Experimental Evaluation of an Integrated Ad-Hoc Network

João Paulo Barraca, Miguel Almeida, Rafael Sarrô, Susana Sargento , Rui L. Aguiar
{jpbarraca,malmeida,rsarro}@av.it.pt, {ssargento,ruilaa}@det.ua.pt

Universidade de Aveiro, Instituto de Telecomunicações, Portugal

***Abstract*— This paper presents an experimental evaluation of an Ad-hoc network architecture integrated with the infrastructure network, developed in the framework of the IST Daidalos project. This architecture supports the efficient delivery of services, unicast and multicast, legacy and multimedia, to users connected to the ad-hoc network. It contains functionalities of routing and mobility to enable mobility of users inside and between ad-hoc networks, distributed QoS mechanisms to support service differentiation and resources control responsive to nodes mobility, and 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 the proposed mechanisms, and the influence and performance penalty introduced in the architecture, with the incremental inclusion of the proposed mechanisms.**

***Index Terms*—Ad-hoc, Integration, evaluation, performance penalty**

I. Introduction

One of the most important objectives of the IST-Daidalos project [1] is to integrate a number of technologies under a common architecture. These technologies are much diversified covering fields like Broadcast (DVB-T/S/H), Broadband and Metropolitan networks (802.16), Mobile Cellular networks (TD-CDMA), Wireless Access networks using infrastructure and ad-hoc paradigms (802.11), or Sensor networks (802.11, Bluetooth). Integrating all these technologies requires the creation of a complex yet efficient architecture. Daidalos project also aims at integrating new services provided by the enhanced business models allowed by a unified architecture. Ad-hoc networks are integrated as a mean to extend radio coverage from wireless hotspots supporting the Daidalos technologies. The multi-hop characteristic of a Mobile Ad-hoc NETwork (MANET) is much useful in providing the increased range. New issues, non existent on standard ad-hoc networks, are now introduced.

Different network operators can provide services in the same place, providing concurrent access methods. Therefore, users expect to discover and connect to the existing networks in a seamless and automatic manner. This implies discovery of the available networks and associated services, as well as auto-configuration of network parameters. Such users have a contract with one or more network operators and expect the terms of the contract to be fulfilled. Terms may include values like bandwidth quotas, service availability, and QoS parameters. Network and service operators expect to profit from the infrastructure and services provided. Specifically this implies the deployment of proper mechanisms to monitor and charge traffic the usage and service consumption.

This paper addresses the experimental evaluation of an integrated ad-hoc architecture that supports the delivery of a large diversity of services, unicast and multicast, legacy and multimedia, to users connected to the ad-hoc network. The services will be delivered with the required quality, which will both depend on the services and user requirements, and the users will be charged for the requested services and motivated to cooperate in the service delivery. Moreover, users may move between ad-hoc networks, and the provision of a seamless mobility is required. Notice that, although some of the functionalities (similar) were already separately addressed, implemented and evaluated, there is still no study that incorporates all these features simultaneously in a single ad-hoc network. The aim of this paper is three-fold: describe the integrated ad-hoc architecture and its new developed functionalities, experimentally evaluate the performance of the proposed mechanisms, and evaluate the influence and performance penalty introduced in the architecture, with the incremental inclusion of the proposed mechanisms.

The paper is organized as follows. Section II presents the general architecture, and section III addresses the proposed ad-hoc functionalities. The description of the Ad-hoc network testbed deployed in the framework of the Daidalos project is performed in section IV, and the results achieved are depicted in section V. Finally, the main conclusions and future plans are addressed in section VI.

II. Ad-Hoc Integration Network Architecture

Figure 1 depicts the architecture of the ad-hoc network in the extended hotspot scenario. It is composed by ad-hoc nodes connected to the access network through a multi-hop path composed by mobile ad-hoc nodes. We consider that the target mobile nodes (MN) in this network are laptops and personal digital assistants (PDA).

