A QoS Architecture Integrating Mobile Ad-Hoc and Infrastructure Networks

Sérgio Crisóstomo[1], Susana Sargento[2], Marek Natkaniec[3], Norbert Vicari[4]

[1] LIACC, Faculdade de Ciências da Universidade do Porto, Portugal
[2] Instituto de Telecomunicações, Universidade de Aveiro, Portugal
[3] AGH University of Science and Technology, Krakow, Poland
[4] Siemens AG - Corporate Technology, Germany

***Abstract*—This paper proposes an Ad-hoc QoS architecture, using as basis some concepts from SWAN, extending it to fulfill our requirements. It also proposes the modules required in the network elements and its interaction to provide optimized QoS in the ad-hoc network and integration between both networks. We also present the overall integration QoS signaling protocol and the differentiation mechanisms to address end-to-end QoS for real-time multimedia applications. Furthermore, the proposed solution addresses the use of multipath routing in order to provide load balancing and increased network reliability.**

***Index Terms*—Ad-hoc networks, infrastructure networks, integration, quality of service.**

I. Introduction

DRIVEN by the increasing users' requirements to be connected to the Internet at anytime and from everywhere, there is a vast amount of research in the area of ad-hoc networks.

Ad-hoc networks first appeared as autonomous networks. However, currently ad-hoc networking is becoming a promising solution to increase the radio coverage of broadband wireless systems [1], extending the coverage area of hotspots. This scenario is particularly useful for telecom operators, especially when associated with wireless subscriber access, since it allows the users to access the services offered, and it allows some operator control on the services accessed by the users. To support the users and service requirements, the ad-hoc network needs to support differentiated QoS, which is a major challenge in ad-hoc networks. First, the network topology may change very frequently since all ad-hoc nodes may be mobile. Second, there is no central node with knowledge of the network resources. Therefore, any ad-hoc QoS protocol needs to work in a distributed way (in all ad-hoc nodes), with mechanisms for reacting in a responsive way to topology changes.

Although the work in QoS support for ad-hoc networks is in its early stages, there are already some proposals: INSIGNIA [3], SWAN (Stateless Wireless Ad-hoc Networks) [5] and FQMM (Flexible Quality of service Model for Mobile ad-hoc networks) [4]. There are, however, no well-known QoS proposals addressing the integration of ad-hoc networks with infrastructure networks.

In this paper we propose a QoS architecture for integration between mobile ad-hoc and infrastructure networks. Due to the intrinsic different nature of both networks, the QoS approaches will be different in the two sides. The smooth integration will be provided by the means of gateways interfacing the networks and supporting both QoS architectures. We propose the ad-hoc QoS architecture, using as basis some concepts from SWAN, extending it to fulfill our requirements. We also propose the modules required in the network elements and its interaction to provide optimized QoS in the ad-hoc network and integration between both networks. We present the overall integration QoS signaling protocol and the differentiation mechanisms to address end-to-end QoS for real-time multimedia applications. The proposed solution addresses the use of multipath routing in order to provide load balancing and increased network reliability.

The paper is organized as follows. In section II, the overall network architecture is presented. Section III addresses strategies for the QoS integration between ad-hoc and infrastructure networks. The extensions of the SWAN proposal and the overall QoS signaling framework to provide integration between ad-hoc and infrastructure networks are presented in section IV. The description of the proposed QoS architecture is performed in section V. Section VI presents our main conclusions and future work.

II. Network QoS Architecture

In the scenario, where ad-hoc networks are used to increase the radio coverage of wireless systems [1], they are not independent networks, but are connected to the Internet through infrastructure access networks.

This work is in part supported by the EU FP 6 for Res. and Dev. Daidalos (IST-2202-506997).

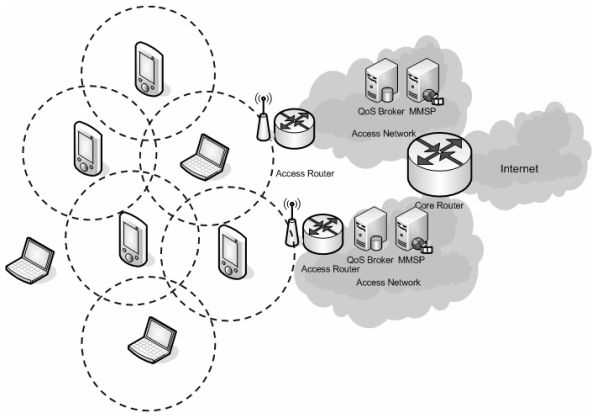

Fig. 1. Network QoS architecture.

