OLSR it could be noticed that the protocol spent a lot of time to produce stable configurations at the beginning of the experiments and to execute the necessary configuration updates upon nodes' insertions and removals; moreover it lacked the implementation of the Djikstra algorithm for calculating the path with the minimum number of hops between couples of nodes. As the work required to update HUT-OLSR could not be done immediately, it was decided not to continue to utilize it as routing protocol for the subsequent tests (See Section 2.2).

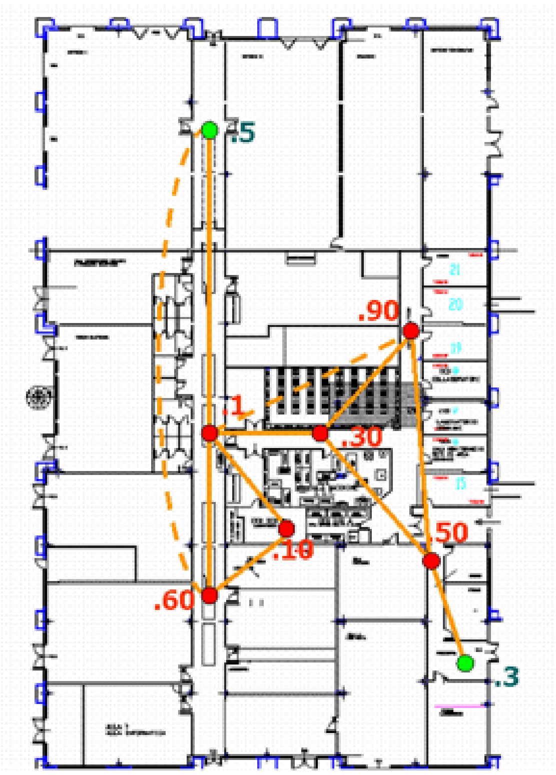


Figure 13

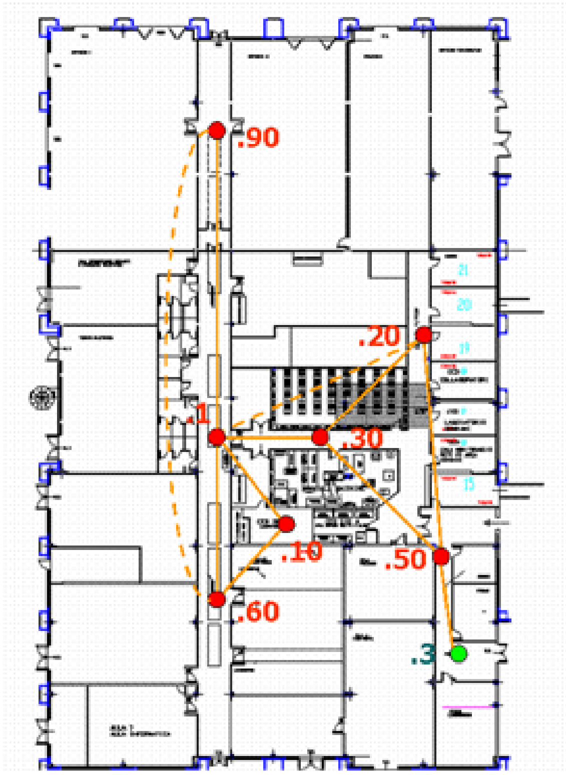


Figure 14

Testing UPPSALA UU-AODV

Due to the lack of time one single type of experiments (Experiment 1, below) was performed to test the AODV implementation by the Uppsala University. The experiment was performed over a 9-node wide network which was the biggest network that could run UU-AODV. In fact, in order to run UU-AODV the Linux kernel has to provide the netfilter support and this has to be necessarily set up during the recompilation of the kernel itself. Unfortunately, some laptop kernels lacked the netfilter support and couldn't be used to test UU-AODV.

For the experiment, as the protocol is proactive, 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 so to have routes toward other nodes generated.

- Experiment 1: the utilized network consisted of the nodes .2, .3, .4, .5, .6, .10, .40, .50 and .90 (Figure 15). All the nodes started together (t = 0s), ping-ed each one of the other participants of the network, one at a time, for 60s and finally stopped. On each node the protocol could find out the correct paths towards chosen destinations, however the discovery of the paths was very time-expensive.

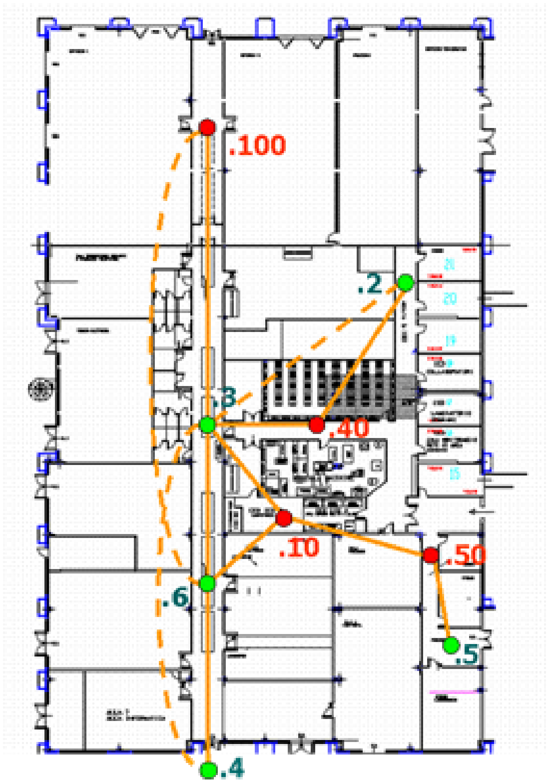


Figure 15

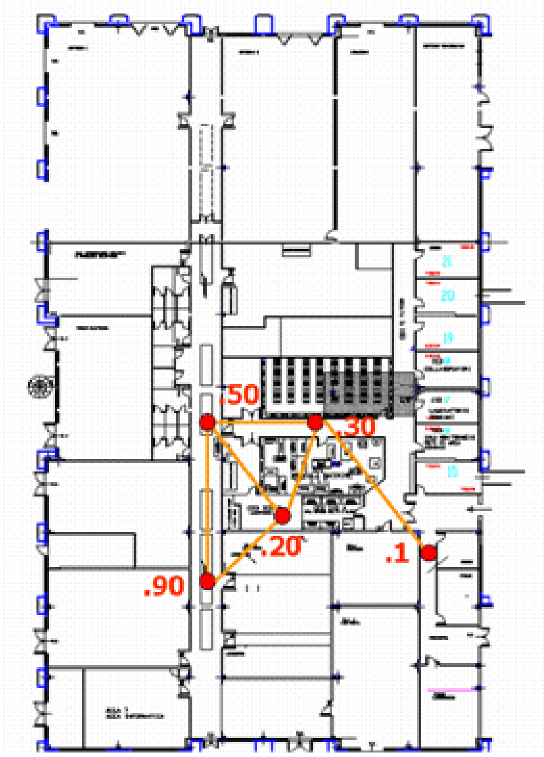


Figure 16

Testing the routing heaviness

A set of experiments was performed aimed at evaluating the impact of a running routing protocol on the performance parameters of an application running over it. Three set of experiments were carried out: the first two with UNIK-OLSR, the last with the UU-AODV.

- Experiment 1: the network consisted of five nodes (.1, .20, .30, .50, .90), all laptops (Figure 16). 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 launched on the host .1 towards the host .90. The whole file to be transferred was 34MB long. The destination was at three hop distance from the source and the path was through the hosts .30 and .50. After a short while the host .50 had problems and stopped working, so the transmission changed path and passed through .30 and .20, however the whole transmission crashed later again because .20 also stopped working. The destination host .90 had received only the first 15MB of the file. The throughput during the transmission had just been about 180Kbps.

Differently from the first experiment the next two were performed on a different day after the set of experiments on FreePastry so the same unchanged network topology as for FreePastry was used (see following section for more details).

- Experiment 2: the network used this time was 8-node wide (Figure 17). As in the previous experiment all the participants started running UNIK-OLSR together and after a slight delay the FTP file transfer was started too, from the source host .20 to the destination host .1 (which was 3 hop distance far). This time the whole file was 2MB long. 427KB arrived

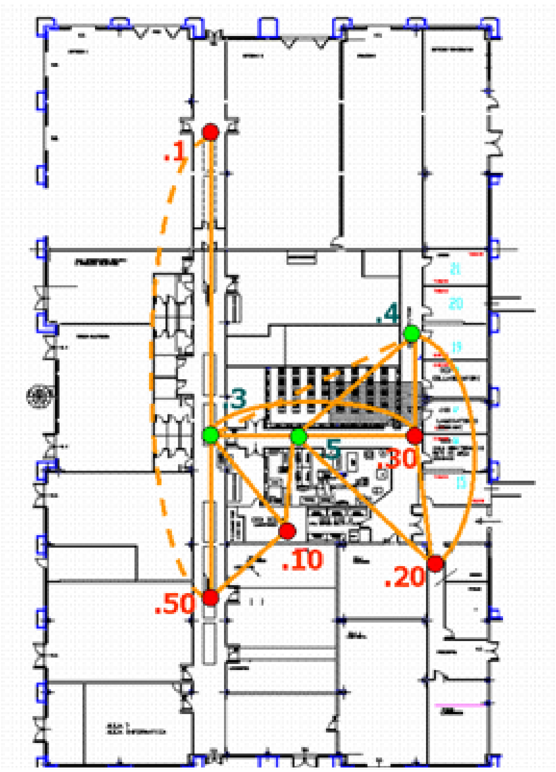


Figure 17

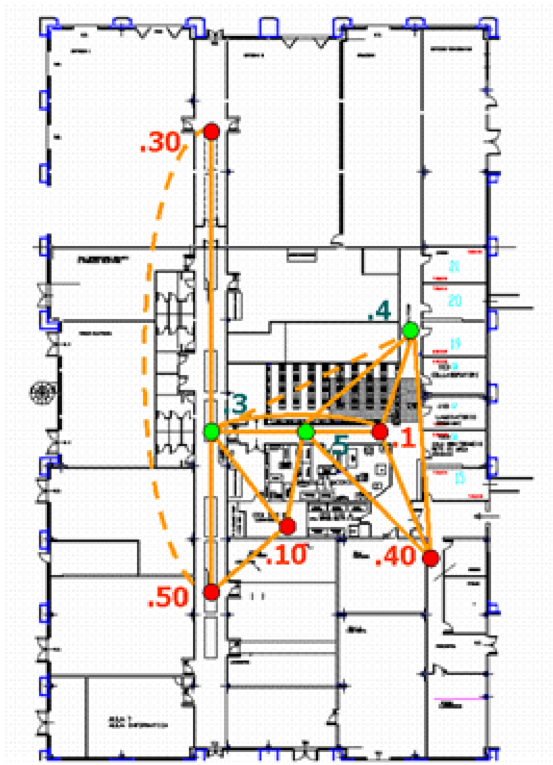


Figure 18

to the destination quite soon, then something got wrong and the experiment had to stop after 5 minutes because the destination host .1 had the whole kernel routing table blanked out and gave up working.

- Experiment 3: the network used was 8 node big (Figure 18) as in the previous experiment. This time the routing protocol was changed. All the participants started running UU-AODV together; the FTP file transfer was started, later, from the source host .40 to the destination host .30 (3 hop distance far). The whole file was 5MB long. The transfer was definitely too slow and after 16 minutes only 140KB had reached the destination; the experiment was interrupted then.

In conclusion both routing protocols seemed to find the correct path towards the destination however data transfers had lots of problems resulting in either interrupting the entire transmission or going on very slowly.

Testing FreePastry on the ad hoc routing protocols

The last set of experiments was carried out in order to evaluate the overhead introduced by a middleware platform based on the original Pastry model [DR01] and to validate its functionalities on the ad hoc networks. The middleware platform used in these experiments was FreePastry-1.3 which is the open-source implementation of the original Pastry model developed by the RICE University [FreeP]. The environment for the experiments was set up by installing FreePastry on a set of laptops which had been previously equipped with the j2sdk-1.4.02 [Java] Java Virtual Machine. On the same laptops, a simple application of Distributed Messaging (DM) was installed too, aimed at testing the main Pastry functionalities. This application implements the P2P common API proposed by [DKSDZ03] and defined in FreePastry. It defines a common interface between the application and middleware platforms based on the overlay network concept. Each instance of the messaging application defines an IDentifier for the local node and requires the user to specify the relationship between the local node and the Pastry ring, i.e., it asks the user if the local node is the first node of the ring or if it knows another node to connect to. In the first case, the node creates a new ring, while in the latter it has to specify the IP-address of the known physical neighbour so that the local node can connect to it, collect its middleware routing table and enter the ring. Once the local node has created/joined a ring, the application provides to the user the possibility to create a mailbox, to delete a mailbox, to send a message to a mailbox and finally to receive a message from a mailbox. The Pastry subject-based routing of the application messages requires that a logical IDentifier has to be assigned to each message, in order to distribute the content on one of the participating nodes following the proximity logic. For this reason, in order to create a mailbox, the user has to specify its IDentifier that represents the key of the message on which the hash function is computed. Our testing DM application also provides a function to generate 100 random mailboxes and the associated IDs sending the related messages on the network. In this way, FreePastry functionalities can be validated in terms of workload, data distribution and overlay construction. Moreover, it is also possible to evaluate the overhead introduced by this platform on ad hoc networks. Particularly, the large number of remote connections needed to maintain the overlay structure is expected to represent a big overhead on ad hoc networks, especially in case of mobility and frequent topology updates.

In order to compare and contrast results, the same set of experiments was repeated running UNIK-OLSR and UU-AODV as routing protocols and the same network topology was used too. The network topology was kept as much similar as possible to the one used for previous experiments. In this case the global number of nodes was decreased to 8: 6 nodes ran FreePastry and the others just worked as routers. The point was that in the usual topology PDAs were all external nodes and laptops internal nodes. However it wasn't possible for the lack of time provide the complete set up of the FreePastry environment on PDAs and they could be used just as routers. Since in the standard topology PDAs were always located in external positions, if they had been maintained they wouldn't have been involved in the middleware traffic and wouldn't have added any results to this set of experiments. For this reason the number of PDAs was reduced to 2 (.3 and .4) and they were located in central positions. The network used in these experiments was always 8 nodes wide, nevertheless there was a continuous turnover of devices for each single node and the exact configurations are illustrated in Figures 19-23. As previously said, each node running FreePastry could join the ring overlay by directly connecting to another node already present in the ring. In order to simplify the remote connections needed to establish the ring, in these experiments each node was forced to contact one of its physical neighbours. In this way the first connection aimed at joining the ring was a one hop connection and as a result not highly influenced by the routing protocol. In order to better distribute the Pastry joining operations another node was added in the hallway inside the CED (the horizontal one in Figure 1). This created a redundant network topology, however it wasn't relevant to the middleware behaviour because the ring overlay always represents a logical topology which is completely independent from the physical network topology.