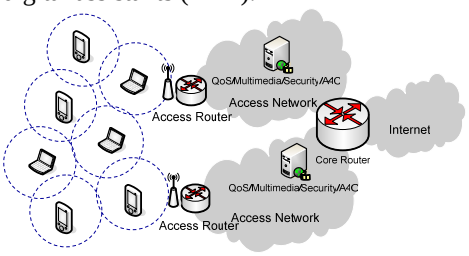

Figure 1: Daidalos ad-hoc network architecture

Inside the ad-hoc network, the traffic is routed through unicast (Ad-hoc On demand Distance Vector routing – AODV [8]), multipath (AO Multipath DV – AOMDV [9]), or multicast (Multicast MANET Routing Protocol – MMARP [10]) routing protocols. The ad-hoc network is connected to

the infrastructure network through an Access Router (AR). This element is a node (fixed router belonging both to the infrastructure and to the MANET) that routes packets between the external networks and the ad-hoc cloud, and provides the interface to the infrastructure network, in terms of routing, mobility, QoS, security and charging procedures.

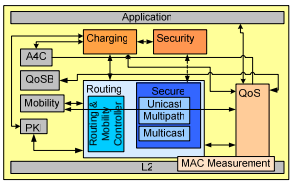

Figure 2: MN and AR general architecture

Figure 2 depicts the general architecture of the MNs and the AR, in terms of its elements (colored) and its interactions with infrastructure elements or non-ad-hoc specific elements (in grey). Notice that the application box is only available in the MN. In the next section we will address in more detail each one of the ad-hoc modules and its integration aspects, to provide the efficient and scalable integration of routing and mobility, QoS, security and charging and rewarding mechanisms, between ad-hoc and infrastructure networks.

III. AD-HOC NETWORK FUNCTIONALITIES

This section presents an overview of the functionalities developed in the ad-hoc integration network, referring to the modules depicted in Figure 2. For more information on the integration of all functionalities refer to [2]. For more individual information refer to [4][10][12][16].

The Routing & Mobility Controller manages the overall routing procedure and address configuration. Located at the Gateway this module periodically broadcasts a packet (GW_Info) with network prefix information, an indication of the distance to the gateway and some other useful information [3][4]. If the node is not connected, it learns the existence of the new network, chooses the network with the lowest hop count, and configures a proper address to start communicating. If the node is already connected, it forwards the message to its 1 hop neighbors.

After the reception of a GW_Info message, a connected node may choose to change to a neighbor network. An inter-ad-hoc mobility protocol was proposed, which results from the integration with the Fast Mobile IPv6 [7] and Context Transfer solutions, to support seamlessly handover to the new access network without need for sessions' re-establishment [2]. This functionality is currently supported in the ad-hoc architecture; as expected, the multi-hop characteristic of ad-hoc networks increases the delays of handovers.

The routing is provided both for unicast and multicast flows. Unicast routing is provided by the AODV protocol [8] for IPv6 and is able to route messages in a standard approach using single routes or, in a more resource efficient manner, choosing the best route for each flow through multipath routing [9]. In order to increase the efficiency of routing and avoid the stack of different unicast routing solutions, AOMDV was chosen to fulfill this task. Multicast routing is provided by the MMARP [10] protocol, which is a new multicast ad-hoc routing protocol that interoperates with fixed IP networks. The interoperation with the ARs is performed by the Multicast Internet Gateways (MIGs) which are the ad-hoc nodes situated just one hop away from the AR. The MMARP protocol is extended to interwork with the mechanism used for GW discovery and address auto-configuration previously described. It further allows the GW to inform all ad-hoc nodes about the path towards multicast sources in the fixed network. All routing protocols were modified in order to avoid common attacks usually found in their original proposals [5][13][14].

In order to allow the QoS interoperation among ad-hoc and infrastructure networks, the base SWAN [11] proposal was adapted and extended [12]. SWAN signalling was adapted to interoperate with infrastructure QoS signalling based admission control, and to support multipath probing. The differentiation model was extended to support four classes of service and congestion feedback between each other. To provide the QoS interworking, a GW interconnecting the ad-hoc with the infrastructure network has the required functions related to the mapping of QoS functions.

The proposal for an extended differentiation model considers four different traffic classes: critical real-time traffic, less demanding real-time traffic, non real-time traffic and regular best-effort traffic. Each of these classes will have assigned a certain amount of bandwidth, except the best-effort that serves as a "buffer zone" or absorber for higher priority traffic bursts introduced by mobility. Figure 3 presents the differentiation model composed by a classifier and by a cascade of priority schedulers, shapers and queues associated to each traffic class. The limited access delay to higher priority traffic is achieved by every node giving priority access to this traffic and using the measured MAC delay (all packets) as feedback to control the rate of lower priority traffic, therefore controlling the shared medium load. The limited delays are applied through a leaky bucket shaper, whose rate is controlled by an AIMD (Additive Increase Multiplicative Decrease) algorithm having the lower level classes delay as feedback.

