Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2007, Article ID 62967, 14 pages
doi:10.1155/2007/62967

Research Article

Experimental Evaluation of the Usage of Ad Hoc Networks as Stubs for Multiservice Networks

Miguel Almeida, Rafael Sarrô, João Paulo Barraca, Susana Sargento, and Rui L. Aguiar

Instituto de Telecomunicações, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal

Received 1 July 2006; Revised 22 October 2006; Accepted 11 January 2007

Recommended by Marco Conti

This paper describes an experimental evaluation of a multiservice ad hoc network, aimed to be interconnected with an infrastructure, operator-managed network. This network supports the efficient delivery of services, unicast and multicast, legacy and multimedia, to users connected in the ad hoc network. It contains the following functionalities: routing and delivery of unicast and multicast services; distributed QoS mechanisms to support service differentiation and resource control responsive to node mobility; security, charging, and rewarding mechanisms to ensure the correct behaviour of the users in the ad hoc network. This paper experimentally evaluates the performance of multiple mechanisms, and the influence and performance penalty introduced in the network, with the incremental inclusion of new functionalities. The performance results obtained in the different real scenarios may question the real usage of ad-hoc networks for more than a minimal number of hops with such a large number of functionalities deployed.

1. INTRODUCTION

Nowadays, user communication requirements are much more than simple connectivity: it is required to assure full service connectivity with high quality, independently of the user's location, and providing the best access at every time. The concept of mobile ad hoc networks (MANET), which include spontaneous grouping of nodes using wireless technologies and collaborating in order to provide communication facilities, gives an alternate path to these full connectivity requirements. The nodes in MANETs are typically PDAs, laptops or even sensors (with limited battery, reduced processing and wireless capabilities), sharing each other communication facilities in order to achieve overall system connectivity. One node by itself, with such limited characteristics, is not capable of a large communication range. When nodes collaborate helping each other in forwarding information from source to destination, the total value of the network is much higher than the sum of the communication span of each node. For such spontaneous networks to operate, address configuration mechanisms and routing protocols are the base mechanisms that need to be in place. There are already many proposals (e.g., [1–10]) covering both these topics and presenting resource efficient mechanisms to allow the creation of a MANET. These proposals appeared mainly due to the high interest in self-organisation networks, and to the requirement of solutions able to operate on resource constrained environments, for example, sensor networks. Many of these proposals have been evaluated using simulation tools [11], with the most popular being ns2 [12] and GloMoSim [13]. Some were further tested in limited testbed environments where real issues concerning program concurrency, hardware implementation, or real wireless interference are present. Simulations are useful to test network behaviour, and have been widely accepted as valid research tools mainly during the last decade, albeit their known deficiencies [14]. Ad hoc networks are typically simulated in scenarios [11] with tens to thousand of nodes distributed over an area sometimes reaching a few thousands of square meters. Generating the desired mobility and traffic patterns of so many nodes distributed over such large area is impractical in real environments, presenting unattainable costs. Simulators can help testing such scenarios by replacing the entire environment by a cluster of servers running a preprogrammed simulation set. Moreover, simulators have the capability to repeat simulations a large number of times with the same parameters or with subtle changes in one or more variables. Such level of control over the entire environment

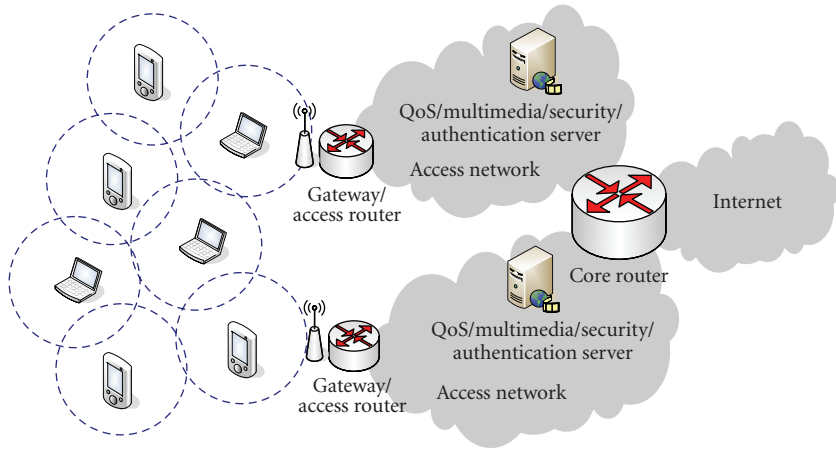

Figure 1: Extended hotspot scenario.

is of vital importance to early validation of new models and protocols.

There are several problems inherent to evaluation through simulation only, which range from limitations of models used, credibility of the proposal results in real environments, or even the (frequent) lack of statistic treatment applied to results. Simulators are dependent on run-time environment and tools, which can obtain different results depending on the architecture or compiler version used [11]. Proposals are sometimes based on scenarios considering situations which are unlikely to be feasible on real environments, or where other proposals already showed to have different types of problems [15]. Moreover, the details of the simulation are often not made available to the research community. In addition, results are sometimes simply dumped into the publication without further analysis, presenting situations where packet loss or delay could make a real application almost impossible to communicate [11]. For all these reasons, experimental evaluation of ad hoc protocols behaviour is essential, even if this is made only in controlled and simple environments.

In this paper, we aim to analyse the usage of MANETs as stubs for accessing complex multiservice networks, similar to those commercially expected today. Thus, we present the results obtained by deploying a multiservice ad hoc network integrated in an infrastructure network eventually managed by a 4G operator [16]. This corresponds to the often referred to as "extended hotspot" scenario (Figure 1), where the ad hoc network acts like a stub to the operators network and is able to provide external communication links, sharable by all users in the ad hoc cloud. In the conventional "hotspot" scenario, all users are directly connected to the access point, which limits coverage and increases radio interference, but provides easy access to multiple services. In the "extended hotspot" scenario, multiple services are required widely for correct integration of the ad hoc cloud within such operator business architecture. More important, these are supposed to execute simultaneously in all nodes, due to the dynamic nature of such ad hoc environments. Understanding the result of the cumulative effect of stacking different modules is of vital importance to the research community developing proposals for these integrated environments. Particularly, it allows a better understanding of the inherent limitations of ad hoc wireless networks and of the potentially multiple functionalities deployed there. This promotes more realistic expectations on features to be supported in this environment, as well as limitations resulting from each solution or from the interactions presented by the several functionalities.

Notice that, in this "extended hotspot" concept, the services to be offered to the users should be similar to the ones offered through a direct connection to the operator managed network and we expect the size of the ad hoc network has a large impact on its feasibility for these types of scenarios. Thus, in our study, we addressed issues associated with typical multimedia networks: connectivity (address autoconfiguration and support of routing, both unicast and multicast), QoS, and charging mechanisms. Since we are focusing on multimedia applications, no analysis is made on transport protocols. We rely on software developed mostly inside the EU project Daidalos [17] and followed an architecture similar to the one proposed by this project.

There are already in the literature many ad hoc network evaluation studies through real testbed deployments ([18–20]). However, most of the studies address routing or QoS issues, with single functionalities evaluated. To our best knowledge there is no study addressing simultaneously all the functionalities required to properly integrate an ad hoc stub in an operator environment. Because individual proposals are effective and capable of providing the expected set of functionalities, interoperation issues arise from integrating several of them. In particular, network overhead and delay accumulate, reducing the network usage experience.

The paper is organised as follows. Section 2 presents the state-of-the-art of some ad hoc protocols proposed in the literature, considering also the ones implemented in our prototype network. Section 3 addresses the software implementation used and the protocols chosen to support the envisioned functionalities. The description of the relevant parts of the

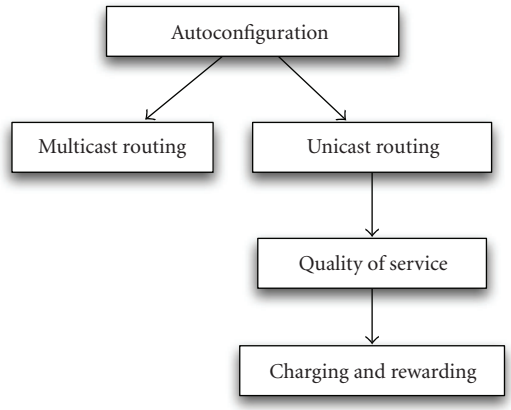

Figure 2: Functional architecture.

ad hoc network testbed is performed in Section 4, and the results achieved are depicted in Section 5. In Section 6, we discuss the impact of the "extended hotspot" scenario, evaluating the drawbacks and benefits of adding certain functionalities to the network. Finally, the main conclusions are presented in Section 7.

