

MAC Layer Measurements for Supporting QoS in IEEE 802.11 Ad-Hoc Networks

Andrzej Glowacz¹, Marek Natkaniec¹, Susana Sargento², and Sérgio Crisóstomo³

¹ AGH University of Science and Technology, Al. Mickiewicza 30,
30-059 Cracow, Poland
{Glowacz, Natkaniec}@kt.agh.edu.pl

² University of Aveiro/Instituto de Telecomunicações, Campus Universitário de Santiago,
3810 Aveiro, Portugal
SSargento@det.ua.pt

³ University of Porto/LIACC, Rua do Campo Alegre, 823,
4150-180 Porto, Portugal
Slc@dcc.fc.up.pt

Abstract. This paper presents a measurement module for Wireless LAN card working in IEEE 802.11b together with an implementation, measurements and overall QoS architecture in ad-hoc mode. In this mode the wireless card allows for best-effort services transfer using IEEE 802.11 MAC layer DCF function. In order to support services with defined QoS level in IEEE 802.11b network, prior implementation of accurate measurement module to obtain the congested status of the shared medium access cooperating with higher layer applications is required. To evaluate the performance of the measurement module, we present the results obtained during measurements and integration tests.

Keywords: MAC Measurements, QoS, IEEE 802.11 WLAN, Ad-Hoc

1 Introduction

Wireless Local Area Networks (WLANs) are becoming increasingly popular due to their flexibility and convenience. They are being widely implemented in many venues from hospitals and airports to retail, manufacturing and corporate environments. WLANs are beginning to be available in public spaces such as schools, hotels, restaurants, malls and shops. This technology offers the highest level of performance and capability features among other local wireless solutions. WLANs play a very important role in the network architecture as a provider of easy and unconstrained access to the wired infrastructure. As a consequence, hotspots are appearing all around the globe and in the most different and remote places. This business strategy is both profitable for the provider and the user. Since radio range is strongly affected in closed spaces or in areas with dense radio interferences, the resilience provoked by the multi-hop characteristics of mobile ad-hoc networks, makes these especially appropriate to provide increased radio coverage with low cost and easy deployment. Therefore, ad-hoc networks are also playing an increasing role in network access.

Ad-hoc networks are characterized by very dynamic changes of topologies and hence their design requires deep attention. To support the users and service requirements, the ad-hoc network needs to support differentiated QoS, which is a major challenge. The QoS protocols for ad-hoc networks need to operate distributed along the ad-hoc nodes, with proper mechanisms for reacting in a responsive way to topology changes. There are some proposals for QoS support in ad-hoc networks: SWAN (Stateless Wireless Ad-hoc Networks) [6], INSIGNIA [4], and FQMM (Flexible Quality of service Model for Mobile ad-hoc networks) [5] are only some of the examples.

To address the ad-hoc network integrated with the infrastructure, which is the scenario addressed in this paper to provide increased radio coverage with low cost and easy deployment, the Daidalos project [1] addressed a new proposal, taking SWAN as its basis, that integrates QoS in ad-hoc with infrastructure networks [8]. The SWAN solution uses MAC layer measurements to support the proper level of QoS for two different traffic classes in ad-hoc networks. The work presented in this paper addresses the concept of advanced MAC layer measurement module implementation to provide information on the congestion network status. We also present the tests applied to this module when used in a test-bed ad-hoc network. This solution provides end-to-end QoS for four traffic classes in sessions between ad-hoc nodes and between ad-hoc and non ad-hoc nodes (QoS integration between ad-hoc and infrastructure networks). Furthermore the proposed architecture addresses the use of multipath routing to perform load balancing and increase network reliability.

The paper is organized as follows. The Chapter 2 shows the proposed QoS architecture, the mobile node and gateway node schemes and the main SWAN extensions for ad-hoc network integration. Chapter 3 contains the details about MAC Measurement Module (MMM) software implementation. Its structure in modified hostap [9] driver and functions is showed. The measured QoS parameters are enlisted in Chapter 4. Chapter 5 presents some measurement results from real test-bed where up to 10 stations contended for access to the medium. At the end, we summarize the paper.

2 Proposed QoS Architecture

The proposed architecture addressed in this paper, aiming at providing increased radio coverage in hotspot scenarios, considers ad-hoc networks connected to the Internet through infrastructure access networks. 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 infrastructure network and 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. The QoS integration requires a special network entity, a Gateway (GW) that interconnects the ad-hoc network with the infrastructure. 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.