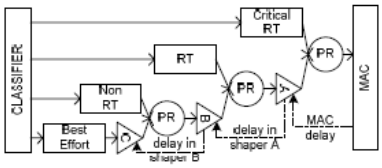

Figure 3 – Extended differentiation model

Since Daidalos infrastructure is driven by operator expectations and business models, it is imperative to have a proper support for charging the users. The operators need to be able to have profit from development of the network and services. The multi-hop and distributed nature (and instability) of MANET requires the existence of distributed charging mechanisms. Most important, these mechanisms need to be compatible and integrated with the A4C architecture already deployed. Ad-hoc networks also require incentives to users to participate in the forwarding process. Such incentives can be provided in many forms, like, for example, credit or service discounts. The developments present in the state of the art address these issues by creating a distributed mechanism [15]

actively marking packets with a proof which is updated at each forwarding node and then reported to the network operator. The proofs are built and updated using a defined set of rules and supported by cryptographic signing and verification primitives. Since this mechanism requires that all packets include the list of forwarding nodes, which increases the network overhead, a new mechanism was proposed [16] that encodes the route in a polynom, which terms and values (fixed size elements) are 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, like individual user profile, service description, QoS parameters, route length, time frame or data amount.

IV. DAIDALOS AD-HOC TESTBED

The Daidalos Integrated Ad-hoc testbed is comprised of several Linux computers running the modules developed accordingly to the architecture previously specified inside the project. All machines have, at least 1.2Ghz CPU and 256Mb RAM, and enough storage space. They do not reflect typical, resource limited, ad-hoc nodes, but are more suited to the required extensive testing. Mandrake 10.0 Official was selected as the official distribution for the entire project and was also used in this testbed. The kernel used was the vanilla 2.6.8.1 with some additional modifications required by some of the tested modules. These extensions are the following: support for DSCP marking using Netfilter, the Hostap wireless driver, a Netlink multiplexer, an IP6_QUEUE Multiplexer, support for Token Bucket Queues, the Mobile IPv6 RC2 stack [7] and a customized version of MACKILL. With the exception of the Mobile IPv6 stack and HostAP driver, all additional functions were developed inside the Daidalos project. Some of these modules operate in Kernel space while others are standard User space applications. The usage of Kernel space modules was limited in an attempt to make the developed modules portable and easy to deploy on different machines with different distributions and kernel versions.

All machines are equipped with 2 network interfaces; one wireless and one wired. The wired interface is used to remote access during the tests and to perform administrative tasks. Two of the nodes are used to interconnect the ad-hoc cloud with the infrastructure network; the wired interface will also be used to transfer data to or from the ad-hoc network. These nodes are referred in this document as the Gateway Nodes.
Wireless interfaces are comprised of Prism2.5 802.11b cards with the following configuration parameters: ad-hoc and promiscuous modes, channel 12, rate fixed to 2Mbits and RTS/CTS threshold of 1 byte. The ad-hoc network is limited to the wireless interfaces and protocols only operate on them. Figure 4 depicts the described ad-hoc testbed. Node1 is directly connected to Node2 which is also directly connected to Node3. Node4 connects directly to Node3 with the exception of mobility tests, in which case Node4's movement induces routing changes.

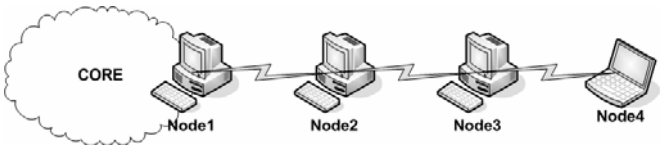

Figure 4: Integrated Ad-hoc network testbed

V. EXPERIMENTAL RESULTS

The results presented in this section are two-fold. From one side, we aim at addressing the performance of the mechanisms proposed and implemented in the integrated ad-hoc network. On the other side, we aim at evaluating the load of the mechanisms in the network, that is, the performance penalty in terms of maximum throughput achieved, overhead introduced and packets delay (and jitter), with the incremental inclusion of the mechanisms in the network, so that the influence of each module can be coherently weighted. To evaluate the influence of the size of the ad-hoc network in the performance penalty, we also change the number of hops for end-to-end conversations inside the ad-hoc network.