2. AD HOC PROTOCOLS FOR 4G SOLUTIONS

Bringing ad hoc networks into a 4G scenario [16] implies interconnecting them with the infrastructure network and supporting basic mechanisms. These mechanisms ensure the creation of such a spontaneous network as a valid extension of the overall operator architecture. Thus, it is essential to evaluate performance on major functions: autoconfiguration (including gateway awareness), routing, QoS, and charging. Although not all of these functions are necessary in traditional ad hoc networks, this basic set of mechanisms must exist for the operators to supply existing services (e.g., voice). Research for the support of the above mechanisms has already led to a large number of publications. The next subsections briefly address various proposals to provide the functionalities required. The set of mechanisms being addressed and their dependency, are represented in Figure 2.

2.1. Autoconfiguration and gateway awareness

In order to effectively communicate in a given network, nodes must have valid and unique identifiers inside the network prefix they belong to. At physical and MAC layers, the wireless card must associate with the network, after which, at network layer, a routable IP address must be obtained. Although the infrastructure network already supports functionalities such as DHCPv6 [21], a node entering the ad hoc network usually has several nodes around, and probably several independent networks to use, and needs to choose one of them (either by traffic or cost considerations).

Proposals [5–10] present some of the possible methods used to disseminate network configuration in this type of networks. Perkins [9] proposes a simple mechanism for autoconfiguration where nodes simply choose a random address and perform a duplicate address detection based on a given network prefix. Jeong et al. [8] propose a solution that differs from the previous by specifying mechanisms more suited to AODV, both for IPv4 and IPv6; it supports the existence and mergers of different network partitions. Laouiti [10] describes an autoconfiguration mechanism for isolated networks with OLSR. Wakikawa et al. [6] propose a method to propagate the network prefix inside the network by means of an Internet Gateway Discovery process similar to the router discovery process of IPv6, and include the integration of MANET routing protocols with Mobile IPv6. Jelger and Noel [7] propose a method where the gateway providing connectivity to the Internet periodically broadcasts a message (GW_INFO), which is then forwarded by all nodes in the ad hoc network. It further supports multiple gateways in the same ad hoc and the ability to choose one of them based on specific metrics, such as the number of hops to the infrastructure.

Secure operation of these protocols is very important in commercial environments, especially when dealing with self-configuration solutions. This prevents the advertisement of any node as a gateway, disrupting the network or increasing the chances of an eavesdropping or black hole attack. Jelger's proposal has been further extended in [22], adding support for security and integration with handover mechanisms. The information GW_INFO messages are signed by the operator and nodes are able to verify this signature using the public key infrastructure.

2.2. Unicast routing

The routing protocol is the element responsible for determining the best route from a source to a given destination. After route is determined, the forwarding mechanism processes the packets according to the information in the routing tables. Topology may change during session lifetime, requiring the routing protocol to react and update routes between end-points. Because of the nature of ad hoc networks, routing protocols should be highly dynamic and robust. Ad hoc routing protocols are often classified regarding its method of finding and maintaining routes, namely: proactive, reactive or hybrid. Popular solutions providing routing in ad hoc networks are AODV [1], OLSR [2], DSR [3], and DYMO [4]. OLSR is a proactive protocol while AODV, DSR, and DYMO are reactive. The first keeps a multipoint relay (MPR) graph in the network, which is responsible for optimizing the routing messages flooding process. OLSR seems to be adequate to networks with high concentration of nodes, although its overhead increases directly with the number of nodes. AODV and DSR calculate routes on-demand and usually deliver better performance, especially in networks with stalled nodes. Overhead is not directly dependent on the number of nodes, making it more suitable to large scenarios where nodes have power limitations. DYMO is a more recent proposal aggregating concepts from both AODV and DSR.

2.3. Multicast routing

Streaming services, such as IP Television, require network conditions to be stable, with low jitter and delay. Because consumption of these services is based on membership rules, and the same content is distributed to a large number of clients, multicast is an important method to consider. Multicast routing is able to deliver the same content to multiple clients upon proper service subscription. The cost to the network is some additional signalling required to maintain the distribution tree and client subscriptions. However, the load on the network as the number of clients increase is close to $O(1)$ instead of the typical $O(N)$ presented by unicast.

Several proposals, [23–27], are able to provide multicast delivery optimized to ad hoc environments. MAODV [23] and MOLSR [24] are, respectively, the multicast versions of AODV and OLSR. ODMRP [25] and ADMR [26] are multicast ad hoc routing proposals that reduce the overhead of maintenance of the multicast tree in the ad hoc network. However, none of these proposals is directly adapted to integration with an external infrastructure.

In multicast communications, a tree extends from the content source to the receivers. In hotspot scenarios, the source can be located in a node on the ad hoc stub; however, usually will be a server on the operator core or on another access network. Thus it is of vital importance that protocols running in the core and ad hoc stub are integratable.

To provide interconnectivity to the Internet, MMARP [27] proposes special nodes (ad hoc nodes directly connected to the gateway) responsible for adapting traffic between MMARP and IGMP [28] formats. Besides, supporting natively infrastructure connectivity, MMARP exhibits lower overhead when compared to the previous proposals.

2.4. Quality of service

Network infrastructure is expensive and has very well-known limitations in terms of bandwidth. As network load increases, QoS traffic parameters like delay, jitter, or packet loss also increase, degrading network conditions. In order to provide the best possible service, while maximising profit, operators have a strict control over the QoS characteristics of their networks and keep their backhauls over provisioned.

When integrating an ad hoc network with an existing commercial network, operators expect to apply the same QoS levels to users. Traditional hotspots can perform this easily by a set of rules at the access point. However, since the ad hoc stub is a distributed and unstable environment, QoS has to be sustained in a distributed manner. Several protocols have already been proposed to support the delivery of adaptive services in mobile ad hoc networks [29–32]. INSIGNA [29], one of the best known, uses a soft state resource management mechanism to enhance network usage. Packets transport an extra field for QoS information, which is used as an in-band signalling. The protocol supports Best Effort services and services requiring reservation with per-flow QoS support. QOLSR [30] is a QoS routing protocol defined to enhance OLSR. Each node gathers information related to QoS parameters such as available bandwidth, delay, jitter, or loss probability. These parameters are reported to OLSR, based upon which the MPRs create or change routes. However, QOLSR is not able to limit the traffic in the network. SWAN [31] uses distributed control algorithms to handle two types of traffic, Best Effort and Real Time, through shaping. It performs rate control for Best Effort traffic, in which traffic marked with less priority can occupy up to the maximum bandwidth left by the Real Time traffic usage. The bandwidth usage by the Best Effort traffic raises according to an additive increase, multiplicative decrease (AIMD) rate control algorithm. SWAN uses source-based regulation algorithms in which congested nodes send messages informing intermediate nodes to wait for a random amount of time before trying to re-establish connectivity. Dynamic regulation is also performed to deal with mobility and false admission issues. In [32] an extension of SWAN was proposed to make it interoperable with the infrastructure and to support four classes of traffic.

2.5. Charging and rewarding

Operators need to be able to profit from the development of the network and services. Since infrastructure networks are driven by operator business models, it is mandatory to support for charging the users. The multihop and distributed nature (and dynamics) of ad hoc networks requires the existence of distributed trust mechanisms, able to provide adequate information for charging and traffic authorization. Most important, these mechanisms need to be compatible and integrated with existing network authorization and charging architectures. Furthermore, ad hoc networks also require incentives for users to participate in the forwarding process, otherwise, nodes may not forward others traffic without any benefit. Such incentives can be provided in many forms, like, for example, credit or service discounts.

