Mobile Ad-Hoc Networks Integration in the Daidalos Architecture

Susana Sargento[1], Tânia Calçada[2], João Paulo Barraca[1], Sérgio Crisóstomo[3], João Girão[4], Marek Natkaniec[5], Norbert Vicari[6], Francisco Cuesta[7], Manuel Ricardo[2], Andrzej Glowacz[5]

[1] Universidade de Aveiro, Instituto de Telecomunicações, Portugal
[2] INESC Porto, Faculdade de Engenharia da Universidade do Porto, Portugal
[3] DCC & LIACC, Porto University Porto, Portugal, [4] NEC Europe Network Lab., Heidelberg, Germany
[5] AGH University of Science Technology, Krakow, Poland, [6] Siemens AG - Corporate Technology, Germany
[7] University of Murcia, Spain

Abstract— **This paper describes the Ad-hoc network integration architecture being developed inside the IST project Daidalos. This architecture supports the efficient delivery of services, unicast and multicast, legacy and multimedia, to users connected to the ad-hoc network. For this purpose, several functionalities need to be in place. First, efficient routing and mobility mechanisms are proposed to enable mobility of users inside and between ad-hoc networks with decreased overhead. Second, distributed QoS mechanisms need to be developed to support service differentiation and resources control responsive to nodes mobility. Finally, security, charging and rewarding mechanisms are proposed to guarantee that only authorized users access the requested services, to increase the operators interest, and to ensure the correct behaviour of the users in the ad-hoc network.**

Index Terms—**Ad-hoc, Integration, routing, mobility, QoS, security, charging**

I. Introduction

The DAIDALOS project [1] aims at seamlessly integrating heterogeneous network technologies that allow network operators and service providers to offer new and profitable services (voice, data, multimedia). The architecture integrates both wired and wireless technologies, with quality of service (QoS) capabilities under a common authentication, authorization, accounting, auditing and charging (A4C) framework and in a secure communication environment. Mobile Ad-hoc networks (MANET) integration is also one of the main Daidalos achievements. Ad-hoc networks in Daidalos are not considered standalone networks, but are used as an extension of the radio coverage of broadband wireless systems, increasing the coverage area of, e.g., hotspots. This business strategy is profitable both for the provider, which increases its revenues, and for the user, that can be connected to the Internet anytime and anywhere. Therefore, Daidalos addresses the main aspects of the integration between ad-hoc and infrastructure networks.

This paper describes the Daidalos ad-hoc network architecture and its integration with the infrastructure networks. More specifically, it describes the modules required in the ad-hoc network, its functionalities and communication to provide the delivery of a large diversity of services, unicast and multicast, legacy and multimedia, to users connected in the ad-hoc network. This efficient communication requires the following *new* functionalities:

- Discovery of a gateway to the infrastructure network to obtain an auto configured global address, and efficient routing mechanisms to support the mobility of users inside the ad-hoc networks with decreased overhead. Moreover, since mobility is a key aspect of next generation networks, the ad-hoc architecture supports handovers of mobile ad-hoc nodes between different ad-hoc networks;
- QoS support in terms of differentiation, admission control and recovery from mobility and congestion situations. Due to the dynamic topology changes and due to the absence of a central node with knowledge of the network resources, QoS support is a major challenge in ad-hoc networks. Therefore, the ad-hoc QoS protocol needs to work in a distributed way, with mechanisms for reacting in a responsive way to topology changes;
- Security mechanisms to guarantee that only authorized users access the ad-hoc resources and the services available, and to make sure that the information of the nodes in the path is not modified in transit;
- Charging and rewarding of mobile ad-hoc nodes. Beyond charging users in an operator environment, an essential issue in ad-hoc networks is the requirement for mobile nodes to cooperate in traffic forwarding, avoiding the selfishness of forwarding. A basic economic idea aims to provide some rewards to nodes that behave appropriately. Due to the dynamic nature of ad-hoc networks, with nodes dynamically joining and leaving, the charging is not trivial. Moreover, in the rewarding process, the information of the overall path needs to be available, which increases the challenge.

This paper is organized as follows. Section II presents the ad-hoc network architecture. The routing and mobility integration are described in section III, and QoS integration is addressed in section IV. The security, charging and rewarding mechanisms are addressed in section V. The main conclusions are depicted 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).