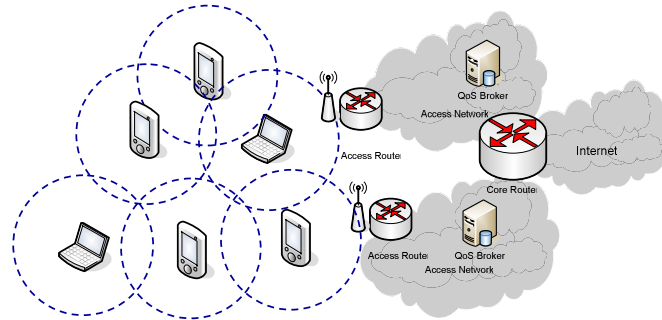


Fig. 1. Ad-hoc Integration architecture

The proposed QoS approach is based on an extension [8] of the SWAN [6] QoS model, and abstracts the ad-hoc path between an ad-hoc node and the gateway as a virtual link in the infrastructure side. Admission control is performed with collaboration of the ad-hoc source nodes, as in normal SWAN protocol. Basic SWAN 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 resorts to dynamic regulation of real-time sessions when congestion/overload conditions occur: when a mobile node detects congestion it starts marking ECN (Explicit Congestion Notification) bits in the IP header of real-time packets; the destination monitors the ECN bits and notifies the source sending a regulate message. The SWAN extensions are briefly addressed in the following subsections.

2.1 QoS functionalities

The QoS functionalities of the mobile node and the GW in this QoS model demand a special design of their QoS stack, which are presented in Fig. 2 and Fig. 3, respectively. The solid lines interconnecting the modules correspond to the data packet processing inside of a node. The dashed lines correspond to control information.

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 retrieval of the application QoS parameters is performed by the QoS client; the Ad-hoc QoS Controller checks for QoS resources along the ad-hoc path and the available resources in its wireless medium. This is performed through the interaction with the MAC Measurement Module. The classification and marking of the packets is addressed in the Classification and (Re)-Marking module, and the QoS differentiation is carried by the Traffic Control (TC) module. To address congestion situations in the ad-hoc network, the node has an ECN Marking module that obtains the congestion status information from the MAC Measurement module and marks the ECN bits to trigger the dynamic regulation of the flows.

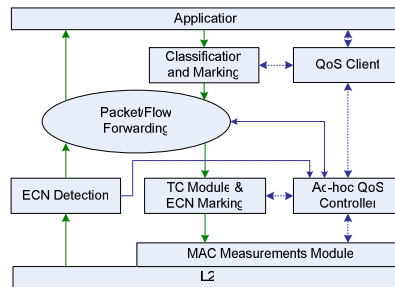


Fig. 2. Mobile Node Scheme

The GW is able to support the same functionalities of the mobile nodes, but does not have interaction with the application signaling (since it works only at the IP layer and below). Instead, it needs to perform interoperation between the QoS signaling in the ad-hoc and the infrastructure side.

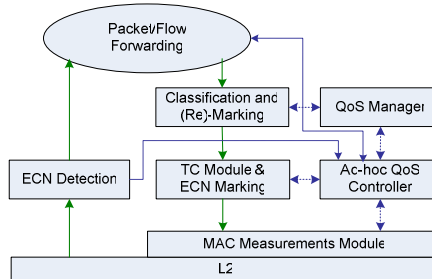


Fig. 3. Gateway Scheme

2.2 QoS Signalling, Dynamic Regulation and Interaction with Multipath

This sub-section makes a brief overview of the QoS signaling extensions performed to SWAN. For more information on these signaling extensions, see [8].

The QoS signaling is performed between ad-hoc nodes and the gateway, in the case of communication with a node in the infrastructure network. The QoS request message (probing request in the SWAN terminology) is received in the gateway (when the sender is in the ad-hoc network). If there are sufficient resources inside the ad-hoc network (through the bottleneck and requested bandwidth fields in this message), 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 message to the receiver. Simultaneously, it also replies to the QoS request with a QoS response message with indication of the available bandwidth in the ad-hoc path. Otherwise, an error message is sent to the sender.

When an ad-hoc node detects an overload condition in a class (target bandwidth for the class exceeded and detected by the MAC measurements module), 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 QoS request process.

To support multipath (with the discovery and maintenance of routes, and control of the path on which data is forwarded) an extension of AOMDV (Ad hoc On-demand Multipath Distance Vector Routing) [8] is used. 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, and the path that will be used is the best one for the flow.

2.3 Extended QoS Differentiation

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 and re-routing.

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. 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. 4.

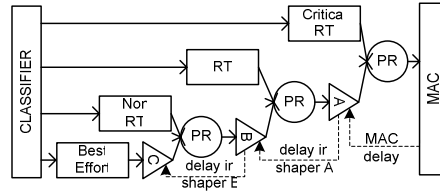


Fig. 4. Extended Differentiation Model

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 (Additive Increase Multiplicative Decrease) 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, through the feedback of measured packet delay, 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. In the 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_i .