During the experiments the nodes started running either UNIK-OLSR or UU-AODV then, after a delay of a few seconds to have the network topology stabilized, they ran the DM application trying to build a single pastry-ring. Details about each single experiment follow.

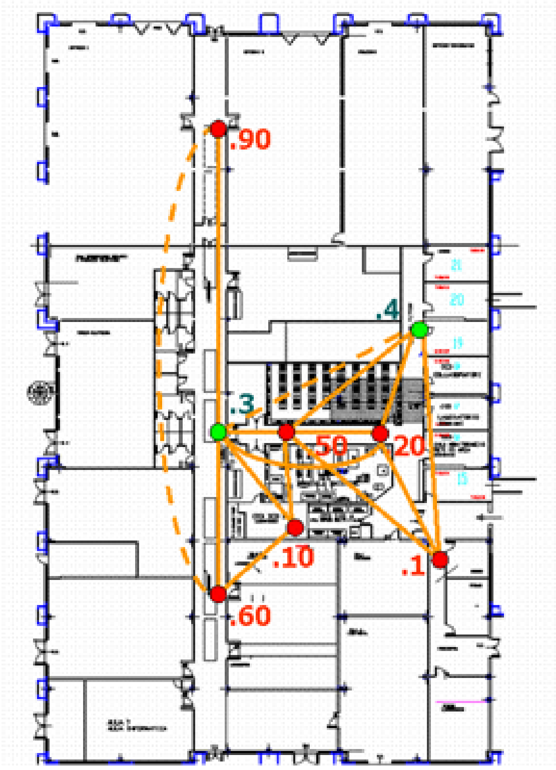


Figure 19

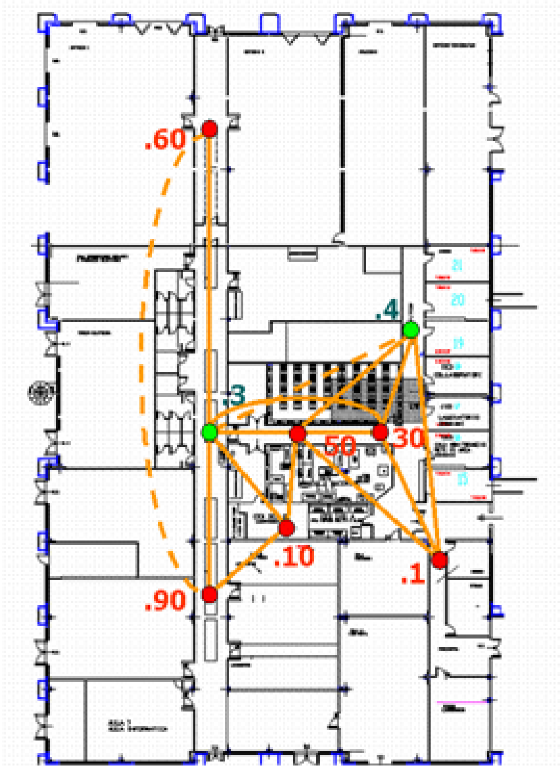


Figure 20

- Experiment 1: the exact configuration is illustrated in Figure 19; the routing service was provided by UNIK-OLSR. The DM application was started first on .60 that created the ring then .10 joined the ring. When .50 tried to join too connecting to .10 it failed and started a new ring standalone. Nodes .20, .90 and .1 joined the ring created by .50; .20 entered the ring through .50 while .90 and .1 through .20. As soon as the two rings had been created, the nodes started to distribute mailboxes all over their rings following the sequence: .20, .50, .1, .60, .10 and .90.

- Experiment 2: it also was carried out using routing provided by UNIK-OLSR; details about the spreading of devices are showed in Figure 20. The node .90 created the pastry ring however no nodes could join it because .90 was too slow and when other nodes attempted to connect to it a Java "connection refused" exception raised from the socket caused by the socket timeout elapsing. So .90

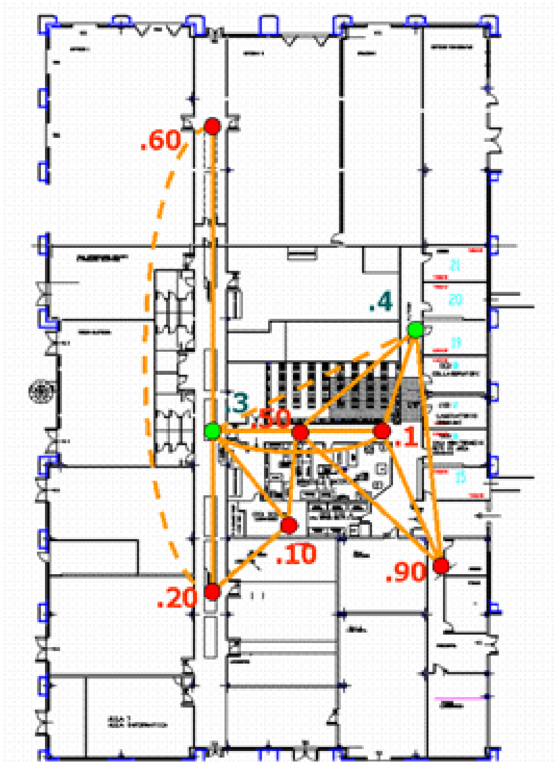


Figure 21

was stopped and .50 created the new pastry ring then .10, .90, .1, .60 joined the ring through .50. In the end nodes participating to the ring started to distribute the mailboxes. The sequence was: .90, .10, .60, .1, .50. The node .30 crashed running the DM application.

- Experiment 3: (Figure 21) it was the last experiment performed with routing service provided by UNIK-OLSR. The pastry ring was started by .20 then .10 and .60 joined the ring through .20, .50 through .10, .1 through .50 and .90 through .1 in the end. The sequence for distributing mailboxes was: .60, .10, .20, .50, .1 and .90.

From this set of experiments it could be noticed that messages were distributed on distant nodes with a low probability, and rarely application messages had to execute a multi-hop middleware path. This was due to the 160-bit size of the local IDs and to the middleware routing table dimensions. For FreePastry the standard dimension of a Pastry routing table is

$$\frac{ID\_length}{Base\_length} = \frac{160}{4};$$

(*Base_length* represents the number of bits of the base chosen to represent the logical ID) rows and ($2^{Base\_length} = 2^4$) 16 columns (because each column represents a possible value assumed by a digit of the logical ID). Hence using a network of 8 nodes, with only 6 nodes taking part to the ring there is a high probability that each node knows the others and the subject-based middleware routing

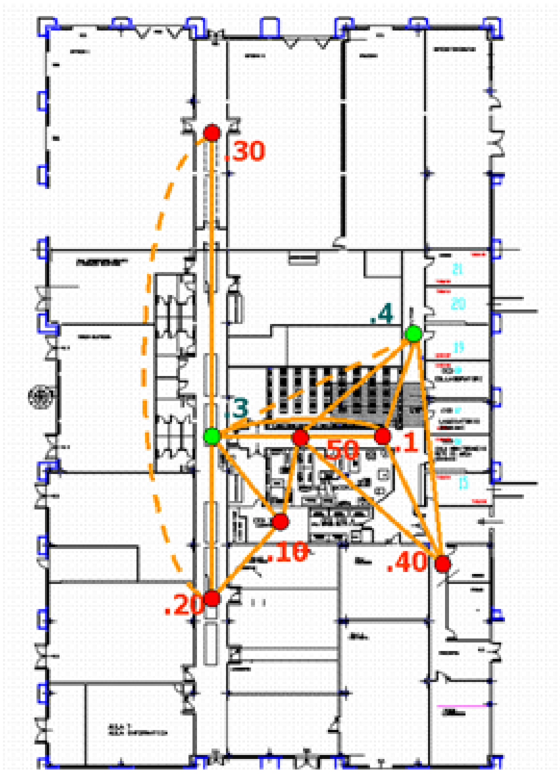


Figure 22

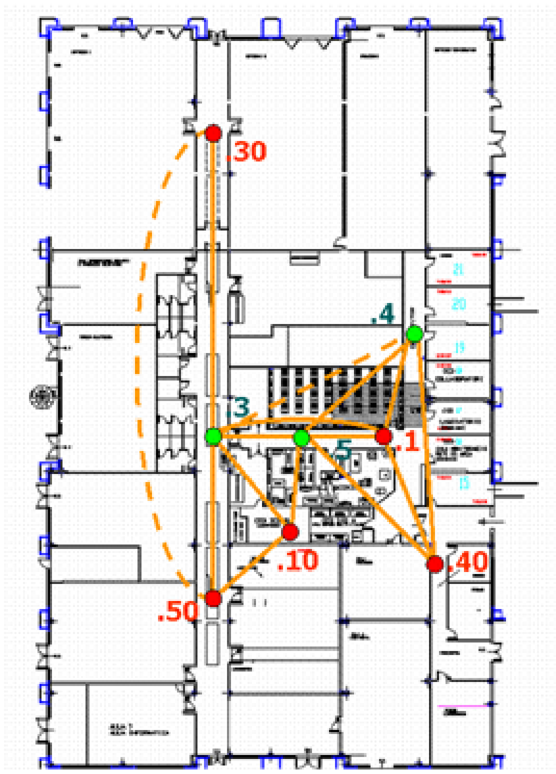


Figure 23

becomes a direct peer-to-peer connection. The real overhead introduced by FreePastry on ad hoc networks is due to the periodical remote connections need to maintain the ring topology.

- Experiment 4: (Figure 22) it was carried out with UU-AODV at network layer. The pastry ring was created by .50 then .10 joined the ring through .50 so as .30 and .1 then .20 tried through .10 and .40 through .1 but they both failed. The nodes of the established ring distributed messages in the following sequence: .50, .30, .10, .1.
- Experiment 5: (Figure 23) UU-AODV was used to provide the routing service as in the previous experiment. The node .50 created the ring first then .10 joined the ring through .50 and .30 through .10. Subsequently .1 tried different times to join the ring: it failed through .10 and .30 too and finally succeeded through .50. The node .40 did it through .1. The distribution of mailboxes was executed, in sequence, by: .40, .1, .10, .30, .50.
- Experiment 6: (Figure 23) it was the last experiment to be carried out and ran UU-AODV at network layer. The pastry ring was created by the node .1 then .40 and .50 joined the ring through .1. .30 followed and joined through .50 and, in the end, .10 tried twice: it failed the first time through .30 and succeeded the second time through .50. In sequence .10, .40, .1, .30 and .50 distributed their mailboxes all around the ring.

The main problem of this set of experiments, running UU-AODV as routing protocol, was represented by the high number of connection failures. This can be caused by the reactive procedure to discover a route towards a specified node. When a node tries to connect to another one, FreePastry generates a TCP connection in order to recover middleware routing table information. If the local node has no routes to the destination AODV generates a Route Request and waits for the answer. In the meanwhile the timeout of the Pastry connection can expire causing the raising of a “Connection Refused” exception on the Java socket. TCP remote connections are also periodically executed by FreePastry in order to maintain the overlay structure and the “Connection Refused” problem can cause the notification of the death remote nodes, even if they are still connected to the ring. The implementation we used of the AODV stores routes into the kernel routing table for only 15s. In this way, using Pastry, the overhead reduction introduced by reactive protocols is cancelled by periodical remote connections. UNIK-OLSR experiments didn’t suffer these problems thanks to the continuous updates of the kernel routing table.

2.2 Quantitative evaluation

Starting from the experience we gained in the experiments presented in the previous section, the CNR group, (integrated with three master students[3] performing their training stage at CNR), in the second half of July, set up a new testbed with the aim to integrate previous qualitative results with a quantitative evaluation of a small area MobileMAN. The second set of experiments was performed, in the same environment used so far, to provide a quantitative estimation of the most interesting phenomena observed. Again, we first measured the performance of OLSR and AOVD in isolation, and then we analyzed the testbed integrating FreePastry on top of the ad hoc network.

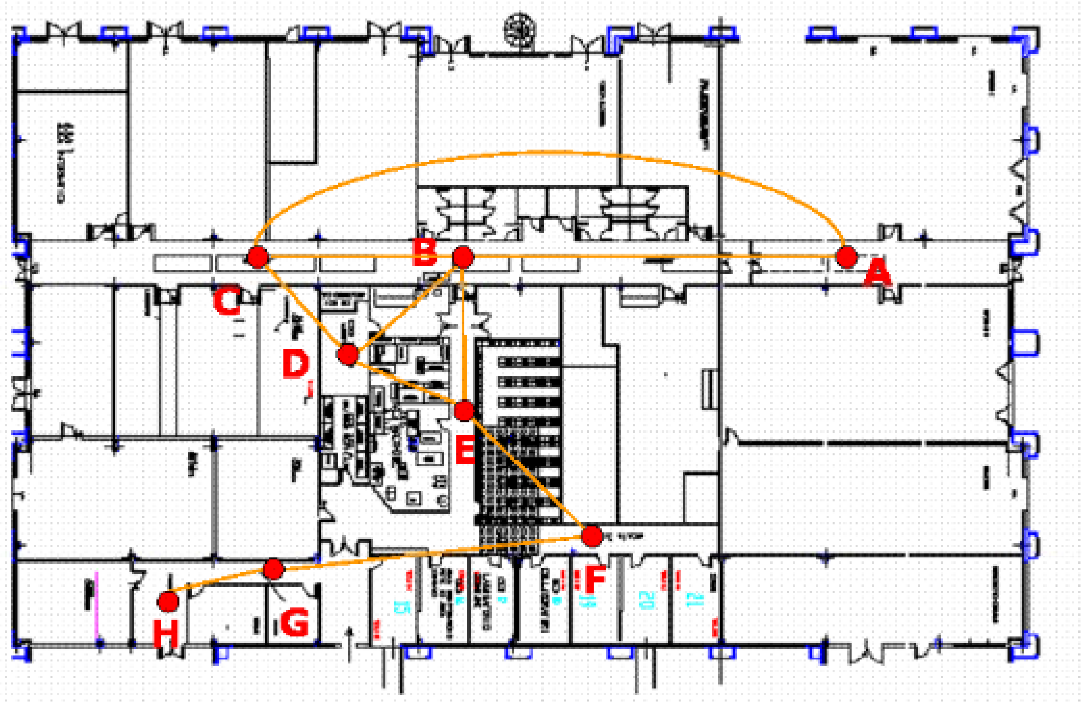


Figure 24: Network Topology

2.2.1 AODV and OLSR Performance

The experiments were made in the environment shown in Figure 2. For ease of reading, in Figure 24 we report the same scenario in which we label the MANET nodes in order to identify them in the following discussion. The figure shows the 8-node scenario on which the results reported below have been obtained. A line among a couple of nodes indicates that a link exists among them.[4]

The aim of the experiments was to compare the two routing protocols in terms of:

[3] Master in Internet Technology jointly organized and managed by IIT-CNR and department of Information Engineering of the University of Pisa.