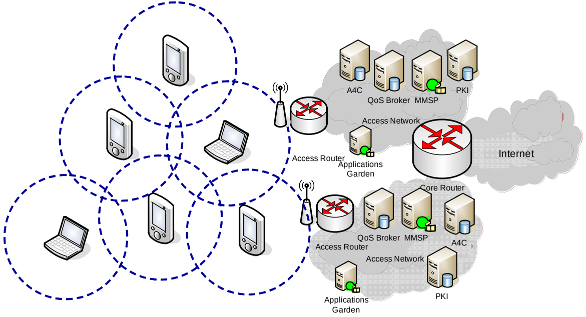


Figure 1: Daidalos ad-hoc network architecture

Inside the ad-hoc network, the traffic is routed through unicast (Ad-hoc On demand Distance Vector routing – AODV [6]), multipath (AO Multipath DV – AOMDV [7]), or multicast (Multicast MANET Routing Protocol – MMARP [8]) 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. The A4C server handles all authorization, authentication and charging issues. Since routing and charging mechanisms are secured, the ad-hoc nodes need to maintain cryptographic material to be able to send and receive the traffic. This key management is provided by the Public Key Infrastructure (PKI) server. The QoS Broker is an element that performs admission control and manages network resources; it controls the network routers according to the active sessions and their requirements. The provision of multimedia services is also supported by the MultiMedia Service Proxy (MMSP). The ad-hoc nodes can access operator services available in the infrastructure network (located in the Application Garden), and can access any node in the Internet.

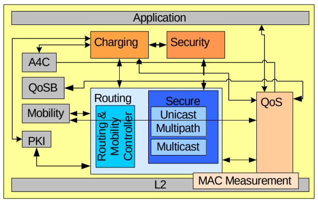


Figure 2: MN and AR general architecture

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 sections 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. Routing and Mobility Integration

The routing functions in the ad-hoc sub-system address several problems. First, ad-hoc nodes need to discover a gateway (GW) to the infrastructure network, auto-configure a global address, and execute handovers between different ad-hoc networks. Second, efficient unicast, multipath and multicast routing between ad-hoc nodes have to be provided. Finally, the mobility of ad-hoc nodes between different ad-hoc networks is also supported.

A. Routing and Mobility Controller

Both MNs and AR contain a Routing and Mobility Controller (AHRoutC) module to: (1) discover a GW to the infrastructure network, (2) execute the authentication and authorization processes, and (3) manage the mobility of the node executing the handover between ad-hoc networks.

1) Gateway Discovery

In order to discover the GW, a proactive gateway discovery protocol based on [2] is used. Each GW sends information periodically to ad-hoc nodes in a GW_INFO message, configured with a hop limit of 1. By receiving it, the ad-hoc node (1) becomes aware of the GW and its network prefix, (2) configures its global address using the prefix announced, and (3) forwards the message to other nodes. The information is then propagated hop-by-hop among the subset of ad-hoc nodes that decided to use this prefix and GW. Together, they form an ad-hoc network sharing the same prefix. In this phase it is also established a tunnel between the mobile nodes and the GW, which will be used for security purposes (this is described in section V).

2) Ad-Hoc Nodes Mobility

When a mobile node moves and starts receiving information messages from multiple GWs, it must select one of them, using criteria that minimize the distance in hops to the GW, and maximize the network stability (or other criteria). Changing GW may imply handover. The Daidalos nodes directly connected to the infrastructure (1 hop distance) use the Fast Handover [5] mechanism, which provides mobility and very low packet loss probability. The ad-hoc handover process proposed differs from the Fast Handover process: it minimises the handover related signalling messages (since it traverses multiple nodes) at the expense of some packet loss.

As shown in Figure 3, the mobile node receives information messages from different GWs, and decides to handover. The new network prefix information is then delivered to the registration module, which creates and configures a Cryptographically Generated Address (CGA) [4] (but does not notify Mobile IP version 6 – MIPv6), verifies the authenticity of the AR using Secure Neighbour Discovery (SEND) [3], and validates the CGA using Duplicate Address Detection (DAD). This process will be detailed in section V. The QoS Broker, the element in charge of managing resources, is in charge of authorizing the handover, taking into account the resources in the new network. The request of approval is made through the old access router (oAR), which currently acts as GW for the moving node, and it is triggered by the GW_SWAP message; the GW_SWAP_Ack message informs the node about the request success. In order to execute the handover, the MN sends the GW_ATTACH message to the new AR, and requests the QoS Broker to release the resources in the oAR. The registration module is informed about the handover, verifies the authenticity of the new AR using SEND, performs DAD to the new global IPv6 address, and notifies MIPv6 in order to configure the new CGA as its primary care-of address; if something goes wrong in these procedures, a REG_FAILED is sent.