<p>if ($n > 0$) $s_i \leftarrow s_i \times (1 - r_i)$ else $s_i \leftarrow s_i + c_i$ if ($(s_i - a_i) > a_i \times g_i$) $s_i \leftarrow a_i \times (1 + g_i)$</p>	<p>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.</p>
---	---

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. Besides that, in the case of congestion situations the higher priority classes are regulated. The bandwidth utilization of each of these classes will be continuously monitored through the MAC Layer measurements. 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.

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). Another advantage of this probabilistic marking is that it can lead to traffic equilibrium between nodes, since the more congested nodes will be more aggressively subjected to regulation.

The best-effort traffic class is expected to convey TCP sessions. Since this class acts as a "buffer zone" to the other traffic classes, its buffer occupancy is expected to severely increase in re-routing situations. A RED (Random Early Detection) queue management discipline should be applied to this class in order to faster regulate TCP sessions to face adverse network conditions. Queues of higher priority classes should also have a RED based queue management discipline. In congestion situations, this queue management will mitigate the delay of non dropped packets and intelligent applications may switch to more appropriate codecs alleviating the congestion.

2.4 Measurements Processing

The Ad-hoc QoS Controller resorts to MAC layer measurements to determine the per-class bandwidth occupancy in the local shared wireless link and the mean delay of the packet transmission to its neighbors, besides other parameters, in order to participate in the distributed ad-hoc resource management and assure the service differentiation. The MAC layer measurements are also used to detect congestion in the wireless medium. This is the main aim of this paper: address the specification, implementation and evaluation of the MAC layer measurements to aid in admission control and dynamic regulation of the extended SWAN proposal.

Fig. 5 illustrates the control of the TC operation. As can be seen in the diagram, the MAC layer measurements module (MMM) periodically sends MAC layer measurements to the Ad-hoc QoS Controller (*Report-MMM*). The TC module also sends periodically queue occupancy reports (*Queue-Information*) to the Ad-hoc QoS Controller. This module computes the reported values and updates the configuration of the TC module (*TC-Rate-Conf*), as described in the extended SWAN service differentiation (sub-section 2.3).

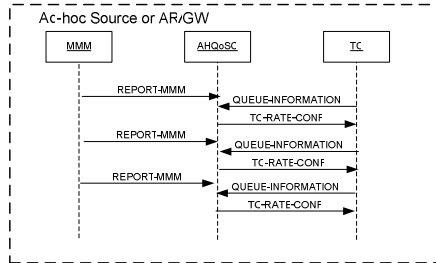


Fig. 5. MMM information for TC configuration

The detection of QoS violations is performed by the Ad-hoc QoS Controller based on the local (shared medium) per-class bandwidth utilization and delay obtained from the MMM (*Report-MMM*). The Ad-hoc QoS Controller detects the class that is being violated and instructs the ECN Marking module to mark the ECN bit of packets (*ECN-Mark-Request*) belonging to the class where the resource violation was detected (Fig. 6).

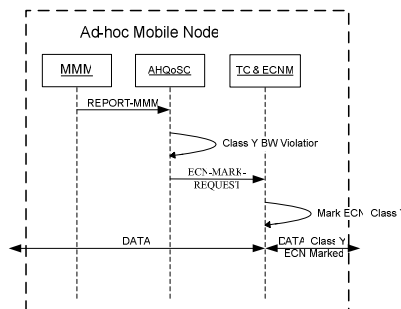


Fig. 6. MMM information for ECN detection

The next sections detail the implementation and evaluation of the elements that provide the shared medium measurements obtained in the ad-hoc network.

3 Software Implementation of MAC Layer Measurement Module

MAC layer Measurement Module has been implemented in C language as part of wireless card driver operating in Linux kernel space. We consider that the mobile nodes will use the IEEE 802.11b standard. IEEE 802.11e [10] is an alternative when the WLAN cards with support for EDCA under Linux 2.6 will be available. In the presented architecture we have resorted to Linksys Instant Wireless Cards WPC11 based on Intersil PRISM version 3 chipset with station firmware v1.7.4. This card was selected because the detailed PRISM driver programmer manual can be obtained from Intersil. There are three different open source drivers for WLAN cards based on PRISM chipset available today for the Linux OS. We chose HostAP driver, because it is the most documented driver (its mailing list has more than 40 MB of posts) and it has all required functions.

The structure of this module is presented in Fig. 6. MMM is located between the IP layer and the wireless card firmware. In the presented solution, MMM has been included in HostAP version 0.2.6 wireless driver working under Mandrake Linux with kernel 2.6.8-1. This approach allows for quick and efficient operation on packets transmitted and received between IP and MAC layer. We avoided the use of additional libraries for packet capture operating in layers higher than MAC which could compromise the performance and the results reliability. The placement of the measurement module in the device driver provides the necessary accuracy and makes possible code changes easier. An alternative idea would be to implement MMM directly in wireless card firmware. However, the functionality of this solution is worse regarding software accessibility.