Solutions like [33–38] envision scenarios where ad hoc networks are integrated with an infrastructure supporting authentication, authorization, and charging mechanisms. SPRITE [33] assumes that nodes have enough storage capacity to store traffic proofs. These proofs are later traded at a bank for credit when the node is connected to a high bandwidth medium. Salem et al. [34] envisions ad hoc extended cellular networks, where base stations are capable of charging, rewarding, and enforcing profile policies on packets generated. In order to achieve this level of control, it proposes all traffic to cross the base station, independently of its origin and destination. SCP [35–37] proposes the creation of a distributed mechanism, actively marking packets with a proof that is updated at each forwarding node and then reported to the network operator, with intrinsic class differentiation. The proofs are built and updated using a defined set of rules and supported by cryptographic signing and verification primitives. PACP [38] improves many of SCP deficiencies (overhead, variable packet size) by encoding the route in a polynomial included in the packets, and securely updated at every node. Upon reception of the charging information on the infrastructure network, the appropriate charging and rewarding actions may be applied. These actions can take in consideration many individual parameters,

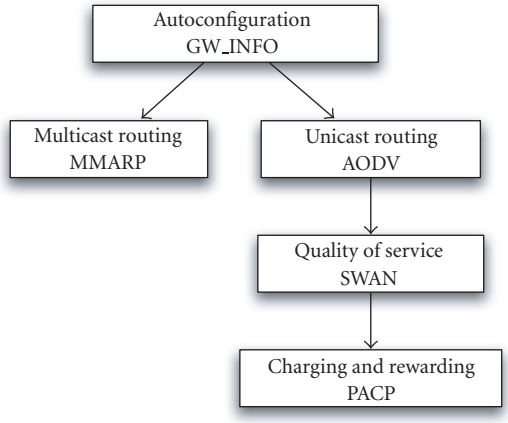

Figure 3: Functional architecture with protocols included.

like individual user profile, service description, QoS parameters, route length, time frame, or data amount. Also, PACP supports distributed access control, allowing the operator to control which flows are allowed between each nodes, without sacrificing routing.

3. MOBILE NODE ARCHITECTURE

The above-mentioned functionalities need to be present in an operator "extended hotspot" aiming at providing multimedia services (real-time voice and video) mixed with bulk traffic. First, autoconfiguration mechanisms are required to enable the nodes to discover hotspots and autoconfigure correctly. After nodes are properly configured, unicast and multicast routing is required to support basic network access. Enhanced services such as voice calls require some form of differentiation from bulk traffic. Finally, operators must be able to apply contracted profiles agreed with each user. Since traffic should not be forced to cross the gateway, both QoS and charging must be performed in a distributed manner, but without disclosing the user profile. The described functional architecture and the protocols chosen to address each functionality are depicted in Figure 3. The next subsections detail the mechanisms implemented.

3.1. Autoconfiguration and gateway awareness

The proposal presented in [7] and then extended in [22] was found to be the most appropriate to our environment, as the others lacked either in security [6–10], dependence on the routing protocol [9, 10], or adequacy to hybrid scenarios [9, 10].

In [22] nodes are able to choose which network to connect and handover between gateways using Mobile IPv6 [39] for global connectivity. Nodes build a tree starting at the gateway and spreading to all nodes in the ad hoc cloud. Independently of the routing protocol, the tree is proactively maintained and nodes always search for the shortest (best) path to the gateway. Besides disseminating configuration information, the integration of this tree with the routing protocol brings clear benefits: whenever a route is not found because the destination node is located on an external network, nodes can use this tree as the optimal path to the gateway.

3.2. Multicast routing

From existing common proposals, only MMARP [27] is able to deliver multicast traffic on the ad hoc stub maintaining compatibility with the rest of the Internet, which typically runs IGMP/MLD [28]. MMARP allows the provision inside the ad hoc cloud the same multicast services provided to infrastructure nodes, without any change to the existing architecture, in a secure manner [40].

To achieve this, MMARP proposes the creation of multicast Internet gateways (MIG), ad hoc nodes directly connected to the gateway. These nodes are responsible for adapting traffic between MMARP and IGMPv6 formats. They communicate with the gateway by notifying it about the interest revealed by other MMARP enabled ad hoc nodes. The MIG sends periodic advertisements to the ad hoc nodes, assigning itself as a default multicast gateway, and informing about the path towards multicast sources in the fixed network.

Any of the ad hoc nodes may become an MIG at any time, and does so when directly connected to the gateway. Besides this proactive behaviour, MMARP includes a reactive component to create and maintain the distribution tree over the ad hoc network, using *Join* messages towards the source to create a multicast shortest path.

3.3. Unicast routing

In our extended hotspot scenario, the envisioned number of nodes is expected to be low, particularly due to limitations arising from concurrence in medium access. In this sense, AODV and DSR or DYMO are better suited for the scenario envisioned.

Due to interoperation issues with the implementations available, AODV was chosen for our prototype. The implementation used was the one available at Upsala University—AODV-UU [41]. Some changes had to be performed to the base implementation in order to support mobile nodes' self-configuration and dynamic change of interface address (support of node mobility within an ad hoc access network and between gateways is required). Moreover, some modules (e.g., charging) needed the nodes to report their routing tables. Although information about the next hop for a given destination could be retrieved from the Linux Kernel routing tables, AODV is able to provide more useful information, like alternative routes. Finally, since we consider that nodes need to be authorized, interfaces need to be in place between AODV and the authorisation modules (when the authorisation arrives, the route previously computed can be invalid and AODV must be triggered to initiate a new route discovery process).

All these changes, related with maintenance operations, are expected to have little or no impact in the resulting performance or operation of the AODV implementation, especially in low mobility scenarios.

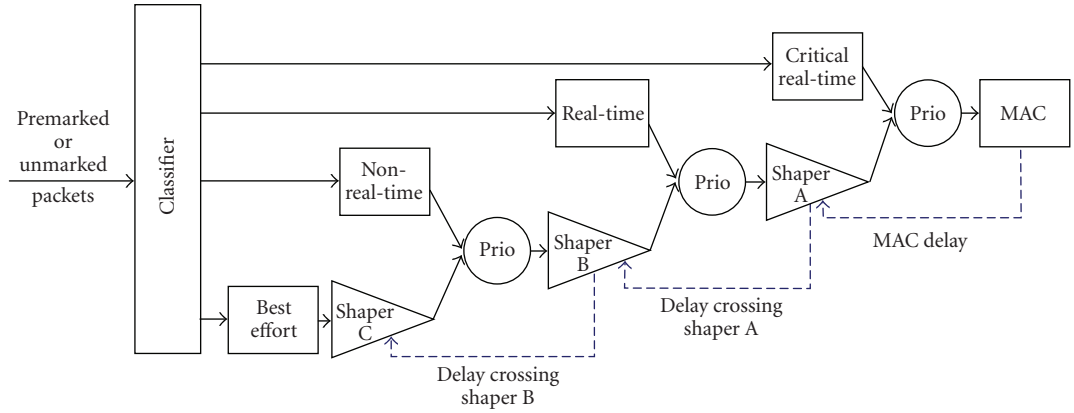

Figure 4: Differentiation model.

3.4. Quality of service

According to studies on the well-known protocols to deliver QoS in ad hoc networks [42], SWAN proves to be one of the best choices: it has lower overhead than ISIGNIA and is the QoS protocol that performs better with AODV. In order to allow the QoS interoperation among ad hoc and infrastructure networks, the base SWAN signalling was adapted and extended [32] to interoperate with infrastructure QoS signalling based admission control, and to support multipath probing. The differentiation model was extended to support several service classes and congestion feedback between them. This extended differentiation model considers four different traffic classes: critical real-time traffic, less demanding real time traffic, nonreal-time traffic, and regular best-effort traffic. Each of these classes will have assigned a certain amount of bandwidth, except the best-effort, that uses the leftovers. Figure 4 presents the differentiation model composed by a classifier and by a cascade of priority schedulers, shapers and queues associated to each traffic class. The delays are applied to each packet through a leaky bucket shaper, whose rate is controlled by an AIMD algorithm having the lower-level classes delay as feedback.

The implementation used follows this extended model supporting 4 classes. This software also provides extended session admission and integration with external authentication and authorization servers.

3.5. Charging and rewarding

Although several solutions provide means to charge for traffic in ad hoc networks, not all are appropriate for the extended hotspot scenario: no proper interoperation with the infrastructure network [33], large overhead in the ad hoc network [35] as the number of hops increase, or use of nonoptimal routes [34].

