

QoS Support in MANETs: A Modular Architecture Based on the IEEE 802.11e Technology

C. T. Calafate, M. P. Malumbres, *Member, IEEE*, J. Oliver, J. C. Cano, and P. Manzoni, *Member, IEEE*

Abstract—Providing quality-of-service (QoS) in wireless *ad hoc* networks is an intrinsically complex task due to node mobility, distributed channel access, and fading radio signal effects. This goal can be successfully accomplished only through the cooperation of the different protocol layers involved. In this paper we propose a novel QoS architecture that is able to support applications with the bandwidth, delay, and jitter requirements in MANET environments. The proposed architecture is modular, allowing the plugging in of different protocols, which offers great flexibility. Despite its modularity, we propose optimizations based on interactions between the media access control (MAC), routing, and admission control layers which offer important performance improvements. We validate our proposal in scenarios where different network loads, node mobility degrees, and routing algorithms are tested in order to quantify the benefits offered by our QoS proposal. In particular, we have also used real H.264/AVC video traces to simulate video sources in order to measure the quality in terms of peak signal to noise ratio of the received video, so that the benefits of applying our QoS scheme to video sources can be assessed in terms of user satisfaction (from the applications perspective).

Index Terms—Cross-layer optimization, distributed admission control, QoS architecture for MANETs.

I. INTRODUCTION

MOBILE *AD HOC* NETWORKS, which are also known as MANETs, are composed of independent mobile terminals that communicate wirelessly to conform to a network. All nodes within an *ad hoc* network provide a peer-level multihop routing service where all nodes simultaneously act as both traffic sources/sinks and as traffic forwarders.

Research in the *ad hoc* networking field has received much attention recently because these networks offer many benefits, such as self-reconfiguration and adaptation to highly variable characteristics—power and transmission conditions, network topology and traffic load—without requiring a fixed infrastructure. When attempting to enhance these networks to support applications with quality-of-service (QoS) requirements, we find that there is still more research to

be done. At the physical layer there are interference problems, adaptive data rates, and fading signal effects, which make it very difficult to offer QoS guarantees—especially when operating on free radio bands, such as ISM (industry, science and medicine) bands. At the MAC layer, we have distributed channel access which causes the well-known hidden and exposed node problems. Such problems make bandwidth reservations a complicated issue (“coupled capacity” problem) that was proved to be NP-hard [1], [2]. At the network layer, routing protocols have to deal with fast topology changes and, if QoS support is required, they must discriminate among the available paths to meet the QoS requirements. At upper layers, applications should be MANET-aware to adapt their behavior and thus improve performance.

In this paper we propose a flexible architecture for MANETs that is able to offer end-to-end QoS support to mobile *ad hoc* network environments. Our proposal builds upon the IEEE 802.11e standard [3] by adding a probe-based admission control system, along with an enhanced version of the dynamic source routing (DSR) protocol [4] to make it efficient also at high degrees of mobility. The proposed architecture includes cross-layer optimizations to improve the performance of the different protocols that conform to it. Since our proposal is based on off-the-shelf wireless devices and optimizations to widely available routing protocols, its development and deployment can be done within a short period of time. It can also be used in heterogeneous MANETs where not all the terminals participate in QoS tasks, while maintaining a high degree of effectiveness in homogeneous MANETs. In terms of performance assessment, in our experiments we have used real video traces encoded with the H.264/AVC [5] codec to obtain performance results at the user level.

This paper is organized as follows. In the next section, we refer to significant related works in the field. In Section III we offer an overview of the proposed QoS architecture. We offer the details on the most relevant architectural elements in Sections IV (admission control layer) and V (routing layer). We then proceed to show the experimental results obtained in Section VI. Finally, in Section VII we present our conclusions.

II. RELATED WORK

Despite the issue of QoS support in MANETs is a relatively novel subject, it has recently received much attention from researchers worldwide. In the literature we can find works that focus on QoS issues related to a single protocol layer (e.g., MAC layer, routing layer) along with works that propose a QoS framework that combines more than one layer. We now proceed to detail some relevant works on each of these areas.

Manuscript received October 24, 2007; revised July 21, 2008. First version published March 16, 2009; current version published June 10, 2009. This work was partially supported by the *Ministerio de Educación y Ciencia*, Spain, under grants TIN2005-07705-C02-01 and DPI2007-66796-C03-03, and the *Universidad Politécnica de Valencia* under programme PAID-06-07. This paper was recommended by Associate Editor W. Zhu.

C. T. Calafate, J. Oliver, J. C. Cano, and P. Manzoni are with the Department of Computer Engineering at the *Universidad Politécnica de Valencia*, Spain (e-mail: calafate@disca.upv.es; joliver@disca.upv.es; jucano@disca.upv.es; pmanzoni@disca.upv.es).

M. P. Malumbres is with the Department of Physics and Computer Engineering at the *Miguel Hernandez University*, Elche, Spain (e-mail: mels@umh.es).

Digital Object Identifier 10.1109/TCSVT.2009.2017405

In terms of MAC layer protocols for *ad hoc* networks, the IEEE 802.11 Working Group E [3] has recently completed a new MAC standard, also denoted as *IEEE 802.11e*, to enhance Wi-Fi networks with QoS support. In [6] Romdhani *et al.* propose enhancements to the IEEE 802.11e technology to offer relative priorities by adjusting the size of the contention window (CW) of each traffic class, taking into account both applications requirements and network conditions. Sobrinho and Krishnakumar propose Blackburst [7], which is a novel distributed channel access scheme that is more efficient than the IEEE 802.11e technology. Other works such as [8]–[10] also propose alternate QoS MAC schemes designed specifically for *ad hoc* network environments.

Concerning routing layer proposals offering QoS support in MANETs, Lin and Liu [11] propose a QoS routing protocol that includes end-to-end bandwidth calculation along with bandwidth allocation schemes. Shigang and Nahrstedt [12] define a distributed QoS routing scheme that selects a network path with sufficient resources to satisfy a certain delay (or bandwidth) requirement. In [13], Xue and Ganz propose a resource reservation-based routing and signaling algorithm (AQOR) that provides end-to-end QoS support in terms of bandwidth and delay. Also, Chen and Heinzelman [14] propose a QoS-aware routing protocol that incorporates admission control and feedback schemes to meet the QoS requirements of real-time applications by offering an estimate of available bandwidth.

Concerning QoS frameworks for MANETs, Lee *et al.* propose INSIGNIA [15], an approach to integrated services support in MANETs through a flexible signaling system. Ahn *et al.* propose SWAN [16], an approach to differentiated services support in MANETs using plain IEEE 802.11 plus rate-control for best effort traffic; traffic acceptance is dependent on local bandwidth estimations and admission control probes.

More recently, the issue of cross-layer QoS solutions has received much attention. Concerning proposals using directional antennas, Hamdaoui and Ramanathan [17] exploit the benefits of MIMO antennas to enable multihop wireless networks with flow-level QoS capabilities. Li and Man [18] propose a multipath routing scheme including an analytical and numerical analysis of link breakages in a multihop directional network, along with a differentiated service (DiffServ) framework for layered video transport using QoS-aware multipath routing.

Other cross-layer QoS proposals including multipath routing schemes are [19]–[21]. Kompella *et al.* [19] develop a formal branch-and-bound framework to determine how to perform multipath routing for multiple description (MD) video in a multihop wireless network. Loscri ; *et al.* [20] propose a QoS multipath routing scheme called QAOMDV which can take advantage of both a well-known multipath routing protocol (AOMDV) and an evolutionary and distributed TDMA MAC layer, called the E-TDMA. Finally, Shiang and Schaar [21] present a distributed cross-layer streaming algorithm for the transmission of multiple videos over a multihop wireless network whose essential feature is the use of priority queuing.

Our proposal differs from the existing proposals since it can operate on top of any routing protocol (both single and multipath) and it completely avoids resource reservations,

while adequately supporting applications with bandwidth, delay, and jitter requirements under high mobility. Instead, periodic end-to-end assessment of path conditions through packet probes is used, thus offering a much more flexible QoS framework that adapts to MANETs with heterogeneous terminals. Notice that our probing technique measures network resources directly, contrary to most works that use probes merely to convey measurements/decisions from intermediate nodes on the path (e.g., SWAN [16]). Additionally, our evaluation is done using the novel H.264/AVC video coding standard to obtain user-level performance indexes.

III. OVERALL QoS ARCHITECTURE

In this section we introduce our modular QoS architecture supporting real-time applications in MANET environments. The purpose is to provide a flexible framework offering end-to-end QoS support to *ad hoc* networks that is both efficient and easily deployable with currently available technology.

Our architecture does not rely on intermediate stations along an end-to-end path for admission control or signaling purposes, avoiding resource consuming tasks such as continuous channel measurements, traffic shaping, and resource reservation. By restricting requirements to a minimum, we are able to use devices with reduced computing power—e.g., PDAs—without performance degradation.

As shown in [7], MAC-level QoS support is a *sine qua non* condition for global QoS support with distributed channel access. Though other QoS-enabled MAC layers, such as Blackburst, could be used, in our architecture we rely on the IEEE 802.11e technology due to its widespread availability in commercial products [which typically implement Wi-Fi Multimedia (WMM), a partial implementation of this standard which offers enough functionality for our purposes].

The IEEE 802.11e standard defines four Access Categories: *Voice*, *Video*, *Best Effort*, and *Background*, offering prioritized channel access. By mapping IP ToS values to MAC priorities, we are able to achieve enhanced distributed coordination function (EDCF)-based traffic differentiation [3] in a simple and straightforward manner.

Another important design issue is the impact of mobility on real-time streams. Though such aspect is often disregarded when focusing on QoS issues, we have shown [22] that high connection disruption times are prone to occur in MANET environments, which have a very negative influence on the fluidity of a real-time session as experienced by users. In that previous work we have also shown that multipath routing is effective at solving this problem. So, within our framework, improvements at the routing layer do not aim at meeting QoS requirements but focus, instead, at minimizing the frequency and duration of communication disruptions due to mobility.