Each presented value is the result of the mean of 5 test runs.

A. Routing

Unicast and multicast routing behavior was the target of the first performance test. With the routing protocols active, we measured the throughput, delay, jitter and overhead.

1) Unicast Routing

As mentioned before, we used AODV for unicast routing.

Table 1 presents the maximum throughput achieved in the network without losses, as a function of the number of hops between the sender and receiver. As can be observed, throughput decreases by a factor of two with the increase in the number of hops. For a 3 hops network, only 340 Kbps are available for communication, which is an indicative that there exists a limit in the size of the ad-hoc network.

Table 1 – Throughput achieved with unicast routing

	Throughput (kbps)
1 Hop	1200
2 Hops	600
3 Hops	340

The delay and jitter increase linearly with the number of hops (Table 2). We notice that, under non heavy-loaded conditions, the absolute values are in the order of some msec, which does not compromise the communications in the ad-hoc network.

Table 2 – Delay and jitter achieved with unicast routing

	Delay (ms)	Jitter (ms)
1 Hop	1.96	3.30
2 Hops	4.04	6.62
3 Hops	7.73	11.68

In terms of overhead generated by the routing protocol, it is only 1.05% of the total traffic (traffic of 600 Kbps flowing from Node3 to Node1).

Below, we analyze the time required for routing re-configuration when mobility of the nodes is in place. In this test, traffic was generated from Node4 to the fixed terminal Node1, and the capture was performed in Node1. The mobile node was initially sending traffic through Node3 and, as it began its movement, it started receiving signal from Node1. When this happens, a route is created directly to Node1. This behavior can be observed in Figure 5, through the analysis of the existing gaps on the bit rate of received traffic. The mean value of the re-configuration time is 2.3 seconds. This value is in the order of some seconds because AODV is configured in such a way that, when a route changes, a node needs to receive 3 HELLOs to make sure that the route has really changed, before re-configuring its route.

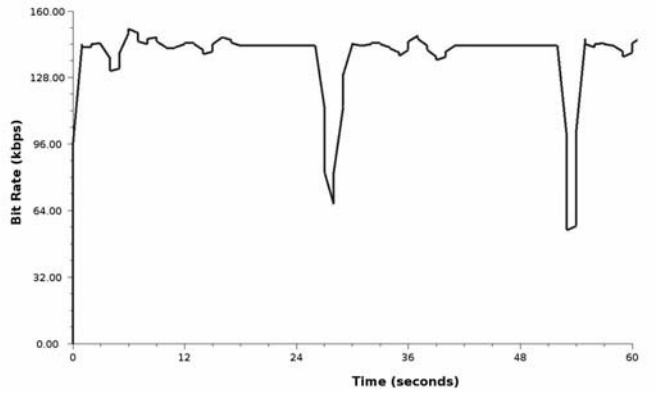

Figure 5: Throughput vs time in the presence of Node4 mobility

2) Multicast Routing

In order to test MMARP's performance, multicast traffic was generated from Node1 node and received at both Node4 and Node3. Figure 6 depicts the traffic rate received at both Node4 and Node3. We observe that, although Node4 is farther from the sender (in terms of number of hops) than Node3, they both receive the traffic at a similar rate.

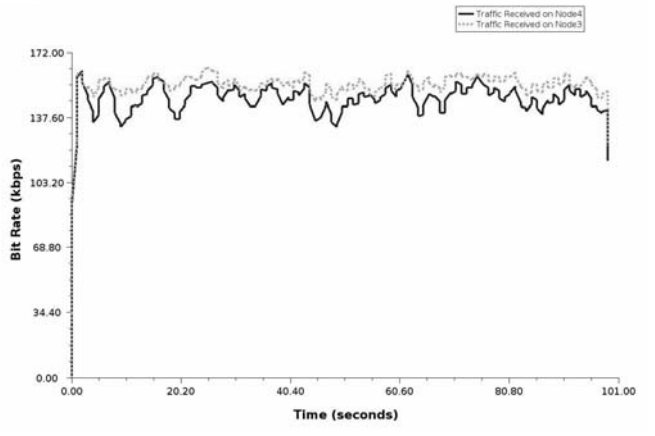

Figure 6: Traffic rate received at both Node3 and Node4 in multicast scenario

In terms of delay and jitter (Table 2), it can be seen that both values are slightly larger on the last hop node (Node 4) than on Node3.