When nonideal rewarding is acceptable (i.e., guarantees are for "approximately," but not exactly, 100% of the traffic), then PACP is one of the best proposals, able to provide correct charging and rewarding information, securing the processes of proofs creation and delivery, without the need of suboptimal routes, and with small network overhead and processing requirements in all nodes in the path. PACP implicitly includes in data packet the identification of the route (in a fixed size field) that will be updated in each node in the ad hoc network towards the destination. The fields containing information on the route are cryptographically secured, so they cannot be wrongly modified along the path. If a malicious node corrupts this information, the next hop will detect the packet is invalid and will drop it. The node belonging to the flow's path one hop way from the receiver, which we denote as the last forwarding node, is responsible for sending the proofs to the gateway, which contain information on the path(s) of the flow. When receiving the proofs, the gateway sends them to the authentication and accounting server to verify the truthfulness of the information, through the cryptographic information contained in the proofs, and retrieves the information of the ad hoc route. PACP associated with proper gateway control processes can provide the tools required to check the behaviour of nodes inside the ad hoc network, eventually leading to creditation/reputation schemes, developed with the aid of the network operator.

3.6. Implementation environment

Software for the nodes was developed on a Linux environment. Mandrake 10.0 Official was selected as the distribution to be used in this testbed, with the vanilla 2.6.8.1 kernel, enhanced with modifications required by some of the tested modules. These enhancements are the support for DSCP marking using Netfilter, the Hostap wireless driver, a Netlink multiplexer, an IP6 QUEUE Multiplexer, support for Token Bucket Queue, an enhanced Mobile IPv6 RC2 stack, and a customised version of MACKILL [43], for testing purposes. With the exception of (parts of) the Mobile IPv6 stack, AODV-UU and the HostAP driver, all additional modules were developed inside the Daidalos project. Kernel space modules were sparsely used in an attempt to make the developed modules portable and easy to deploy on different machines with different distributions and kernel versions. A partial vision of the software modules used is depicted in Figure 5, mostly focusing in the customised

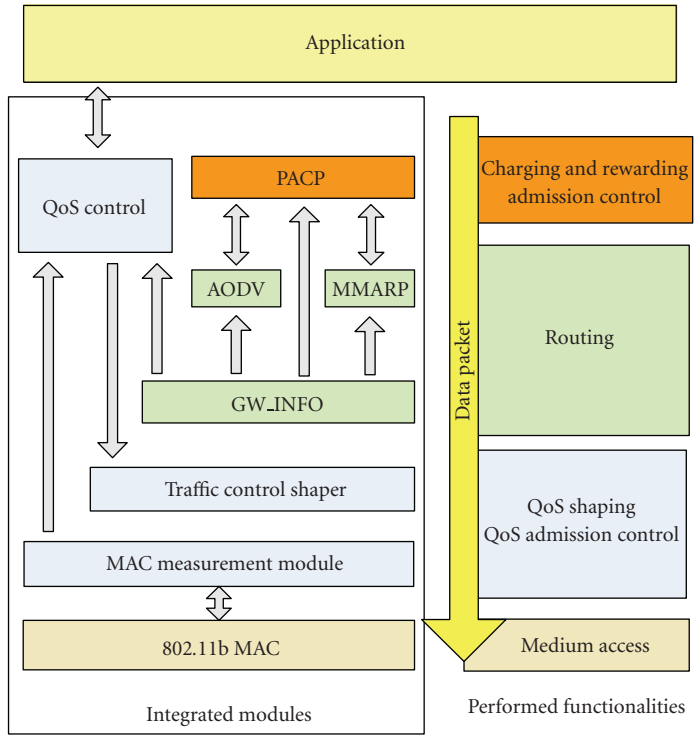

Figure 5: Software architecture.

functions described above. Note that these modules can be mostly turned on or off, according to the specific test to be performed (in some cases, activating some dummy modules). Furthermore, note that other software was also present, but is not discussed due to paper limitations.

The overall integrated system proved difficult to manage, but reached an integration level adequate for controlled trials. However, even considering this as research prototype software, the large number of interactions identified has raised some concerns on the development cost for reaching reliable, integrated software usable in commercial devices. This is especially relevant when taking in consideration the low resources available on a typical terminal.

4. TESTBED DESCRIPTION

The integrated ad hoc testbed is comprised of several Linux computers running the modules previously described. All machines have, at least 1.2 GHz CPU, 256 Mb RAM, and enough storage space; one of the problems identified with the extended hotspot concept is the fact that the mobile node was not even able to run effectively if its specifications were worse than these. These specifications do not reflect typical, resource limited, (current) ad hoc nodes, but are only suited to the extensive testing possible in a lab, or to yet-to-be-developed small form factor PDAs. All machines are equipped with 2 network interfaces: one wireless and one wired. The wired interface is used to provide remote access during the tests and for administrative tasks. Ad hoc networking is limited to the wireless interfaces, and the developed protocols operation is restricted to these interfaces. One of the nodes (acting as a gateway) is used to interconnect the ad hoc cloud with the infrastructure network, and here the wired interface will also be used to transfer data to or from the ad hoc network. The wireless interfaces used were Prism 2.5 802.11b cards with the following configuration parameters: ad hoc and promiscuous modes, channel 12, rate fixed to 2 Mbits and RTS/CTS threshold of 1 byte. The bit-rate limitation was in place to increase reliability, avoiding bit-rate changes and support a channel with bit-rates easily handled by the mobile nodes. Channel 12 was selected for interference minimisation.

We have conducted the test in two different topologies, one indoor and one outdoor. Both are string topologies, that is, the nodes are connected sequentially to only two neighbours, in order to maximize the number of hops (see Figure 6). Six machines were used in a multihop linear structure (string topology). Node 1 is the gateway and is directly connected to the infrastructure network, and Node 6 is at the other end of the network. The results are stable enough with six machines, and the idea of using a string topology, without interfering traffic, is to show a bound on the maximum achievable performance. In real world scenarios, results will be consistently worse than the ones achieved in these tests.

In the indoor topology, the nodes have been deployed in a roughly square building with around 36 m size, and normal office/lab divisions ("IT" building in Figure 7). Many WiFi access points exist inside the building, mostly on channels 1, 6, and 11. Since there is not enough physical space

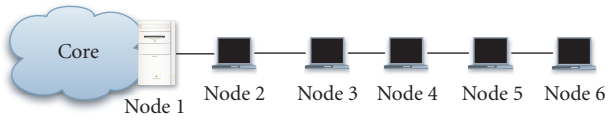

Figure 6: Topology of the indoor tests.

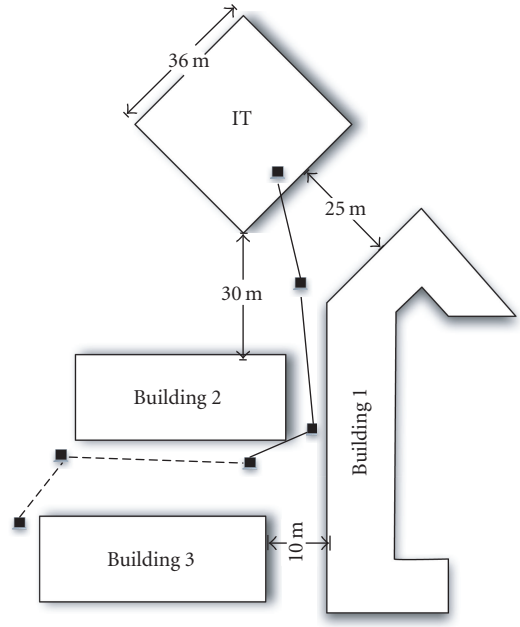

Figure 7: Topology of the outdoor tests.

to create the desired topology without nodes interfering with each other, the MACKILL tool was used to perform filtering (in kernel), based on the source MAC address, ensuring a logical string topology. Most of the tests were performed without traffic in the building (weekends), and in some cases, with the nearby access points (those on channel 11) powered off. Figure 7 shows the node's placement for outdoor tests. The topology is now physically a multihop linear structure (MACKILL tool was not used, since the nodes are sufficiently far away from each other for the routes to be stable). This topology was used to analyse the impact caused by wireless interference.

5. EXPERIMENTAL RESULTS

In this section, we present the results obtained with individual functionalities in the ad hoc network. We aim to test delay, jitter, and overhead, for a specific set of traffic profiles, targeting multimedia communications. We further evaluated network throughput with the incremental addition of nodes.