In terms of admission control, Georgiadis *et al.* [1] have shown that performing resource reservations in multihop wireless environments is an NP-hard problem, even under simplified rules for bandwidth reservation. This means that the per-node local measurements do not offer enough data for end-to-end bandwidth reservation, which makes the implementation of bandwidth reservation schemes for MANETs difficult

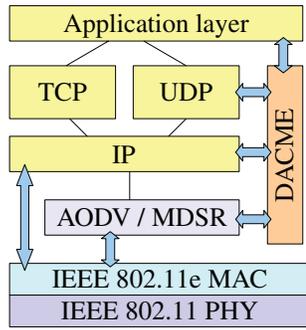


Fig. 1. Diagram of the proposed QoS architecture, including cross-layer interactions.

(e.g., the one proposed in the INSIGNIA [15] framework). So, we complete our QoS framework by including a novel admission control system based on end-to-end probes that avoids making strict bandwidth reservations, thereby offering soft QoS guarantees to real-time flows. Within our framework, this element is referred to as distributed admission control for MANET environments (DACME).

The different architectural elements, shown in Fig. 1, conform to a modular QoS architecture characterized by several cross-layer optimizations among its components; these inter-dependencies are represented in Fig. 1 as double-headed arrows. Throughout the paper we will offer details about the different cross-layer optimizations proposed.

In the next two sections we will describe the novel admission control and routing protocols that integrate our QoS architecture, and we then proceed to validate the feasibility of the global QoS proposal experimentally.

IV. DISTRIBUTED ADMISSION CONTROL ELEMENT

The main component of our QoS architecture is DACME, a probe-based admission control mechanism that performs end-to-end QoS measurements according to the QoS requirements of multimedia streams. In order to operate under optimal conditions in IEEE 802.11-based MANETs, it is recommended that all radio interfaces are IEEE 802.11e enabled. However, this is not a strict requirement since DACME will still operate correctly independently of the MAC layer used. In terms of the software required for MANET nodes, the sources and destinations of QoS flows must have a DACME agent running. The rest of the nodes will simply treat DACME packets as regular data packets, being unaware of the mechanism itself.

Concerning DACMEs components, Fig. 2 shows the functional block diagram of a DACME agent. The main elements of DACME are the *QoS measurement module* and the *packet filter*. The QoS measurement module is responsible for assessing QoS parameters on an end-to-end path, while the *packet filter* blocks all traffic that is not accepted into the MANET according to these end-to-end measurements.

An application that wishes to benefit from DACME must register itself with the DACME agent, indicating the desired destination IP address and the source and destination UDP ports, along with a QoS specification (Q_{SPEC}), stating the requested bandwidth, delay, and jitter: (B_R, D_R, J_R). If any among the available bandwidth, the end-to-end delay, or the

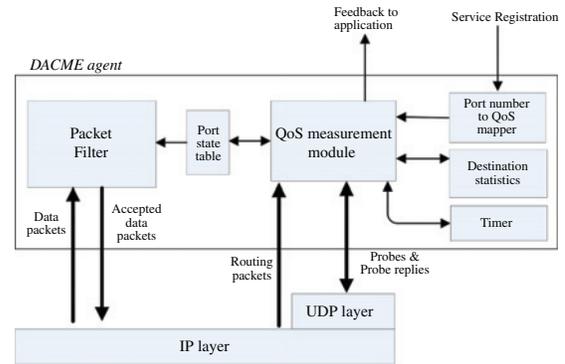


Fig. 2. Functional block diagram of the DACME agent.

jitter values does not meet the application's requirements, DACME will notify this event to the application.

Once registration is successfully completed, the QoS measurement module is activated; it will periodically perform path probing between the source and destination. The purpose is to assess if the path can meet the QoS requirements (Q_{SPEC}), which may be defined in terms of end-to-end bandwidth, delay, and/or jitter. The destination agent, upon receiving probe packets, will update the destination statistics table where it keeps per-source information of the packets received during the current probing period. After receiving the last packet of a probe (or if a timeout is triggered), the destination agent will send a reply back to the source DACME agent. The QoS measurement module, upon receiving each probe reply, will update the state of the path using per-connection bandwidth, delay, and jitter flags. Once enough information is gathered, it checks all the registered connections towards that destination, and then decides whether a connection should be accepted, preserved, or rejected, updating the Port state table accordingly (defining a connection status flag as accept or drop). In the scope of a single terminal, if only part of the registered connections can be allowed, preference is given to those that have registered first.

QoS support becomes effective when the packet filter module, according to the port state table, interacts with the IP layer by configuring the TOS header field of packets pertaining to accepted data flows. The IEEE 802.11e MAC must then map the service type defined in the IP TOS packet header field to one of the four MAC access categories that it makes available.

We now comment on the interactions between DACME and the routing and MAC layers. Afterwards, we proceed by defining the bandwidth, delay, and jitter probing algorithms.

A. Interaction Between DACME and the Routing Protocols

The DACME agent can benefit from routing layer information to assess the current state of end-to-end paths, avoiding probe packets when no path is available. It can also measure the QoS of new paths as soon as they become available through route discovery procedures. The assessment of routing states can be done by communicating directly with the routing agent, or by intercepting routing packets arriving through the wireless interface.

Concerning the interaction between DACME and reactive routing protocols such as *ad hoc* on-demand distance vector

AODV [23], we achieve optimum performance by re-assessing the end-to-end QoS conditions as soon as a routing *RREP* message from a destination of a QoS flow is received. Such a message indicates that a new path to that destination is available, and so the admission control mechanism is able to react earlier if the new path cannot meet the QoS requirements.

Concerning DSR [4] and our multipath extension to DSR (MDSR), presented in Section V, similar strategies do not offer significant benefits due to the intensive use of cache made by these routing protocols. Still, other cross-layer optimizations between DACME and MDSR are required to achieve optimum performance; as an example, the calculation of optimum timeout values at the receiver DACME agent must take into account that traffic is arriving through multiple paths. The source DACME agent must also take multipath routing into account when estimating end-to-end delay.

As a final remark, we wish to emphasize that the routing protocol remains agnostic about the functioning of DACME and, in the case of AODV and MDSR (non-QoS-aware routing protocols), it remains agnostic about QoS traffic too.

B. Interactions Between DACME and the IEEE 802.11e Layer

The QoS strategy proposed in DACMEs framework requires MANET stations to handle packets according to the priority tagging in their IP header. Similar to data packets, probe packets should be handled by the MAC layer according to their priority (using the IP TOS header field). Although the IEEE 802.11e MAC layer offers four ACs, only two of them (voice and video) are adequate for real-time services. In our framework, to avoid the stolen bandwidth problem described in [24], we assigned bandwidth probing packets to the video access category (AC_VI). Relative to end-to-end delay and jitter probing packets, these should be assigned to the same access category as the application packets, so that measurements are accurate and meaningful. As a final remark, the contention-free bursting mechanism—part of the IEEE 802.11e functionality—should be turned off in order to avoid jitter peaks and, more importantly, to make the probing measurement process more reliable.

C. Bandwidth Probing

Relative to the support for bandwidth-constrained applications (B_R), DACME relies on an end-to-end bandwidth probing process, which consists of sending probes to the destination periodically. In our experiments we have set the inter-probe time (IPT) to 3 s (± 0.5 s of jitter to avoid synchronization effects), which offers a responsiveness similar to that provided by, e.g., AODV [23].

Each bandwidth probe is composed of n packets generated back to back. When all the packets from a probe i arrive to the destination node (or only a subset N_i , if the timeout is triggered), the DACME agent at the destination will measure the average interarrival time (AIT_i) of incoming probe packets

$$AIT_i = \frac{\Delta t_{rec_i}}{N_i - 1} \quad (1)$$

and it will then calculate the available bandwidth by doing

$$BM_i = \frac{8 \times p}{AIT_i} \quad (\text{bit/s}) \quad (2)$$

where p is the packet size used; p should be similar to the one used by the application (on average). This bandwidth measurement is then sent back to the source. The DACME source agent, when receiving probe reply packets, will collect the BM values sent by the destination agent to be able to reach a decision on whether to admit the connection or not.

We will now detail the process followed to determine the optimum number of packets per probe (n), along with the bandwidth refinement and correction processes required, to accurately estimate available bandwidth. Afterwards, we present the actual admission control mechanism used in DACMEs framework.

Due to space limitations, in the next three sections we will omit the comparative analysis on the accuracy of the various probing possibilities for bandwidth, delay, and jitter. This preliminary study can be found in [25], [26].

1) *Probe Size Tuning*: Our tuning aims at medium-sized MANETs, in scenarios with a number of nodes between 30 and 100, and where the average number of hops is between 2 and 4. This choice was made considering the expected path lifetimes for MANETs, which typically become too low to offer acceptable QoS levels beyond this threshold.

According to [27], the expected path length of a squared scenario sized L is $2 \cdot L/3$. So, we want that: $2 \times R \leq 2 \cdot L/3 \leq 4 \times R$; for a radio range $R = 250$ m (default one), the target scenario side length is in the range $750 \leq L \leq 1500$ m.

We consider that the most important factors affecting the average interarrival time of probe packets and, therefore, the measured bandwidth, are the end-to-end path congestion (c), the number of hops on the path (h), and the number of packets per probe (n). Therefore, we use function $\Gamma(n, c, h)$ to represent the measured bandwidth in a certain environment as a function of these three parameters.

In order to gain insight into this function, we devised a static scenario that allows us to make measurements in a controlled environment and develop an analytical framework that can later be applied to mobile scenarios. For this preliminary tuning process we require the scenario to be static to obtain meaningful comparisons between reference values and those obtained with probes, avoiding interferences due to routing traffic and variable number of hops. These would cause available bandwidth to vary with time, making any measurements meaningless for our purpose.

The static scenario we selected is presented in Fig. 3. When choosing it we took into account several factors. First of all, contention among different data sources had a clear impact on performance. This was achieved by aggregating several stations within data communication range of each other (traffic sources and first intermediate station). Our aim was also to create a multihop environment, since it will also affect available bandwidth. That was achieved by including several intermediate stations on the end-to-end path. With this setting we also experience hidden terminal effects and the

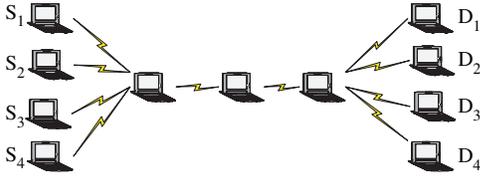


Fig. 3. Static scenario used to tune the probing process.

impact of carrier sensing ranges (considerably larger than data communication ranges).