Every packet incoming from the IP layer is analyzed by MMM for the sake of its Differentiated Services Code Point (DSCP) code. The time instant of each packet reception is recorded and the packet is then sent to the wireless card. The packet transmission time, obtained by the measurements, is the time between packet reception in MMM and positive MAC layer acknowledgement reception from receiver station. The later means that packet is received correctly. Thus, for each packet, MMM measures the time the packet is stored in the wireless card memory, increased by the station's contention time and data frame transmission time in DCF mode – sequence RTS, CTS, DATA and ACK frames duration. MMM allows for measurements precision of a few microseconds which seems sufficient for the presented application. For example, the propagation of single 1500 byte data frame takes about 1.1 ms using transmission rate 11 Mbps, or 12 ms using 1 Mbps.

In order to assure that the admission control is operating properly, it is required to provide information about unused network resources to the decision module. This is accomplished by measuring the bandwidth utilization represented in bps. To allow this functionality, the wireless card is set to promiscuous mode. In this mode, the wireless card is capable of receiving all MAC frames (except management frames), also addressed to any other wireless cards. The advantage of this method over using monitor mode is that only one wireless card is required, which is common in most wireless devices. For monitor mode, two cards are required: one for monitoring which cannot transmit at the same time, and a second one for simultaneous transmission. With the following two pieces of information, the current transmission rate and the bandwidth utilization, we are able to determine the amount of unused resources.

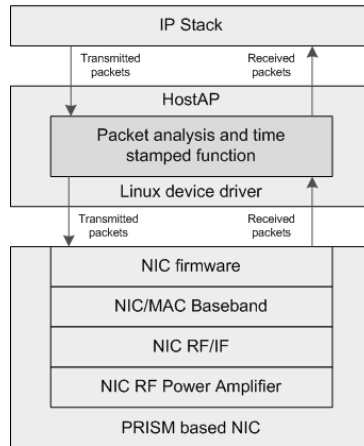


Fig. 6. MAC Layer Measurement Module structure and location

4 Supporting QoS by MAC Measurements

MMM measurement module allows the collection of information concerning overall bandwidth utilization in the wireless channel, average transmission delay for outgoing frames, number of frames transmitted, received, lost and exceeding a given QoS delay threshold. The samples are periodically reported and, additionally, the exact time intervals of the samples are determined and reported in each sampling procedure for the accuracy of the calculations. Also, idle intervals, where no transmission occurs, are calculated and reported as well. All these parameters enlisted are collected for each defined traffic class as well as for entire traffic in the wireless channel. In the later case, the summary is larger than the direct sum of all components, because there is also some non-categorized traffic which uses network resources. Note that the number of stations is not predefined but also measured by our module.

Considering MMM Station's outgoing data, various detailed parameters are collected for each frame: ID, times of transmission, average delay, DSCP code, destination MAC. However, due to processing considerations, only the 100 last frames statistics are recorded, which effectively corresponds to a transmission history of at least 30 ms at full transmission rate of 11 Mbps. This is a calculation assuming that small 50 byte frames are being transmitted. Actually, due to medium access technique using CSMA/CA access and RTS-CTS-DATA-ACK transmission scheme, even small frame transmission takes at least 2 ms, which gives a total of 200 ms history. An example of frame statistics is showed in Table 1 below.

Table 1. An example of Transmitted Frames Statistics. Collected statistics cover all information for identifying transmitted frames. Transmission time is computed using two preceding timestamps. Although this value is in microseconds we assume lower accuracy but still suitable for MAC measurements. DSCP value of 0 is default for IPv4 and non-class traffic, 96 is default value for IPv6 packets. For setting other values it is required to use DSCP marking module. Valid transmission rates for IEEE 802.11b are 1, 2, 5.5 and 11 Mbps. Hexadecimal MAC addresses are limited here to 24 bytes (of 48) due to limited space

Frame ID	Tx start [μs]	Tx stop [μs]	Delta [μs]	Length [bytes]	DSCP Code	Rate [Mbps]	MAC Address
0193	20452974	20458100	5126	1536	0	11	2B-2D-21
0194	20458100	20461379	3279	1536	0	11	2B-2D-21
0195	20459379	20462532	3153	1536	0	11	2B-2D-21
...							
0288	76474071	76476346	2275	1084	96	11	26-94-85
0289	76871204	76873658	2454	1084	96	11	26-94-85
0290	77087372	77089761	2389	1084	96	11	26-94-85
0291	77287851	77290427	2576	1084	96	11	26-94-85
0292	77444820	77447267	2447	1084	96	11	26-94-85

MMM uses the above statistics to consecutively compute QoS parameters. In order to achieve this functionality it is important to filter all frames which are not exactly data frames. This includes broadcast and multicast frames. QoS parameters are enlisted in Table 2. MMM builds report messages concerning overall and per-class (four classes) statistics.