These ad-hoc networks can be seen as an extension to access networks, where nodes can access the Internet through other mobile nodes towards the Internet (Fig. 1).

The purpose of this composed access network is to deliver and support any type of services and applications (e.g. audio and video conferencing and streaming) to the end users, located in the ad-hoc network. Therefore, both ad-hoc and infrastructure networks need to be closely integrated to provide the adequate service delivery and support of differentiated QoS in an integrated way for the users and services. We consider that both networks are Differentiated Services (DiffServ) [2] based to achieve scalability and performance. In the infrastructure side, there is an element, the QoS Broker that performs admission control and manages network resources; it controls the network routers according to the active sessions and their requirements. The provision of advanced multimedia services is also supported by the Multimedia Service Proxy (MMSP).

In the ad-hoc side, the same advanced services need to be provided with the same quality. For the support of real-time advanced services in these scenarios, the integration of these networks considering QoS aspects needs to be in place. Since the nature of the ad-hoc and infrastructure networks is very different, the QoS protocols will need to be modified in order to provide smooth integration between the two different types of networks. Thus, functional gateways, co-located with the access routers, will be required to interface the two types of networks.

In the next sections we will address the QoS support in wireless ad-hoc networks, its integration aspects, and a solution will be proposed for the support of real-time advanced services.

III. Integration Strategy

Although much work has been done in the area of QoS in IP networks, this work has mostly assumed wired networks and infrastructure wireless networks. Even in infrastructure based wireless networks, the network resources are very scarce, and therefore special QoS mechanisms need to be in place when mobile nodes change their point of attachment to the network. When the use of mobile wireless networks is extended further to include support for ad-hoc networks, the support of QoS becomes an even more difficult problem. In ad-hoc networks, all nodes may be mobile and the network topology may change very frequently. This unstable topology and the fact that there is no central node with knowledge of the network resources introduce a great challenge in the QoS support in this type of networks. Therefore, any ad-hoc QoS framework needs to work in a distributed way (in all ad-hoc nodes), with mechanisms for reacting in a responsive way to topology changes.

In order to provide end-to-end QoS for the services being delivered through the ad-hoc networks connected to the Internet, it is required to provide interoperation between the distributed QoS solutions inside the ad-hoc networks and the fixed IP networks.

This integration requires a special network entity, a Gateway (GW), co-located or not with the element (usually the access router) that interconnects the ad-hoc network with the infrastructure one (see Fig. 2). This entity needs to perform, beyond other functions not related to QoS, the QoS inter-working in terms of service admission control and service differentiation. This gateway is targeted to: (1) cooperate in the admission control decision; (2) perform signaling mapping between the ad-hoc and infrastructure networks; (3) perform service differentiation mapping; (4) perform service differentiation enforcement (classification, monitoring, policing and shaping); (5) resort to user profiles for admission control decisions; and (6) perform traffic regulation (in overload conditions) on a priority differentiation basis. Since gateway has no means to efficiently manage resources in the ad-hoc the admission control has to be performed with the ad-hoc nodes collaboration.

In the next section we will present a proposal for QoS integration between ad-hoc and infrastructure networks based on the extension of the SWAN model.

IV. SWAN Extensions for Integration

SWAN [5] is composed by a QoS model for service differentiation, an associated QoS negotiation procedure, and a dynamic regulation mechanism to react in case of congestion situations (e.g. due to mobility and route changes). This QoS model addresses two traffic classes: real-time and best-effort traffic. The mechanism is stateless in the sense that intermediate nodes do not keep any per-flow state information. Instead, SWAN uses local rate control for UDP and TCP best-effort traffic based on MAC delay measurements, and admission control for real-time traffic is performed by the source, based on the result of an end-to-end request/response probe that senses the available bandwidth through the path from the source to the destination. SWAN also resorts to dynamic regulation of real-time sessions when congestion/overload conditions occur (e.g due to node mobility).

