On The Limits of Ad-Hoc Networks: Experimental Evaluation

Miguel Almeida, Susana Sargento, João Paulo Barraca, Rafael Sarrô, Rui L. Aguiar
Instituto de Telecomunicações- Polo Aveiro, Universidade de Aveiro, Campus Universitário de Santiago, 3810-193 Aveiro, Portugal
Phone: +351-234 377900, Fax: +351-377901, e-mail: {malmeida,jpbarraca,rsarro}@av.it.pt, {ssargento,ruilaa}@det.ua.pt

Abstract— **This paper presents an experimental evaluation of a multi-service Ad-hoc network architecture, interconnected with a fixed 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 support for routing with delivery of multicast services, mobility, QoS mechanisms to support service differentiation and resource control responsive to node mobility as well as 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 network, with the incremental inclusion of new functionalities. Results, although biased by the software implementations, may question the usage of ad-hoc networks for more than a minimal number of hops.**

***Index Terms* — ad-hoc networks, experimental characterization, multi-service, QoS.**

I. INTRODUCTION

Mobile Ad-hoc networks (MANETs) are traditionally networks composed by the spontaneous grouping of nodes using wireless technologies and collaborating in order to provide communication facilities. These nodes are typically PDAs, laptops or even sensors (with limited battery, reduced processing and wireless capabilities). One node by itself with such characteristics is not capable of a large communication range. If they 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 there is only need for some address configuration mechanisms and routing protocols (for basic communication). Many proposals (e.g. [1], [2], [3], [4]) already cover both these topics and present resource efficient mechanisms to allow the creation of MANET. A multitude of other proposals also exist, covering different issues, mainly due to the high interest in self-organization networks, and to the requirement of solutions able to operate on resource constrained environments, such as sensor networks.

Many of these proposals are evaluated using simulation tools such as ns2 [5] or GloMoSim [6]. Some are further tested in real world environments where real issues concerning program concurrency or real wireless interference are present. In this paper we present the results obtained by deploying a multi-service ad-hoc network integrated in an infrastructure network eventually managed by a 4G operator. This corresponds to the often referred as “hotspot extension” scenario [7], where the ad-hoc network is able to provide external communication links, sharable by all users in the ad-hoc cloud.

Multiple services are required at each node for correct integration of the ad-hoc cloud within such an operator business architecture. More important, they 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 low resource environments. Particularly, it allows a better understanding of the inherent limitations of ad-hoc wireless networks and realistic expectations on features to be supported. Also, the limitations resulting from each solution or from the interactions presented by the several solutions are of vital importance. In our study, we addressed routing, unicast and multicast, address auto-configuration, QoS and charging mechanisms. We rely in software developed mostly inside the EU project Daidalos [8].

This paper is organized as follows. Section II presents the protocols we focused in our prototype network, and section III addresses the software implementation and the description of the relevant parts of the ad-hoc network testbed. The results achieved are depicted in section IV. Finally, the main conclusions are presented in section V.

II. AD-HOC PROTOCOLS FOR 4G SOLUTIONS

One of the scenarios of usage of ad-hoc networks is associated with the extension of wireless cellular infrastructures, in 4G scenarios [8], where a node will act as gateway. We addressed this scenario, which has led to a large number of previously published research works, and considered a set of technologies that could be exploited in this “futuristic” environment. Routing, auto-configuration with gateway information, QoS, and charging are some of the features we developed and integrated in our testbed.

For unicast routing purposes we used the Ad-hoc On-demand Distance Vector routing protocol (AODV) [1]. The base implementation was AODV-UU as provided by University of Upsala at [9]. Some modifications were required to the original implementation in order to fulfil some requirements from other modules, namely gateway information and charging. The gateway information module can change the address of the wireless interfaces in real time and the original version had no support for dynamic addresses. The charging module [10] requires information about the next hop for a given destination; while this information could be retrieved from routing tables, AODV is able to provide more information like alternative routes. These changes however are expected to have little or no impact in the resulting performance or operation of the AODV implementation.

Multicast routing is provided by the Multicast MANET Routing Protocol (MMARP) [11], which is a solution that interoperates ad-hoc Multicast routing with standard multicast used in fixed IP networks. One big advantage of this proposal is that end-nodes only need to support standard Multicast Listener Discovery (MLD). The interoperation with the ad-hoc gateway routers 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 interoperate with the mechanism used for gateway discovery and address auto-configuration previously described. It further allows the MIG to inform all ad-hoc nodes about the path towards multicast sources in the fixed network.