We define three UDP traffic profiles according to different bit rates, 64 Kbps, 128 Kbps, and 256 Kbps, to evaluate the network without being in stressful situations. These traffic profiles emulate envisioned voice and video communications supported in these stub networks (these are the services with more requirements). Packet size used was 512 bytes, unless otherwise specified. In the QoS tests, four classes of traffic were addressed: real-time, less demanding real-time, nonreal-time, and best effort. Traffic is generated with the aid of the MGEN tool [44].

For each configuration, 5 tests were made, with 300 seconds runs. The presented results are the mean of the 5 tests. The incremental addition of nodes in the network allows the evaluation of real deployment possibilities and drawbacks of each mechanism in the ad hoc network, when used to deliver multiservices in an operator environment.

In Section 5.3, we perform a comparison between the performance of unicast routing for both the indoor and outdoor scenarios. The remaining results were obtained using the indoor topology.

5.1. Autoconfiguration

The Jelger mechanism to autoconfigure the addresses of the ad hoc nodes was evaluated. We addressed the overhead introduced in the network and the time needed for self-configuration, which represents a period of nonconnectivity.

Measured overhead is 922 bps per link which, for a 64 kbps bit rate, represents 1.44% of the data traffic. Autoconfiguration time takes an average of 2 seconds and represents the time between the reception of the first GW_INFO message and the transmission of the first GW_INFO message to other ad hoc nodes (when the node is fully configured). When a node moves inside the ad hoc network, it receives a new GW_INFO message, from a potential new upstream neighbour, after 1 second, in the worst case scenario. After the reception of that message, the new default gateway is configured and new routes can be calculated by the routing protocol. Generally, autoconfiguration was seen not to have a large impact in network performance.

5.2. Multicast routing

Multicast tests were performed with MMARP, also in a string topology. In this scenario, Node 1 is the multicast sender. First, Node 2 sends a *Join* message to start receiving the multicast traffic; then, Node 3 sends a *Join* message. Upon this process, Node 2 becomes an MIG; all the other nodes *Join* to the source to receive the same multicast service. Finally, Node 1 sends the traffic that flows in the entire network. It is worth noticing that apart from the MMARP protocol, only the autoconfiguration protocol was running in order for the nodes to get an IPv6 address.

The first metric evaluated is the throughput achieved. In Table 1 we show the variation of the throughput with the addition of more nodes to the network, both with multicast and unicast routings active. The traffic source is the Node 1, which sends a flow to all other nodes.

We observe that, in a direct connection between two hops, throughput is 1223 Kbps. This bit rate corresponds to the effective user data transmission. The real throughput in the network would be slightly higher due to additional headers and RTS/CTS mechanism. In a five-hop connection the

Table 1: Throughput: routing and autoconfiguration.

Hops	multicast (Kbps)	unicast (Kbps)
1	1223	1222
2	672	559
3	291	322
4	191	204
5	76	122

Table 2: Delay: multicast routing and autoconfiguration.

Delay (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	3.527	4.184	4.809
2 Hops	8.910	9.912	31.642
3 Hops	13.194	45.474	113.267
4 Hops	16.941	67.027	194.941
5 Hops	21.619	82.823	252.608

Table 3: Jitter: multicast routing and autoconfiguration.

Jitter (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	0.227	0.224	0.221
2 Hops	1.669	1.930	10.586
3 Hops	0.841	25.286	20.306
4 Hops	1.142	25.119	22.246
5 Hops	1.374	21.743	23.683

Table 4: Packet loss: multicast routing and autoconfiguration.

Loss (%)	64 Kbps	128 Kbps	256 Kbps
1 Hop	0.24	0.35	0.54
2 Hops	3.13	2.39	3.40
3 Hops	2.06	8.00	11.85
4 Hops	2.38	8.04	22.89
5 Hops	2.82	11.72	33.03

Table 5: Delay: unicast routing and autoconfiguration.

Delay (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	4.474	4.606	4.535
2 Hops	9.058	9.242	9.045
3 Hops	13.968	15.036	17.691
4 Hops	19.578	20.924	97.502
5 Hops	23.619	24.248	1333.563

Table 6: Jitter: unicast routing and autoconfiguration.

Jitter (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	0.560	0.741	0.697
2 Hops	1.254	1.236	0.997
3 Hops	1.248	1.434	1.835
4 Hops	2.205	1.975	13.456
5 Hops	1.452	2.228	21.474

throughput comes down to 76 Kbps. This behaviour is expected since all nodes are close to each other and radio interference exists.

Tables 2 and 3 present the delay and jitter for each traffic profile. The objective of these tests is to evaluate the impact of multicast routing in the traffic for different configurations of the testbed when the network is not fully congested. Taking these two parameters into account, it can be seen that the performance for the first hop is very similar for the three traffic profiles, since the available bandwidth is much larger than the one used. However, when the number of hops increases, delay increases for the two lowest bit rates studied (64 and 128 Kbps). The third flow (256 Kbps) shows large delays for hop counts larger than 3, when maximum throughput is exceeded. This increase is, obviously, larger for high traffic values. Notice that the delay value for a direct connection is smaller than the delay increase with the number of hops. This shows the penalty of multihop communications in shared environments. For 256 Kbps flows, delay reaches values larger than 100 ms, which is not acceptable for voice; however, video streaming can still be supported.

Table 4 shows packet loss results, reaching values larger than 10% for communications of 256 Kbps traversing more than 2 hops, due to excessive collisions in the shared media.

The overhead introduced by MMARP and GW INFO is 3.94% in 64 Kbps of traffic, which indicates that the overhead added by MMARP alone is almost twice the one introduced by the autoconfiguration protocol.

5.3. Unicast routing

In this section we evaluate the impact of introducing AODV in the network. Here we have also to include autoconfiguration to support IPv6 addressing autoconfiguration.

The first set of tests address the indoor emulated topology. In this scenario, we first measure the maximum available throughput without losses. These results are presented in Table 1 as a function of the number of hops between the sender and the receiver. As expected, the throughput decreases with the number of hops. It can be seen that for one- and two-hop counts the achieved throughput with AODV is below the presented throughputs for MMARP. This is because traffic is sent to the MIG which is one hop away from the gateway. Since it is sent directly, no join messages need to be issued and hence we save bandwidth. This effect is greatly attenuated for the remaining hop counts, as MMARP's overhead is substantially bigger than AODV's.

The second set of tests evaluates the packet delay (Table 5) and jitter (Table 6) of the different traffic profiles flowing between the ad hoc network and the infrastructure. Both delay and jitter values slightly increase with the increase of the flows' bandwidth and with the number of hops. It can be observed that for the 5th hop in the 256 kbps traffic profile, the delay introduced by AODV is higher than the one introduced by MMARP. This is related to the way the protocols work. On one hand, MMARP discards packets which cannot be delivered, but on the other hand, AODV buffers them, hence introducing more delay.

Table 7: Unicast routing and autoconfiguration: indoor and outdoor throughputs.

Hops	Outdoor (Kbps)	Indoor (Kbps)
1 Hop	1223	1222
2 Hops	432	559
3 Hops	258	322

The overhead introduced by the AODV and autoconfiguration protocols is of 2.38% per hop with 64 Kbps of traffic in the network, which is similar to the one of MMARP.

This means that the additional overhead introduced by AODV alone is of 0.94% for a 64 kbps traffic profile. This value was obtained by reducing the overhead of GW_INFO alone (obtained in Section 5.1) to the 2.38% of cumulative overhead presented in this section.

This scenario was further used to evaluate the impact of using the indoor or the outdoor topology. Table 7 shows the maximum throughput achieved in the same test conditions for both topologies. We notice that there is no large difference between the indoor or outdoor topologies, except a throughput decrease with the number of hops for values slightly smaller with the outdoor topology. This seems to result in the opposite to what would be expected, since the outdoor topology would reduce the radio interference. This fact is related with the increase of the nodes' distance between them, which introduces more errors and reduces the payload throughput.

For the first hop throughput, the connection is established between two nodes. This induces a big similarity between the conditions for both scenarios. For this case there is no interference caused by other nodes and hence the throughputs presented for both indoor and outdoor topologies do not differ significantly. Similar results were also obtained for delay and jitter, as well as for autoconfiguration only. Based on these results, we decided to use the indoor topology to run the remaining tests.

