Testing an Admission Control Module for MANETs in real devices

Álvaro Torres, Carlos T. Calafate, Juan-Carlos Cano, Pietro Manzoni [1]

Abstract— **Despite all the research efforts in the two previous decades, only a few mobile ad-hoc networks (MANETs) have been actually deployed, and even fewer were able to offer Quality of Service (QoS) support. The main problems hindering actual deployment have to do with the distributed effects of mobility, channel contention, and interference. Using simulation or analytical models, several QoS protocols, architectures and algorithms have been presented with the aim of improving QoS support in these environments; specifically, several works address admission control mechanisms since these are fundamental to provide QoS support in ad-hoc networks. When attempting to translate these research efforts to real testbeds, though, issues like feasibility in real systems, implementation complexity, node deployment and experiment repeatability have prevented their validation. In this paper we present a real implementation of DACME, our distributed admission control system for mobile ad-hoc networks. We test its effectiveness in an IEEE 802.11e enabled testbed, and the experimental results show that the solution developed is able to achieve good QoS levels, offering sustained bandwidth levels and bounded delay.**

Keywords— **Quality of Service; MANETs; testbed; distributed admission control; performance evaluation.**

I. Introduction

NOWADAYS, mobile ad hoc networks (MANETs) provide a cheap and infrastructureless form of communication. Typical MANET users share messages and collaborate with each other [1]. When combined with an appropriate routing protocol, the IEEE 802.11 standard [2] allows to easily deploy a MANET, which can be very useful in areas where the provision of a central infrastructure is limited or not possible.

More recently, interest has grown in enabling MANETs to support real-time services such as videoconferencing or VoIP calls among participants. However, this type of network lacks a good quality of service (QoS) support. In this scope, the IEEE 802.11e working group [3] provided QoS support at the MAC level, although the standard has been used mostly in wireless LAN environments. The 802.11e extension introduces four new traffic categories: Voice, Video, Best Effort, and Background (ordered according to their priority). These four categories provide traffic differentiation by adopting per-category values for the Contention Window (CW) and Inter Frame Space (IFS) parameters.

Despite the enhancements that IEEE 802.11e has brought, it is still not enough when facing QoS flow concurrency. In fact, one of the most crucial components of a system attempting to provide QoS guarantees is the Admission Control Module (ACM). This module should be able to estimate the resources of the network and decide when application flows should be admitted or rejected, avoiding to interfere with previously active flows. Unfortunately, this is not an easy task since MANETs are highly dynamic environments, and thus flow admittance does not guarantee good QoS conditions throughout time.

This paper provides the results of a real implementation (i.e., using a real IEEE 802.11e enabled testbed) of DACME [4], a low power consumption and low complexity end-to-end admission control module. The proposed solution imposes no constraints on the intermediate nodes of the communication other than being able to route packets. Experimental testbed results confirm the goodness of DACME at enhancing QoS support in wireless multi-hop environments, although some differences between simulated and real environments were detected. To improve DACME effectiveness in real deployments, we propose an enhancement to DACME's decision module.

The rest of this paper is organized as follows: in section II we review different admission control solutions for MANETs found in the literature. Section III presents a description of DACME, the admission control system we selected to implement and test, along with the proposed enhancement to DACME's decision module. Section IV offers a brief description of the testbed used for testing. Then, section V presents the experimental results and discussion. Finally, section VI presents the conclusions of this work along with future works.

II. Related works

In the literature we can find several admission control (AC) algorithms for MANET environments. Following the guidelines provided by the survey made by Hanzo et al. [5], the different AC algorithms available can be divided in two large groups: *routing coupled* and *routing decoupled*. In the first group we can find different algorithms such as ACRMP [6] and MACMAN [7]. Tthese AC algorithms require modifying the routing algorithm to support the AC extension. This strategy has some benefits, such as shorter admission times and less overhead of the AC protocol, mainly because they use routing packets to measure the state of the network. However, this first group also presents several drawbacks because, since they are coupled with a specific routing algorithm, a different routing protocol can not be used without losing the AC module. Another drawback is the strong requirements imposed on nodes, forcing every node in the network to adopt the modified routing protocol to support the AC module. This means that

[1]Departamento de Informática de Sistemas y Computadores - Universitat Politècnica de València, Spain {altorco1, calafate, jucano, pmanzoni}@disca.upv.es

even low power nodes, and nodes which do not require the AC module, will have to dedicate additional resources to support it.