The basic SWAN model needs to be enhanced and changed to cope with our requirements.

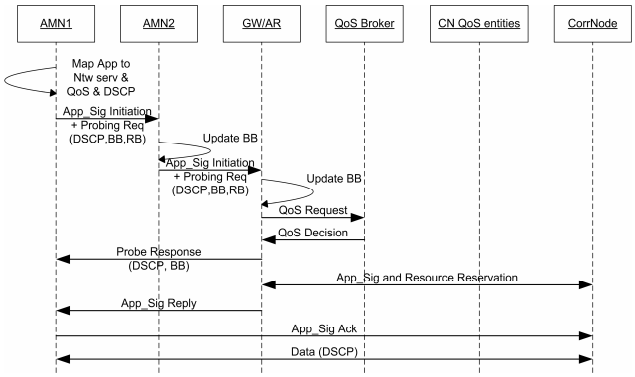

Fig. 2. Ad-hoc initiated session setup.

First, the SWAN QoS signaling and dynamic regulation protocol need to be integrated with the QoS signaling in the infrastructure side; signaling mapping needs to be in place in the GW. Second, we aim at addressing multipath support and load balancing inside the ad-hoc network (it is planned to further extend the multipath support to perform load balancing among GWs). Finally, we will extend the SWAN model to address four traffic classes in the ad-hoc cloud; the differentiation model will be optimized. In the next sub-sections we describe the proposed SWAN extensions.

A. QoS signaling and dynamic regulation protocol

Our proposal is to abstract the ad-hoc path between a source (or destination) Ad-hoc Mobile Node (AMN) and the GW as a virtual link. This abstraction has the main purpose of allowing the integration with QoS signalling in infrastructure networks, in which the source and destination AMN and the GW are the only ad-hoc nodes that participate in the QoS signalling process from the point of view of the infrastructure network.

Being based on an adaptation of the SWAN model, the ad-hoc nodes in this QoS architecture do not keep per-flow state information, which means that there are no per-flow resource reservations. We extend the SWAN probing process to interoperate with the infrastructure network.

An ad-hoc node wishing to establish a session through the infrastructure network should gather information about the available resources in the virtual link to the gateway resorting to the SWAN probing process (Fig. 2). In our example, we consider a general application protocol, App_Sig, that requires 3 messages for a session setup: Initiation, Reply and Ack. These messages can be mapped into real protocols: in Session Initiation Protocol (SIP) [6], for example, they correspond to INVITE, 200 OK and ACK. If the sender node is an ad-hoc mobile node, it sends an App_Sig Initiation message with DSCP value corresponding to the requested class of service piggybacked with a probing request message (Fig. 2). This request contains a Bottleneck Bandwidth (BB) field located in an IPv6 extension header that is updated in a hop-by-hop basis with the minimum available bandwidth of the corresponding class in the path, and the Requested Bandwidth (RB) for the flow (the mobile node includes a QoS client module that maps application to network QoS parameters).

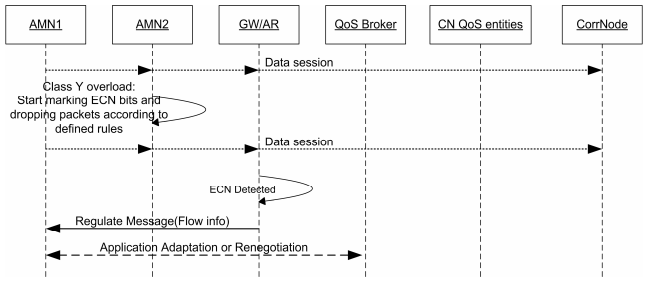

Fig. 3. Dynamic regulation.