[4] It can be noted that some links that were marked as unstable in Figure 2 are now marked with solid lines to point out that they are now stable links. The reason for this is that we removed some obstacles to signal propagation (e.g., fire doors were opened).

- overhead introduced in the network due to the routing messages;
- the delay introduced for path discovery.

To have a meaningful comparison we used the ping application to generate some user-level traffic. Without any traffic, the reactive protocol (AODV) does not perform any activity. Two set of experiments were performed depending on the way the ping operation was performed.

2.2.1.1 Experiment 1

In this set of experiments, the central node E performed a ping operation towards the other nodes of the network according to a randomly selected sequence: A,H,D,F,G,B,C. For each node in this sequence, the node E performed the ping operation for 1 minute, then it moved to ping the next node in the sequence. The same sequence was used in all the experiment of this set. We performed several experiments that produced similar results. Hereafter, we present the results obtained from one of these experiments. These results are summarized in Figure 25 and Figure 26 for OLSR and AODV, respectively. In the figures we report the amount of traffic (expressed as number of Bytes per second) forwarded by each node of the network. More precisely, this traffic includes both the routing traffic generated by the node itself and the routing traffic generated by the other network nodes and forwarded by this node.

As expected, the traffic depends on the part of the network a node is located. By looking in details to Figure 25 we can note that

- nodes B and D (the two upper curves) have to forward the highest quantity of traffic (about 1.1 Kbps);
- nodes C, E, and G (the curves around 0.8 Kbps) have an intermediate load;
- nodes A and F have a 0.4 Kbps load.
- node H is lightly loaded (its traffic, 300 Kbps in average, is about 1/4 of node B and H traffic).

This traffic partitioning shows a good correspondence with the role of the nodes inside the network. H is a leaf in the network graph and it has only one link with node G. Nodes F and A are leaves, as well but they are connected to the network via two links. The other nodes are inside the core of the network and have several neighbors thus their traffic is higher.

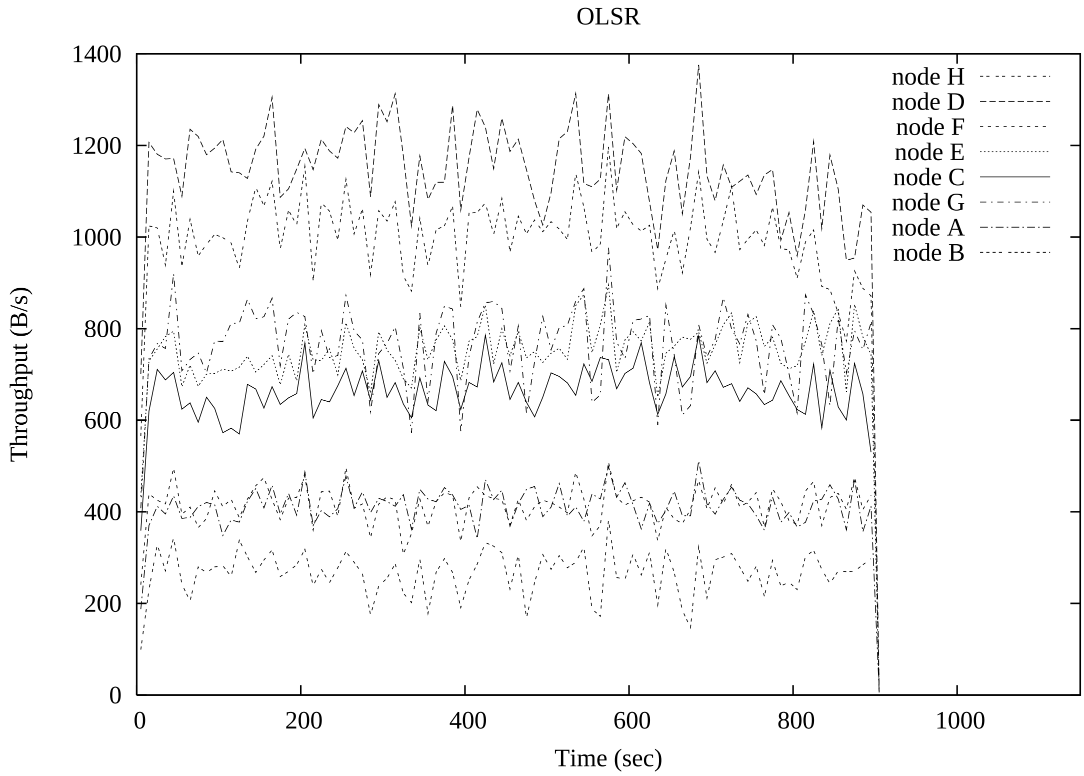


Figure 25: OLSR overhead

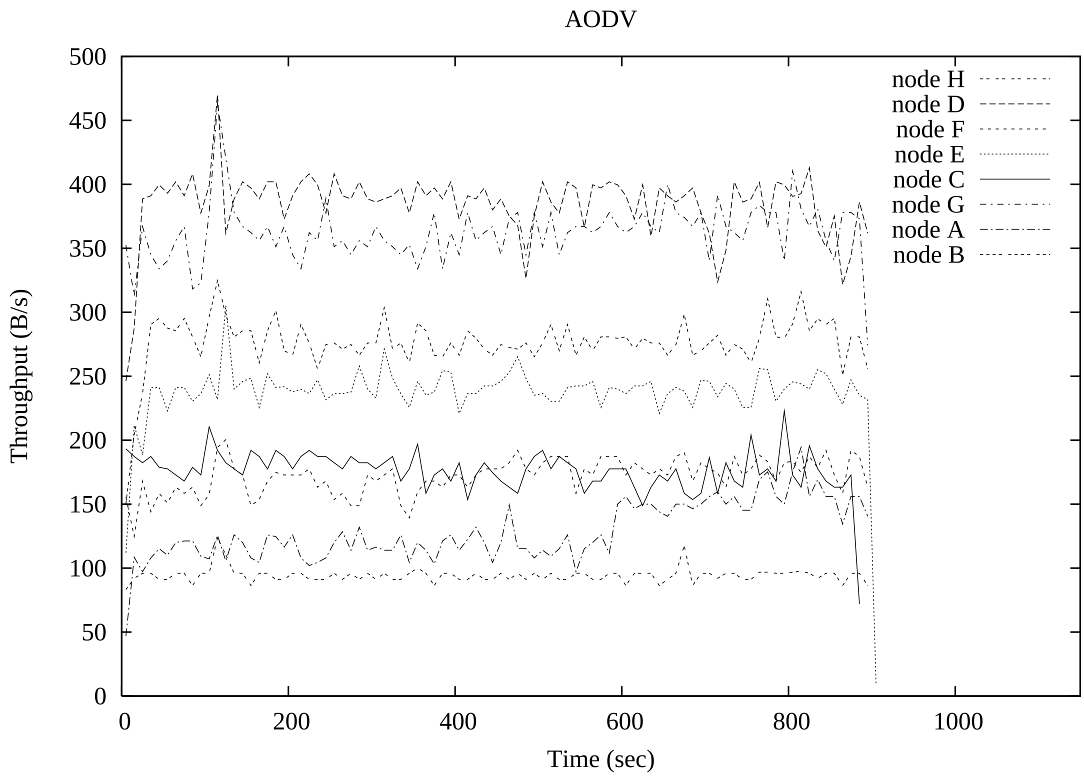


Figure 26: AODV Overhead

It is interesting to note that, in the AODV case these relationships between a node position and the traffic it observes do not always hold. For example, see Figure 26, while node H still experienced the lowest load, node B (which has the highest traffic with OLSR) had an intermediate load. This is due to the reactive nature of the AODV protocol that makes the routing traffic dependent on the traffic flows at the application level.

As expected, OLSR generated a traffic that was significantly higher than that produced by AODV. Specifically, while with AODV the traffic range was [100 – 400] bytes/sec, with OLSR was [200 – 1200] bytes/sec. However, even though in percentage this difference is big, it is important to note that in both cases the impact on the utilization of the 802.11b bandwidth is almost negligible. A node observed at most 1.2 Kbps of routing traffic.

On the other hand, delay measurements pointed out possible severe problems on QoS when using AODV. Specifically, with AODV, if the route was not yet in the nodes cache we measured, for completing a simple ping-operation between a couple of nodes at a 2-hop distance, delays in the order of 19-20 seconds. To perform the same operation, OLSR requires 1 second (or less) as routing tables are generally updated.

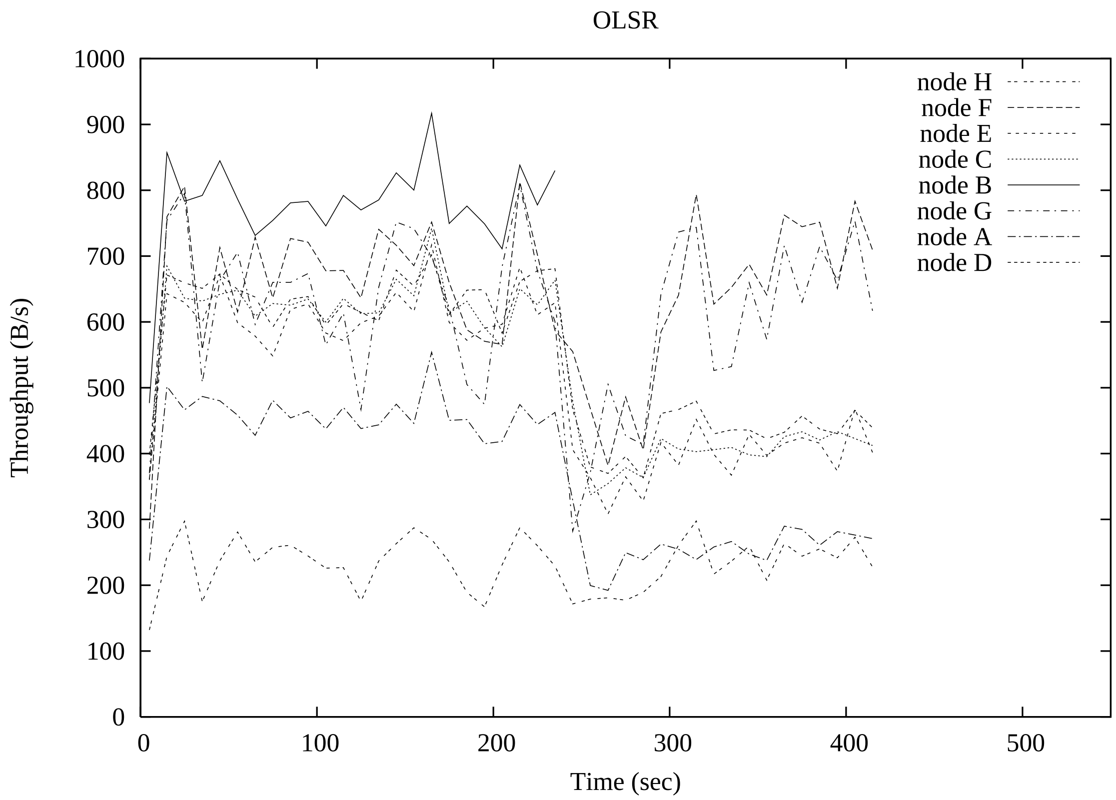


Figure 27: OLSR traffic

2.2.1.2 Experiment 2

In this set of experiments, the external node *H* continuously (for 400 seconds) performed the ping operation towards the node A. In this case the shortest path is: H-G-E-B-A. After *x* seconds (*x* equal to 250 and 180 seconds in OLSR and AOVD experiments, respectively) from the beginning of the experiment, the node B disconnects from the network and this cause a change in the route from H to A that now is crossing D and C, instead of B.

Again, we performed several experiments. Hereafter we report, as an example, the results from one of them. The results are summarized in Figures 27 and 28 for OLSR and AODV, respectively.

As far as OLSR, we can note a load distribution similar to that observed in the Experiment 1, e.g., node B and node H experienced the highest and lowest load, respectively. Differences (in terms of amount of traffic) between Experiment 1 and 2 can be explained by the different application-level traffic flows. After node B shut down, there was a transient phase during which the node-traffic decreased (due to some missing routes), after which a new steady state was achieved. In this new steady state, we observed a significant decrease of the traffic in the nodes that were connected with node B (A, E, D, C).

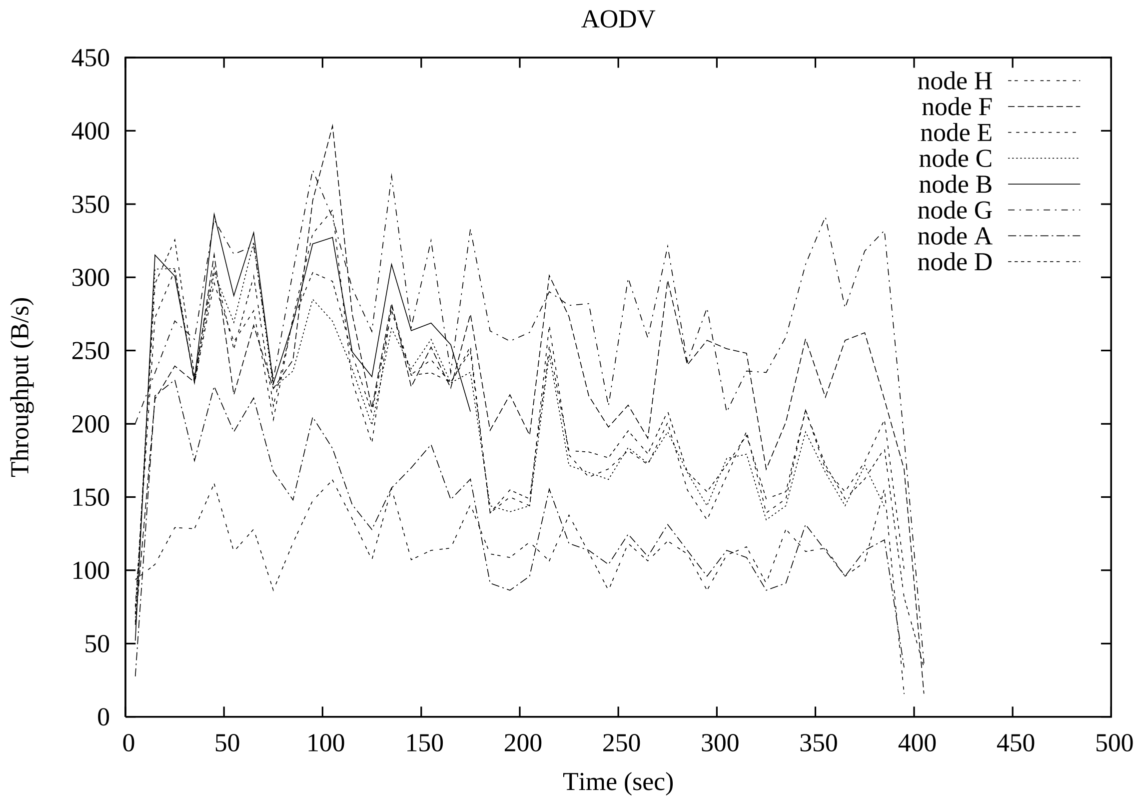


Figure 28: AODV Traffic

Less marked differences, before and after B shut down, can be observed by looking at AODV results. After the transient state following B shut down, the active nodes almost observed the same load experienced before the shut down event.