Address auto-configuration and gateway discovery were implemented by modifying the solution proposed at [12] by Jelger et al to support multiple gateways. The proposal specifies that the node connecting the ad-hoc cloud to the infrastructure network should advertise information allowing network discovery and address auto-configuration using IPv6. This is performed by sending a GW_INFO message containing a Neighbour Connectivity indicator, a Prefix Length, a Distance, a Sequence Number, the Gateway Address and an additional reserved field.

In order to allow the QoS interoperation among ad-hoc and infrastructure networks, the base SWAN [13] proposal was adapted and extended [14] in order 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. The classes were defined as: 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 also serves as a “buffer zone” or absorber for higher priority traffic bursts introduced by mobility. Figure 1 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 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.

Since infrastructure is expected to be driven by operator expectations and business models, it is imperative to have a proper support for charging the users. Furthermore, ad-hoc networks also require incentives for users to participate in the forwarding process. Such incentives can be provided in many forms, such as credit or service discounts. 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. To satisfy such purposes we used PACP [15], a charging and rewarding protocol.

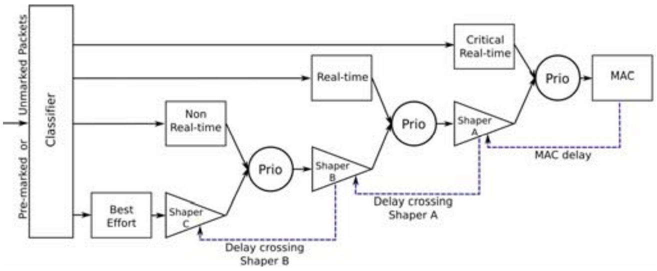

Figure 1: Differentiation model

III. TESTBED DESCRIPTION

The integrated ad-hoc testbed is comprised of several Linux computers running the modules previously described, according to Figure 2. The nodes were developed over Linux software. 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 some additional modifications required by some of the tested modules. These extensions are: the support for DSCP marking using Netfilter, the Hostap wireless driver, a Netlink multiplexer, an IP6_QUEUE Multiplexer, support for Token Bucket Queues, enhanced Mobile IPv6 RC2 stack and a customized version of MACKILL. With the exception of (parts of) the Mobile IPv6 stack and HostAP driver, all additional functions were developed inside the Daidalos project.

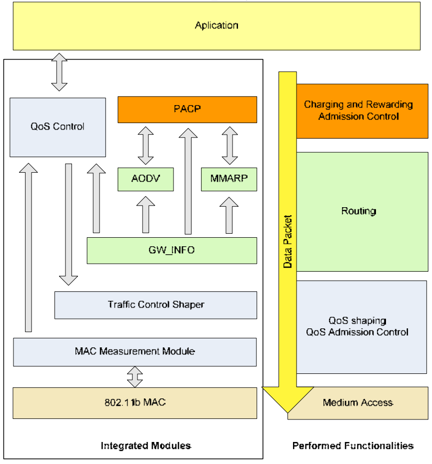

Figure 2: Software architecture

All machines have, at least 1.2Ghz CPU, 256Mb RAM, and enough storage space (in fact, the mobile node was not even able to run with specs much worse than these). One of the nodes is used to interconnect the ad-hoc cloud with the infrastructure network (GW), and here the wired interface will also be used to transfer data to or from the ad-hoc network. Wireless interfaces were Prism2.5 802.11b cards with the configuration parameters: ad-hoc and promiscuous modes, channel 12, rate fixed to 2Mbits and RTS/CTS threshold of 1 byte. Ad-hoc networking is limited to the wireless interfaces, and the developed protocols operation is restricted to these interfaces.

Figure 3 depicts the testbed infrastructure used in the tests reported in this paper. Six machines were used, in a multi-hop linear structure.

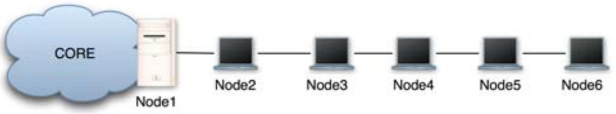

Figure 3: Integrated Ad-hoc network testbed

These nodes have been deployed inside a roughly square building with around 36m size, and normal office/lab divisions. Many WiFi access points exist inside the building, mostly on channels 1, 6 and 11. MACKILL was used to avoid logical interference. Many tests (but not all of them) were performed without traffic in the building (weekends), and in some cases, with the nearby access points powered off.