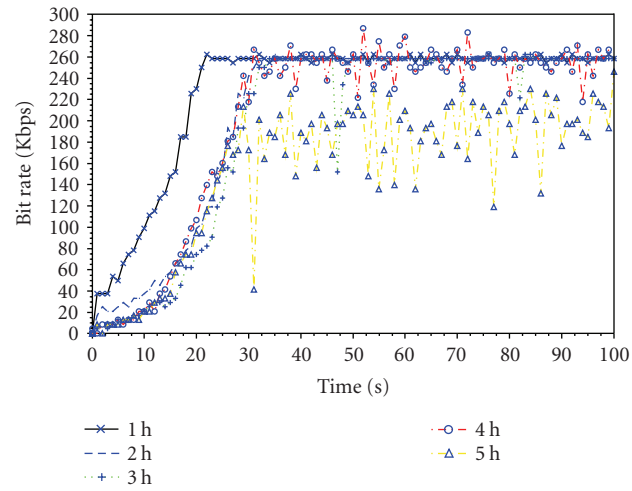

Figure 8: "Less demanding real time" rate variation.

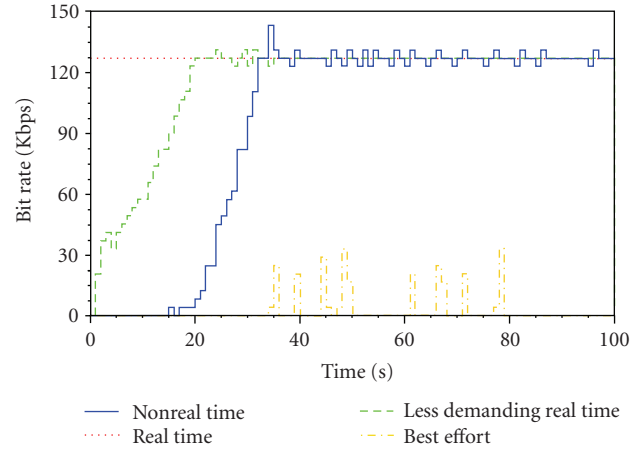

Figure 9: QoS initial setup differentiation for the first hop connection.

5.4. QoS

In this section, we summarize results obtained when traffic control and differentiation modules are activated in the nodes. In terms of traffic control, we evaluate the maximum achievable throughput and the influence of the number of hops in the ad hoc network between the sender and receiver. The maximum throughput decreases with the number of hops: its value decreases from 1.2 Mbps (one hop) to 120 Kbps (5 hops), similar to the previous results.

Figure 8 presents the rate of the less demanding real-time class (class with priority just below the real-time) in communications with different number of hops between sender and receiver. In all cases, the flow bandwidth is 256 Kbps, and it starts at time 0 seconds. First, we notice that, in all cases, the requested rate is achieved after a significant amount of time (between 30 and 40 seconds). This behaviour is introduced by the AIMD shaper that linearly increases the maximal transfer rate when no congestion is noticed in the network. Note that this would create strong problems for traditional TCP traffic. Second, we observe that the rise of the curve decreases with the increase in the number of hops. This illustrates the influence of shaping also at intermediate nodes.

Figure 9 shows the classes differentiation when generating the same bit rate (128 Kbps in this case) for all classes, and starting all flows at the same time. In the order of decreasing priorities, we have real-time, less-demanding real-time, nonreal time and best effort. We observe that real-time class starts at its maximum rate and lower classes take more time to reach the required throughput (time increases with decreasing priority). In the extended SWAN model, the real-time traffic class does not have any shaper and initiates its service at the maximum rate, since its usage has absolute priority. Best effort class uses the remaining bandwidth, which in this case is almost none.

Through the previous results we conclude that the extended SWAN model is able to support service differentiation and regulation of the flows. Unfortunately, the number

of hops in the ad hoc network has a large influence both in the maximum achievable throughput and in the time to achieve the requested rate. System behaviour is very dependent on the environment, even for such controlled tests as we have here reported. Typically tested uncontrolled situations, with mobility patterns, lead frequently to behaviours not easily understandable in terms of service differentiation.

The cumulative overhead for real time classes is of 2.11%, 1.20%, and 0.67%, respectively, for the 64 kbps, 128 kbps, and 256 kbps traffic profiles. Again, these values are similar to the ones of unicast and multicast routing. For the target multimedia services we want to deploy, only Real Time seems not to compromise network performance, so remaining integration tests will only concentrate on the analysis of this traffic class, even when QoS modules are active. For that reason, we do not present values on delay and jitter in this subsection: these parameters are not influenced by the QoS modules under these conditions.

5.5. Charging and rewarding

Finally, we evaluate the performance of the charging and rewarding mechanism (PACP [38]), the last remaining feature. We address here the overhead resulting of charging procedure only, and other aspects will be tackled on the complete multiservice analysis in next section. In the scenario used, the flows are sent from Node 5 to Node 1 (the gateway). PACP reports and PACP proofs generate almost the same rates of control bytes. However, PACP reports are sent in bursts every 37 data packets (each report contains the proof of 37 packets), while PACP proofs are of constant size in all packets (48 or 88 bytes). We notice these results (please refer to Table 8) are dependent on the packet rate, which is due to the constant proof size and the constant number of reports issued per data packet forwarded. We also see that the number of control bits introduced in the network will increase linearly as the number of packets increases. The overhead is presented for two distinct situations: with and without security processing, with the latter situation a bound on the performance of a mobile node with a cryptographic co-processor. We used elliptic curve digital signature algorithm (ECDSA) as the cryptographic algorithm, and it is our belief that this choice will only be realistic if special hardware exists in the (low-power) ad hoc nodes, due to the high computational requirements required for this.

Naturally the inclusion of security mechanisms increases the overhead of the charging and rewarding protocol. A real node with cryptographic coprocessor would have a performance in the middle of the two curves presented.

6. IMPACT ON THE USAGE OF AD HOCS AS STUB NETWORKS

When interconnecting an ad hoc network to an operator network and providing it with a set of functionalities and services, some performance drawbacks are to be expected.

After the evaluations performed in the last section it seems that the overhead introduced by the autoconfiguration functionality seems reasonable, since this feature is essential on a stub ad hoc network.

Table 8: Charging overhead (in Kbits) versus bitrate and usage of cryptographic mechanisms.

Overhead	64 Kbps	128 Kbps	256 Kbps	Average (%)
No ECDSA	10.98	21.96	43.90	17.15%
With ECDSA	16.33	32.67	65.33	25.52%

Multicast routing is of interest to our scenario in order to optimize resources when delivering typical broadcast services (streaming multimedia contents, such as audio and video). However, experiences have shown that ad hoc multicasting, in real scenarios, should be carefully considered. Still, for a five-hop connection, although there is a significant percentage of loss, delay is of 253 ms. Delay is not particularly relevant for the targeted services, but jitter impacts the size of the cache that must be reserved at the terminals. Jitter is 24 which is considered to be acceptable. As expected, all values increase with bandwidth and hops count.

When performing unicast routing tests, it was clear that its addition introduces some performance penalty in terms of delay, jitter, and overhead. Considering that voice communications require a jitter and delay lower than 50 ms and 150 ms, respectively, one could expect to be able to use all traffic profiles in a voice call, except the one of 256 kbps for a 5-hop connection. Here we already point a figure to the maximum number of hops allowed in an ad hoc network. The autoconfiguration and routing functionalities introduce a total cumulative overhead of 2.38% for the same traffic profile as shown in Figure 12, which is still acceptable.

Despite its apparent adequateness for multiservice networks, the QoS control mechanism used in ad-hocs is inefficient unless traffic belongs to the Real Time class (no shaping). Shaping of other classes takes tens of seconds to achieve maximum throughput, which is obviously inadequate. TCP connections, for instance, would not easily live under these circumstances. In normal usage (e.g., http traffic) this impairs QoS support in a multiservice network. Thus only priority traffic (e.g., voice) is able to be usefully differentiated from other best effort traffic, in these stub networks with a cost of a little percentage bandwidth.

For the complete multiservice network, with unicast routing, autoconfiguration, Real-Time, and charging control active, we expect jitter and delay values to increase. Tables 9 and 10 depict the jitter and delay in this network. We observe that, without security methods, the values are slightly increased when compared to the ones using only routing, as a consequence of the packet processing and inclusion of proofs (as discussed, real-time QoS impact is negligible). Also, because PACP is implemented in user space, an additional context switch must be performed, as packets flow between kernel and user space. PACP directly controls buffering and queuing mechanisms. When not considering cryptographic authentication, PACP control leads to more regulated traffic output, which slightly improves network behavior under congestion. The 256 kbps test for 5 hops (Table 9) leads to a heavily congested network (much larger than the

Table 9: Delay both with and without data authentication Unicast routing, autoconfiguration, Real-Time, and PACP are active.