As far as the delay is concerned, we got results similar to those observed in Experiment 1: very large delays with AODV (about 20 seconds) when the path is not already in the cache. On the other hand, OLSR introduced very small delays. The only exception was after B shut down which caused a routing-table reconfiguration. In this case the ping operation performed while OLSR was updating the routing tables experienced a delay of about 8 seconds to be completed.

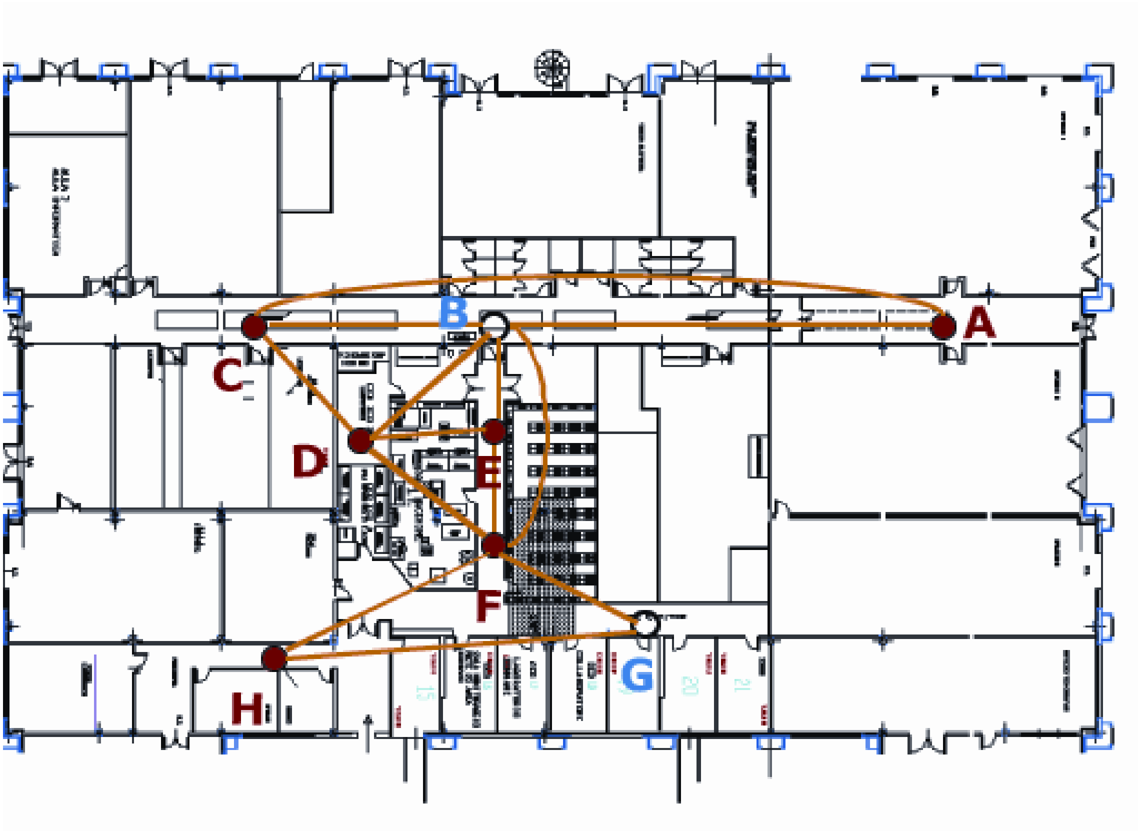


Figure 29: Network Topology for FreePastry Experiments

2.2.2 Performance of FreePastry on Ad Hoc Networks

This set of experiments were made by adopting a network topology (see Figure 29) which slightly differs from that used for the analysis of the routing algorithms. We decided this modifications as experiments shown in Section 2.1 indicated that central nodes in the network tends to become saturated. To avoid this, we moved one node towards the centre of the network. In this way we increased the redundancy in this area. Among the eight nodes of the ad hoc network, 6 provided the Pastry service, while nodes B and G (the empty circles in the figure) were only involved in routing and forwarding operations.

As far as the Pastry operations, node E was the first to start and then it initialized the overlay ring related to the distributed messaging service. Then the following actions were performed in sequence: F joined the ring by connecting to E, D connected to E, C connected to D, A to C, and last H connected to C. At this point all the nodes providing the Pastry services were connected to the overlay ring.

In the next figures we investigate the costs (in terms of traffic) required to maintain the Pastry overlay ring on top of our ad hoc network. It is worth remember that Pastry generates management traffic (to maintain the overlay ring) both when a node join the network (and hence it needs to acquire the information about the other nodes belonging to the same service), and periodically to check the status of the overlay ring. The latter operation performed by each node is implemented by opening TCP connections from that node towards all the other nodes belonging to the ring.

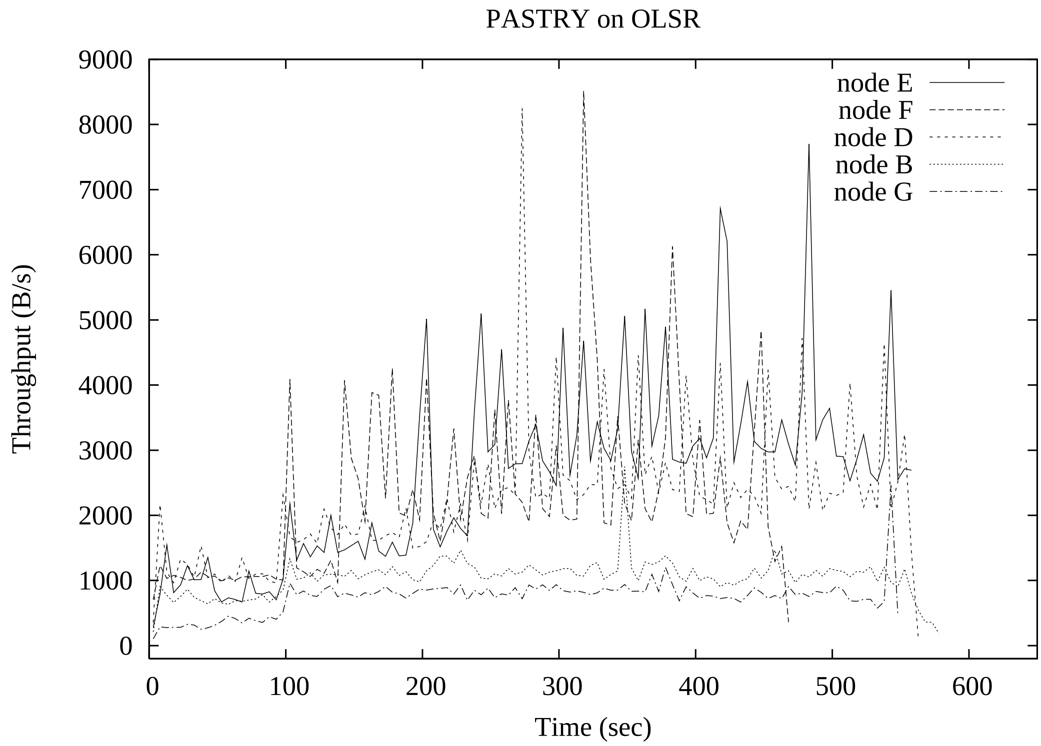


Figure 30: Traffic observed by each node using OLSR

Figures 30 and 31 present the amount of traffic forwarded (including their own generated traffic) by the network nodes using OLSR and AODV, respectively. As expected, node B and G that only participate to the routing and forwarding operations generate the same amount of traffic observed in Section 2.2.1 in which the traffic was mainly due to routing operations. On the other hand, the nodes belonging to the ring Pastry periodically experience a burst of traffic produced by the overlay-ring maintenance operations. To better investigate this aspect, in the following graphs we focus on a single node and analyze the type of traffic it generates. Specifically, we identify four traffic classes:

- the traffic due to the ARP-protocol operations;
- the routing traffic;
- the UDP traffic generated by Pastry ring maintenance operations, e.g., ping operations performed at Pastry level by a node to test the other nodes in the ring; and
- the TCP traffic used by the nodes belonging to the Pastry ring to exchange their routing tables and other ring-related information.

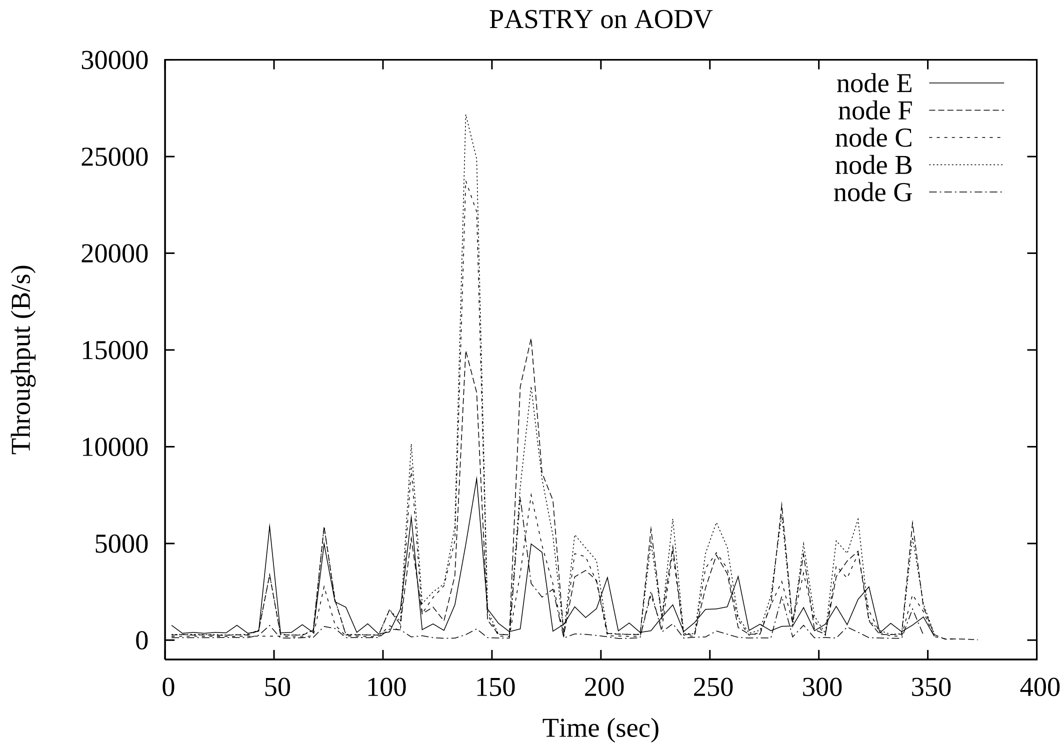


Figure 31: Traffic observed by each node using AODV

In the figures below we report the total traffic generated by a node. More precisely, while the curve ARP indicates only the total traffic produced by ARP protocol, the curve OLSR indicates the sum of ARP + OLSR traffic. The curve UDP denotes the sum of ARP+OLSR+UDP traffic, and the TCP curve is the node total traffic (i.e., ARP+OLSR+UDP+TCP).

Figures 32 and 33 clearly pointed out that the traffic pecks observed in the traffic are due to the overlay ring management (i.e., TCP and UDP traffic). Indeed, from these figures we can observe that the ARP traffic is almost negligible, while routing traffic is quite regular and provides links utilization levels similar to those observed in Section 2.2.1 without FreePastry. On the other hand, the traffic burstiness is almost only due to TPC/UDP traffic required by the overlay ring maintenance operations.[5] These observations are confirmed by observing other network nodes. In particular, by observing node A, when using AODV (see Figure 34), it clearly appears that Pastry operations may produce big traffic pecks, mainly during ring initializations.

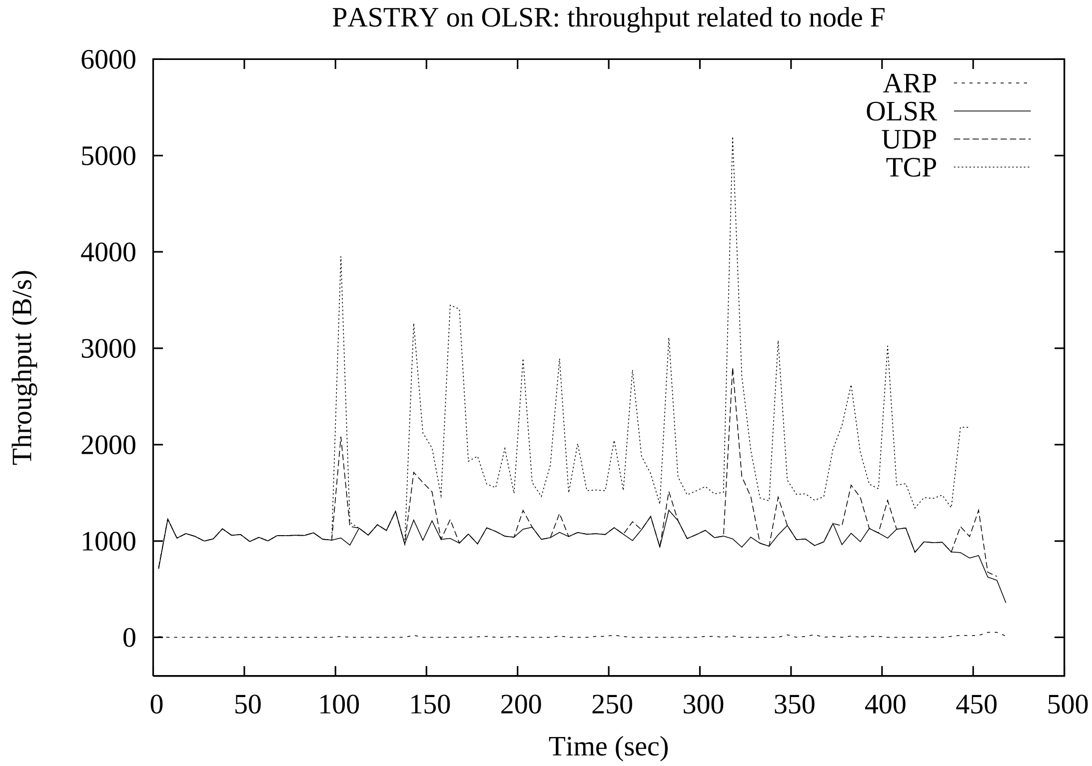


Figure 32: Node F traffic on an OLSR network

5 The flat behavior in the first 100 seconds of the OLSR figure was due to our choice to start the ring Pastry only after a 100 seconds delay in order to have the OLSR routing tables initialized. We did not add any delay in the AODV network because when no upper layer traffic is generated AODV does not add any information in the routing tables.