The number of hops (4) aims at achieving a worst case behavior for the target scenario sizes (see above). Therefore, we can simplify function Γ as follows:

$$\Gamma(n, c, h) \cong \Gamma(n, c)|_{L \min < L < L \max}. \quad (3)$$

Concerning the capacity of a multihop *ad hoc* network, in [27] the authors determine that the theoretical upper limit for the capacity of a chain is defined as $C_{\max} = W_{\max}/4$, where W_{\max} is the maximum raw bandwidth achievable by the technology. W_{\max} for IEEE 802.11g is of 54 Mbit/s, and so $C_{\max} = 13.5$ Mbit/s. So, we can now proceed to find the optimal value for probe size n using our static scenario configuration.

When probing for bandwidth we have two conflicting interests: more packets per probe offer more accuracy; but at the same time we want to minimize overhead. So, the cost function used must represent this conflict of interests and allow us to find an adequate tradeoff. The chosen cost function is

$$\Lambda(n) = \text{Max} \left(\frac{\sqrt{V_c [\Gamma(n, c)]}}{2 \cdot E_c [\Gamma(n, c)]} \right) + \frac{n}{Q_{\text{lim}}} \quad (4)$$

where V_c and E_c refer to the variance and the expected value with respect to variable c , and Q_{lim} refers to the maximum queue size per MAC access category (which is set to 50 packets). Notice that, in order for both magnitudes to be comparable, we normalize them so that we are able to minimize percent differences. As shown in Fig. 4 (top), a minimum for Λ is reached for $n = 10$. Hence, each bandwidth probe will consist of 10 back-to-back packets.

Experimental results showed that, for $n = 10$, the accuracy of a single probe has an error ranging from 4% to 16%.

2) *Bandwidth Refinement Through Multiple Probes*: For each probe sent, the new value for the measured bandwidth obtained at the destination is sent back to the source only once. The information offered by these consecutive probes is used to refine the mean and standard deviation values for the bandwidth. Upon arrival of the first measurement, it merely stores that value by setting $\mu_0 \leftarrow BM_0$ and $\sigma_0 \leftarrow \infty$.

By using the measurements that follow, it recalculates the mean and the standard deviation as follows:

$$\begin{cases} \mu_i \leftarrow \frac{(i-1) \cdot \mu_{i-1} + BM_i}{i} \\ \sigma_i \leftarrow \sqrt{\frac{(i-2) \cdot (\sigma_{i-1})^2 + (BM_i - \mu_i)^2}{i-1}} \end{cases}, \quad i > 0 \quad (5)$$

thus allowing us to refine both values iteratively.

3) *Correction of the Estimation Bias*: MANETs conform a system with memory and, for that reason, any short-term measurement values must be corrected in order to accurately

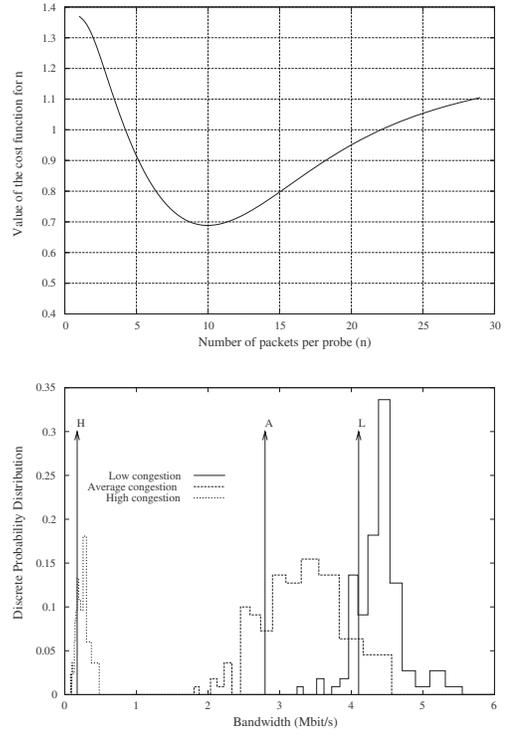


Fig. 4. (Top) Value of cost function Λ for different values of n and (bottom) discrete probability distribution for the probing process under low, average, and high levels of congestion.

reflect long-term values. To understand this phenomena, let us take as an example the results of Fig. 4 (bottom), where we show the discrete probability distribution of the probing process for our static scenario under low, average, and high congestion.

The arrow/letter pairs refer to available bandwidth measurements made with real traffic, and are used as a reference for the comparison. As can be seen, the three probability distributions are not centered around the reference values (H, A, and L), which explains why their mean is superior to the actual bandwidth availability. Also, we notice that the average levels of congestion tend to favor lower kurtosis values.

So, our purpose is to obtain, for all values of end-to-end path congestion (c), an unbiased estimator $v_p(c)$ for the long-term available bandwidth $B_p^{LT}(c)$ achievable with a certain packet size p .

We have tested with different correction functions and found that we are able to achieve high degrees of accuracy by merely relying on short-term measurements of both mean and standard deviation

$$v_p(c) = \alpha \cdot \mu_p(c) + \beta \cdot \sigma_p(c). \quad (6)$$

Notice that α and β are parameters whose optimal value can be obtained through regression. Our purpose is to find values for α and β that apply to a wide range of path congestion values. This purpose can be met since we find experimentally that the expression

$$\exists \alpha, \beta \in \mathbb{R}: \left| 1 - \frac{v_p(c)}{B_p^{LT}(c)} \right| < \varepsilon, \quad (7)$$

$$\forall c \in [0, C_{\max}], \quad \forall p \in [p_{\min}, p_{\max}]$$

is true for very low error values (ε).

Algorithm 1 Probabilistic admission control mechanism for bandwidth-constrained applications

After receiving each BM_i **do** {
 calculate μ_i and σ_i
 find the unbiased bandwidth estimator $v_{p,i}$
 if ($v_{p,i} - t_{n-1,0.95} \frac{\sigma_i}{\sqrt{n}} > B_R$)
 then $Flag(BW) \leftarrow 1$
 else if ($v_{p,i} + t_{n-1,0.95} \frac{\sigma_{p,i}}{\sqrt{n}} < B_R$)
 then $Flag(BW) \leftarrow 0$
 else if ($n < N_{max}$)
 then send a new probe }

4) *Admission Control Under Bandwidth Constraints:* In terms of an actual implementation of a DACME agent, the decision on whether to accept, maintain, or refuse a certain connection is based on the probabilistic admission control mechanism described as Algorithm 1.

This algorithm is executed every time a probe reply is received. Decisions are based on statistical confidence levels; therefore, $t_{n-1,0.95}$ refers to a Student's t-distribution with $n-1$ degrees of freedom and for a confidence level of 95%. The proposed algorithm allows us to reduce the number of probes required to perform a decision to a value as low as two probes; it occurs often in those situations in which it becomes quickly evident that the available bandwidth is either much higher or much lower than the requested one. If the application is solely bandwidth constrained, the source DACME agent will then notify it whether the connection can currently be admitted or not. If the application also has requirements on end-to-end delay and delay jitter, the DACME source agent will perform more tests, up to a limit of n_{max} . These issues are addressed in the two following sections.

D. Delay Probing

To support applications with bandwidth and delay requirements (B_R, D_R), or delay requirements alone (D_R), DACME offers a measurement technique similar to the measurements made by a ping application. However, to reduce as much as possible the time used to perform measurements, a new echo request packet is sent immediately after receiving an echo reply packet. Also, and for the sake of accuracy, the echo reply packet should have the same length and the same value for the IP TOS field as the echo request one.

In [25] we find that at least three consecutive round-trip times are required to obtain a reliable delay measurement. Therefore, the technique we use to handle applications with delay requirements is the following: We start with four consecutive probe request/probe reply rounds to assess the end-to-end delay. The value of the first round is discarded since it is used as a warm-up round to trigger routing and find the end-to-end bidirectional paths. The results from the three remaining probing rounds are averaged and stored as $D_e(0)$. In case any of the packets is lost, the end-to-end path is considered to be broken and the traffic is blocked.

We will now describe how, after estimating $D_e(0)$, we will estimate the delay at different levels of path

utilization u ($0 \leq u \leq 1$), for traffic in both voice and video MAC categories.

According to [28], a simple but commonly used predictor of delay based on the aforementioned parameters is

$$D_e(u) = \frac{D_e(0)}{1-u}. \quad (8)$$

For our study we find that this approach offers little accuracy in MANET environments, and so we propose an alternative delay estimation function

$$D_e(u) = \alpha \cdot (e^{\beta \cdot u} - 1) + \vartheta \cdot u^2 + \eta \cdot u + \gamma \quad (9)$$

which allows achieving much higher degrees of accuracy. Notice that it relies on the parametrization vector $v = (\alpha, \beta, \gamma, \eta, \vartheta)$. Since we know that $D_e(0) = \gamma$, we are able to obtain a normalized representation for delay instead

$$\bar{D}_e(u) = \frac{D_e(u)}{D_e(0)} = a \cdot (e^{b \cdot u} - 1) + c \cdot u^2 + d \cdot u + 1 \quad (10)$$

where $a = \alpha/D_e(0)$, $b = \beta$, $c = \vartheta/D_e(0)$ and $d = \eta/D_e(0)$.

Our goal with (10) is to model the delay behavior in a way that is independent of the measured end-to-end delay values. Hence, we normalize using $D_e(0)$ so that it can be adapted to each specific case. Therefore, we obtain an equation similar to (8), though more accurate in multihop MANET environments. We are now able to estimate what the end-to-end delay would be if QoS traffic starts flowing; this is particularly important when the bandwidth consumed approaches the currently available capacity.

Based on this estimating function, we must find the values for parameters a , b , c , and d that offer an accurate prediction of delay for traffic belonging to both video and voice MAC access categories. Therefore, we rely on a curve-fitting process to obtain two distinct parameter sets: $v_{voice} = (a', b', c', d')$ and $v_{video} = (a'', b'', c'', d'')$, that apply depending on the type of traffic. To understand the need for two parameter sets, notice that utilization is measured with respect to the value for the expected available bandwidth which, as explained at the beginning of the section, must always be done with probes in the video access category of IEEE 802.11e.