Delay (ms)	64 Kbps	12 Kbps	256 Kbps
1 Hop	6.27	5.20	5.36
2 Hops	9.15	9.18	10.50
3 Hops	14.92	15.07	18.66
4 Hops	23.28	21.26	246.82
5 Hops	31.71	32.87	1106.62
Delay w/ECDSA (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	10.08	10.09	11.16
2 Hops	18.20	18.24	18.39
3 Hops	26.74	27.00	33.37
4 Hops	35.77	35.48	592.29
5 Hops	53.62	67.15	1702.05

Table 10: Jitter both with and without data authentication Unicast routing, autoconfiguration, Real-Time, and PACP are active.

Jitter	64 Kbps	128 Kbps	256 Kbps
1 Hop	0.47	0.60	0.74
2 Hops	0.50	0.57	0.83
3 Hops	0.50	0.66	0.92
4 Hops	1.16	0.97	15.68
5 Hops	1.26	2.13	21.22
Jitter w/ECDSA (ms)	64 Kbps	128 Kbps	256 Kbps
1 Hop	0.50	0.62	0.95
2 Hops	1.00	1.01	1.09
3 Hops	1.02	0.95	5.79
4 Hops	1.43	1.33	16.90
5 Hops	1.34	7.36	26.62

throughput), situation where PACP queue management actually leads to an improvement of performance.

Enabling PACPs' cryptographic authentication methods significantly increase the one-way delay. For each packet sent into the network, it will be signed once (by the sender), verified once (by the receiver) and also verified by all forwarding nodes (if any). Producing ECDSA signatures is not expensive (below 1 ms); however, verifying them has some cost. The tests performed in the testbed show that verifications take between 3 and 5 ms on the current hosts. Notice that these values are valid using ECDSA 163 bits and other key sizes will change this processing time. In a 5 hop scenario, a packet is verified 5 times and signed once. The minimum delay added to packets due to ECDSA will then be approximately between 20 and 25 ms. The real value measured in this scenario is 21.91 ms, which is according to the expected. This additional delay resulted by reducing the values obtained without ECDSA to the values obtained with ECDSA for the 5th hop corresponding to the 64 Kbps traffic class (refer to Table 9).

Figures 10, 11, and 12 show results for delay, jitter, and overhead, structured according to incremental addition of modules, for some scenarios. The remaining equivalent

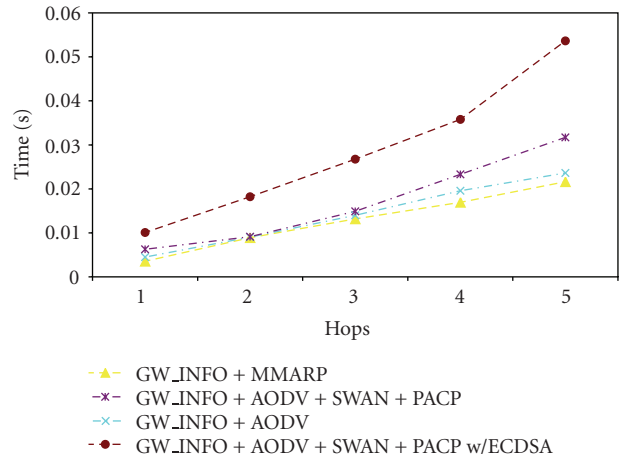

Figure 10: Cumulative delay for the 64 kbits traffic profile.

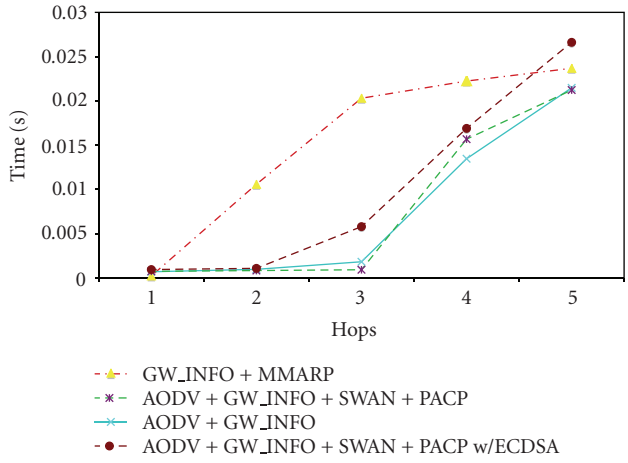

Figure 11: Cumulative jitter for the 256 kbits traffic profile.

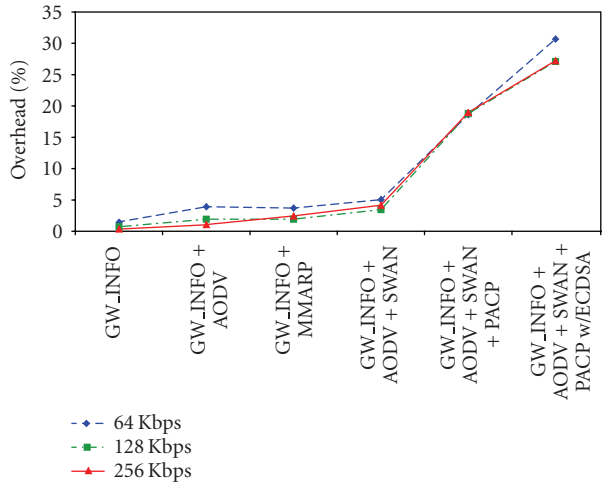

Figure 12: Cumulative overhead with the increase of functionalities.

results, for other traffic profiles, present similar performances, and all considerations below are generally valid for the tests performed. We can observe that the main delay source is the charging and rewarding mechanisms, and more specifically the security mechanisms introduced. All the other mechanisms do not significantly impact on the delay in the network. The processing introduced by MMARP for the multicast routing and by the verification of the signatures in PACP is significant, increasing the variation of the delay achieved by the data packets and hence, considering jitter, we observe that both MMARP and PACP with ECDSA introduce the higher penalties.

Finally, we observe that the increase in the overhead is also mainly due by charging and rewarding capabilities and its security mechanisms. The inclusion of an ad hoc network in the operator environment requires that some significant control information is introduced in the network to enable the "revenue" from the ad hoc network deployment. However, this significantly extra overhead supports security and charging/rewarding mechanisms, and therefore, the operator needs to balance all these issues. The results may suggest that other charging and control mechanisms should be researched for commercial networks.

7. CONCLUSIONS

This paper shows the measured effects of introducing several functionalities into ad hoc networks serving as stubs for multiservice networks. These results are part of a much larger work being performed for the integration of ad hoc networks in extended hotspot scenarios.

The results obtained, overlaying multiple ad hoc networks functionalities (unicast and multicast routing, self-configuration of gateways, QoS, charging) in a very simple scenario raise several concerns. A very basic concern was the overall complexity of the software to be deployed in the nodes, and the large number of potential interactions. This makes the system quite prone to errors, and raises some interoperability concerns in a commercial environment with multiple software providers.

Of a wider conceptual concern, we found a large behaviour variability, when routes are changing and QoS mechanisms are trying to regulate the network. In fact, it seems hard to expect a stable, smooth, behaviour of such a mobile network. For small mobility scenarios, the effective usage of ad hoc networks seems not to go further than a couple of hops, as already seen in studies focusing in single features.

The incremental addition of software modules showed the tradeoffs that an operator needs to face when adding extra functionalities to its network, namely the impact that trust and QoS have on network performance. Overhead values in multiservice ad hoc networks become large when imposing trust in the communications, and communications are throttled as soon as QoS regulation is taking place. These results show that a carefully scenario analysis should be developed before deploying ad hoc stubs in any multiple-service network: not all of features will be effective in complex environments.

In our opinion, using ad hoc as stub networks, the so-called "extended hotspot scenario," introduces an interesting concept and results show that the operators' network coverage can be extended for a few number of hops. This number may vary according to the mechanisms that the operator chooses to deploy, but will nevertheless be small if voice-alike services are considered. A full functional stub network can support all features presented before, and still be able to maintain an acceptable performance with delays lower than 50 ms and jitters lower than 10 ms for a maximum of two hops.