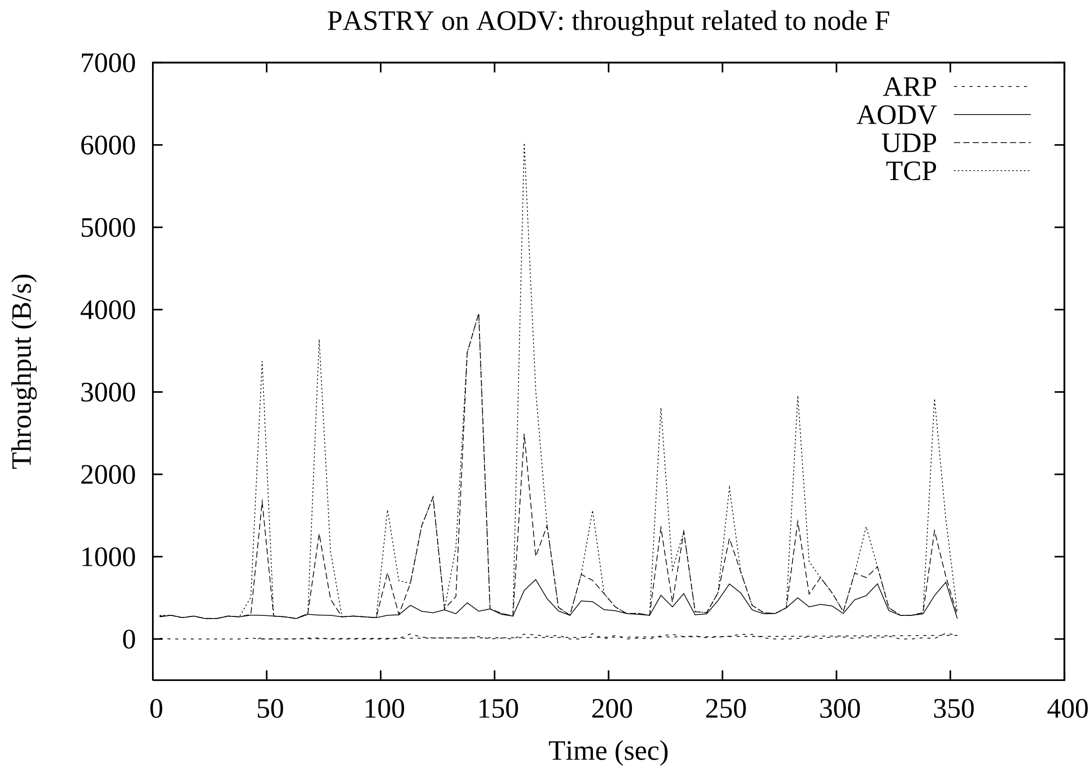


Figure 33: Node F traffic on an AODV network

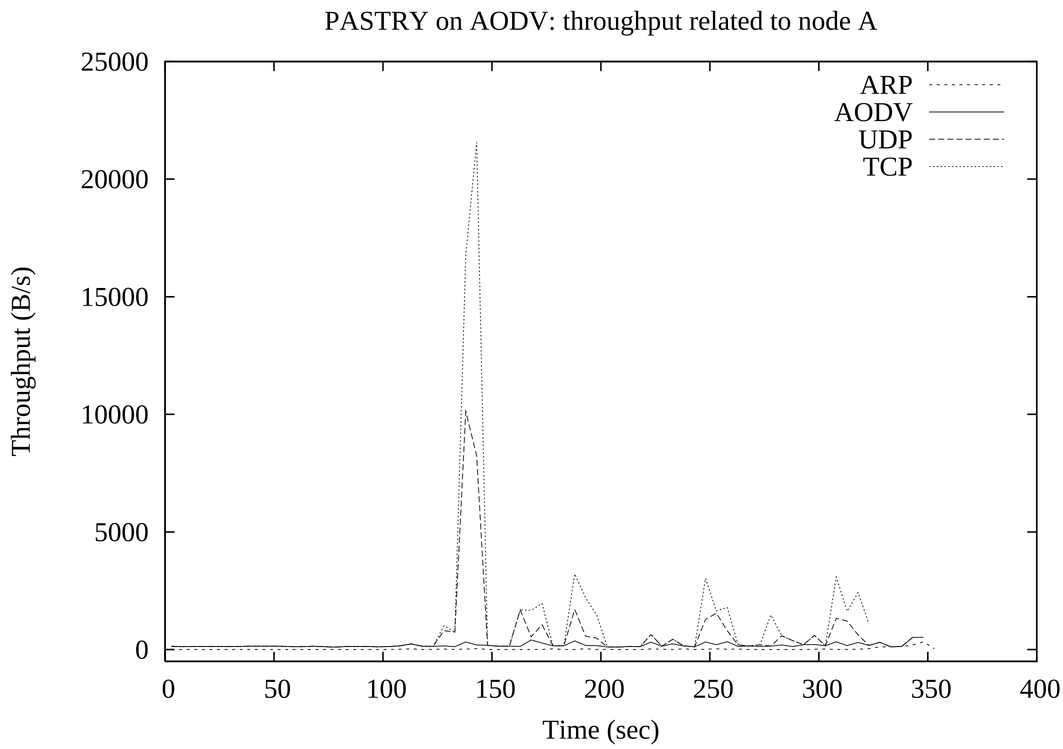


Figure 34: Node A traffic profile (AODV case)

3. INTERACTION WITH USERS

The reaction of the users to this new networking philosophy (multi-hop ad hoc networking), which impacts the application level as well, is an important measurement tool for tuning and modifying the ad hoc networking paradigm. At the same time, this activity is essential for discovering the ad hoc networking potentiality in terms of users' needs and requirements, as well as social and economic input.

In the current phase of MobileMAN it is not possible to have a direct users' access to the MobileMAN technology but we felt important to construct tools and methods to facilitate the interaction of users with the MobileMAN world. To this end, some initiatives have been taken:

- Tools for users' interaction. The main initiative in this area is carried out by SUPSI-DSAS that implemented a website to facilitate the interaction with users (http://mobileman.projects.supsi.ch). The website structure was divided into two parts: Static part, and interactive WIKI section. The first part has the objective to inform about the technology and the project. The WIKI part is a collection of prepared pages (minimal structure) editable by any user with the goal of gaining information by them. Users had also the possibility to add pages wherever they wanted. The software used was phpWiki.

- Direct interactions with "expert users" and dissemination material. This activity is currently carried out by CNR though a set of initiatives (presentations to class of users, users' mailing list, a web site) for collecting feedbacks/comments and suggestions from expert people.

CNR activities complement those performed by SUPSI. Indeed the aim of CNR is to interact with expert people to obtain feedbacks on the technical solutions, while SUPSI has a more general view to interaction with users. Not-expert users being the main target of SUPSI activities.

3.1 User-oriented Website

One of the objectives of the MobileMAN project is the validation of the paradigm from a social point of view through an active participation of potential end-users. To this aim we have developed a user-oriented project website that will serve as a platform for a dialogue between system developers and a community of potential end-users, and as a tool for gaining information about attitudes and opinions towards the MobileMAN technology and its applications. In the previous months, we focused on the design and implementation of such website and the conduction of the pilot test phase. The tool to allow this information gathering is phpWiki, a software that allows editing of web pages by users and therefore shared and collective content-creation. In this deliverable we focus on the description of the results of the pilot tests and on the forthcoming activities.

3.1.1 End-users participation in creative scenario building and applications development through Wiki

3.1.1.1 Objectives and strategies

The objective of the website is first of all to give information to non-expert potential end-users about the project and the technology being developed within it. This information is basic and leaves out technicalities and details. Another main goal of the website is to assess end-users' responses to the MobileMAN paradigm. This goal was to be achieved through the Wiki section of the website, which proposes a dynamic interaction with potential end-users. More precisely, we aimed at involving users in the development of use and application scenarios for MobileMAN. This section briefly presents the activities carried out within his framework by SUPSI-DSAS.

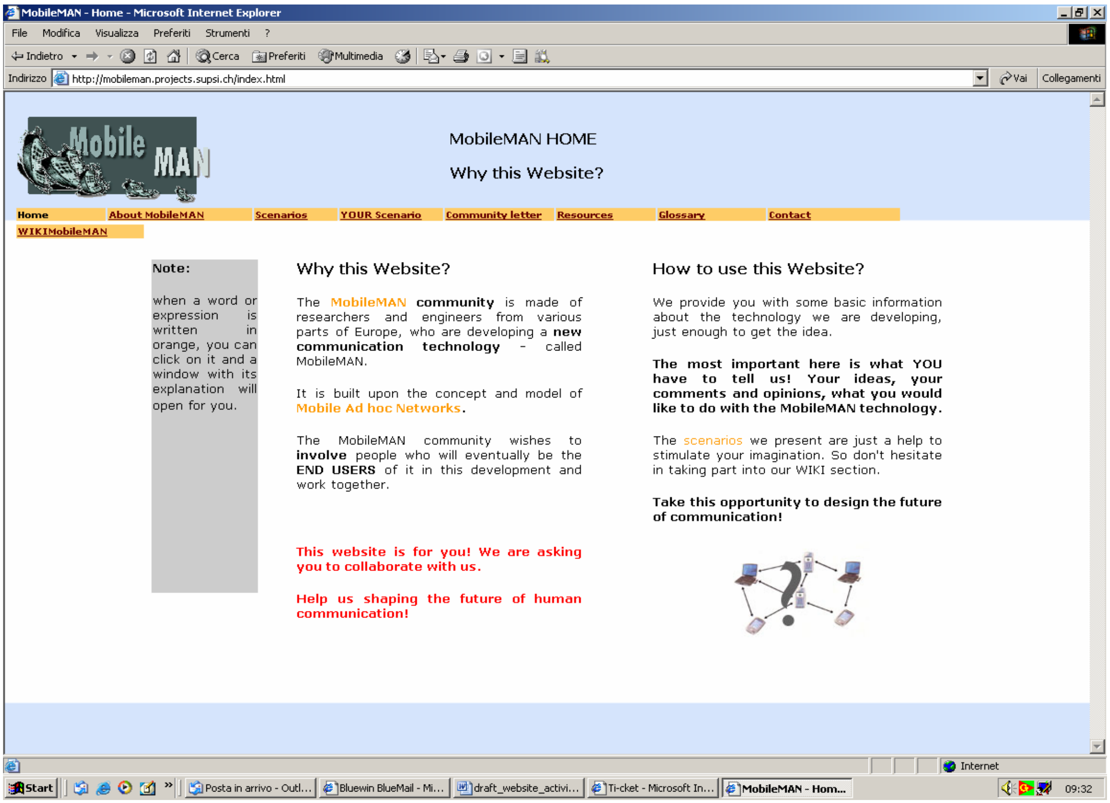


Figure 35: Homepage of the interactive web site (static part)

3.1.1.2 The Wiki website

3.1.1.2.1 Description and Objectives

The website (http://mobileman.projects.supsi.ch) consists of two distinct parts with two different objectives: one static (descriptive) and one editable (Wiki section). The **static part** entails a description of the project and the developed technology. The explanations do not go into details; aim of this part is to give enough information to allow users to be able to understand the concept and provide information that will be useful for us. A glossary with the main necessary terms was developed as well as a resources page with links to websites with related information (for users interested in knowing more). The **Wiki section** consists of a collection of pages that are editable by anyone visiting the site. Users can create new pages and link them in the desired way, they can add content to existing text or can modify parts which they consider to be wrong. By intervening in the existing structure, users can collectively create and shape content. Wiki pages are developed using specific software; it provides the main structure that can be adapted (titles, page names, content). The consequence is that some parts of the page structure are fixed and cannot be modified (positions of buttons e.g.).

3.1.1.2.2 Sample pages of the Website

This section illustrates through some figures the content of the Website. Figure 35 presents the static part of the homepage of the interactive website. To ensure an easy structure, a navigation bar was provided at the top of the pages. The most important text was put in red in order to catch the attention of users. This was the main goal and the reason of the website. This layout was used throughout the entire static part of the website.

Figure 36 illustrates the shopping mall scenario, which was provided as an example of what users were invited to do with the Wiki section, namely to develop similar scenarios of use for MobileMAN. This scenario was a short version of the more detailed shopping mall scenario developed by Cambridge Laboratory of Computers, University of Cambridge.

Figure 37 shows the initial page of the Wiki section. The page structure is provided by the Wiki software and is fixed. The content and links between pages were adapted to the purposes of our own website.