Probabilistic admission control for delay-bounded applications is done according to Algorithm 2. The strategy followed consists of making decisions based on worst and best case estimations for the delay. If the application is bandwidth-constrained and the traffic is blocked, these estimations are made according to bandwidth usage. When traffic is flowing, or when the application is delay-bounded alone (which suggests that bandwidth requirements are minimal), there is no need to perform such adjustments, and the measured value is directly used. In this case, we introduce a small margin of uncertainty of $D_e(0) \pm 10\%$ to provoke hysteresis, thereby avoiding frequent on/off traffic fluctuations in worst-case scenarios.

E. Jitter Probing

In this section, we complete DACMEs QoS framework by including basic support for jitter-constrained applications. During the jitter measurement period, the source agent must

Algorithm 2 Probabilistic admission control mechanism for delay-bounded applications

Execute code from algorithm 1 if appropriate. Then **do** {
if (application is bandwidth-constrained && traffic is currently blocked) **then** {
 $u_{\min} \leftarrow \frac{B_R}{v_p + t_{n-1, 0.95} \frac{\sigma_p}{\sqrt{n}}}$
 $u_{\max} \leftarrow \frac{B_R}{v_p - t_{n-1, 0.95} \frac{\sigma_p}{\sqrt{n}}}$
 $d_{\min} \leftarrow \bar{D}_e(u_{\min}) \times D_e(0)$
 $d_{\max} \leftarrow \bar{D}_e(u_{\max}) \times D_e(0)$
else { $d_{\min} \leftarrow 0.9 \times D_e(0)$
 $d_{\max} \leftarrow 1.1 \times D_e(0)$ }
if ($d_{\min} > D_R$) **then** $Flag(Delay) \leftarrow 0$
else if ($d_{\max} < D_R$) **then** $Flag(Delay) \leftarrow 1$
else if (application is bandwidth-constrained && $n < n_{\max}$)
then send a new bandwidth probe }
}

send packets with the same size, IP TOS field, and data rate as the application being served for a certain period of time (t_j). The receiving end, aware of the source's packet sending rate by explicit notification, calculates the standard deviation values for the jitter and returns them to the source. To optimize bandwidth consumption, jitter probing is performed after delay and bandwidth probes if neither test denied the connection, and only if the application's traffic is blocked. In case traffic from the target application is flowing through the network, there is no need to send jitter probes since the destination agent can measure the jitter using actual traffic, and then send those measurements back to the source.

Independent of the method used to measure jitter (probes or actual traffic), the standard deviation for the jitter (σ_j) alone will be helpful to assess compliance with the maximum value defined (J_R) because jitter follows a normal distribution with a mean value of zero; therefore, about 95% of the cases fall in the interval between $\pm 2\sigma$.

As shown in Algorithm 3, we only accept traffic if 95% of the packets experience a jitter value lower than the maximum requested. We also introduce hysteresis by defining a margin of uncertainty of $2 \cdot \sigma \pm 10\%$, where the strategy consists of maintaining the previous value to reduce traffic fluctuations.

The only issue that is still to be defined is the duration of the probing period (t_j). We know that the accuracy in estimating σ_j will depend on it, and well as on the source load (r) for a certain end-to-end bandwidth availability (B_{\max}). Therefore, we have a three-dimensional distribution function for the standard deviation defined as $\theta(t_j, r, x)$, where variable x represents randomness in the system.

We seek to find the optimal value for the duration (t_j^{opt}) of the jitter probing process that allows us to estimate σ_j with a reasonable accuracy at different data rates, that is

$$\sigma_j(r) \simeq E_x \left[\theta \left(t_j^{opt}, r, x \right) \right], \forall r \in [0, B_{\max}]. \quad (11)$$

The choice for (t_j^{opt}) should offer an equilibrium between the occupation of the end-to-end path and both the precision and variability of the measurements made. We will now

Algorithm 3 Probabilistic admission control mechanism for jitter-bounded applications

After receiving a jitter reply **do** {
if ($2.1 \times \sigma_j < J_R$)
then $Flag(Jitter) \leftarrow 1$
else if ($1.9 \times \sigma_j > J_R$)
then $Flag(Jitter) \leftarrow 0$ }
}

define mathematically each of these parameters so that we can integrate them into a cost function and find t_j^{opt} .

Starting with the occupation of the path, we measure it as the ratio t_j/IPT , where IPT is the periodicity of the probing processes, which was defined earlier as being 3 s by default.

If we obtain estimations for the jitter's standard deviation with M encompassing different source load values (r), we may define the average relative error for estimator θ as a function of t_j as

$$\xi [\theta(t_j)] = 1 - \frac{1}{B_{\max}} \times \int_0^{B_{\max}} \frac{E_x [\theta(t_j, r, x)]}{\sigma_j(r)} \partial r. \quad (12)$$

Simplifying (12) through discretization we obtain

$$\xi [\theta(t_j)] \cong 1 - \frac{1}{M} \times \sum_{r_i \in [0, B_{\max}]}^M \frac{E_x [\theta(t_j, r_i, x)]}{\sigma_j(r_i)}. \quad (13)$$

Concerning variability, we define it as half the average of the normalized inter-quartile range (IQR); to calculate it we will use $\Theta(d_j, r, x)$, which is the cumulative distribution function for $\theta(d_j, r, x)$ with respect to variable x . So

$$\zeta [\theta(t_j)] = 1 - \frac{1}{B_{\max}} \times \int_0^{B_{\max}} \frac{\Theta(t_j, r, 0.75) - \Theta(t_j, r, 0.25)}{2 \cdot \sigma_j(r)} \partial r \quad (14)$$

to which we apply discretization, obtaining

$$\zeta [\theta(t_j)] \cong 1 - \frac{1}{M} \times \sum_{r_i \in [0, B_{\max}]}^M \frac{\Theta(t_j, r_i, 0.75) - \Theta(t_j, r_i, 0.25)}{2 \cdot \sigma_j(r_i)}. \quad (15)$$

After the path occupation, the precision of σ_j , and its variability are defined, we can now relate them through a cost function that will allow us to find t_j^{opt} . As in the case of bandwidth, in the case of jitter larger probing periods are associated with more accuracy, but also with more overhead. In the case of jitter probes we measure both the percent error of the value obtained ($\xi [\theta(t_j)]$) and its variability ($\zeta [\theta(t_j)]$) to quantify the degree of accuracy. The chosen cost function, which offers a balance between resource consumption and accuracy, was therefore

$$\Psi(t_j) = \frac{t_j}{IPT} + \frac{\xi [\theta(t_j)] + \zeta [\theta(t_j)]}{2}. \quad (16)$$

Fig. 5 shows that cost function Ψ reaches a minimum when $t_j = 250$ ms for traffic in both video and voice MAC access categories (B_{\max} for the environment used is of 3 Mbit/s, and

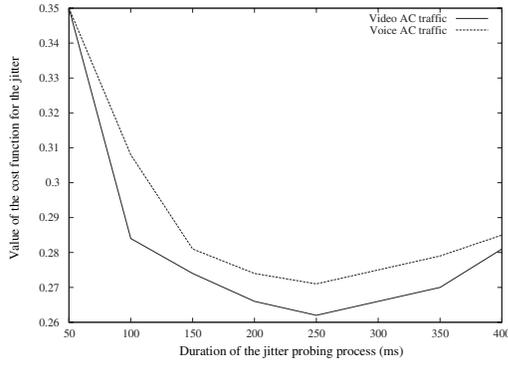


Fig. 5. Determination of t_j^{opt} using cost function $\Psi(t_j)$.

M is equal to 10). Therefore, in DACMEs framework, this value is chosen as our t_j^{opt} .

V. EFFICIENT ROUTING IN MANETS

Despite the fact that the architecture proposed in Section III is flexible enough to accommodate to virtually every routing protocol currently available, achieving optimum performance requires a routing protocol that is highly responsive to interruptions of on-going communications, detecting path losses and finding new ones as quickly as possible. Such responsiveness is heavily dependent on link-layer awareness, since only by receiving MAC layer feedback are we able to achieve low response times when link breaks occur. Hence, other routing protocols that rely on Hello messages for link breakage detection are too slow and considered inadequate for our purpose (see [29] for more details).

To mitigate the impact of mobility on real-time sessions we developed MDSR, a multipath extension to the DSR routing protocol [4] that offers increased performance to real-time video streaming in the presence of mobility. QoS-aware routing protocols would be able to offer further result improvements (if link layer detection of broken links is used and efficiently handled), though they are not required to achieve good QoS within the current QoS framework, as we will show later. So, our focus is solely on mitigating the impact of mobility.

A. MDSR Routing Protocol

The design of the MDSR routing protocol seeks an optimal integration of route discovery and route assignment processes by achieving maximum path disjointness under low additional routing load. In [22] we analyzed the performance of the MDSR routing protocol in legacy MANET environments at high speeds. Here we include a formal description of the MDSR routing protocol based on set theory. To begin with, let $\mathfrak{R}_{n,s} = \{R_{n,s}^k\}$ denote the paths corresponding to propagated route request packets (RREQs) originated at source n with sequence number s , and let δ_{\max} be the maximum value allowed for $|\mathfrak{R}_{n,s}|$.

To aid us in comparing the disjointness of paths we define operator \mathcal{N} , which allows obtaining the set of nodes conforming a certain path. Given a MANET of N nodes represented

Algorithm 4 RREQ propagation strategy

```

if ( $i < s$ ) then Discard( $\check{R}_{n,i}$ )
else if ( $i > s$ ) then {
   $s \leftarrow i$ 
   $\mathfrak{R}_{n,s} \leftarrow \{\check{R}_{n,i}\}$ 
} else if ( $|\mathfrak{R}_{n,s}| < \delta_{\max}$ ) then {
  if ( $\mathcal{N}(\check{R}_{n,i}) \cap \mathcal{N}(R_{n,s}^k) = \{\varphi_S\} \wedge |\check{R}_{n,i}| \leq |R_{n,s}^k|, \forall k: R_{n,s}^k \in \mathfrak{R}_{n,s}$ ) then {
    Propagate( $\check{R}_{n,i}$ )
     $\mathfrak{R}_{n,s} \leftarrow \mathfrak{R}_{n,s} \cup \{\check{R}_{n,i}\}$ 
  } else Discard( $\check{R}_{n,i}$ )
} else Discard( $\check{R}_{n,i}$ )

```

as $\bar{\varphi} = \{\varphi_i\}$, $i \leq N$, every path between source (S) and destination (D) can be represented as

$$\mathcal{N}(P_{\varphi_S \rightarrow \varphi_D}) \doteq \{\{\varphi_k\} \in \bar{\varphi} \mid \{\varphi_S, \varphi_D\} \subset \{\varphi_k\}\}.$$

Upon arrival of a new RREQ packet $\check{R}_{n,i}$ at an intermediate node, it propagates the packet conditionally, according to the strategy described in Algorithm 4. The algorithm ensures that additional route requests are only propagated if their route lengths are equal or shorter than the length of the first RREQ received. More important, it further restricts RREQ propagation by allowing only routes that are node disjoint with respect to the previous ones.