Table 2. MMM Report Message. Three initial parameters are common for report. Remaining statistics are duplicated for overall statistics and four traffic classes

Parameter	Description
interval	Report Interval
nrStations	Number of Stations in Network
rate	Current Transmission Rate
DSCP threshold	DSCP Code assigned to given traffic class QoS Threshold defined for given class
avgDel	Average Frame Transmission Delay
nrTx	Number Frames Transmitted
nrExc	Number Frames Transmitted and exceeding QoS Threshold
nrLost	Number Frames Lost
nrRx	Number Frames Received
Bw	Bandwidth Utilization
idleTx	Time since last frame transmission
idleRx	Time since last frame reception

Table 3. MMM Configuration Message

Parameter	Description
MMM_Interval	Time interval for MMM reporting
ov_Threshold	QoS Threshold defined for overall statistics
DSCP_Codes[4]	List of DSCP codes for all classes
Thresholds[4]	List of QoS Thresholds for all classes

MMM is fully configurable with a single configuration message, as shown in Table 3. This provides modification of MMM behavior or its adjustment to the current network state. In typical operations, this message is triggered by Ad-Hoc QoS Controller in mobile node or the gateway. The communication between MMM

module in kernel space and these modules in user space is accomplished using *netlink* sockets.

5 Measured QoS Parameters

The measurement module has been tested in homogeneous IEEE 802.11b environment. The primary goal of MMM is to directly support higher layer modules in order to get proper admission control. We have built an ad-hoc network consisting of ten mobile nodes and two wireless stations as in Fig. 7. In our testing topology, we have run large volume traffic FTP applications due to their aggressive resources utilization. One station acts as an FTP Server while the MMM measurement module is installed on another station. That MMM Station requires Prism wireless cards, while all the other nodes use their default drivers. After connection establishment, MMM Station is capable of measuring wireless channel parameters: the user connected to this station can take advantage of this possibility.

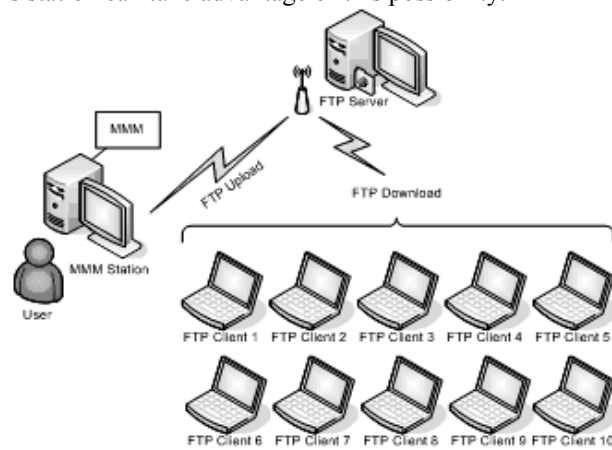


Fig. 7. Testing Configuration of IEEE 802.11b Ad-hoc Network

The behavior of wireless nodes is as follows: one to ten FTP Clients start their download sessions to reflect mass traffic and reach the network capacity. The MMM Station could behave in the same way but, taking its additional outgoing frames recording into consideration, its behavior is set to FTP upload transmission. Even though FTP Server is the bottleneck in this scenario, the server and all the nodes remain in the same ad-hoc network during the whole experiment. In fact, the same functionality is required in the managed infrastructure network, and the server can be easily replaced by any access point in these considerations.

We collect the measurements obtained in the presented ad-hoc topology. The FTP upload from MMM Station always starts after the connections establishment of other stations. Thus, MMM traffic always has to compete for network resources. We have run two experiments: one without MMM Station's transmission to evaluate and adjust bandwidth measurements, and a second test with MMM Station transmitting FTP data

in order to evaluate this another kind of measurements. The intention of the following graphs described in this section is to show the measured samples as received by QoS controllers, and not any exact characteristics.

Each single experiment has 140 seconds of transmission in the established state; the report interval has been set to 30 ms. The report interval has to be carefully chosen. This value must be set low enough to allow efficient admission control. However, having in mind that transmission of single frames takes at least 2 ms, it cannot be too small in order to avoid errors during samples averaging.

5.1 Bandwidth Utilization

We compare the distribution of the bandwidth samples over the transmission time. The bandwidth utilization BW including physical layer overhead and protocol inter-frame space times can be estimated owing to changing transmission rates using the following formula:

$$BW = \frac{L_{DATA}}{T_{DIFS} + 3 \times T_{SIFS} + \frac{L_{DATA}}{R_{DATA}} + \frac{L_{RTS} + L_{CTS} + L_{ACK}}{R_{BR}} + \frac{4 \times L_{PPLCP}}{R_{PPLCP}} + \frac{4 \times L_{HPLCP}}{R_{HPLCP}}} [bps] \quad (1)$$

where:

T_{DIFS} is the DIFS time;

T_{SIFS} is the SIFS time;

L_{DATA} is the length of data frame;

L_{RTS} , L_{CTS} , L_{ACK} are lengths of RTS, CTS, ACK frames;

L_{PPLCP} , L_{HPLCP} are lengths of PLCP preamble and header;

R_{DATA} is the payload rate;

R_{BR} is the basic rate;

R_{PPLCP} , R_{HPLCP} are PLCP preamble and header rates.