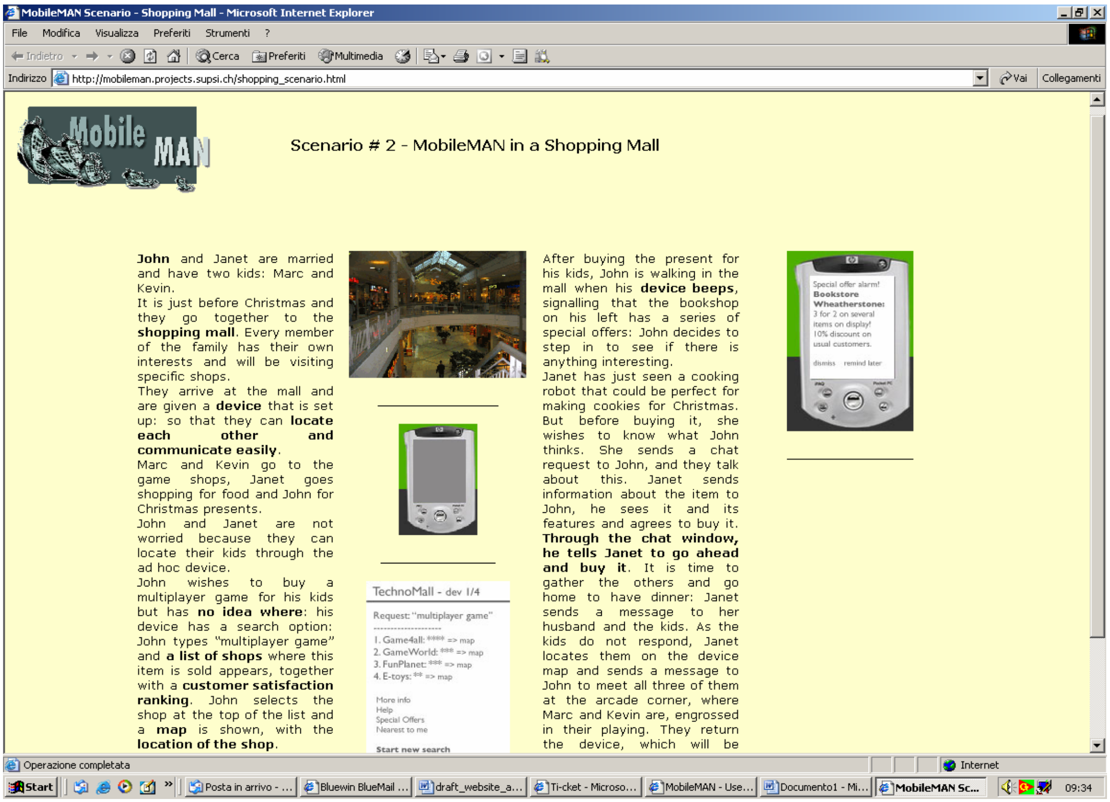


Figure 36: Example of Scenario: Shopping Mall Scenario

3.1.1.2.3 Usability tests

Usability tests were performed prior to the publication of the website. These had the objective of revealing possible usability issues and checking if the provided information was understandable.

The usability tests in particular focused on the following aspects:

- clarity of the website objectives;
- clarity and completeness of presented information;
- usability of the Wiki section for first users;
- clarity of the website structure (in particular for the Wiki section).

Tests were performed with 4 individuals during the second week of May 2004. They were given the material with the description of tasks to perform and some questions to answer within a timeframe of 30 minutes. After the test they were requested to compile one questionnaire with open questions and one with a number of statements to agree with on a 5-elements Likert scale[6]

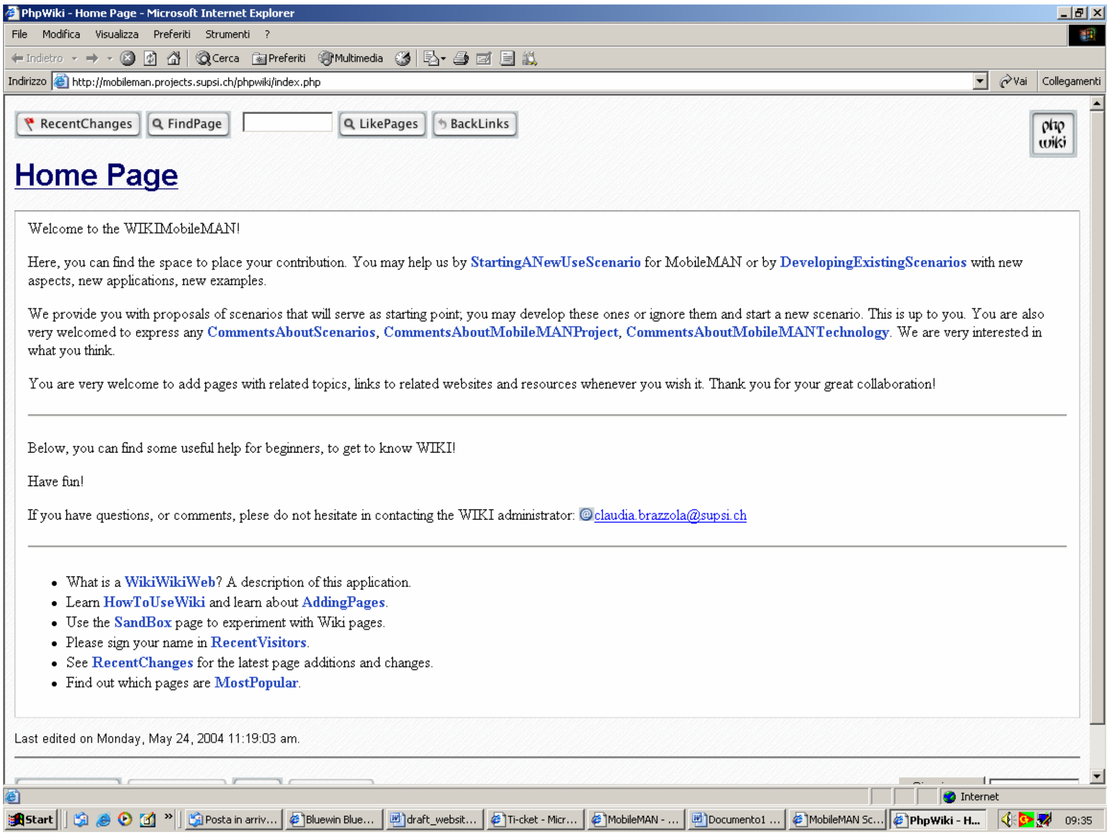


Figure 37: Initial page of the Wiki section

No time restrictions were imposed in relation to the compilation of these two questionnaires. At the moment of the tests, the users saw the website for the first time and none of them had previous experience with Wiki websites. All four testers had a fairly good level of computer literacy.

As illustrated in Table 1, the results of the usability tests with regard to the static pages clearly indicate that some changes had to be made in order to underline the goal of the website, as well as to the glossary (extension and linking to texts). Other aspects, such as basic concepts (meaning of ad hoc networks e.g.) and where to find more information were considered successful.

[6] A Likert scale is a rating scale often used to assess users' subjective reactions to a system, involving the assertion of a statement about use followed by a judgement of strong to moderate agreement or disagreement.

Table 1: Evaluation of the content of the static pages (information).

	Tester 1	Tester 2	Tester 3	Tester 4
Questions on content				
a) Goal of website	**Wrong/Mis**	**Wrong/Mis**	Correct	Correct
b) Where to find more info?	Correct	**Wrong/Mis**	Correct	Correct
c) Difference scenario/YOUR scenario	Correct	Correct	Correct	Correct
d) What does MobileMAN stand for?	Correct	Correct	Correct	**Wrong/Mis**
e) What does Ad hoc network mean?	Correct	Correct	Correct	**Wrong/Mis**
f) Glossary extended or ok?	**Links***	**Links***	**Extended**	**Extended**

* links means that the tester suggested an inline glossary (opened on click over the term).

The evaluation of the interactive Wiki tool indicates, however, that usability issues prevented the users from completing some of the tasks and, more importantly, that users did not experience the exercise to be particularly pleasant. Quite to the contrary it was partly perceived as frustrating. As shown in table 2, the frequency of wrong or missing completion of tasks is rather high.

Table 2: Evaluation of the interactive Wiki tool (tasks and questions).

	Tester 1	Tester 2	Tester 3	Tester 4
Test of Wiki tool				
a) Task 1	Correct	Correct	**Wrong/Mis**	**Wrong/Mis**
b) Task 2	**Wrong/Mis**	**Wrong/Mis**	**Wrong/Mis**	**Wrong/Mis**
c) Task 3	Correct	Correct	Correct	**Wrong/Mis**
d) Task 4	Correct	Correct	Correct	**Wrong/Mis**
e) Task 5	Correct	**Wrong/Mis**	**Wrong/Mis**	Correct
f) Task 6	Correct	Correct	Correct	**Wrong/Mis**
g) Task 7	Correct	Correct	Correct	Correct

Note: Mis = Missing; task was not completed.

The Wiki evaluation focused on 7 tasks:

- ✓ Task 1 invited the visitor to edit a predefined page of the Wiki section. This task was carried out successfully by 2 individuals, whereas 2 did not complete it. Being their very

---

Usually, a 5 points scale is used, where the points are strongly agree, partially agree, neither agree nor disagree, partially disagree, strongly disagree.

first use of the Wiki features, this result could be expected and explained with the difficulty related with the first impact of an unknown object.

- ✓ Task 2 refers to the registration as new user. This task was not carried out by anyone of the 4 testers. Only later it became clear that this was due to a problem (bug) of the software itself.

- ✓ Task 3 invited the user to make a list of the most recent changes of the Wiki pages. 3 testers out of 4 were able to complete this task correctly.

- ✓ Task 4 required the editing of another predefined page. Again, 3 testers out of 4 were able to do it correctly.

- ✓ Task 5 was about adding a new page to the Wiki section. 2 out of 4 testers were successful. This task was a little more complex than the previous ones, because it required the understanding of using a "WikiWord", that is, joined words with the first letters written in capitals so that the system would recognize it as hyperlink and create a linked page with that name. For the development of the Wiki it is important that users are able to add new pages.

- ✓ Task 6 was a question about the usage of the sandbox. 3 out of 4 testers understood correctly that it is a dedicated area to experiment with Wiki that can easily be cleaned from its content.

- ✓ Task 7 was a question about what is a back link. All 4 testers understood it correctly.

Although the successful task completion was not too low (17/28), the overall qualitative evaluation of the experience showed that users considered frustrating their inability to do the tasks with the Wiki tool. Its use was considered too difficult, not user-friendly and confusing. Navigation within the Wiki tool was seen as non-intuitive. The use of English was perceived as a difficulty. With regard to comprehensiveness of the content opinions diverged, with one user stating that the language was too scientific and complicated, and another the opposite.

The evaluation indicated that major difficulties only turned up in relation to the Wiki tool. Being a package only partially customizable, the problems could only partly be solved. However, the following useful modifications to the website were made according to the suggestions given by the testers:

a) the glossary was expanded to cover all specific words and expressions;

b) words in the text were converted in hyperlinks that opened a small window with an online glossary (the same as the dedicated page, only in a smaller window, more easily accessible);

c) on the page that explained more in detail the creative part with the use of the Wiki tool, a "quick guide" with the more basic uses was provided (a sort of help before use);

d) direct access to the Wiki section was provided with a link present on any website page;

e) the objective of the website was written in red colour to draw attention to it.

As illustrated in table 3, the statement agreement (Likert) tool to evaluate the test experience showed that feelings like being lost and inability to understand the Wiki tool were experienced at least to some degree by all of the testers.

Table 3: Statement agreement on a 5-elements scale (Likert) about the test experience.

Issue	Disagree			Agree	
	1 (totally)	2 (partially)	3 (neither nor)	4 (partially)	5 (totally)
I always feel in control using it.	A	D C	B		
It is very easy to understand straightaway.	A	B C	D		
It was always clear what would happen when I clicked on something.		A B C D			
The information was excellent quality, concise, clear and understandable.*			C	A B	D
The font sizes are right.*				A C	D B

A B C D identify the four testers;
* These two statements refer to the static part of the website.

Table 3 reveals that users felt confused when using the Wiki section, and did not understand at once how to use it. Navigation was also considered not optimal. The answers regarding the quality of information were spread – main complaint was about the incomplete glossary, which was enriched after the tests. Moreover, the glossary was also copied into a pop up window accessible directly from the terms in the text.

To conclude, the usability tests performed showed that the objectives of the **static part** of the website were fairly reached. The navigation structure was perceived as clear and the fact that no superfluous information was provided as good. The suggestion to improve the glossary was immediately followed.

The **Wiki section** tests results required more concern, since this interactive part of the website was the source of difficulties. The quick guide for first users was provided on the static pages that lead to the Wiki section with the idea that it would help overcoming the confusion experienced by first users. With these modifications to the website it was published for the pilot test phase with a community of students based on the assumption that using the Wiki could be compared to some type of entertaining game; from this point of view, students using it would gain benefits in enjoying its use.

3.1.1.3 The Pilot Test

The website was put online at the beginning of June 2004 with the informal request directed to some project partners to encourage their students to visit the website and provide some contribution through the Wiki section. The fact that the request was passed from project partners to some students in informal way had also the goal of testing what level of collaboration would have been reached without any form of incentives. However, the level of voluntary participation to this pilot test turned out to be rather limited.

3.1.1.4 Lessons Learnt

The period of pilot testing of the website and reflection on the results allows us to make some interesting considerations. With regard to voluntary collaboration aspect, the lesson learnt is that collaboration is not to be taken for granted, but is a complex concept that we are currently studying in more details. With regard to the pilot test of the Wiki website, the limited collaboration level was due to a number of factors.

3.1.1.4.1 Difficulties in using Wiki

First of all, the difficulty of **using the Wiki** for first comers (difficulty shown by the usability tests performed before publishing the website) was identified as an important cause. If the collaboration does not involve some sort of reward it should at least not require too great an effort. The fact that it is confusing for first users prevented them from collaboration since the effort to learn how to apply it was too great.

3.1.1.4.2 The role of incentives

Students were asked to give their contribution without gaining any benefits in exchange. The consideration of the exercise as a sort of entertaining game did not prove to be correct. Unless the students consider the project and the technology very worth of collaboration, there is no incentive that justifies the effort. This resulted in a limited willingness to cooperate. Another factor that may have had a negative impact on the pilot test was its timing. Indeed, June is the last month of the academic year and students are occupied with other tasks, such as examinations and compilation of academic papers. As said, the role of incentives – and the type of incentives – has proved to be very interesting to study in more depth.

3.1.1.4.3 The language factor

The website was in English. Being a foreign language for many of the involved students, it worked as an obstacle to collaboration. Being required to read and write in a language that was not their own constituted a further obstacle to collaboration.

3.1.1.5 Planned activities

To ensure the collaboration by the students, during the Project Meeting held in Helsinki at the beginning of June, it was suggested that our activity was to be placed within two courses held at HUT during the next academic year. This would mean that the activity would be carried out later than first planned. However, it seemed a good solution to gain collaboration from a community of students by formally place the activity into a curriculum course and reward the students for their collaboration in credits or similar ways. To do this, the request was to be able to track the users' movements and contributions. PhpWiki presented the inability to guarantee user identification and login before being able to make any change to the pages (editing). This gap would not allow us to provide a secure way to identify what user made what changes and consequently to allow an evaluation of the contribution, as incentive to the students to participate. For these reasons, the strategy to carry out the planned scenario building activity has been reviewed.