Table 3 – Delay and jitter achieved with multicast routing

	Delay (ms)	Jitter (ms)
Node4	58.34	26.38
Node3	68.819	27.67

The overhead introduced by MMARP is 15.8%, which means that the inclusion of multicast in ad-hoc networks introduces a significant performance penalty.

B. Auto-configuration and Routing

Gateway advertisement is performed once every second to allow for the auto-configuration of ad-hoc nodes addresses. The setup scenario and experiment is similar to the one in sub-section A with unicast routing.

The throughput values achieved are the same as in unicast routing only. This means that the auto-configuration functionality does not introduce throughput penalty. Delay and jitter results are presented in Table 4. They are just slightly larger than the previous ones with routing.

Table 4 – Delay and jitter achieved with auto-configuration and unicast routing

	Delay (ms)	Jitter (ms)
1 Hop	2.04	3.33
2 Hops	4.13	6.66
3 Hops	7.99	11.78

The overhead introduced is 1.48%, also slightly larger than the previous one.

The configuration time is an average of 2 seconds. This time value is the time between the reception of the first GW_INFO message and the first GW_INFO message sent (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.

C. QoS

In this section we present the main results related to traffic control and differentiation.

In terms of traffic control, we address the maximum achievable throughput (regulated by the shaper) and the influence of the number of hops in the ad-hoc network between the sender and receiver. This study was performed through the generation of UDP CBR traffic. Figure 7 presents the maximum supported throughput in an intermediate class (the one just below real-time) for different number of hops. Note that in the extended SWAN model, the real-time traffic class does not have shaper and initiates its service at the maximum rate.

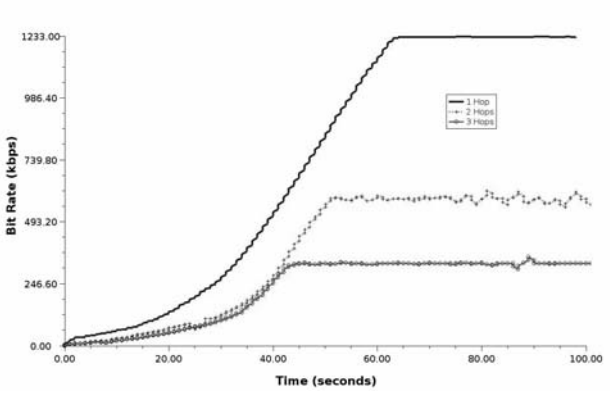

Figure 7: Intermediate class shaping and throughput

First, we notice that, in both cases, the maximum throughput is achieved after a significant amount of time (order of 50 sec). This behavior is introduced by the AIMD shaper that linearly increases the maximal transfer rate when no congestion is noticed in the network. This initial time can be significantly decreased with a proper tuning of the shaper parameters.

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 the intermediate nodes.

Finally, the maximum throughput also decreases with the number of hops. Its value decreases from 1.2 Mbps (one hop) to 340 Kbps (3 hops). These values are similar to the ones achieved in the previous sub-sections: QoS 'per se' is not decreasing throughput; it just increases the time to reach the maximum value.

The following figure shows the differentiation in terms of time required to achieve a specific throughput, when generating the same bit rate (100 Kbps in this case) for all classes and starting all flows at the same time. Class identified by DSCP 0x2a is for real-time traffic, 0x52 is the one just below the real-time one, and 0x7a is the one just above best effort. We observe that lower classes take more time to reach the required throughput.