IV. EXPERIMENTAL RESULTS

This section presents the measured results obtained. The performance penalty introduced, by each functionality was measured in terms of maximum throughput, overhead, packet delay and jitter. We measured such metrics through 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 ad-hoc network size, we changed the number of hops between the sender and receiver in the ad-hoc network. In all experiments, unless otherwise specified, the bandwidth of the packet flows was 64Kbps, 128Kbps and 256 Kbps. UDP traffic was generated with *iperf* and *mgen*, according to the specific test. The same scenarios were used to test all functionalities, to allow for the comparison of their impact in the network.

a) Auto-configuration

The Jelger mechanism to auto-configure 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 non-connectivity.

Measured overhead is of 922 bps per link which, for a 64kbps bit rate, represents 1.44% of the data traffic. Auto-configuration time takes an average of 2 seconds and represents the time between the reception of the first GW_INFO message and the transmission of this 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, auto-configuration was seen not to have a large impact in network performance.

b) Multicast Routing

MMARP's evaluation started by the analysis of the throughput achieved in the topology described in Figure 3. These results are shown in Table 1 and we can observe that, in a direct connection between two hops, throughput is 1223 Kbps. This bit rate corresponds to the effective payload transmission (there is an overhead of 777 Kbps from IP, UDP and MAC headers, and from the CTS, RTS packets). In a five hop connection the throughput comes down to 76 Kbps. This behaviour is expected since all nodes are close to each other and radio interference exists.

In Table 2 and Table 3, it can be seen that the jitter and delay values increase significantly with the bandwidth of the flows. It is noticeable the penalty incurred by the multicast routing protocol. Packet losses (Table 4), increases both with the number of hops and flows' bit rate. Notice that these losses achieve 11.85 % in a 3 hops scenario with only 256 Kb/sec bandwidth flow. These values enforce the previous statement on the performance penalty incurred by multicast traffic in ad-hoc networks. In fact, ad-hoc multicasting in real scenarios should be carefully considered.

Table 1: Throughput for routing with auto-configuration

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 auto-configuration

Delay (ms)	64Kbps	128Kbps	256Kbps
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 auto-configuration

Jitter (ms)	64Kbps	128Kbps	256Kbps
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 auto-configuration

Loss (%)	64Kbps	128Kbps	256Kbps
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

The overhead introduced by MMARP and GW INFO is 3.94% in 64Kbps of traffic, which indicates that the overhead added by MMARP alone is almost twice the one introduced by the auto-configuration protocol. Notice that the first node is a MIG, and therefore, the number of control messages traversing this node is larger. This difference is actually reflected in the way the code is developed, and further explains the large jitter values for 1 hop.

c) Unicast Routing

In this section we evaluated the performance metrics of unicast routing with auto-configuration of the network. The maximum throughput achieved in the network losses is 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 the one of AODV.

We observe that both delay (Table 5) and jitter (Table 6) values are very small, where jitter values are about 5-10% of the delay, and increase slightly with the increase in the flows' bandwidth. This increase is, obviously, larger for high traffic values and the increase in the number of hops induces an increase in both delay and jitter. 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 multi-hop communications in shared environments.

Table 5 – Delay: unicast routing and auto-configuration

Delay (ms)	64Kbps	128Kbps	256Kbps
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

Jitter (ms)	64Kbps	128Kbps	256Kbps
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

The overhead introduced by the AODV and auto-configuration protocols is of 2.48% per hop with 64Kbps of traffic in the network, which is similar to the one of MMARP.

d) QoS

Figure 4 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. 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, in the extended SWAN model, the real-time traffic class does not have any shaper and initiates its service at the maximum rate. 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.