The probing request message is updated by every intermediate node in the ad-hoc network with the BB of the corresponding class (minimum bandwidth in the path). Every mobile node has a layer 2 measurement module that measures the occupied bandwidth and the delay in each class in the wireless medium. After receiving this message, the gateway checks the BB and the RB (optionally, it can also check the delay values). If the BB is larger than the RB, this means that the ad-hoc network has sufficient available bandwidth. In this case, the gateway checks for resources in the infrastructure network, issuing a QoS request to the QoS Broker [7], and in the case of a positive answer, it forwards the App_Sig Initiation message to the receiver (correspondent node – CorrNode) through the core network (CN) entities (CN QoS entities). Simultaneously, it also replies to the probing request with a probing response message with indication of the available bandwidth in the ad-hoc path. Otherwise, an error message is sent to the sender. The correspondent node replies to the App_Sig Initiation with an App_Sig *Reply* message and, if the session parameters are allowed in the two terminals, the setup process ends with an *App_Sig Ack* message.

Fig. 3 depicts the case of dynamic regulation. When an ad-hoc node detects an overload condition in a class (target bandwidth for the class exceeded), this node starts marking ECN bits in packets of the affected class. The gateway monitors the ECN bits, and upon its detection notifies the sources by sending *Regulate* messages. When a source receives a *Regulate* message it should perform application adaptation, or else, should re-start the probing process.

B. Interaction with Multipath

In this section we describe how the model can support multipath routing in the ad-hoc network, i.e., multiple paths established between the ad-hoc nodes and the GW. The multipath routing is responsible for the discovery and maintenance of routes; moreover it is responsible for the control of the path on which data is forwarded. In multipath, the packets of a specific flow have to be forwarded on the selected path with the more adequate QoS characteristics.

The route discovery and maintenance process is based on AOMDV (Ad hoc On-demand Multipath Distance Vector Routing) [8]. Link disjoint paths are discovered and described by the next hop and last hop at each node on the path. They ensure that packets once assigned to a path will be forwarded on this path at each node. The assignment process is detailed in section V.

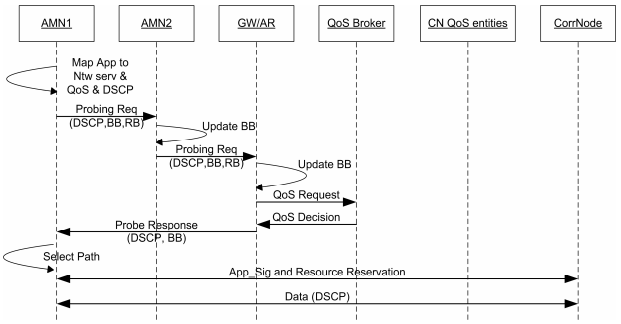

Fig. 4. Multipath session setup.

The standard packet/flow forwarding mechanism of AOMDV is extended to provide load balancing and QoS support. Usually, in AOMDV the first discovered path is used. Alternative paths are only utilized as backup. In the modified solution, all paths are utilized according to the following rules: (1) packets of an existing flow are scheduled for the same path as the preceding packets of the flow; (2) a new QoS flow is assigned to the best path according to the result of the probes; (3) a new best effort flow is assigned to a path selected taking into account the utilization of the alternative paths. To achieve this, a flow-forwarding table keeps track of the paths in which the flows are forwarded. The table is maintained at the source node (respectively, destination node or GW for the return flow). For intermediate nodes the path is fixed by the *next_hop* and *last_hop* notion in the routing table. The load entry in the table keeps track of the load imposed on the paths. The load entry takes only recent packets into account.

To integrate the multipath in the QoS signaling, the mobile node (in the ad-hoc network) or the GW (in the infrastructure network) need to start the probing process in the different paths. Upon receiving the several *probe response* messages, the path that will be used is the best one for the flow. Therefore, the application session setup signaling message can only be sent after the probing process, and these processes need to be decoupled. This process is shown in Fig. 4: after receiving the last *probe response*, the mobile node selects a path and sends the application signaling in this path.

C. Measurements Processing

As was referred before, every mobile node has to perform some MAC layer measurements in order to support admission control and service differentiation functions.

We consider that the ad-hoc mobile nodes will use the IEEE 802.11b standard[1] operating in DCF (Distributed Coordination Function).