Each RREQ packet arriving to the destination will generate a route reply (RREP) packet back to the source. The source, upon receiving information about alternative routes to destination ($P_{\varphi_S \rightarrow \varphi_D}$), stores that information in its cache $\mathcal{C}_{\varphi_S \rightarrow \varphi_D} = \{P_{\varphi_S \rightarrow \varphi_D}\}$.

When selecting an optimal path, a source node can adopt either node disjointness or link disjointness criteria. Operator \mathcal{N} allows us to obtain the set of nodes conforming a path, as defined previously. We now define operator \mathcal{L} , which allows us to obtain the set of links conforming a certain path. If we define $\bar{\mathcal{L}}$ as the family of all two-element sets over $\bar{\varphi}$

$$\bar{\mathcal{L}} \leftarrow \{\{\{\varphi_i, \varphi_j\}\} \subset \mathcal{P}(\bar{\varphi}) \mid \varphi_i \neq \varphi_j, \forall \varphi_i, \varphi_j \in \bar{\varphi}\}$$

then a path can alternatively be represented in terms of the links that conform to it as

$$\begin{aligned} \mathcal{L}(P_{\varphi_S \rightarrow \varphi_D}) &\doteq \{\{\{\varphi_i, \varphi_j\}\} \in \bar{\mathcal{L}} \mid \varphi_{i+1} \\ &= \varphi_j, \forall \varphi_i, \varphi_j \in \mathcal{N}(P_{\varphi_S \rightarrow \varphi_D})\}. \end{aligned}$$

For each destination, operators \mathcal{N} and \mathcal{L} also apply to the cache of known routes $\mathcal{C}_{\varphi_S \rightarrow \varphi_D}$. So, we define $\mathcal{N}(\mathcal{C}_{\varphi_S \rightarrow \varphi_D}) \doteq \{\mathcal{N}(P_{\varphi_S \rightarrow \varphi_D})\}$ and $\mathcal{L}(\mathcal{C}_{\varphi_S \rightarrow \varphi_D}) \doteq \{\mathcal{L}(P_{\varphi_S \rightarrow \varphi_D})\}$.

Based on the concepts define above, we now define the route selection strategy for the MDSR routing protocol. Supposing that a source A must send a packet to a destination B , Algorithm 5 applies. This algorithm gives preference to node disjoint routes first, followed by link disjoint routes, and finally routes with least number of links in common with respect to the previously used route. On each category, preference goes to the shortest route available.

The combined use of both algorithms guarantees that, for a given video flow, consecutive paths used are disjoint most of the time. This is very effective at reducing the impact of

Algorithm 5 Route selection strategy

Pick an optimal path $P_{\varphi_A \rightarrow \varphi_B}^* = P^*$ with respect to the previous one used ($\tilde{P}_{\varphi_A \rightarrow \varphi_B}$) such that:

if ($|\mathcal{C}_{\varphi_A \rightarrow \varphi_B}| = 0$) **then** *FindNewRoute*(φ_B)

if ($\mathcal{L}(\tilde{P}_{\varphi_A \rightarrow \varphi_B}) \equiv \emptyset$) **then** *Return* ($\{P^* \mid |\mathcal{N}(P^*)| \leq |\mathcal{N}(P_{\varphi_A \rightarrow \varphi_B}^m)|, \forall P_{\varphi_A \rightarrow \varphi_B}^m \in \mathcal{C}_{\varphi_A \rightarrow \varphi_B}\}$)

$\mathcal{D}_{\varphi_A \rightarrow \varphi_B} \leftarrow \{P_{\varphi_A \rightarrow \varphi_B}^d \in \mathcal{C}_{\varphi_A \rightarrow \varphi_B} \mid \mathcal{N}(P_{\varphi_A \rightarrow \varphi_B}^d) \cap \mathcal{N}(\tilde{P}_{\varphi_A \rightarrow \varphi_B}) = \{\varphi_A, \varphi_B\}\}$

if ($\exists P^* \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B} : |\mathcal{N}(P^*)| \leq |\mathcal{N}(P_{\varphi_A \rightarrow \varphi_B}^d)|, \forall P_{\varphi_A \rightarrow \varphi_B}^d \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B}$) **Return** P^*

$\mathcal{D}_{\varphi_A \rightarrow \varphi_B} \leftarrow \{P_{\varphi_A \rightarrow \varphi_B}^d \in \mathcal{C}_{\varphi_A \rightarrow \varphi_B} \mid \mathcal{L}(P_{\varphi_A \rightarrow \varphi_B}^d) \cap \mathcal{L}(\tilde{P}_{\varphi_A \rightarrow \varphi_B}) = \emptyset\}$

if ($\exists P^* \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B} : |\mathcal{L}(P^*)| \leq |\mathcal{L}(P_{\varphi_A \rightarrow \varphi_B}^d)|, \forall P_{\varphi_A \rightarrow \varphi_B}^d \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B}$) **Return** P^*

else **Return** ($\{P^* \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B} \mid |\mathcal{L}(P^*) \cap \mathcal{L}(\tilde{P})| \leq |\mathcal{L}(P_{\varphi_A \rightarrow \varphi_B}^d) \cap \mathcal{L}(\tilde{P})|, \forall P_{\varphi_A \rightarrow \varphi_B}^d \in \mathcal{D}_{\varphi_A \rightarrow \varphi_B}\}$)

mobility since the chances that disjoint paths break at about the same time are low.

B. Interaction Between the Routing and the IEEE 802.11e Layers

The routing protocols used in our framework (MDSR, and also AODV and DSR for comparison) were modified to be aware of IEEE 802.11e availability; therefore, they adjust the priority of their own packets to benefit from high priority channel access. This is done by setting the IP ToS/TC header field to obtain a mapping to the voice access category (highest priority) at the MAC layer since low response times to link breaks are of highest significance in MANETs, especially when the protocols being used are reactive.

The IEEE 802.11e layer was also modified to become aware of routing data, making sure that all routing packets are put at the head of the appropriate queue (in our case, voice), thereby reducing routing latency to a minimum.

VI. EVALUATION OF THE PROPOSED QoS ARCHITECTURE

In this section, we perform an incremental evaluation of the proposed QoS architecture. The sequence of tests here presented emphasizes on the benefits offered by each of the different modules composing our framework. We begin by assessing the effectiveness of the IEEE 802.11e technology in segregating QoS traffic from best effort traffic. We then proceed by evaluating the benefits of multipath routing to reduce the impact of node mobility on real-time streams. Finally, we show the effectiveness of the proposed distributed admission control mechanism to regulate the QoS traffic flowing in the MANET. Since our tests are incremental, in this last step we are actually evaluating the performance of the overall architecture proposed.

For application-level measurements we picked real-time H.264/AVC [5] video streams. We consider that the requirements of real-time video sessions in terms of bandwidth and delay, along with the generation of a variable bit rate (VBR) data stream, are adequate to validate the system under demanding conditions. Our video sequence of choice is the well known Foreman sequence (see Fig. 6, top) in the CIF format (352×288 pixels), which is adequate for video-conferencing. We concatenate 30 copies of this Foreman sequence (10 s

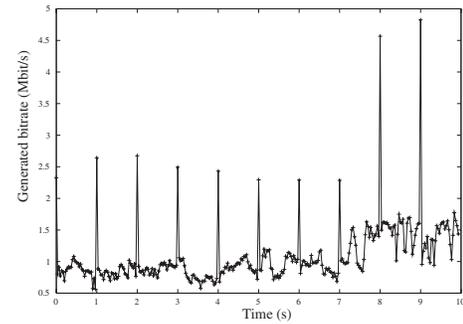


Fig. 6. (Top) Foreman sequence and (bottom) a 10-second snapshot of the bit rate for the H.264-encoded stream.

long) to obtain a 300-s long video. The frame rate used is of 30 Hz, and the number of RTP packets generated per second is 210. The global quantization parameters for the sequence were adjusted to get a target bit-rate slightly above 1 Mbit/s and an average PSNR value of 38.2 dB.

In Fig. 6 (bottom), we show the bit rate generated by the H.264/AVC codec for the Foreman sequence during a 10-s period. As can be seen, every second there is a peak on the instantaneous bit rate generated. This is because we configure the video codec to generate one I frame every second to reset error propagation, and so the video GOP size is of 30 frames. The remaining frames are predictively coded (P frames). The different codec parameters were tuned for optimum performance in MANETs (see [30] for further details).

To conduct our experiments we used the ns-2 simulator [31]. Simulations are made in a square area sized 870×870 m where the radio range is set to 250 m. The number of nodes used is 50, and each of them has an IEEE 802.11g radio

interface and a routing agent running (either MDSR or the standard AODV and DSR routing protocols). Concerning mobility, it is generated according to the random way-point mobility model adjusted so that all MANET nodes are constantly moving at a fixed speed of 5 m/s (no pause times), unless stated otherwise, for all the scenarios generated the average number of hops is of 4.

Relatively to the simulation process itself, we begin with a 100-s period during which routes between traffic sources and destinations are found; also during that period we start background traffic so that, when a video streaming session begins, it encounters a steady-state MANET environment. After the 100-s warm-up period, we start injecting QoS traffic, and each experiment runs for additional 300 s. The results obtained are actually drawn from this 300-s period.

The graphs depicted in the following sections represent an average of 10 different executions of the simulation with different randomly generated mobility scenarios.

A. Segregation of QoS Traffic From Best Effort Traffic Using IEEE 802.11e

In this section we devise a set of experiments that evidence how the IEEE 802.11e technology is able to differentiate QoS traffic from best effort traffic. We use standard IEEE 802.11 technology, whose MAC layer is not QoS-enabled, for comparison. At this stage we use neither admission control (DACME) nor enhanced routing mechanisms (MDSR).