Fig. 8 presents the average bandwidth and delay over all 140 s considering 0 to 10 background sessions which equals to the number of active mobile stations FTP1..10; in this experiment, MMM Station is inactive. Although bandwidth parameter changes unpredictably, the constant decrease can be clearly observed.

5.2 Transmission Delay

In this subsection we address the period of time the frame being sent by MMM Station waits in the wireless card cache until it is transmitted. This waiting period includes frame storage in wireless card memory, transmission time and possible MAC layer retransmission or multiple retransmissions. If the frame is lost or corrupted in some way, then no valid acknowledgement (ACK) is received and the overall delay increases.

In a non congested network, the average transmission time varies from 3 to 4 ms as seen in Fig. 9a, although a single delay of around 30 ms has been observed. For quality tests, the delay threshold has been set to 3.5 ms. In this scenario, half of the frames slightly exceed that threshold. Adding more background sessions of mass traffic is a potential problem for efficient QoS accomplishment, as they can dramatically worsen the channel conditions. We observe in Fig. 9b that the transmission delays are much larger than the ones of separated sessions. Along with the increase in the number of stations, the channel properties worsen, but with lower impact as between one and two stations (Fig. 10). The delay samples peak reach 300-950 ms; if the frame is not acknowledged after MAC layer retransmission, the proper timeout indicator will be set and the number of lost frames will be recorded (as seen in Table 2). Fig. 8 shows also the averaged values of transmission delays for different mobile stations in this experiment. We observe a near linear growth in the average delay along with increase in the number of stations (with the number of stations larger than one).

5.3 Transmitted Frames

Assuming that accurate outgoing frames statistics are necessary for QoS support, we measure the number of frames exactly exceeding a given delay threshold, as well as the number of transmitted frames. For non congested network, the number of frames exceeding QoS threshold is about half of all transmitted frames, as presented in Fig. 11a. When at least two connections coexist in the same network, the number of exceeding frames falls along with the amount of successfully transmitted data (Fig. 11b). It can be concluded from Fig. 10 that the rate of frames exceeding the threshold is not strongly dependent on the number of active stations. Instead, the time of exceeding frame transmission is much longer in a congested network (recall Fig. 9). In fact the whole increase in the average delay from 4 ms to 16 ms, seen in Fig. 8, is caused by frames which do not meet the QoS requirements.

5.4 Number of Active Stations in Ad-hoc Network and Transmission Rates

Although the exact number of active stations is known in advance in this experiment, in real applications it might not be the case. Therefore, this information can be important to evaluate the contention. Also, last transmission rate and times of last transmission are recorded for each station. While rate allows for estimation of a given station limit on bandwidth used, the idle time is indicator if the station should be considered inactive and its entry removed from the table. Example estimation is presented in Table 4 below.

Table 4. An example of active stations estimation. Number of estimated stations $nrStations=11$

MAC Address	Idle Time [μ s]	Rate [Mbps]
00-06-25 EB-ED-42	17037	11
00-06-25 EB-E6-44	2620	11
00-0C-F1 13-F6-71	29700	11
00-0C-F1 13-F4-B8	77941	11
00-0D-54 99-0B-80	93418	11
00-0D-54 99-0D-14	12302	11
00-06-25 DB-DC-D8	75411	11
00-0D-54 99-0B-D2	131350	11
00-0D-54 99-09-C5	295201	11
00-0D-54 99-0B-73	39088	11
00-0D-54 99-10-75	4863	11

5.5 Transmission Rate

The current transmission rate for the station using MMM is measured to calculate bandwidth utilized: product of the time of MMM transmission multiplied by transmission rate adds to overall bandwidth calculation. This parameter is substituted for R_{DATA} , see subsection 5.1.

5.6 Traffic differentiation scenario

This test shows the MMM module measurements of traffic with four different class flows. MAC layer measurement module (MMM) should report to AHQoS module overall and per class bandwidth as well as overall and per class delay. These measurements are necessary for admission control decisions and proper TC module control. The overall bandwidth and delay should be a sum of classes and non-classes traffic. This test was performed with all ad-hoc QoS modules present and working.