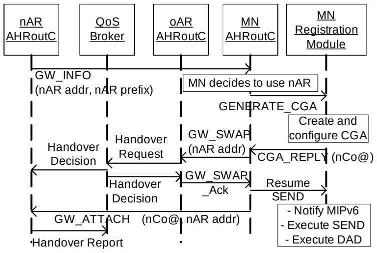


Figure 3: Ad-hoc handover process

B. Unicast and multipath routing

The basic unicast connectivity in the Daidalos ad-hoc architecture is provided by the reactive AODV [6] routing protocol. Routes between ad-hoc nodes are discovered on demand, while the route towards the AR is provided by the GW discovery.

Optional, the multipath extension of AODV, AOMDV [7] routing protocol, provides the discovery of multiple, link disjoint paths to other nodes or the AR. Additional to the higher resilience provided by multipath routing, an efficient usage of the multiple paths is obtained by integration of the QoS mechanisms. Data packets with QoS requirements are forwarded on a path according to the availability of resources. For best effort traffic, the load can be shared evenly on different paths. The assignment between data flows and paths is controlled in a flow forwarding table at the source node. For intermediate nodes the path is determined by the notion of a last hop in the routing table, i.e. the first hop of a path determines the complete path and intermediate nodes do not need to keep a flow forwarding table.

The GW in the AR includes a mapping module to translate between ad-hoc and infrastructure routing messages.

C. Multicast routing and integration aspects

MMARP [8] is a new multicast ad-hoc routing protocol that incorporates additional functionalities, compared to other ad-hoc multicast routing protocols, to deal with the complexity of supporting traditional IP nodes whilst interoperating 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. Every node may become a MIG at any time. The MIG is responsible for notifying the ARs about the multicast groups having interested receivers within the ad-hoc fringe. The key point of the protocol is to confine any new functionality within the ad-hoc fringe, providing ad-hoc nodes with the ability to process standard protocols (i.e. Multicast Listener Discovery - MLD) to interact with non-ad-hoc nodes. Therefore, the protocol does not require any changes in standard IP nodes and routers.

The MMARP protocol is extended to interwork with the mechanism used for GW discovery and address auto-configuration previously described. The messages propagated by the GW discovery module inform the MMARP ad-hoc routing module in each node about its nearest AR. It further allows the GW to inform all ad-hoc nodes about the path towards multicast sources in the fixed network. The MMARP implementation is being further extended with a security extension which provides message authentication and non-repudiation. Messages sent and forwarded by the ad-hoc nodes will be protected against most route manipulation attacks and message forgery.

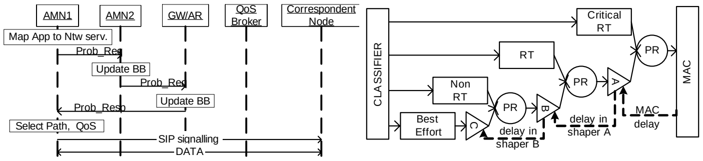
IV. QoS Integration

The integration of the ad-hoc networks with the infrastructure networks raises the requirement of coupling the infrastructure QoS model with the ad-hoc specific QoS model. Since the QoS model for the infrastructure network is Differentiated Services (DiffServ) based, the QoS model proposed for the ad-hoc network is an extension of the SWAN model [9].

In order to allow the QoS interoperation among these networks, the base SWAN proposal was adapted and extended. 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, as it will be later explained. 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.