As referred earlier, our focus is on the support for real-time video streaming applications. We therefore inject into the MANET the trace of a single video stream with the characteristics referred to earlier. Concerning best effort traffic, we inject a variable number of FTP/TCP sources (bandwidth greedy) and CBR/UDP sources generating data at a rate of 1 Mbit/s. We study the performance when increasing the number of best effort sources from 0 to 18. The number of sources is increased with a granularity of 3, maintaining a 2 to 1 relationship between the number of TCP and UDP sources, respectively. This means that there are twice as many TCP sources compared to UDP sources in all tests.

Fig. 7 shows the performance experienced by our reference video stream. We can observe that, when using either AODV or DSR combined with IEEE 802.11e, the throughput it maintained close to the maximum, i.e., slightly above 1 Mbit/s, even when increasing the number of background traffic sources. If IEEE 802.11e is not used, the throughput decays gradually, with loss values of 80% and up.

In terms of end-to-end delay, we observe similar performance benefits: IEEE 802.11e reduces delay by more than one order of magnitude for both AODV and DSR routing protocols.

Concerning the quality of the video session as experienced by the user, we use the H.264/AVC reference decoder to obtain the corresponding PSNR values, along with a confidence interval of 95%. To obtain these values we pass all the simulation results relative to the video stream through the video codec, and then obtain the PSNR values taking the original raw video sequence as reference. Fig. 8 shows that

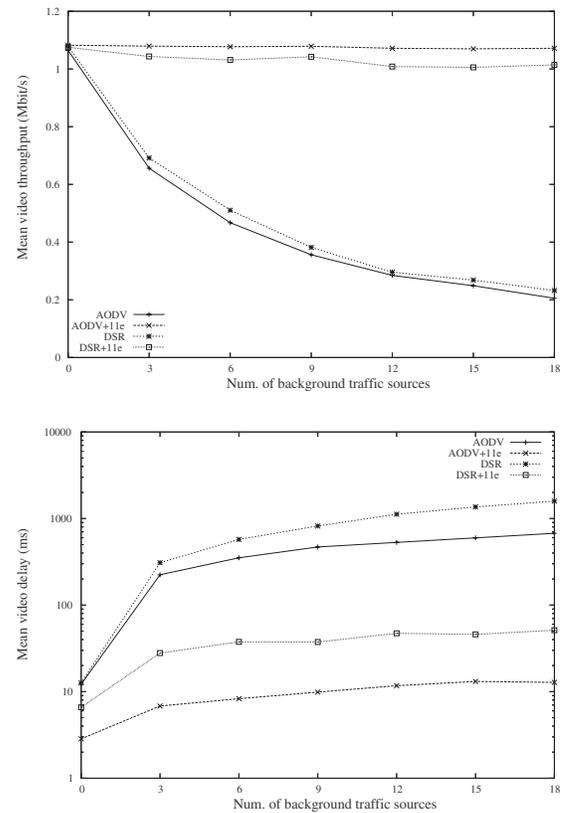


Fig. 7. (Top) Mean values for the video throughput and (bottom) the video delay when varying the number of background traffic sources.

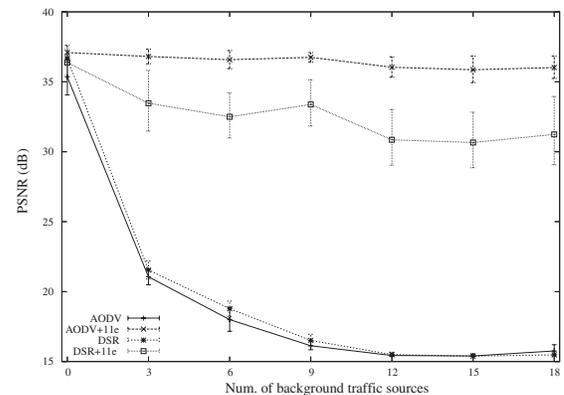


Fig. 8. PSNR confidence intervals when varying the number of background traffic sources.

PSNR values are kept at reasonably good quality levels if IEEE 802.11e is used. When using the legacy IEEE 802.11 technology, though, PSNR values drop below 25 dB just when the first three sources of background traffic are started, getting worse as background traffic is increased. This clearly puts into evidence that MAC-level QoS support is essential to achieve a global QoS framework in MANETs.

In terms of background traffic (see Fig. 9), we find that the aggregated throughput value is increased more than twice when using IEEE 802.11e; such improvements are related to increased routing responsiveness. This means that there is a win-win situation in which both real-time and best effort sources benefit from the IEEE 802.11e technology.

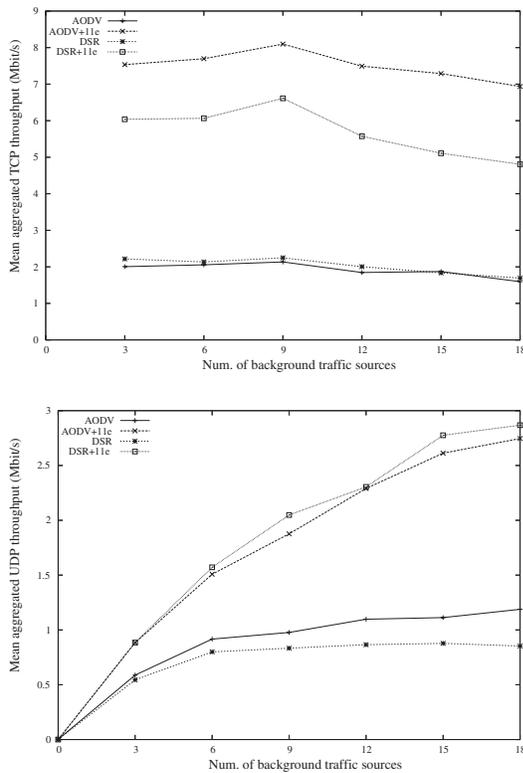


Fig. 9. (Top) Mean values for the TCP throughput and (bottom) UDP throughput varying the number of background traffic sources.

B. Reducing the Impact of Mobility Through Multipath Routing

In this section, we show how our MDSR routing protocol is able to significantly decrease the frequency and size of video streaming gaps. Video gaps are defined as significant interruptions of a video streaming session which, in MANET environments, are typically caused by mobility and related to rerouting processes. We build upon the results of the previous section, and so we will use an IEEE 802.11e-enabled MAC layer for our tests. We also include six best effort sources as background traffic. As before, our focus is on a single real-time video stream injected into the MANET using the real trace of an H.264-encoded video sequence.

Simulation settings are very similar to those of the previous section, but we now alter mobility by varying the node speed in a range between 1 and 9 m/s. Concerning the routing protocols used, we compare the standard DSR routing protocol to our enhanced version of it—MDSR—whose purpose is to boost performance at high levels of mobility.

In terms of mean PSNR values, Fig. 10 (top) shows the robustness towards mobility offered by MDSR, including a confidence interval for the mean; the degree of confidence is of 90%. Though the improvements achieved are clear, we can obtain further insight if we analyze the frame loss pattern. Even for a moderate speed of 5 m/s, where we have a 1.2 dB difference in terms of PSNR, we find that the user experience (i.e., perceptual video quality) is greatly improved by using MDSR (see Fig. 10, bottom) since lengthy connection disruptions are kept to a minimum. Concerning video throughput and end-to-end delay, we also find that

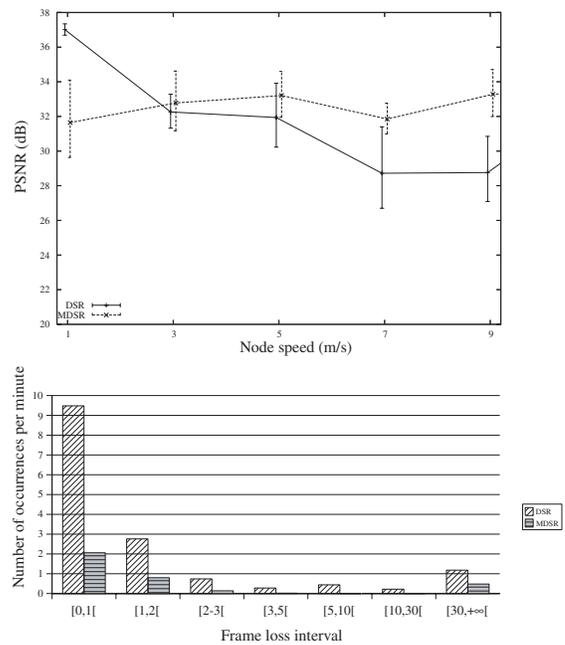


Fig. 10. (Top) PSNR when varying node speed and (bottom) video gap histogram at a speed of 5 m/s.

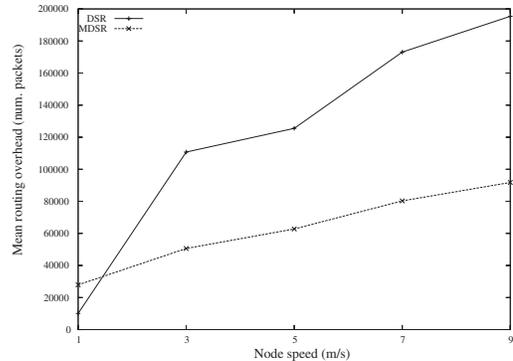


Fig. 11. Mean routing overhead when varying node speed.

MDSR offers a better performance for node speeds of 3 m/s and above (results not shown).

In terms of routing overhead, Fig. 11 shows that DSR requires an excessive routing overhead to offer a performance similar to that of MDSR at moderate and high speeds. So, despite that MDSR employs an enhanced route discovery mechanism that imposes more routing overhead than DSR, it actually requires fewer routing packets than the latter in the long term.

C. Application-Level QoS Support Through Distributed Admission Control

We finally evaluate the performance of the complete QoS architecture as depicted in Fig. 1. Again we build upon the findings of previous sections, now focusing on the benefits of our distributed admission control system in supporting multiple real-time video streams in IEEE 802.11e-based MANETs.