Since DCF mode operates as best effort MAC and an admission control procedure is required, we need the measurements information to assure a proper QoS level. The following parameters need to be measured: (1) Per-class/overall delay – packet delay monitoring for four different classes/overall (from upper layer to the MAC layer and the time of the completion of RTS-CTS-DATA-ACK in DCF); (2) Per-class/overall bandwidth utilization – achieved by sensing the media and constructing periodic statistics about overall and per-class (DSCP code) bandwidth occupancy; (3) Transmission rate - current WLAN card transmission rate (in case of IEEE 802.11b the stations communicate using one of four possible transmission rates: 1, 2, 5.5 and 11 Mbps); (4) Number of stations - the estimation of the number of active stations in the neighborhood to determine the contention and to evaluate the available bandwidth using current rate information.

D. QoS 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. Real-time traffic has priority over best-effort traffic, while best-effort traffic is subjected to a leaky-bucket traffic shaper. In order to assure a limited delay to real-time traffic, the rate of the shaper is controlled applying an AIMD (Additive Increase, Multiplicative Decrease) algorithm that has the MAC delay as feedback.

This MAC delay represents the time it takes to send a packet between the transmitter and the next-hop receiver, including the total deferred time (including possible collision resolution) plus the time to fully acknowledge the packet. However, to fully support real-time traffic, local rate control of best-effort traffic is complemented by admission control of real-time traffic, and by the control of congestion situations which may occur due to network dynamics such as re-routing. The SWAN service level differentiation model can be further expanded to consider a finer service granularity. Our proposal considers four different traffic classes: one for critical real-time traffic, another one for less demanding real-time traffic, one for non real-time traffic service and a last one for 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.

Since no MAC differentiation is assumed, the access to shared medium imposes the same delay for all packets. In order to ensure a limited delay in the MAC access to higher priority packets it is necessary to control in a distributed way the total number of packets accessing the shared medium.

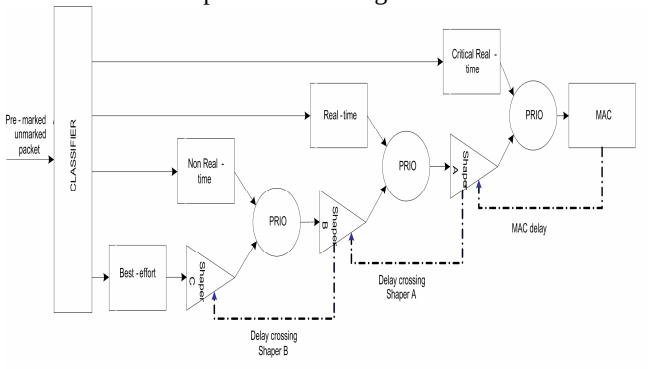

Fig. 5. Extended differentiation model.

[1] IEEE 802.11e is an alternative when the WLAN cards with support for EDCA under Linux 2.6 will be available.

$if\ (n > 0)$

$\quad s_i \leftarrow s_i \times (1 - r_i)$

$else$

$\quad s_i \leftarrow s_i + c_i$

$if\ ((s_i - a_i) > a_i \times g_i)$

$\quad s_i \leftarrow a_i \times (1 + g_i)$

Where:

- n is the number of packets that exceeded the threshold delay d_i;
- s_i is the shaping rate of shaper i;
- r_i is the multiplicative decrease factor;
- c_i is the additive increase increment;
- a_i is the actual rate of traffic crossing the shaper;
- g_i maximum gap of r_i concerning a_i

Fig. 6. AIMD algorithm.

So, 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. Traffic of higher priority classes is limited by admission control and regulation.

The extended differentiation model is composed by a classifier and by a cascade of priority schedulers, shapers and queues associated to each traffic class, as illustrated in Fig. 5. A priority scheduler gives priority to critical real-time traffic over the other classes. A limited delay is targeted to this class by applying a leaky bucket shaper (shaper A) to the other service classes. In order to achieve this limited delay, the rate of shaper A is controlled by an AIMD algorithm having the MAC delay as feedback. The differentiation between the real-time class and the other low priority classes is achieved by the scheduler connected to the input of shaper A. The rate of shaper B is also controlled based on an AIMD algorithm but having as feedback the packet delay imposed by shaper A. Connected to shaper B is a similar stage that differentiates non-real-time and best-effort traffic. In each shaper, the AIMD algorithm having measured packet delay as feedback is periodically applied to control the shaping rate. This is to ensure a limited delay (class dependent) to the traffic of each class. These targeted limited delays are thresholds delays for the algorithm decision criteria in each shaper. The feedback of the shapers are the MAC layer delay, for shaper A, or the time a packet is blocked in the downstream shaper, for the other shapers.