A. QoS Signalling for Session Establishment

Figure 4a illustrates the message sequences of the QoS signaling corresponding to the establishment of a data session triggered by an ad-hoc node (a similar process exists if the sender is outside the ad-hoc network; in this case the AR starts the probing process). Before issuing the application signaling for session setup, the node performs a QoS check in the ad-hoc path. Application characteristics are mapped into network services and QoS parameters (bandwidth, loss, DiffServ Code Point - DCSP). The source node will then perform a hop-by-hop probing process by sending a *probe request* message to the next hop in that path, towards the destination (if the destination is outside the ad-hoc network, the probing is sent towards the AR). In the case of multipath support, a *probe request* message will be sent to each available path to the destination. Every intermediate node updates the *probe request* with the bottleneck bandwidth of the path. Once the *probe request* arrives at the GW, a *probe response* is generated with information on the bottleneck bandwidth of the path.



a) Session setup with QoS (multipath)

b) Extended differentiation model

Figure 4: QoS integration

When the requester node receives the answers of the probing (unicast or multipath), it performs admission control based on the QoS requirements and the available bandwidth on the paths. Then, the session negotiation is in place (we consider the Session Initiation Protocol - SIP) and the resources in the infrastructure network are checked (through admission control in the QoS Broker).

B. *Service Differentiation*

Service Differentiation in SWAN considers only two service classes, one targeted at real-time UDP traffic and another one targeted at best-effort TCP and UDP traffic.

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 4b 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. In case of congestion situations (e.g. due to network dynamics) the higher priority classes are regulated. The bandwidth utilization of each of these classes will be continuously monitored. If the target bandwidth of one of these classes is exceeded, the Explicit Congestion Notification (ECN) bits of the packets belonging to that class will be marked, triggering a regulation procedure.

C. *Measurements*

As referred, the nodes need to feed the admission control and regulation mechanisms with information on the bottleneck bandwidth in the path and congestion state. This is performed through the enhancement/modification of the Linux WLAN card driver (IEEE 802.11b). To address the measurements, the card is set to promiscuous mode. The following parameters need to be measured: (1) per class/overall delay (from the start to the completion of RTS-CTS-DATA-ACK exchange in DCF); (2) per-class/overall bandwidth utilization sensing the media and constructing periodic statistics; (3) transmission rate, one of four possible transmission rates: 1, 2, 5.5 and 11 Mbps; (4) number of stations in the neighbourhood to determine the contention and to evaluate the available bandwidth using current rate information.

V. SECURITY, CHARGING AND REWARDING

This section details the security functionalities of the ad-hoc architecture and the charging and rewarding mechanism developed.

A. *Authentication and authorization processes*

The lowest form of security introduced in the ad-hoc sub-system is the one of address ownership. Ad-hoc nodes are exposed to a highly volatile environment in which addresses are chosen and distributed in a “first come, first served” fashion, without the control of a centralized entity. The first consequence is the open opportunity for address impersonation. Using SEND [3], as shown in Figure 5, the ownership of a node’s address is verified by using Public Key cryptography and associating the public part of the key to the owned address. SEND offers a challenge/response mechanism during the usual Neighbour Discovery Protocol (NDP), which ascertains whether the node also possesses the private key bound to that address, thus proving the ownership.

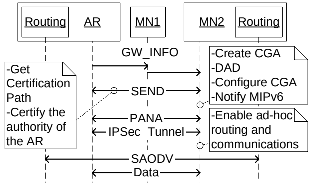


Figure 5: Authentication and Authorization

The proposed ad-hoc authentication and authorization process works as follows (Figure 5). After acquiring a global network prefix via the GW_INFO message, the node requests a certification path to the AR; using it, the node can certify the authority of the AR using the SEND procedure. A CGA [4] is then created by the node and, if validated by the DAD mechanism, it is used to configure the network interface and notify the MIPv6. The node can now authenticate in the network using Protocol for Carrying Authentication for Network Access (PANA) [10]. PANA provides the link between the node in the access network and the A4C infrastructure. One specific problem identified in ad-hoc, when applying this protocol, is the assumption that the node is one hop away from the AR, where the PANA Attendant resides. In order to overcome the multi-hop ad-hoc characteristic, an IP over IP tunnel is established between the node and the AR, providing the mandatory one hop. PANA is then executed and the resulting symmetric key is then used to bootstrap an IP Security (IPSec) association.

B. *Secure routing*

A routing protocol can be divided into two main parts. The first is route discovery and maintenance, during which nodes map paths to other nodes and keep these paths updated with the changes in network topology. The second main part is the actual forwarding of the packets. Even if a node is assured the route is correct, this does not mean that the forwarding will be performed correctly by the subsequent nodes. In this phase of the project we focus on securing the route discovery and maintenance phases of the routing protocol, as it is assumed that if the node has passed the authentication checks, it is not a malicious node. Moreover, as will be shown later, the rewarding mechanism will reduce the maliciously of the forwarding nodes.