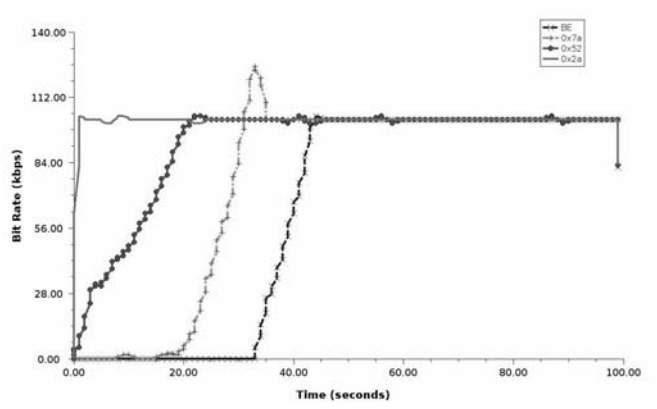

Figure 8: Initial set-up differentiation

Finally, we address the differentiation between traffic classes, as example, real-time and non real-time when the network is saturated (each flow requests 1.024 Mbps). We notice that the service achieved by the real-time traffic class is always better than the other one, although competition for resources is present.

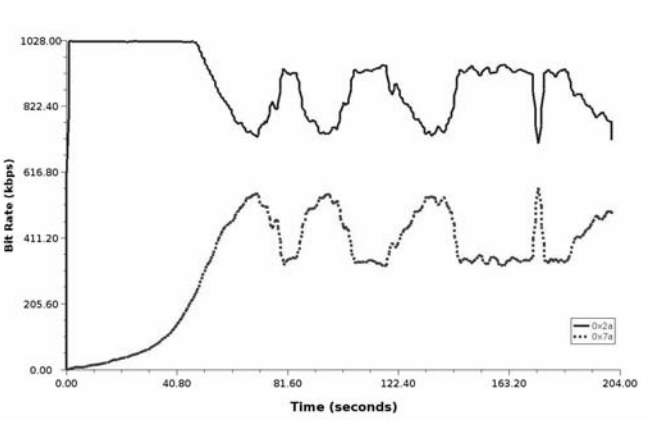

Figure 9: Differentiation in overload conditions

D. Charging and Rewarding

In this sub-section we evaluate the performance of the charging and rewarding functionality (PACP [REF_PACP]). In these tests we evaluate the overhead resulting of charging procedure and the error between the information reported and the actual packets received. Both parameters were evaluated by generating a UDP CBR flow of 250 kbit/s. The flow was sent from a node 3 hops inside the ad-hoc network to a infrastructure node connected to the ad-hoc Gateway. In this situation the PACP Charging Manager is collocated with the receiving node.

Figure 10 depicts the packets received at the receiver. Data flow keeps stable at the expected rate of 250 kbits/s. Overhead resulting from in-band packet proofs, PACP reports and PACP report acknowledges is also represented. PACP reports and proofs generate almost the same rate of control bytes. However, PACP reports are sent in burst every 37 data packets (each report contains the proof of 37 packets), while PACP proofs are constant in all packets. The resulting overhead is not constant varying between 35 and 45kbits/s. We shall notice that these results are dependent on the packet rate and not on the actual bandwidth of the data flows.

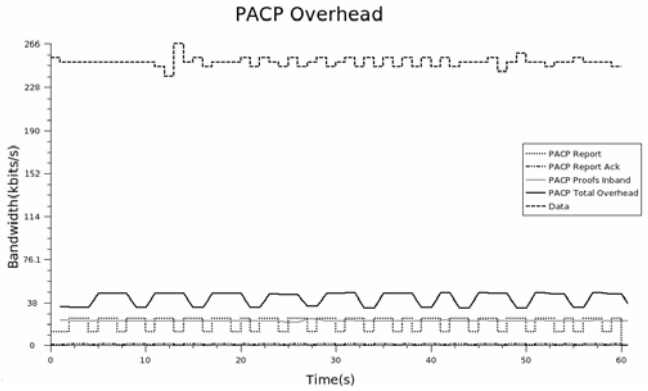

Figure 10: PACP Overhead with a UDP CBR flow

In ad-hoc network charging proposals, due to mobility and instability of the ad-hoc network, the actual rate of packets

charged may differ from the actual service consumed. In the case of PACP, if a packet is dropped after the proof is collected (between last hop and receiver), or if the route is large and change very frequently, charging results of PACP may suffer some deviation from the actual number of packets traversing the network. Dropped packets result in charging and rewarding a higher number of packets, and instability results in not rewarding all forwarded packets.