Being d_1 the target limited delay for the real-time traffic, in normal conditions a packet in the corresponding queue head is expected to be transmitted to the next hop in less than d_1 seconds. This expected time will be (d_1+d_2) *and* $(d_1+d_2+d_3)$ for the real-time and non real-time traffic, respectively. Following we present our proposed algorithm.

Every T time interval, the rate of each shaper is increased by an increment of c_i Kbps until one or more packets exceed the threshold delay d_i. When this is the case, the shaping rate is decreased by multiplicative factor r. When the shaping rate substantially exceeds the actual rate, there is the risk of transmitting data bursts without due control, which may affect delay of higher priority classes. In order to avoid this problem, the rate controller monitors the actual transmission, and regulates the shaping rate in order to not exceed the actual rate in more than a gap percent of the actual rate.

This differentiation model needs to be complemented by per-class admission control of the three higher priority classes, similar to the one used by the base SWAN for real-time traffic (probing). Besides that, 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 ECN bits of the packets belonging to that class will be marked, triggering a regulation procedure, as previously described.

Since marking all packets would have as side effect the readmission of all flows, which can cause unnecessary performance degradation, this ECN marking should be randomly performed according to a probability increasing with the congestion state of the class (queue occupancy).

V. Ad-Hoc QoS Architecture

The previous section addressed the required extensions of the SWAN protocol and the functionalities of the elements to provide QoS in the ad-hoc network and to integrate with the QoS architecture in the infrastructure network. In this section we present and describe the proposed QoS architecture that supports the above mentioned functionalities.

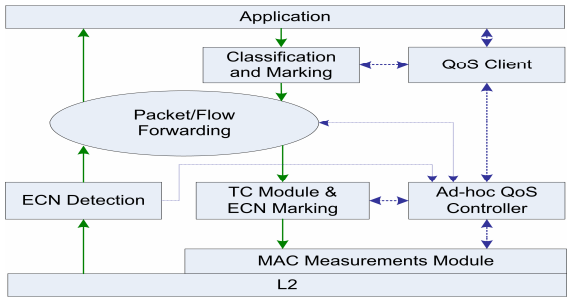

Fig. 7. Mobile node QoS modules.

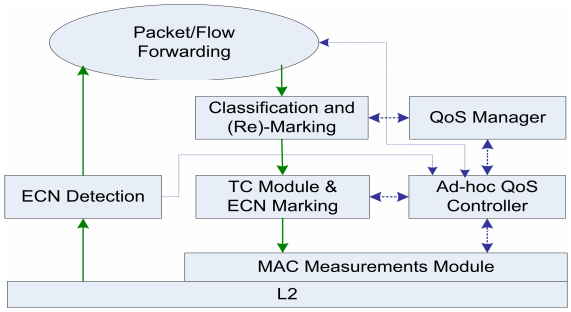

Fig. 8. Gateway QoS modules.

The ad-hoc mobile node has a double role, acting as a host which produces and consumes application traffic and acting as a router that forwards the traffic of other nodes. The mobile node needs to be able to retrieve the QoS parameters from the application characteristics, trigger the check for QoS resources along the ad-hoc path and check the available resources in its wireless medium. It can also classify and mark the packets according to its class, ensure QoS differentiation, mark ECN bits and detect ECN marked packets in the case of congestion. In our architecture, the mobile node supports multipath routing and the choice of an ad-hoc path according to its QoS resources. The GW is able to support the same functionalities of the mobile nodes, but does not have interaction with the application signalling (since it works only at the IP layer and below). Instead, it needs to perform interoperation between the QoS signalling in the ad-hoc and the infrastructure side.

These functionalities demand a special design of the QoS stack of both ad-hoc mobile node and the GW, which are presented in Fig. 7 and Fig. 8. The solid lines interconnecting the modules correspond to the data packet processing inside of a node. The dashed lines correspond to control information.