Results presented in Figures 12 and 13 were obtained in a three-hop ad-hoc scenario. The presented measurements were obtained from MMM report in the source node. Mgen tool was used for background traffic. Mgen was setup with four distinct traffic flows (one in each contemplated traffic class). The measures were collected for about 25 minutes. Each flow sent one 512-bytes packet per second. In Figure 12 it can be observed that every class utilizes the bandwidth allowed through its priorities. The same happens with the delay values.

The mean packet delay considering all traffic classes is 2ms, which shows that the network was non-saturated, see Figure 8.

6 Conclusions

In this paper we presented the implementation of the measurement module, allowing for MAC layer traffic parameters measurements. In consecutive work, differentiation of results obtained for each traffic class allow for implementation of efficient mechanism providing QoS in IEEE 802.11b wireless networks. It can be crucial for access points operation in particularly congested areas as restaurants, hotels, airports. The architecture presented in this paper extends the scope originally proposed in SWAN model, adding the integration of different access technologies, routing and load balancing. An obvious and practical advantage over SWAN is that the whole system uses only one wireless card per station, thus it can be easily deployed in modern networks. Our plans also include the quick addition of the MMM functionality to other standards, the popular IEEE 802.11g and IEEE 802.11e.

Our future plans also concern the verification of developed architecture by performing simulations and by the realization of field tests, as well as integration with security and charging mechanisms. In further tests we plan to also consider higher layers self-similar traffic patterns for better analysis of wireless environment.

References

1. Daidalos IST Project: Designing Advanced Interfaces for the Delivery and Administration of Location independent Optimised personal Services. EU Grant No FP6-2002-IST-1-506997. <http://www.ist.daidalos.org>
2. IEEE 802.11 Standard for Wireless LAN: Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Inc., New York (1999)
3. IEEE 802.11b: High-Speed Physical Layer Extension in the 2.4 GHz Band, IEEE Inc., New York (1999)
4. Lee S. B. et al.: INSIGNIA: An IP-Based Quality of Service Framework for Mobile Ad Hoc Networks J. Parallel and Distrib. Comp., Special issue on Wireless and Mobile Computing and Communications, vol. 60 n°4 (2000) 374-406
5. Xiao, H., Seah, W.K.G., Lo, A., and Chua, K.C.: A Flexible Quality of Service Model for Mobile Ad-Hoc Networks. In Proceedings of the IEEE Vehicular Technology Conference, Tokyo (2000) 445-449
6. Ahn, G.S., Campbell, A.T., Veres, A., Sun L.H: *Supporting Service Differentiation for Real-Time and Best-Effort Traffic in Stateless Wireless Ad Hoc Networks (SWAN)*. In IEEE Transactions on Mobile Computing, vol. 1, no. 3, (2002) 192-207
7. Sargento, S., Calçada, T., Barraca, J.P., Crisóstomo, S., Girão, J., Natkaniec, M., Vicari, N., Cuesta, F., Ricardo, M., Glowacz, A.: Mobile Ad-Hoc Networks Integration in the Daidalos Architecture. In Proceedings 14th IST Mobile & Wireless Communications Summit, Dresden, Germany, (2005)
8. Crisóstomo, S., Sargento, S., Natkaniec, M., Vicari, N.: A QoS Architecture Integrating Mobile Ad-Hoc and Infrastructure Networks. Workshop on Internet Compatible QoS in Ad hoc Wireless Networks, Cairo, Egypt (2005)
9. Malinen, J.: Host AP driver for Intersil Prism2/2.5/3, <http://hostap.epitest.fi>
10. IEEE 802.11e: Medium Access Control (MAC) Quality of Service (QoS) Enhancements, draft version 11.0, IEEE Inc., New York (2004)

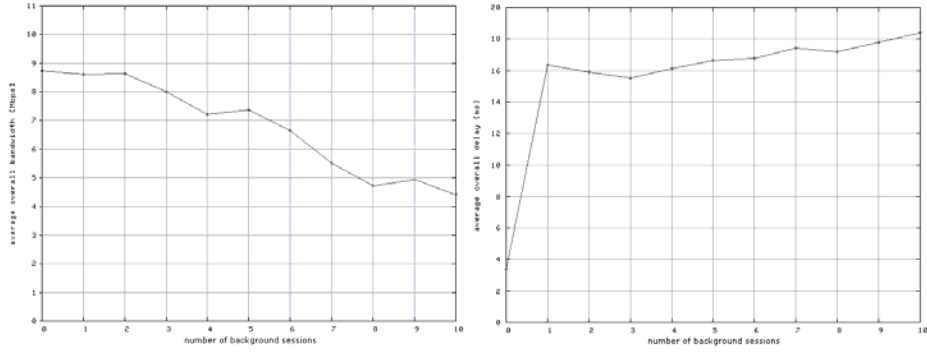


Fig. 8. Overall average bandwidth and delay for different background sessions number