The basic simulation setup is very similar to the one of the previous sections, with the difference that MANET stations include an implementation of DACME. For testing we use

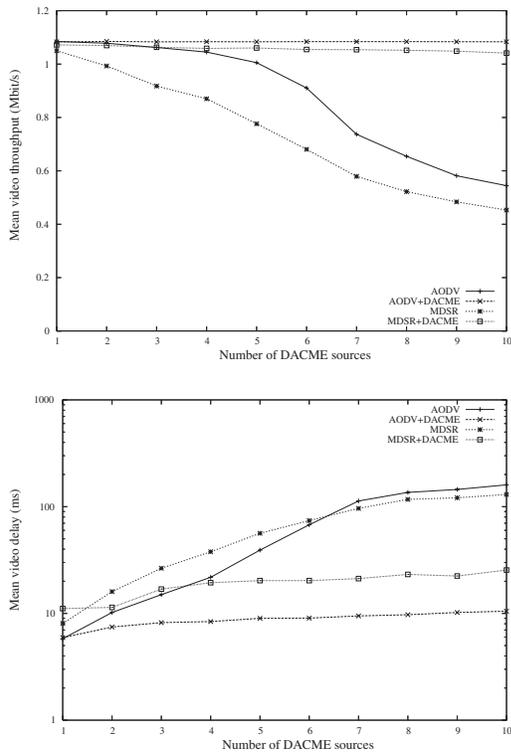


Fig. 12. (Top) Mean values for the video throughput and (bottom) the video delay when varying the number of DACME-regulated QoS sources.

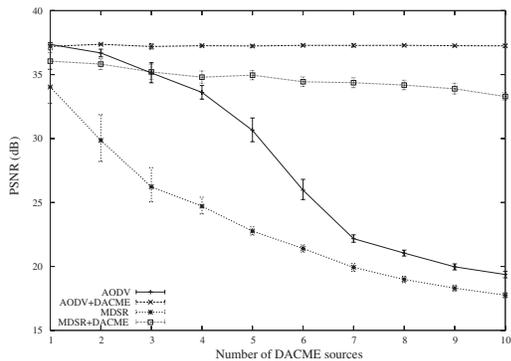


Fig. 13. PSNR confidence intervals when varying the number of DACME-regulated QoS sources.

our enhanced version of the DSR routing protocol, MDSR, as well as AODV, since the cross-layer interactions between DACME and AODV exposed in Section IV-A prove to be quite effective. The number of best effort background sources is fixed at six (four TCP sources and two UDP sources). Concerning node mobility, it is also fixed at 5 m/s.

Relatively to QoS traffic, we will test with up to 10 video sources instead of using a single video source as before. Each source of video traffic registers with the DACME agent, setting a Q_{SPEC} of $(B_R, D_R, J_R) = (1\text{Mbit/s}, 100\text{ms}, 10\text{ms})$.

Fig. 12 shows that the admission control mechanism proposed is highly effective, offering steady throughput and delay values to the VBR video streams as the number of video sessions increases.

Fig. 13 shows the PSNR values including a 95% confidence interval for the mean. We find that DACME-regulated sources are able to maintain very good values for video distortion

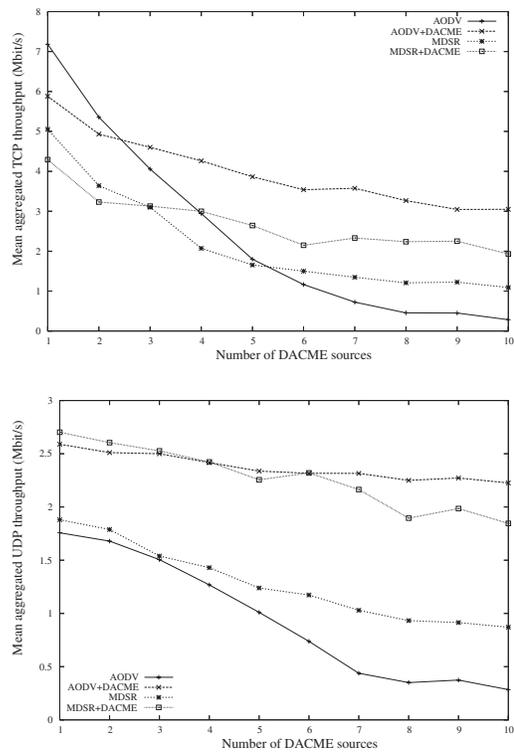


Fig. 14. (Top) Mean values for the TCP throughput and (bottom) UDP throughput when varying the number of DACME-regulated QoS sources.

(above 33 dB), while the video distortion values without DACME drop to very low quality (below 25 dB) or even noise levels (below 20 dB).

Relative to best effort traffic, we find that both TCP and UDP sources achieve higher throughput when DACME is active (see Fig. 14). We therefore conclude that DACME makes resource usage more efficient for both DACME and non-DACME traffic.

Concerning the QoS traffic acceptance rate, Fig. 15 (top) shows that DACME restricts traffic admittance to about 80% (on average), and reaching about 35% of the injected traffic when the number of sources is 10 (for this particular scenario setting). The difference experienced using both routing protocols is only slight, showing that multipath routing algorithms do not cause DACME to misbehave. In terms of aggregated QoS traffic, we find that it increases almost linearly when increasing the number of sources (see Fig. 15, bottom). This means that more number of QoS sources do not cause the admission control mechanism to fail or misuse the available radio resources.

Focusing now on the routing overhead, Fig. 16 shows that DACME has a stabilizing effect on routing mechanisms, avoiding that routing traffic increases too much due to congestion; this negative effect is especially noticeable for the AODV routing protocol.

Concerning the most appropriate routing protocol to use, we found that MDSR offers better performance for high degrees of mobility by avoiding communication gaps, while AODV performs better for low degrees of mobility. Besides taking mobility into account, works such as [32] show that the physical layer modeling not only affects the absolute performance

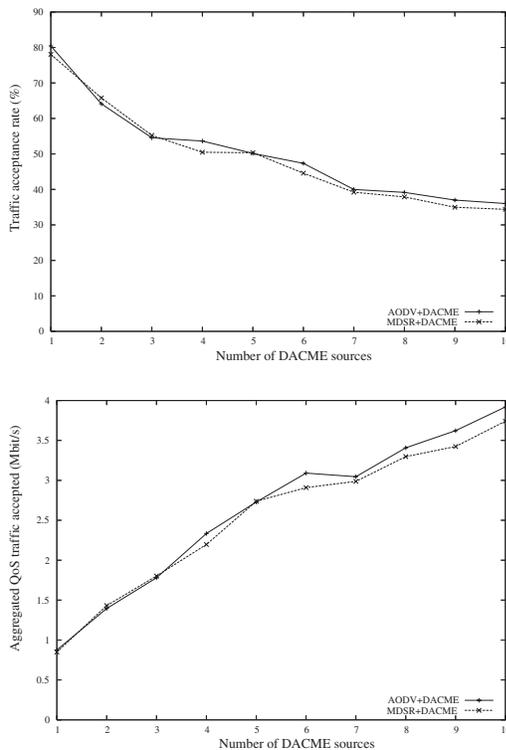


Fig. 15. (Top) Traffic acceptance rate and (bottom) aggregated QoS traffic when varying the number of DACME-regulated QoS sources.

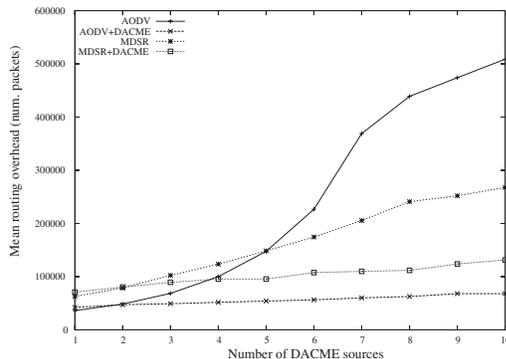


Fig. 16. Mean routing overhead when varying the number of DACME-regulated QoS sources.

of a protocol but, because its impact on different protocols is nonuniform, it can even change the relative ranking among protocols for a same scenario. Therefore, final decisions on routing protocol convenience should be made according to real-life testbed experiments.

In terms of overhead, results showed that the probing traffic injected by DACME agents generate merely 30 to 50 kbit/s per source on average. Additionally, as shown in this section, probing traffic does not affect the performance of real-time sessions negatively.

With respect to DACMEs limitations, we found that its capacity to sustain the degree of QoS is intimately related to the effectiveness of the routing protocol used and the lifetime of routes. When using the MDSR routing protocol, QoS can be sustained for node speeds up to 15 m/s. Moreover, we found that, despite it being able to integrate stations not supporting

IEEE 802.11e, performance will suffer an important decay in the presence of congestion if the percentage of such legacy stations is significant.

VII. CONCLUSION

In this paper we have proposed a novel QoS architecture for MANETs that seeks to alleviate the effects of both congestion and mobility on real-time applications. Our architecture is highly modular and combines our DACME with the IEEE 802.11e technology to offer soft QoS support to MANETs heavily loaded by both best effort and QoS traffic. The proposed architecture relies on MDSR to reduce the impact of mobility on real-time sessions, also offering good performance when the routing protocols used are able to quickly respond to topology changes. By taking H.264 real-time video communication as our target application, we show through simulation that our system is able to solve most of the problems that these streams may encounter in MANET environments, namely TCP-related congestion, node mobility, and unregulated QoS traffic. Experimental results from a medium-sized MANET scenario have shown that the proposed QoS architecture is very effective at maintaining video throughput values high and delay values low. By relying on H.264/AVC reference software, we also have shown that the actual video PSNR values obtained are very good. Besides the aforementioned benefits, the proposed architecture merely requires some optimizations of standard technologies for the PHY/MAC and routing layers, and so it can be easily and quickly deployed. Also, by avoiding resource reservations, we reduced to a minimum the constraints imposed on MANET terminals, thereby offering a solution that can be deployed in both homogeneous and heterogeneous MANET environments.