Due to the difficulties presented above, a modified strategy had to be developed in order to reach the intended objectives and goals of this activity. The following modifications were thus integrated:

1) formal integration of the creative scenario building activity into courses activities, with evaluation of the tasks and contributions of the students;
2) involvement of different types of classes (IT and non-IT);
3) use of offline and online instruments for the various parts of the activity;
4) use of a more robust electronic platform (moodle) to guarantee a more reliable system;
5) compilation of a questionnaire, as part of the activity (required to get the rewarding credits).

The fact that the activity will be part of a defined curriculum course and therefore will be evaluated will ensure collaboration in this activity by the students.

Another important aspect is that the groups could **work in their own languages**. We plan to involve at least 2 classes of students at HUT. We will also consider the opportunity to involve other classes from other institutions (e.g. University of Lugano, University of Zurich, University of Cambridge, University of Applied Sciences of Lugano).

The activity of creative scenario group building will consist of a series of tasks to be carried out some in small groups, some others individually. Students will be divided into small groups and asked to produce (by collaborating) a scenario for MobileMAN. This activity will be completed by half of the groups using Wiki (online) and the other half using paper (offline). After that, students will be asked to choose three scenarios from all the produced scenarios to comment and evaluate.

Table 4: Combination of type of students groups and type of medium used for collective scenario building.

		MEDIUM USED FOR SCENARIO BUILDING	
		Online (Wiki)	**Offline (paper)**
GROUP TYPOLOGY	**IT oriented**	IT groups working in online modality	IT groups working in offline modality
	Non-IT oriented	Non-IT groups working in online modality	Non-IT groups working in offline modality

By combining the two media used (paper and Wiki) and the two typologies of groups, we aim at understanding whether there are differences due to the medium used or not. We finally opted for another version of Wiki that has proved to be more reliable and robust and that is part of a more complete course management system (Moodle). This version of Wiki should also allow for compulsory login of users before they make any changes to the content.

After this pilot phase of collaborative scenario building for MobileMAN with students, we plan to disseminate the knowledge of the website among key people interested in wireless technology development (e.g. those actively involved in related discussion forums) and invite them to virtual discussions about MobileMAN.

3.2 Dissemination Activities

In order to better understand the users' expectations and acceptance of the MobileMAN technology, CNR lunched a set of activities directed to present the project solutions and potentialities to users with a good expertise in information and communication technologies.
Currently, two initiatives of this type have already been performed:

- Interactions with university students
- Interactions with Human-Computer Interaction (HCI) experts

3.2.1 Interaction with University Students

We started this activity by organizing a presentation to university students with a very high skill in information technology. Specifically, on May 13th, we arranged a two-hour presentation in the framework of the course *Applications for Mobile Computers* for the last-year students in Computer Engineering (5-year degree) at the University of Pisa (Engineering Faculty). We decided to use a top down approach (i.e., starting with skilled ICT people) to minimize the communication risks (i.e., using a language and concepts that is not too far from the audience). In this way we intended to learn how to present the concepts of the MobileMAN paradigm. More precisely, we tried to identify which concepts/ideas are more difficult to understand also for a skilled audience, and hence need to be better emphasized in the successive presentations, when addressing people with a lower degree of ICT knowledge.
To present MobileMAN we prepared a set of four presentations (now available on the project web site) in Italian language for making interactions with student easier:

- M. Conti "MobileMAN: Mobile Metropolitan Ad Hoc Network"
- E. Borgia "Routing su reti Ad Hoc"
- F. Delmastro "CROSS-ROAD: CROSS-layer Ring Overlay for Ad Hoc Networks"
- P. Andronico "Design Mobile Communicators"

In addition, to introduce the ad hoc networking concept in a more concrete way, after the presentations we used a very simple demo of a (single-hop) ad hoc network with a chatting application on top. Specifically, we used as a testing application memeChat (http://www.csl.sony.co.jp/project/memeChat) which is an application developed by Sony to start exploring the potentialities of very simple single-hop ad hoc networks.

To collect users' feedbacks, we prepared a questionnaire. The questionnaire was distributed during the presentations. The contents of the questionnaire and preliminary feedbacks available from the users are discussed in the next subsection. Just after the presentation at the University of Pisa, a mailing list (mobileman-users@iit.cnr.it) was set up to established a link with those university students (the majority) that expressed they interest to continue to cooperate with us.[7] Currently, about 20 students are part of this mailing list (most of them being participants to the presentation we gave at University of Pisa). We expect to enlarge significantly this number by organizing other presentations to university students. One presentation has already been planned to the students of the *Internet Technology* Master jointly organized by IIT-CNR and the Department of Information Engineering of University of Pisa. This presentation has been explicitly requested by the students but it has not yet been organized due to the examination period. We expect to organize it at the end of the summer-vacation period.

3.2.1.1 Questionnaire

The questionnaire we prepared is reported in the Appendix (see Figures 42 and 43). Below we report some preliminary results obtained from the questionnaires distributed at University of Pisa on May 13th (the statistics are based on the 16 students that returned the questionnaire). Each figure is labelled with the question the statistics refer to.

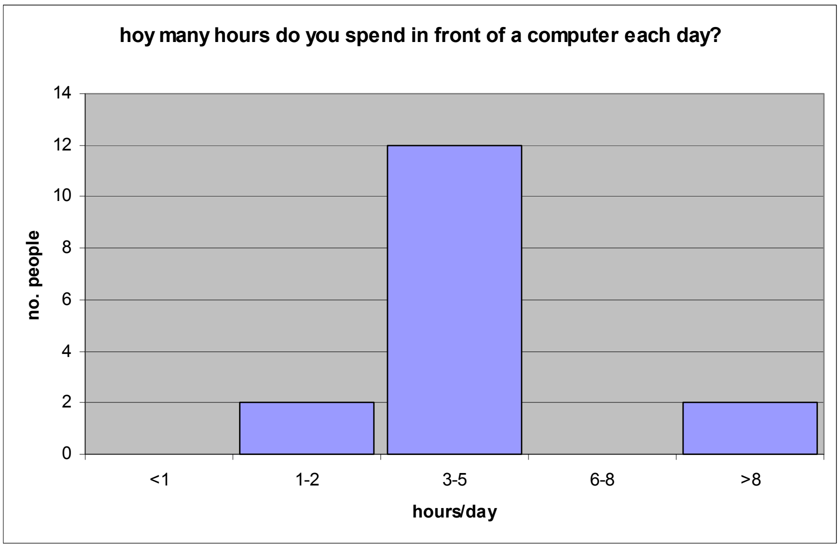


Figure 38: Users ICT profile

As expected, these students were a very technology-oriented community. As shown in Figure 38, they typically spend 3-5 hours per day using a pc/laptop for a wide range of activities: not only for study, but also for fun and work (see Figure 39).

[7] We decided to have a closed mailing list (in this way we avoid receiving spamming messages). Users have to send a request message to the mailing list manager to be become part of the mailing-list community.

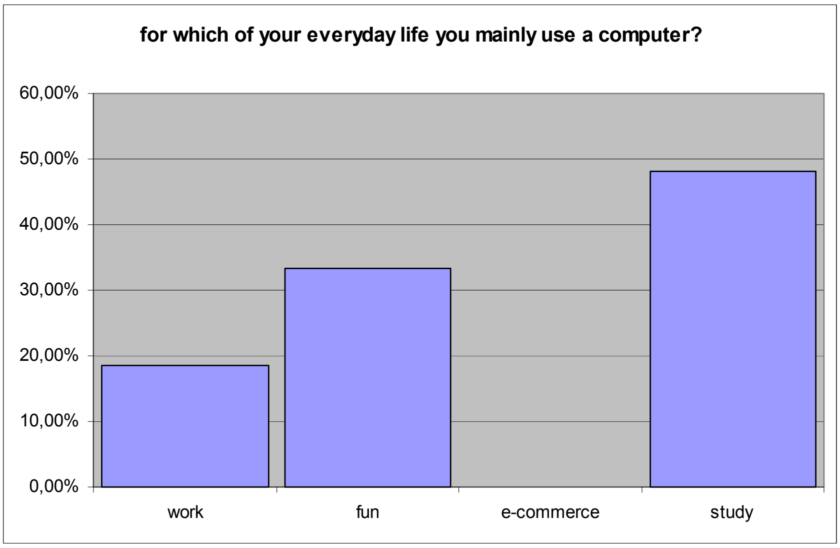


Figure 39: Users profile

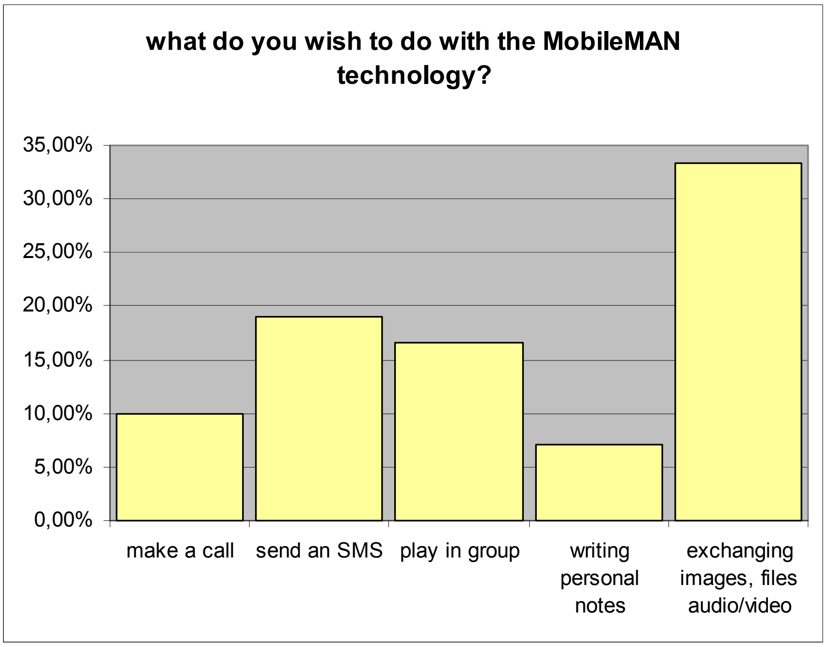


Figure 40: Expected MobileMAN usage

A preliminary, but interesting, indication of which kind of usage the students can envisage for a technology like MobileMAN is summarized in Figure 40. The graph show that students expect to use the MobileMAN technology for collaborative group activities (documents exchanging, messaging or distributed plays). It is interesting to note that these applications are well supported by a middleware platform, like Pastry, that has been selected for the MobileMAN testbed. In addition, messaging and files exchange are the first-phase MobileMAN test applications. At the end of the presentation, the students expressed their interest to learn more about the MobileMAN and for this reason we established the MobileMAN users' mailing list. In addition, they declared their availability to further collaborate with us trying to identify interesting (from their perspective) applications for the MobileMAN platform. Unfortunately, the successive interactions with this group of people was less satisfactory as only two-three of them provided further inputs about applications they envisage suitable and interesting for MobileMAN. The main reason for this was that, just after the presentation, the exams period started and students concentrated all their effort/time on this.

The current conclusion from this experience is that feedbacks/ideas from students with ICT expertise are an important source to improve the MobileMAN framework from the technical standpoint. The major problem to address is to identify suitable ways to involve students in the MobileMAN framework. When these activities are part of the curricula, students are very happy to contribute; no other incentive is necessary in this case to attract their interest. Otherwise, their study load is quite heavy and they have no enough time to dedicate to our project, to be useful. For this reason, we are currently discussing with professors at University of Pisa to have some MobileMAN related activities as a part of the training activities inside computer engineering courses.

3.2.2 Interaction with HCI experts

MobileMAN project was presented at a workshop organized in the framework of the International Conference on Computer Human Interaction in Wien, April 2004 (CHI2004, http://www.chi2004.org/). The theme of the CHI2004 edition was "Connect" to stress the strong interest of the Human Computer Interaction community on mobile device and wireless communications that allow people to be connected everywhere and everytime. As one of the MobileMAN-technology targets is to support the creation of (virtual) user's communities we felt this as an interesting context to discuss MobileMAN social aspects. In particular, using ICT technologies (mainly, small smart devices with wireless technology) for supporting the users' interaction was indeed the main focus of the workshop "Lost in Ambient Intelligence" (http://www.chi2004.org/program/prog_workshops.html#ws9) where we presented the project.[8] "Ambient intelligence is said to consist of ubiquitous computing plus social and intelligent user interfaces allowing social interaction" [Nijholt A. et al. – Lost in Ambient Intelligence? - CHI04 Workshop].

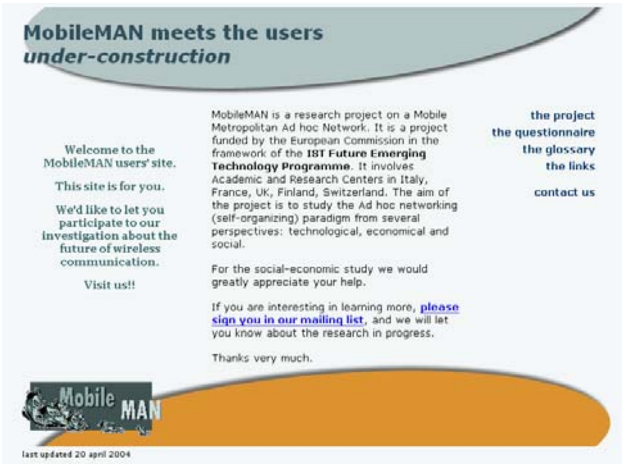
At the full-day workshop there were presentations from industries, as well as academic institutions, from USA, The Netherlands, Austria, Italy, Germany, United Kingdom and Switzerland, and from professionals mainly in the Human Computer Interaction and Computer Science. It is well known

---

[8] P. Andronico "MobileMAN Project".

that ICT field nowadays needs more and more interdisciplinary contributions from different scientific and humanistic fields. For this reason our scope to present MobileMAN project to this type of audience was mainly to try having some feedbacks from this multi-disciplinary community of specialists.

Because we wanted to reach the interest and the comments of a multi-disciplinary community, we gave a very general introduction of the project, omitting the more technical aspects, and focusing the talk on users' standpoint. Just after the presentation, there were doubts and difficulties on understanding something that is under development (e.g., the differences with mobile-phone services, hot spots, and similar technologies were not clear) even though the audience was in average skilled in ICT technologies.[9] The doubts were (at least partially) solved after explaining in a more detail what the Wi-Fi technology is, and how we can use it to construct self-organizing, free of charge, community networks.