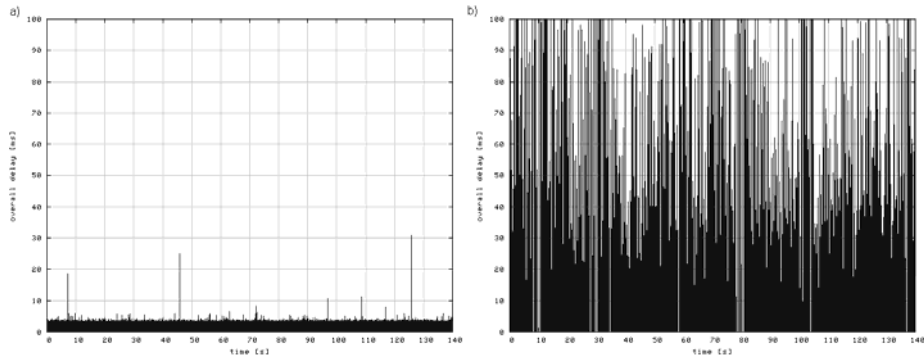


Fig. 9. Average frame delay for: a) single session; b) session with ten background transmissions

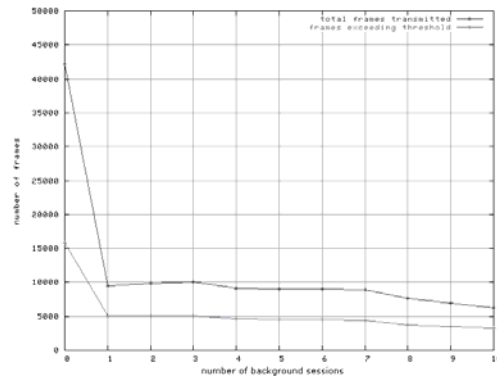


Fig. 10. Total number of frames transmitted and exceeding QoS threshold

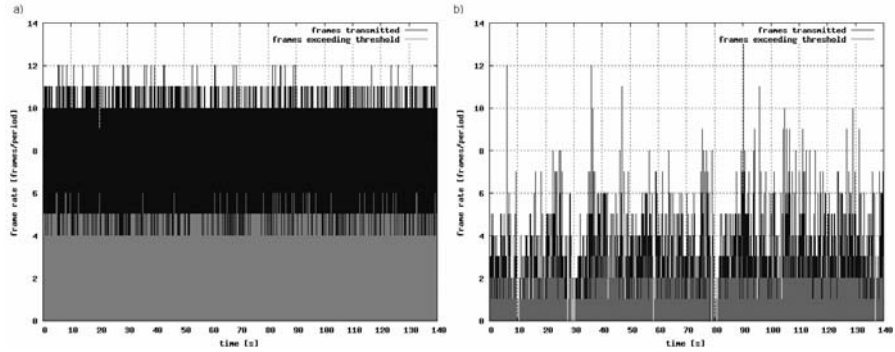


Fig. 11. Number of frames transmitted and exceeding QoS threshold for: a) single session; b) session with ten background transmissions

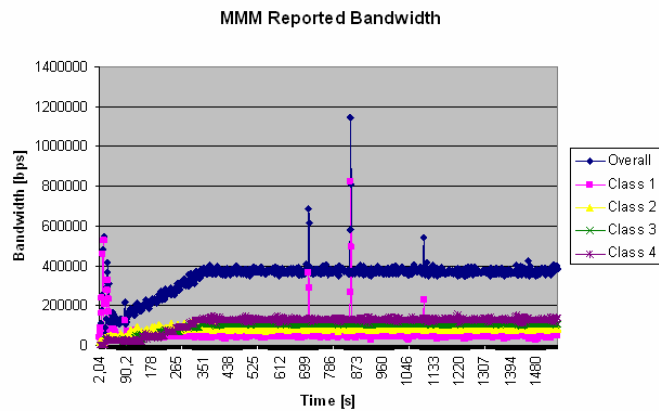


Fig. 12. Overall and per class bandwidth reported by MMM during test

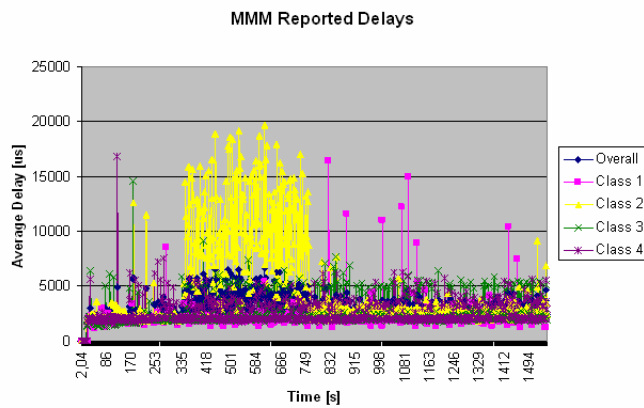


Fig. 13. Overall and per class frame delays reported by MMM during test