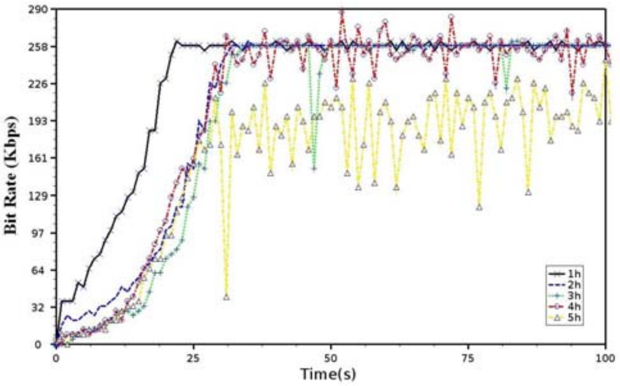

Figure 4: Less-demanding real-time rate variation for a 1 Hop connection

Figure 5 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, non-real time and best effort classes. 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). Best effort class uses the remaining bandwidth.

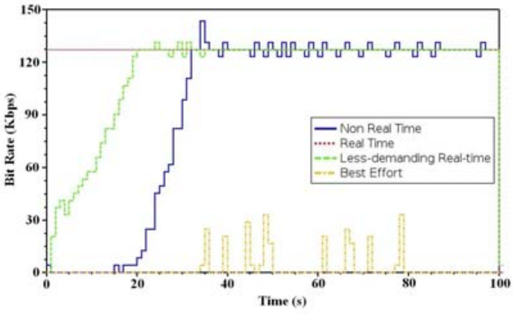

Figure 5: Initial set-up differentiation

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.

e) Charging and Rewarding

Finally, we evaluate the performance of the charging and rewarding mechanism (PACP). We address the overhead resulting of charging procedure and the error between the information reported and the actual packets received. In the scenario used, the flows were sent from Node4 to Node1. In this situation, the PACP charging manager is collocated with the receiving node (GW), which is the most favourable situation. The resulting overhead is not constant varying between 35 and 45kbits/s for the 256 kbps bandwidth flow.

The overhead percentage for each bandwidth flow is presented in Table 7. The total overhead is around 17%. We notice that these results are dependent on the packet rate and not on the actual bandwidth of the data flows. This is due to the constant proof size and the constant number of reports issued per data packet forwarded.

Table 7: Charging overhead

	64Kbps	128Kbps	256Kbps
Report Rate	5.8Kb	9.69Kb	20.78Kb
Marking Rate	5.9Kb	11.2Kb	23.2Kb
Marking Overhead	9.2%	9.2%	9.2%
Report Overhead	9.14%	7.57%	8.12%
Total Overhead	18.34%	16.77%	17.32%

Because additional processing was added to every packet, we expect jitter and delay values to change when compared with the routing-only values. Table 8 depicts the jitter and delay obtained for a 3 hop scenario with different traffic. We observe that the values are slightly increased when compared to the ones presented before, which is a consequence of the packets' verification, processing and inclusion of proofs.

Table 8: Delay and jitter - unicast routing, auto-configuration and PACP are active

Delay/Jitter (ms)	64Kbps	128Kbps	256Kbps
Delay	16.4	16.6	25.8
Jitter	7.8	6.18	3.5

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 may suffer some deviation from the actual number of packets traversing the network.

V. CONCLUSIONS

This paper showed the measured effects of introducing several functionalities into ad-hoc 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) raise several concerns. Of larger concern, is the large behaviour variability that one can find, 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 (unless message passing mechanisms are considered). Overhead values are large for secure charging the communication, and communication is throttled as soon as QoS regulation is taking place.

These results show that a carefully scenario analysis should be developed before deploying ad-hoc functionalities in any network: not all of features will be effective in complex environments.

REFERENCES

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

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

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

[4] I. Chakeres, C. Perkins, "Dynamic MANET On-demand (DYMO) Routing", IETF Internet Draft, draft-ietf-manet-dymo-06.txt, Oct 2006

[5] Network Simulator - ns-2, www.isi.edu/nsnam/ns/ as in Feb. 2006.

[6] R, Ahuja, et al, "GloMoSim: A Scalable Network Simulation Environment", M Takai, L Bajaj - Technical report 990027, UCLA, Computer Science Department, 1999.

[7] 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.

[8] S. Sargento et al., "Mobile Ad-Hoc Networks Integration in the Daidalos Architecture", IST Mobile and Wireless Communications Summit 2003, June 2005.

[9] Uppsala University, Ad Hoc Implementation Portal website: http://core.it.uu.se/AdHoc/ImplementationPortal , as in February 2006.

[10] J. Girão, João Paulo Barraca et al., "QoS-differentiated Secure Charging in Ad-hoc environments, International Conference on Telecommunications", Aug 2004.

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

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

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

[14] 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.

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