An interesting result of the workshop discussion was that people are no more passive with respect to technology. This attitude can help us in trying to involve people and let them reflecting on how their interests and way of life can be supported/enhanced by this technology. However, to achieve this, we had to focus the attention on the users' needs trying to support their daily activities through applications and services.

All the workshop participants seemed to agree on the difficulties we should encounter when involving people, mainly if belonging to not-skilled groups.

The main difficulty for presenting the MobileMAN project to users is the innovative level of its paradigm "citizen's network". To solve this, we must try to explain MobileMAN to people by playing, and/or using other languages and metaphors, avoiding some aspects that are not yet in the common background. For this reason we developed a user website that we intend to advertise with a proper visit card. The card was already distributed at the ACM CHI2004 conference as well as at other international events.

[9] This type of problems was already reported by SUPSI-DSAS by interacting with not-expert users.

4. LESSONS SUMMARY

According to the two objectives of this MobileMAN first-phase evaluation, we can subdivide the lesson learned in two classes:

- technical issues;
- users' interaction issues.

4.1 Technical Lessons

From our experimentation we can conclude that good pieces of software exist, correctly implementing the single functionalities required in a MANET. The implementation of AODV and OLSR we tested are quite robust. They are able to maintain updated routing tables even under frequent changes to the network topology. However, their usage is not yet user friendly. Problems were experienced depending on the release of the LINUX kernel.

The FreePastry implementation we tested properly operated on top of our multi hop ad hoc networks.

On the other hand severe problems have been identified from the performance standpoint. The problems affected almost all MANET layers: network interface, routing and forwarding, TCP, and Pastry.

The quality of the wireless links is highly variable. IEEE 802.11 operates in the ISM spectrum and hence experienced a lot of noise from external sources. In addition, the increasing success of WiFi hot spots tends to saturate all the available channels. In our experimentation we have often to switch our MANET on a different 802.11 channel to avoid the influence of existing WiFi access points.

Routing and forwarding performance problems were experienced when using AODV and are due to the reactive nature of the protocol. The delays due to path discovery and maintenance have a strong negative impact on upper layer protocols that use "connection-oriented" operations. Specifically, with AODV, when no route is in the node cache, we measured, for completing a simple ping-operation between two nodes at a 2-hop distance, a delay of about 20 seconds (values of this order were experienced several times). For the same operation, using OLSR it (typically) takes 1 second (or less) to complete the same operation. OLSR generally has updated routing tables. In rare cases in which the ping operation was performed just after a change in the topology, and hence no updated route was available, we experienced a delay of up to 8 seconds to complete the ping operation.

AODV path-discovery delays have a negative impact on upper layer operations. These delays often caused timers expiration in upper layer protocols (e.g., the FreePastry timeout related to ring maintenance) that, as a consequence, declared failed an operation that is indeed only delayed due to AODV delays.

From the overhead standpoint we observed that, as expected, OLSR produces a higher routing traffic respect to AODV, but at least in the network we analyzed, the percentage of this traffic is small compared to the 802.11b available bandwidth.

At the transport level, we experienced the TCP problems already pointed out in the literature. Long TCP connections show a throughput that decreases with time. This aspect requires further investigation in future experimentation to verify if an appropriate tuning of the protocol parameters (e.g., advertised window) can fix these problems.

Finally, at middleware layer, the FreePastry implementation by operating its own routing ring independently from the underlying ad hoc network introduces a heavy overhead on the ad hoc network. In addition, FreePastry maintenance operations exploit the TCP services and hence poor TCP performances couple with the FreePastry overhead to reduce the overall system performance.

From this experience we gained some indications for solving the performance problems in our MANET. Specifically, these results provide additional arguments to using for a MANET the cross layer architecture as proposed in ([CMTG04], [CCMT04], [D5]).

Results related to the comparison between reactive and proactive routing protocols indicate that, with a proactive protocol: i) the response times are much better, and ii) the protocol overheads, at least inside the ad hoc horizon, are not heavy. Furthermore, results related to FreePastry indicate that significant performance benefits can be expected if routing information (extended with services information) can be used at the middleware layer to implement the overlay ring maintenance operations avoiding the big overheads connected with implementing it via middleware operations.

According to this indication we are currently developing the software modules required to exploit cross layering interactions according to the reference architecture shown in Figure 41. The vertical component, Network Status (NeSt), is introduced to enable indirect interactions between protocols belonging to different layers of the protocol stack. This approach combines the flexibility of the layered architecture (protocols at different layer are designed and maintained independently) with the performance optimizations of a full cross-layer protocols design [CCMT04].

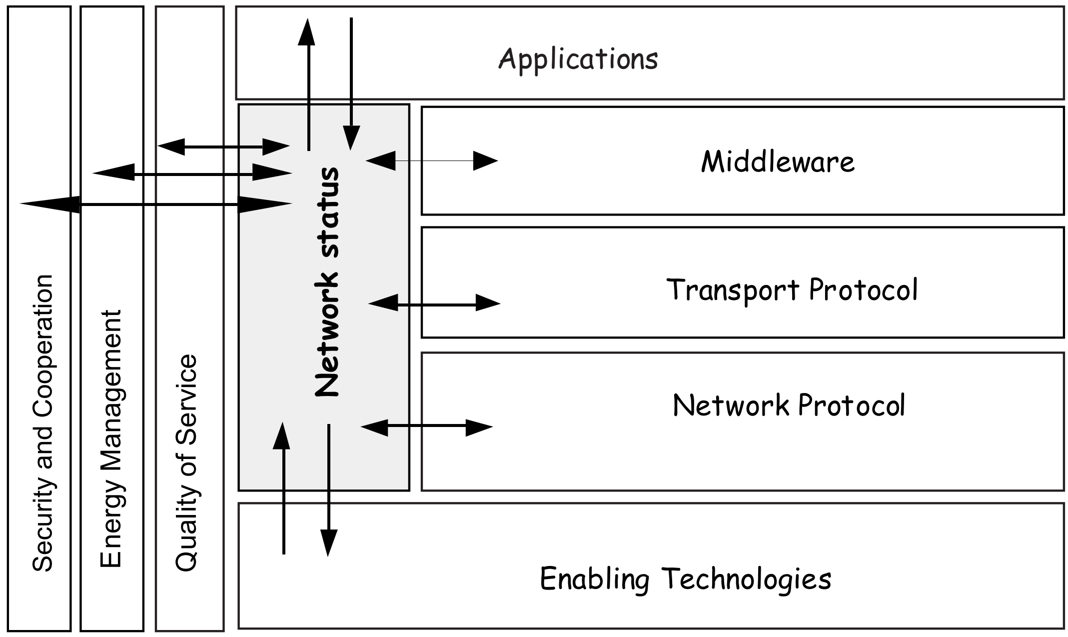


Figure 41: Cross-layer architecture

To implement this architecture we are currently developing three main software modules:

1. A proactive routing protocol optimized from the overhead standpoint by adopting a hazy-sighted link (HSLS) state approach [SSR01][SMSR02]. In addition, our protocol is able to transport (in addition to the link state information) the information required to implement cross-layer interactions, for example the services supported by each node of the network. This information which will be used by higher layers (e.g., middleware) instead of implementing their own service discovery functionalities.
2. An enhanced version of Pastry that exploits cross-layer interactions with the routing protocol to minimizing the overheads due to the overlay ring maintenance. This cross-layer aware version of Pastry has been named CROSS-ROAD: CROSS-layer Ring Overlay for AD Hoc Networks.
3. The Nest module that provides the interactions between the routing (OLSR in a first phase, then our HSLS module) and CROSS-ROAD.

As soon as the above modules will be ready, a new experimental phase will be planned to compare the legacy and the cross-layer MANET architectures. In the meantime, the experimentation with a legacy MANET continues. Among the planned activities are both indoor and outdoor experiments. Indoor experiments will be used to further investigate the most interesting and critical aspects so far observed: i) delays introduced by a reactive protocol; ii) the relationship between these delays and the timer values used at the transport and middleware level; and iii) the causes of the poor performance exhibited by TCP protocol in a multi-hop environment.

REMARK. The experimental activities we performed were also very important to gain experience on how to manage testing activities with a significant number of participants. Indeed, in this preliminary phase of MobileMAN evaluation we involved 10-12 participants almost all aware of MobileMAN technical issues. Nevertheless, during the first experiments disorder and testing problems (software alignment on a large number of machines, coordination of a large number of users, etc.) emerged. We can expect that these problems will increase exponentially with the number of (not-expert) users. This first phase has thus been important to acquire more knowledge about the experimentation problems. Starting with this experience we are now able to plan, in a better way, future experimentations involving a larger number of users.

4.2 Lessons from Users' Interaction

One main lesson emerged from all the activities that involved the class of users (university students) we identified for the first stage of MobileMAN evaluation: voluntary collaboration should not be taken for granted, but is a complex concept that we are currently studying in more details. With regard to the students, the limited collaboration was due to a number of factors: period of the year close to the examination phase, no incentives, etc. On the other hand, we observed that when the MobileMAN activities are part of the curricula tasks the students are willing to participate and dedicate more effort to them then requested. For example, the students spending their Master training activities at CNR are dedicating to MobileMAN more hours per-day then requested by their Master stage, simply because they are attracted by the innovative level of the project. Similarly, students at the University of Pisa participated in a very active way to our talks when they were part of their official courses. For these reasons, for this class of users the best solution we identified for stimulating their participation to the MobileMAN evaluation is to include these activities as a part of university courses. More precisely, SUPSI social evaluation activities will be placed within two courses held at HUT during the next academic year. Furthermore, to get feedbacks on the MobileMAN technical solutions, CNR is currently discussing with professors at the University of Pisa to include some MobileMAN related activities as a part of the training activities inside computer engineering courses.

5. REFERENCES

[APE02] APE: Ad hoc Protocol Evaluation testbed. Department of Computer Systems at Uppsala, Sweden. http://apetestbed.sourceforge.net/

[BCGS04] S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (Editors), *Mobile Ad hoc networking*, IEEE Press and John Wiley and Sons, Inc., New York, 2004.

[CCMT04] M. Conti, J. Crowcroft, G. Maselli, T. Turi, "A Modular Cross-layer Architecture for Ad Hoc Networks" in *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks,* Jie Wu (Editor), CRC Press, New York, 2004.

[CMTG04] M. Conti, G. Maselli, G. Turi, S. Giordano, "Cross Layering in Mobile Ad Hoc Network Design", *IEEE Computer*, February 2004, pp. 48-51.

[D5] MobileMAN Deliverable D5, http://cnd.iit.cnr.it/mobileMAN

[DKSDZ03] P. Druschel J.Kubiatowicz I. Stoica F. Dabek B. Zhao, "Towards a common API for structured peer-to-peer overlays", February 2003.

[DR01] P. Druschel A. Rowston, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems" in *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Heidelberg, Germany, November 2001.

[FreeP] FreePastry, www.cs.rice.edu/CS/Systems/Pastry/FreePastry.

[GLNT04] Per Gunningberg, Henrik Lundgren, Erik Nordstrom, Christian Tschudin, "Lessons from Experimental MANET Research," to appear in *Ad Hoc Networks Journal*, special issue on "Ad Hoc Networking for Pervasive Systems", M. Conti, E. Gregori (Editors).

[HUT] http://www.tct.hut.fi/tutkimus/MobileMan/index.shtml

[Java] *Java API specification*, http://java.sun.com

[OLSR] http://www.olsr.org.

[SMSR02] Cesar A. Santivanez, Bruce McDonald, Ioannis Stavrakakis, Ram Ramanathan, "On the Scalability of Ad Hoc Routing Protocols", Proceedings of INFOCOM2002, New York, June 23-27 2002.

[SSR01] Cesar A. Santivanez, Ioannis Stavrakakis, Ram Ramanathan, "Making LinkState Routing Scale for Ad Hoc Networks", Proc. MobiHoc 2001, Long Beach, October, 2001.

[UU] http://user.it.uu.se/~henrikl/aodv.

6. APPENDIX

Official project website:
http://cnd.iit.cnr.it/mobileMAN/

MobileMAN interactive users website:
http://cnd.iit.cnr.it/mobileMANusers/

This questionnaire is part of a study to identify possible applications for a new technology, named MobileMAN, based on portable devices (e.g.: PDAs) wireless technology (e.g.: WiFi) and self-organizing networking. This technology, called Mobile Metropolitan Ad hoc Network (MobileMAN), is developed within a European project by research institution of several countries (Italy, France, UK, Finland, Switzerland). If you'd like to participate and help us, please fill this questionnaire. We are also designing an interactive users website to interact with you. If you'd like to see the results of this questionnaire and if you'd like to help us in future activities, please visit it. Thanks for your collaboration.

Personal Informations

Sex	Male			Female	
Age	< 25	26-35	36-45	46-55	> 55
Country of residence			**For years**		

ICTs General

How many hours you spend in front of a computer each day?				
< 1	1-2	3-5	6-8	> 8
For which of your everyday life you mainly use a computer?				
Work	Fun	e-commerce	Private	Other uses ....

MobileMAN is a technology that works with human interactions: it allows communicating without cost with colleagues and friends that are close to you, or close to your friends, or close to the friends of your friends...
At this time the implementation is based on WiFi technology. Friends are persons close to you within the WiFi transmission range.

Please, could you think what would you like to do with this technology?				
Make a call	Send an SMS	Play a role game	Writing your personal notes	Others
Please could you think about some applications/services you would like to have implemented with the MobileMAN technology?				

Figure 42: Questionnaire – Page 1

Wireless Devices

Do you use the WiFi technology? Please tell us on which devices.		
PDA	Portable Computer	Others ____________

Which kind of action you used to do with your WiFi technology?

____
____
____
____
____
____
____

Please, think now at you in your city and your daily communications:

How big is your city? (approximately)				
2km radius	5km radius	10km radius	20km radius	I don't know

In your work communications, your communications are within the ...			
Same building	Same quarter	Same city	Others ____________

In your home communications, your communications are within the ...			
Same building	Same quarter	Same city	Others ____________

Which type of relationship you have with the persons you communicate with?

Relationships	When at work	When at home	Elsewhere ____________
Family			
Friends			
Colleagues			
Neighbours			
Other types ____________			

Thanks for your collaboration.

If you want to help us again in the future, please, visit our MobileMAN interactive users website at the URL http://cnd.iit.cnr.it/mobileMANusers/ (under construction at the moment) or leave us your email address or send an email request to the following address to join our special mailing-list (mobileman-users-request@iit.cnr.it).

Your email address:

____________________@____________________

Figure 43: Questionnaire – Page 2