ACKNOWLEDGMENTS

The work presented in this paper was partially funded by the EU project IST-2002-506997 "Daidalos" [17]. Authors would like to thank the anonymous reviewers' comments, which much helped improving this paper.

REFERENCES

[1] C. E. Perkins, E. M. Belding-Royer, and S. Das, "Ad hoc on Demand Distance Vector (AODV) Routing," IETF experimental RFC 3561, July 2003.

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

[3] D. Johnson, D. maltz, and Y.-C. Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet Draft, draft-ietf-manet-dsr-10.txt.

[4] I. Chakeres and C. Perkins, "Dynamic MANET on-Demand (DYMO) Routing," IETF Internet Draft, draft-ietf-manet-dymo-06.txt, October 2006.

[5] T. Aura, "Cryptographically Generated Addresses," IETF RFC 3972, March 2005.

[6] R. Wakikawa, J. Malinen, C. Perkins, A. Nilsson, and A. Tuominen, "Global connectivity for IPv6 Mobile Ad Hoc Networks," IETF Internet Draft, draft-wakikwa-manet-globalv6-05.txt, March 2006.

[7] C. Jelger and T. Noel, "Gateway and address autoconfiguration for IPv6 adhoc networks," IETF Internet Draft, draft-jelger-manet-gateway-autoconf-v6-02.txt, April 2004.

[8] J. Jeong, J. Park, H. Kim, H. Jeong, and D. Kim, "Ad Hoc IP Address Autoconfiguration," IETF Internet Draft, draft-jeong-adhoc-ip-addr-autoconf-06.txt.

[9] C. Perkins, "IP Address Autoconfiguration for Ad Hoc Networks," IETF Internet Draft, draft-ietf-manet-autoconf-01.txt, November 2001.

[10] A. Laouiti, "Address autoconfiguration in Optimized Link State Routing Protocol," IETF Internet Draft, draft-lauoiti-manet-olsr-address-autoconf-01, January 2006.

[11] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: the incredibles," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 4, pp. 50–61, 2005.

[12] "Network Simulator—ns-2," February 2006, http://www.isi.edu/nsnam/ns/.

[13] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla, "GloMoSim: a scalable network simulation environment," Tech. Rep. 990027, Computer Science Department, UCLA, Los Angeles, Calif, USA, 1999.

[14] E. Nordström, P. Gunningberg, C. Rohner, and O. Wibling, "Comparing simulation, emulation, and real-world experimental results in mobile ad hoc networks," in *Proceedings of the 6th Scandinavian Workshop on Wireless Ad-Hoc Networks (ADHOC '06)*, Stockholm, Sweden, May 2006.

[15] D. Bansal and H. Balakrishnan, "TCP-friendly congestion control for real-time streaming application," Tech. Rep. MIT-LCS-TR-806, MIT Laboratory for Computer Science, Cambridge, Mass, USA, May 2000.

[16] S. Sargento, T. Calçada, J. P. Barraca, et al., "Mobile ad-hoc networks integration in the DAIDALOS architecture," in *Proceedings of the 14th IST Mobile & Wireless Communications Summit*, Dresden, Germany, June 2005.

[17] Daidalos ISTProject: Daidalos, (FP6-2002-IST-1-506997). http://www.ist-daidalos.org/.

[18] MIT RoofNet, http://www.pdos.csail.mit.edu/roofnet/doku.php.

[19] Orbit Testbed, http://www.winlab.rutgers.edu/pub/docs/focus/ORBIT.html.

[20] Microsoft Networking Research Group, http://www.research.microsoft.com/mesh/.

[21] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, Eds., *Dynamic Host Configuration Protocol for IPv6*, IETF RFC 4361 , July 2003.

[22] T. Calçada and M. Ricardo, "Extending the Coverage of a 4G Telecom Network using Hybrid Ad-hoc Networks: a Case Study," MED-HOC- NET, June 2005.

[23] E. Royer and C. Perkins, "Multicast Ad hoc on-Demand Distance Vector (MAODV) Routing," IETF Internet Draft, draft-ietf-manet-maodv-00.txt, July 2000.

[24] P. Jacquet, P. Minet, A. Laouiti, L. Viennot, T. Clausen, and C. Adjih, "Multicast Optimized Link State Routing," IETF Internet Draft, draft-jacket-olsr-molsr-00.txt, IETF Internet Draft, November 2001.

[25] Y. Yi, S. Lee, W. Su, and M. Gerla, "On-Demand Multicast Routing Protocol (ODMRP) for Ad Hoc Networks," IETF Internet Draft, November 2002.

[26] J. Jetcheva and D. Johnson, "The Adaptive Demand-Driven Multicast Routing Protocol for Mobile Ad Hoc Networks (ADMR)," IETF Internet Draft,, July 2001.

[27] P. M. Ruiz, A. Gomez-Skarmeta, and I. Groves, "The MMARP protocol for efficient support of standard IP multicast communications in mobile ad hoc access networks," in *Proceedings of the IST Mobile & Wireless Communications Summit*, pp. 478–482, Aveiro, Portugal, June 2003.

[28] B. Cain, S. Deering, I. Kouvelas, and B. Fenner, "Internet Group Management Protocol, version 3," IETF RFC 3376, October 2002.

[29] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: an IP-based quality of service framework for mobile ad hoc networks," *Journal of Parallel and Distributed Computing*, vol. 60, no. 4, pp. 374–406, 2000.

[30] H. Badis and K. Agha, "Quality of service for Ad hoc Optimized Link State Routing Protocol (OLSR)," IETF Internet Draft: draft-badis-manet-qolsr-03.txt.

[31] G.-S. Ahn, A. T. Campbell, A. Veres, and L.-H. Sun, "Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN)," *IEEE Transactions on Mobile Computing*, vol. 1, no. 3, pp. 192–207, 2002.

[32] S. Crisóstomo, S. Sargento, M. Natkaniec, and N. Vicari, "A QoS architecture integrating mobile ad-hoc and infrastructure networks," in *Proceedings of the 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA '05)*, pp. 897–903, Cairo, Egypt, January 2005.

[33] S. Zhong, J. Chen, and Y. R. Yang, "Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *Proceedings of the 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03)*, vol. 3, pp. 1987–1997, San Francisco, Calif, USA, March-April 2003.

[34] N. B. Salem, L. Buttyán, J.-P. Hubaux, and M. Jakobsson, "Node cooperation in hybrid ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 4, pp. 365–376, 2006.

[35] B. Lamparter, K. Paul, and D. Westhoff, "Charging support for ad hoc stub networks," *Computer Communications*, vol. 26, no. 13, pp. 1504–1514, 2003, special issue on Internet Pricing and Charging: Algorithms, Technology and Application.

[36] J. Girão, B. Lamparter, D. Westhoff, R. L. Aguiar, and J. P. Barraca, "Linking ad hoc charging schemes to AAAC architectures," in *Proceedings of the 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS '04)*, Heidelberg, Germany, August 2004.

[37] J. Girão, J. P. Barraca, B. Lamparter, D. Westhoff, and R. L. Aguiar, "QoS-differentiated secure charging in ad-hoc environments," in *Proceedings of the 11th International Conference on Telecommunications (ICT '04)*, pp. 1093–1100, Fortaleza, Brazil, August 2004.

[38] J. P. Barraca, S. Sargento, and R. L. Aguiar, "The polynomial-assisted ad-hoc charging protocol," in *Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC '05)*, pp. 945–952, Murcia, Spain, Junes 2005.

[39] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6," IEFT RFC 2775, June 2004.

[40] F. Galera, P. Ruiz, and A. Goméz-Skarmeta, "Security Extensions to MMARP through Crytographicaly Generated Addresses".

[41] Uppsala University, "Ad Hoc Implementation Portal," February 2006, http://core.it.uu.se/AdHoc.

[42] K. K. Vadde and V. R. Syrotiuk, "Quantifying factors affecting quality of service in mobile ad hoc networks," *Simulation*, vol. 81, no. 8, pp. 547–560, 2005.

[43] MacKill Tool, June 2006, http://www.apetestbed.sourceforge.net/.

[44] "MGEN: The Multi-Generator Toolset," June 2006, http://www.pf.itd.nrl.navy.mil/mgen/.