Section III described our choices in routing protocols. We make use of Secure AODV (SAODV) [11] to protect the signalling of these protocols. The security necessary to protect the multicast signalling messages is a problem of a different nature, which was addressed in III. SAODV is also based on Public Key cryptosystems; it makes use of digital signatures to ensure the authenticity of the message, and a Lamport hash chain to protect the nodes from misusing the protocol and force sub-optimal routes.

C. *Charging and rewarding*

Charging traffic directly routed between ad-hoc nodes requires a distributed, secure and scalable solution. Moreover, the potential revenues of the operator derived from service consumption are affected by factors like the number of users accessing those services, the throughput of the network and the satisfaction of the users. This requires the forwarding in the network to be free from selfish behaviours. One measure to increase the willingness to forward others traffic is to reward forwarding users for the battery and processing power used.

The developments present in the state of the art address these issues by creating a distributed mechanism [12] 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 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.

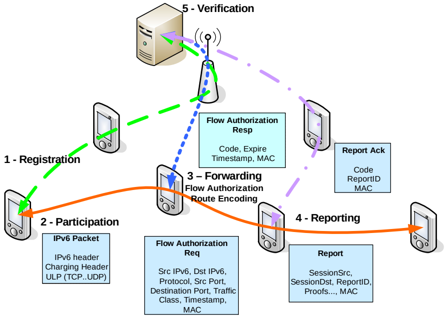

5 - Verification
Flow Authorization Resp
Code, Expire Timestamp, MAC
1 - Registration
Report Ack
Code ReportID MAC
3 – Forwarding
Flow Authorization
Route Encoding
2 - Participation
IPv6 Packet
IPv6 header Charging Header ULP (TCP..UDP)
4 - Reporting
Flow Authorization Req
Src IPv6, Dst IPv6, Protocol, Src Port, Destination Port, Traffic Class, Timestamp, MAC
Report
SessionSrc, SessionDst, ReportID, Proofs..., MAC

Figure 6: Charging phases and messages

Figure 6 presents the different phases in the charging and rewarding mechanism, as well as the relevant messages exchanged. A sending node already authorized to participate in the network (Registration phase) must add a charging header to every sent packet, as an IPv6 Hop-by-Hop extension, composed by some control information, a Route Identifier, a Hash Chain and a MAC. The Route Identifier field implicitly (through polynomial encoding) includes the identification of the route that will be updated in each node in the ad-hoc network towards the destination. This field is fixed size and cryptographically secured (through Hash Chain and MAC) so it cannot be wrongly modified along the path (Participation phase). Each forwarding node updates the charging header by encoding its address to the Route Identifier and its shared secret to the Hash Chain (Forwarding phase). If this node is the last forwarding node, it is responsible for collecting and sending the route identifiers (proofs) to the AR (Reporting phase). These proofs are sent to the AR when the information collected in the node reaches a specific number, or when a timeout expires. If the node does not report the proofs, it will not be rewarded. Upon reception of the proofs, the AR is able to verify the authenticity of the information contained in the message and to send a charging event to the A4C for the correct amount of credit. The A4C verifies the truthfulness of the information, through the cryptographic information contained in the proofs, and retrieves the information of the ad-hoc route (*Verification* phase). It is then able to correctly charge the sending/receiving nodes, as well as correctly reward the forwarding nodes.

Simulation experiments were performed to address the performance of the proposed charging and rewarding protocol. The results confirm that, compared to [12], this polynomial route encoding mechanism has a high charging efficiency and low network overhead.

VI. Conclusions

This paper described the Ad-hoc network integration architecture being developed inside the IST project Daidalos, in terms of its functionalities, modules and interactions to efficiently support the delivery of diversified services to users connected to the ad-hoc network. The proposed architecture is able to efficiently integrate ad-hoc and infrastructure networks, supporting unicast and multicast routing, QoS, security and charging mechanisms with small overhead in the ad-hoc side. Moreover, the architecture allows the mobility of users between ad-hoc networks.

Although the ad-hoc architecture is still under development, there is already a first working demo that supports the delivery of an audio and video streaming to multi-hop unicast and multicast users, with secure routing, and with the required information to allow for the charging and rewarding process.

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] 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.

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

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

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

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

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

[8] 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.

[9] 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.

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

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

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