REFERENCES

- [1] L. Georgiadis, P. Jacquet, and B. Mans, "Bandwidth reservation in multihop wireless networks: Complexity and mechanisms," in *Proc. Int. Conf. Distributed Comput. Syst. Workshops (ICDCSW'04)*, Hachioji-Tokyo, Japan, Mar. 2004, pp. 762–767.
- [2] S. Chakrabarti and A. Mishra, "QoS issues in *ad hoc* wireless networks," *IEEE Commun. Mag.*, vol. 39, no. 2, pp. 142–148, Feb. 2001.
- [3] *802.11e IEEE Std. Inform. Technol.—Telecommun. and Inform. Exchange Between Syst.-Local and Metropolitan Area Networks-Specific Requirements Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment 8: Medium Access Control (MAC) Quality Service Enhancements*, IEEE 802.11 WG, 2005.
- [4] D. B. Johnson, Y. Hu, and D. A. Maltz. (Feb. 2007), "The dynamic source routing protocol (DSR) for mobile *ad hoc* networks for ipv4." *Request for Comments: 4728, MANET Work. Group*, [Online]. Available: <http://www.ietf.org/rfc/rfc4728.txt>
- [5] *Draft ITU-T Recommendation and Final Draft Int. Std. Joint Video Specification*, (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC), Mar. 2003.
- [6] L. Romdhani, Q. Ni, and T. Turletti, "Adaptive EDCF: Enhanced service differentiation for IEEE 802.11 wireless ad-hoc networks," in *Proc. Wireless Commun. Networking Conf.*, vol. 2. New Orleans, LA, 2003, pp. 1373–1378.
- [7] J. L. Sobrinho and A. S. Krishnakumar, "Quality-of-service in *ad hoc* carrier sense multiple access wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1353–1368, Aug. 1999.
- [8] C.-H. Yeh and T. You, "A QoS MAC protocol for differentiated service in mobile *ad hoc* networks," in *Proc. Int. Conf. Parallel Process.*, Kaohsiung, Taiwan, Oct. 2003, pp. 349–356.

- [9] S. Sivavakeesar and G. Pavlou, "Quality of service aware MAC based on IEEE 802.11 for multihop *ad hoc* networks," in *Proc. IEEE Wireless Commun. Networking Conf.*, vol. 3, Atlanta, GA, Mar. 2004, pp. 1482–1487.
- [10] A. Chen, Y. T. L. Wang Su, Y. X. Zheng, B. Yang, D. S. L. Wei, and K. Naik, "Nice - a decentralized medium access control using neighborhood information classification and estimation for multimedia applications in *ad hoc* 802.11 wireless lans," in *Proc. IEEE Int. Conf. Commun.*, May 2003, pp. 208–222.
- [11] C. R. Lin and J.-S. Liu, "QoS routing in *ad hoc* wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1426–1438, Aug. 1999.
- [12] C. Shigang and K. Nahrstedt, "Distributed quality-of-service routing in *ad hoc* networks," *IEEE J. Select. Areas Commun.*, vol. 17, no. 8, pp. 1488–1505, Aug. 1999.
- [13] Q. Xue and A. Ganz, "Ad hoc QoS on-demand routing (AQOR) in mobile *ad hoc* networks," *J. Parallel Distrib. Comput.*, vol. 63, no. 2, pp. 154–165, Aug. 2003.
- [14] C. Lei and W. B. Heinzelman, "QoS-aware routing based on bandwidth estimation for mobile *ad hoc* networks," *IEEE J. Select. Areas Commun.*, vol. 23, no. 3, pp. 561–572, Mar. 2005.
- [15] S. B. Lee, A. Gahng-Seop, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-based quality of service framework for mobile *ad hoc* networks," *J. Parallel Distrib. Comput., Special issue Wireless Mobile Comput. Commun.*, vol. 60, no. 4, pp. 374–406, Apr. 2000.
- [16] G.-S. Ahn, A. T. Campbell, A. Veres, and L. Sun, "Supporting service differentiation for real-time and best effort traffic in stateless wireless *ad hoc* networks (SWAN)," *IEEE Trans. Mobile Comput.*, vol. 1, no. 3, pp. 192–207, Sep. 2002.
- [17] B. Hamdaoui and P. Ramanathan, "Cross-layer optimized conditions for QoS support in multi-hop wireless networks with MIMO links," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 667–677, May 2007.
- [18] Y. Li and H. Man, "Video transport over multi-hop directional wireless networks," *Wiley Wireless Commun. Mobile Comput.*, vol. 7, no. 2, pp. 217–233, Jan. 2007.
- [19] S. Kompella, S. Mao, Y. T. Hou, and H. D. Sherali, "Cross-layer optimized multipath routing for video communications in wireless networks," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 831–840, May 2007.
- [20] V. Loscri, F. De Rango, and S. Marano, "Ad hoc on-demand multipath distance vector routing (AODM) over a distributed TDMA MAC protocol for QoS support in wireless *ad hoc* networks: Integration issues and performance evaluation," *Eur. Trans. Telecommun.*, vol. 18, no. 2, pp. 141–156, May 2006.
- [21] H.-P. Shiang and M. van der Schaar, "Multiuser video streaming over multihop wireless networks: A distributed, cross-layer approach based on priority queuing," *IEEE J. Select. Areas Commun.*, vol. 25, no. 4, pp. 770–785, May 2007.
- [22] C. T. Calafate, M. P. Malumbres, and P. Manzoni, "Mitigating the impact of mobility on H.264 real-time video streams using multiple paths," *J. Commun. and Networks*, vol. 6, no. 4, pp. 387–396, Dec. 2004.
- [23] C. E. Perkins, E. M. Belding-Royer, and S. R. Das. (2003, Jul.). "Ad hoc on-demand distance vector (AODV) routing," *Request for Comments 3561, MANET Work. Group*, [Online]. Available: <http://www.ietf.org/rfc/rfc3561.txt>
- [24] L. Breslau, E. Knightly, S. Shenker, I. Stoica, and H. Zhan, "Endpoint admission control: Architectural issues and performance," in *Proc. ACM Sigcomm 2000*, Stockholm, Sweden, Sep. 2000, pp. 57–69.
- [25] C. T. Calafate, P. Manzoni, and M. P. Malumbres, "Supporting soft real-time services in MANETS using distributed admission control and IEEE 802.11e technology," in *Proc. 10th IEEE Symp. Comput. Commun.*, La Manga del Mar Menor, Cartagena, Spain, Jun. 2005, pp. 217–222.
- [26] C. T. Calafate, "Analysis and Design Efficient Techniques Video Transmission IEEE 802.11 Wireless Ad Hoc Networks," Ph.D. thesis, Tech. Univ. Valencia (UPV), Valencia, Spain, 2006. [Online]. Available: <http://www.grc.upv.es/calafate/>.
- [27] J. Li, C. Blake, D. S. J. De Couto, H. I. Lee, and R. Morris, "Capacity of *ad hoc* wireless networks," in *Proc. 7th ACM Int. Conf. Mobile Comput. Networking*, Rome, Italy, Jul. 2001, pp. 61–69.
- [28] D. E. Comer, *Comput. Networks and Internets*. Englewood Cliffs, NJ: Prentice Hall, 2004.
- [29] C. T. Calafate, M. P. Malumbres, and P. Manzoni, "Performance of H.264 compressed video streams over 802.11b based MANETS," in *Proc. Int. Conf. Distributed Comput. Syst. Workshops (ICDCSW'04)*, Tokyo, Japan, Mar. 2004, pp. 776–781.
- [30] C. T. Calafate and M. P. Malumbres, "Testing the H.264 error-resilience on wireless ad-hoc networks," in *Proc. 4th EURASIP Conf. Focused*

Video/Image Process. Multimedia Commun., Zagreb, Croatia, Jul. 2003, pp. 789–796.

- [31] K. Fall and K. Varadhan, "Ns notes and documents," *The VINT Project. UC Berkeley, LBL, USC/ISI and Xerox PARC*, Feb. 2000, Available: <http://www.isi.edu/nsnam/ns/doc/>
- [32] M. Takai, J. Martin, and R. Bagrodia, "Effects of wireless physical layer modeling in mobile *ad hoc* networks," in *Proc. 2nd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Long Beach, CA, Oct. 2001.



Carlos T. Calafate graduated with honors in electrical and computer engineering from the University of Oporto, Oporto, Portugal, in 2001. He received the Ph.D. degree in computer engineering from the Technical University of Valencia, Valencia, Spain, in 2006.

Since 2005, he is with the Department of Computer Engineering, Universidad Politécnica de Valencia, as an Assistant Professor. His research interests include mobile and pervasive computing, security and QoS on wireless networks, and video coding

and streaming.

Dr. Calafate is a member of the Computer Networks research group.



Manuel P. Malumbres (M'01) received the B.S. degree in computer science from the University of Oviedo, Oviedo, Spain, in 1986, and the M.S. and Ph.D. degrees in computer science from University of Valencia, Valencia, Spain, in 1991 and 1996, respectively.

In 1989 he joined the Computer Engineering Department at the Technical University of Valencia, Valencia, Spain, as an Assistant Professor. In 2000, he became an Associate Professor and established the Computer Networks Group (GRC), and was the Group Director until June 2005. In September 2005, he moved to Miguel Hernandez University, Elche, Spain, where he holds the same position. He is currently with the Department of Physics and Computer Engineering at the Miguel Hernandez University. He has published more than 100 papers in journals and conference proceedings and has written several books on networking for undergraduate computer science courses. Currently, his research and teaching activities are related to multimedia networking (audio/video coding and network delivery) and wireless technologies (MANETs and sensor networks).

Dr. Malumbres has been a member of the technical committee of IEEE Multimedia Communications Group since 2004.



Jose Oliver received the MS and Ph.D. degrees in computer engineering from the Technical University of Valencia (UPV), Valencia, Spain, in 1998 and 2006, respectively.

Since 1999, he has been with the Department of Computer Engineering, UPV, where he holds the position of Lecturer gained by means of the "programa cantera" in 2001. He teaches computer networks and multimedia networks at the Faculty of Computer Science. He has served as a referee for several major conferences and journal papers. His research interests include image and video coding and wireless transmission.



Juan Carlos Cano obtained the M.Sc. and Ph.D. degrees in computer science from the Technical University of Valencia (UPV), Valencia, Spain, in 1994 and 2002, respectively.

He is now an Associate Professor with the Department of Computer Engineering, UPV. From 1995 to 1997, he worked as a programming analyst at IBMs manufacturing division in Valencia. His current research interests include power aware routing protocols, quality of service for mobile *ad hoc* networks, and pervasive computing.



Pietro Manzoni (M'96) received the M.S. degree in computer science from the Università degli Studi di Milano, Italy, in 1989 and the Ph.D. degree in computer science from the Polytechnic University of Milan in 1995.

He is currently an Associate Professor with the Department of Computer Engineering, the Universidad Politécnica de Valencia, Spain. His research activity is related to wireless network protocol design, modeling, and implementation.