Table 5 presents the results of traffic traversing the network and the number of proofs received to charge the traffic. The total deviation is a results of 3 packets lost between the last hop and the receiving node. As the congestion increases, UDP flow will suffer from higher deviation. TCP flows will self adjust and contain the packet loss, resulting in a lower increase of the charging error.

Table 5 – Traffic in the network and proofs received

	Packets	Bytes
Received	3661	2050160
Reported	3664	2051840
Deviation	0.083%	0.083%

VI. CONCLUSIONS

This paper presented an experimental study of the Ad-hoc network integration architecture being developed inside the IST project Daidalos. This architecture is able to efficiently integrate ad-hoc and infrastructure networks, supporting unicast and multicast routing, QoS, security and charging mechanisms. Moreover, the architecture allows the mobility of users between ad-hoc networks.

The results obtained in terms of performance show that the network is able to fulfil its requirements in terms of service delivery, unicast and multicast, with its required quality, and correctly charge the users for the services accessed. In terms of performance penalty, the results show that the main bottleneck in the amount of useful traffic that can coexist in the network is the increase in the number of hops, and therefore, the increase in the ad-hoc network size.

Future plans include the support of multiple gateways in the ad-hoc network for load balancing purposes, and the mobility between ad-hoc, moving and infrastructure networks.

ACKNOWLEDGMENT

The work presented in this paper was partially funded by the EU project IST-2002-506997 “Daidalos” [1].

REFERENCES

[1] Daidalos IST Project: Daidalos: “Designing Advanced Interfaces for the Delivery and Administration of Location independent Optimised personal Services”. (FP6-2002-IST-1-506997). http://www.ist.daidalos.org.

[2] S. Sargento et al., “Mobile Ad-Hoc Networks Integration in the Daidalos Architecture”, In proc. of the IST Mobile & Wireless Comm. Summit 2003, June 2005.

[3] C. Jelger et al., “Gateway and address autoconfiguration for IPv6 ad-hoc networks” IETF Internet-Draft: draft-jelger-MANET-gateway-autoconf-v6-01.txt, Oct 2003.

[4] 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.

[5] J. Arkko, J. Kempf, “Secure Neighbour Discovery (SEND)”, IETF Internet-Draft: draft-ietf-send-ndopt-03, Jan 2004.

[6] T. Aura, “Cryptographically Generated Addresses (CGA)”, IETF Internet-Draft: draft-ietf-send-cga-06, Apr 2004.

[7] R. Koodli, “Fast Handovers for Mobile IPv6”, IETF Internet-Draft: draft-ietf-mipshop-fast-mipv6-03.txt, Oct 2004.

[8] C. Perkins et al., “Ad hoc On-Demand Distance Vector (AODV) Routing”. IETF experimental RFC 3561, July 2003.

[9] M. Marina, S. Das, “Ad hoc On-demand Multipath Distance Vector Routing”. Technical Report, CS Dep., Stony Brook Univ., April 2003.

[10]P. Ruiz et al., “The MMARP Protocol for Efficient Support of Standard IP Multicast Communications in Mobile Ad hoc Access Networks”. In proc. of the IST Mobile & Wireless Comm. Summit 2003, June 2003.

[11]G.-S. Ahn et al., “Supporting Service Differentiation for Real Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks.” In IEEE Trans. on Mob. Comp. vol. 1, no. 3, 2002.

[12]S. Crisóstomo, S. Sargento et al, “A QoS Architecture Integrating Mobile Ad-Hoc and Infrastructure Networks”, 3rd ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-05), January 2005.

[13]A. Yegin et al., “Protocol for Carrying Authentication for Network Access (PANA) Requirements”, IETF Internet-Draft: draft-ietf-pana-requirements-08.txt, June 2004.

[14]M. Zapata, “Secure Ad hoc On-Demand Distance Vector (SAODV) Routing”, IETF Internet-Draft: draftguerrero-manet-saodv-00.txt, Aug 2001.

[15]J. Girão, J. Barraca et al., “QoS-differentiated Secure Charging in Ad-hoc environments”, International Conference on Telecommunications, Aug 2004.

[16]J. P. Barraca et al., “The Polynomial-Assisted Ad-hoc Charging Protocol”, 10th IEEE Symposium on Computers and Communications (ISCC 2005), June 2005.