A. QoS Modules and Interactions

The ad-hoc mobile node has two main controlling components, the QoS Client (QoSC) and the Ad-Hoc QoS Controller (AHQoSC). The QoSC is active only when the mobile node is a sender or receiver, being responsible for the end-to-end (non ad-hoc specific) QoS negotiation. When an application is about to start, it triggers the QoSC for the QoS check and reservation. The QoSC then requests the AHQoSC to check for resources in the wireless medium or in the path of the ad-hoc network (the end-to-end check, including the infrastructure network, will be provided through the GW), depending if the architecture only deals with unipath (1st case) or considers multipath (2nd case). Upon the reception of a positive answer on the available resources (in the wireless medium or in the ad-hoc path, respectively, in unipath or multipath cases) from the AHQoSC, the QoSC matches or adapts the application QoS needs and the session establishment is started. The QoSC is also triggered by the AHQoSC when congestion situation is detected; in face of this situation it will contact the application module to re-negotiate the session and QoS parameters.

The AHQoSC is the module responsible for attending the QoSC requests, controlling the admission of new flows taking into account the per-class available bandwidth and delay values in the ad-hoc path. It triggers and performs the probing process of the SWAN model and controls the Traffic Control (TC) module for QoS differentiation assurances in the ad-hoc network. AHQoSC also takes proper actions in face of network congestion. When congestion in some class occurs, the AHQoSC instructs the ECN marking module (collocated with the TC module) to perform ECN marking in the packets belonging to that class (see sub-section IV.D).

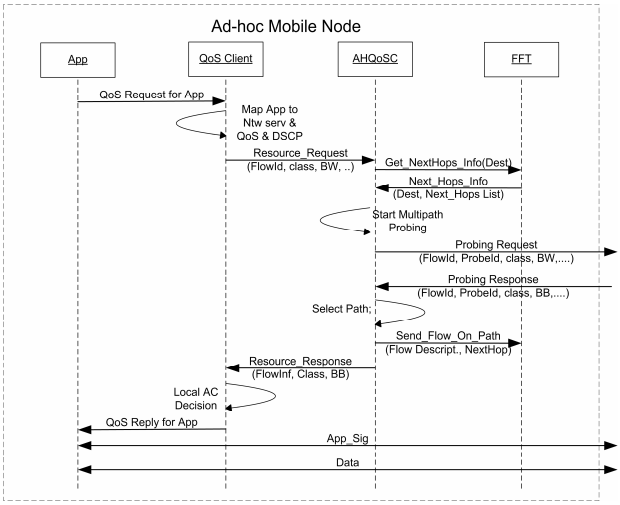

Fig. 9. Interaction between modules (MN).

On the other hand, when the AHQoSC receives a notification from the ECN detection module (this means that the node is the destination or the AR/GW), of a flow ECN marked, it will send a *regulate* message (sub-section IV.A) to the source node of the flow. When an AHQoSC in a source node receives a *regulate* message, it will inform the QoSC about the resource violation, carrying the flow description. The AHQoSC resorts to MAC layer measurements in order to determine the per-class bandwidth occupancy in the local shared wireless link and the mean delay of the packet transmission to its neighbours, besides other parameters, in order to participate in the distributed ad-hoc resource management and assure the service differentiation. Based on this information, the AHQoSC will control (configure) the TC module in order to enforce that each class will have the appropriate service level. The AHQoSC is also responsible for the SWAN probing process. In the sender node, when the AHQoSC receives a request for resources from the QoSC, it will evaluate the local available resources, and then will perform resource probing towards the destination. After receiving the probing response, it will re-evaluate the local resources, and send the answer to the QoSC. In an intermediate node, the AHQoSC updates the probing message with the minimum of the bottleneck bandwidth carried in the probe, and the local bandwidth. In the destination node, when a probe reaches the destination (or the GW), the AHQoSC sends back a probing answer to the source of the request with the result of the probing process.

The traffic differentiation is achieved by means of appropriate TC functions performed by every ad-hoc node, described in detail in sub-section IV.D. Fig. 9 illustrates the communication between internal QoS components and to external peer QoS entities of an ad-hoc node working as a source node, in a multipath session setup process. Before issuing the application signaling, the QoSC is triggered for QoS check in the ad-hoc path. The QoSC maps the application characteristics into network services (bandwidth, etc.) and QoS parameters (loss, DCSP, etc.), and issues a

Resource_Request to the AHQoSC with the service class, the required bandwidth and a flow identifier, which is the primary key identifying the new data session context in the mobile node. The AHQoSC consults its information about the local bandwidth availability for that class. In the case of unicast routing, if the available local bandwidth is sufficient, the Flow/Forwarding table (FFT) is consulted to check the next hop to the destination, and a hop-by-hop probing process is initiated by sending a *probe request* message to the next hop in that path. When the AHQoSC receives the *probing response*, it sends a *Resource_Response* with the bottleneck bandwidth to the QoSC. In case of multipath support, the probing process will be slightly different. The AHQoSC will retrieve from FFT the list of the *Next_Hops* and correspondent *Hop_Counts*, and will send a *probe request* message through three shortest paths (if available). This restriction to the number of probes is necessary in order to limit the probing overhead. At the end of the probing process, the AHQoSC will receive multiple *probing responses*, and will choose the path to use based on the QoS requirements, and on criterions such as load balancing. After the decision, it will set in the FFT the appropriate filters to accommodate the flow in the chosen path[2]. Then, the AHQoSC sends a resource response to the QoSC. The QoSC determines the available resources and proceeds with the session negotiation to the infrastructure network.

The GW is the entity that interfaces the ad-hoc cloud with the infrastructure network. As was previously shown, the QoS components and its functions are very similar to the ones of an AMN. The main difference is the non existence of a QoSC, since there is no interaction with application, and instead it contains a QoS Manager. The QoS Manager performs admission control with respect to the QoS requests it receives from the infrastructure nodes to ad-hoc destinations or vice-versa. When receiving a QoS request from the infrastructure to the ad-hoc network, the QoS Manager triggers the AHQoSC to check for QoS resources in the ad-hoc path. The AHQoSC will then start the probing process in the ad-hoc side. When receiving a *probing request* message from the ad-hoc side, the QoS Manager triggers the request for resources in the infrastructure side through the infrastructure QoS signalling.

The Classification and Marking module in the GW performs (re)marking functions to the classes supported in the ad-hoc network.

VI. Conclusions

This paper presented a QoS architecture for integration of ad-hoc networks with infrastructure networks, building a scenario very useful in hotspot environments. The proposed architecture includes the QoS differentiated and control support in ad-hoc networks and its mapping and integration with the QoS support in the infrastructure networks in order to provide end-to-end QoS for the services addressed. The ad-hoc QoS solution is based on the SWAN model with extensions to provide integration with other networks, multiple paths support and load balancing, and with a new QoS differentiation model able to efficiently support four traffic classes. The overall architecture, modules, interactions and signaling flows between modules and elements are presented.

This architecture is being simulated in ns-2 and implemented in the Linux OS (kernel 2.6). Our future work concerns the validation of this QoS architecture through simulations and real experiments, and its integration with ad-hoc security and charging and rewarding solutions.

[2] The session is not yet admitted. If the QoS Client fails to admit the session, the FFT filter 'will be cleaned'

References

[1] B. Xu, S. Hischke and B. Walke, "The Role of Ad Hoc Networking in Future Wireless Communications", Proceedings of ICCT 2003

[2]] S. Blake (ed) et al., "An Architecture for Differentiated Services", IETF RFC 2475, December 1998.

[3]] S. B. Lee *et al.* , "INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks" J. Par. and Dist. Comp., SI Wirel. and Mob. Comp. and Comm., vol. 60 nº4, Apr. 2000, pp.374-406.

[4]] Hannan Xiao et al., "A Flexible Quality of Service Model for Mobile Ad-Hoc Networks." In Proceedings of the IEEE Vehicular Technology Conference, Tokyo, Japan, May 2000, pp. 445-449.

[5] 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, pp. 192-207.

[6] J. Rosenberg et. al., "SIP: Session Initiation Protocol", IETF RFC 3261, June 2002.

[7] V. Marques et al., "An IP-Based QoS Architecture for 4G Operator Scenarios", IEEE Wireless Comm., June 2003

[8] M. Marina and S. Das, "On-demand Multipath Distance Vector Routing for Ad Hoc Networks", in Proc. of the International Conference for Network Protocols (ICNP), Riverside, USA, November 2001.