With respect to routing decoupled algorithms, their main advantage is that they allow using any routing protocol for MANET environments. Within the routing decoupled algorithms group, an additional division can be made between *stateful* and *stateless* protocols. Stateful algorithms save certain information about the state of the links in every node. Such strategy allows AC algorithms such as INSIGNIA [8] to store information about QoS conditions relative to past flows, and decide, based on this information, whether to accept or reject the flow. Thus, similarly to what occurs for routing coupled protocols, they impose several restrictions on intermediate nodes and require more computing power from these nodes. Within the stateless, routing decoupled protocols group, we can find solutions such as DACME [4], which do not impose any restriction on intermediate nodes since they need not store any information about past flows, nor must they have a high computing power.

The main drawback associated with all the aforementioned solutions is that they have only been tested on simulated environments, and, to the best of our knowledge, none has been implemented and tested in a real environment. In this paper we will develop and test DACME. We chose this proposal since it has been proved to be a powerful and efficient AC protocol, and yet easy to implement and deploy.

III. DACME Overview

In this section we present a brief description of DACME (Distributed Admission Control for MANET Environments) [4]. DACME is a distributed admission control system which allows achieving per flow QoS requirements in terms of bandwidth and delay. One of the main advantages of DACME is that it does not impose any specific requirements on MANET nodes besides the use of IEEE 802.11 and having routing support; in fact, DACME agents are only required at the communication endpoints. Since DACME does not have MAC level constraints, it can be implemented in an easy way on all systems supporting the standard TCP/IP architecture.

A. Admission decision algorithms meeting bandwidth requirements

DACME learns about the network status using a probe/response strategy. In particular, to achieve an accurate bandwidth estimation, DACME uses a burst of probe packets periodically generated by the source in a back-to-back fashion. These packets arrive to the destination node with an average inter-packet time gap which allows the destination node to make an estimation of the available end-to-end bandwidth. When the destination gathers all the data needed, it sends a response packet with the current bandwidth estimation (BM) back to the source. An illustration of this strategy can be found in figure 1.

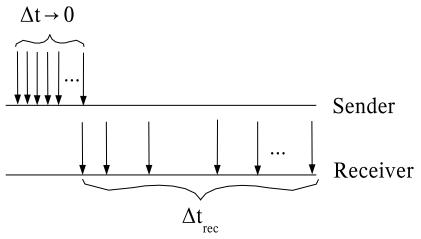

Fig. 1. Bandwidth estimation strategy.

After receiving each BM_i ***do*** *{*

$\mu_i \leftarrow \frac{(i-1)\cdot\mu_{i-1}+BM_i}{i}$, $\sigma_i \leftarrow \sqrt{\frac{(i-2)\cdot(\sigma_{i-1})^2+(BM_i-\mu_i)^2}{i-1}}$

if $(\mu_i - t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}} > B_R)$

then *Flag(BW)* $\leftarrow$ *1*

else if $(\mu_i + t_{i-1,0.95}\frac{\sigma_{p_i i}}{\sqrt{i}} < B_R)$

then *Flag(BW)* $\leftarrow$ *0*

else if *(i < 5)*

then *send a new probe }*

Fig. 2. The CIB-DA decision algorithm.

When receiving a response, the source applies an algorithm to decide whether the flow is accepted or rejected. The admission control algorithm adopted by DACME heavily relies on bandwidth estimations to decide whether to admit or deny a flow. For this reason, in this section we will focus explicitly on this DACME algorithm. In particular, we first describe CIB-DA, the original decision algorithm proposed in [4], and we then propose HRCI-DA, a novel mechanism we have developed based on experimentation, which offers an improved behavior in real environments compared to the former.

A.1 CIB-DA: Confidence Interval Based Decision Algorithm

The CIB-DA algorithm uses the values of up to five probes to obtain a 95% confidence interval for the available bandwidth value, based on which flow acceptance/denial decisions are made. Algorithm 2 shows the pseudo-code that describes the behavior of the CIB-DA algorithm. Previous works [4] have shown that this algorithm is highly effective in simulated MANET environments.

CIB-DA is executed every time a probe reply is received. Decisions are based on statistical confidence levels; therefore, i refers to the current iteration, $t_{i-1,0.95}$ to a Student's t-distribution with $i-1$ degrees of freedom, and for a confidence level of 95%. Parameter B_R refers to the bandwidth required by the application, while BM refers to the bandwith measurement explained previously. If the application is solely bandwidth constrained, the value of the bandwidth flag - *Flag(BW)* - will determine whether the QoS flow can be accepted.

A.2 HRCI-DA: Hybrid Range/Confidence Interval Decision Algorithm

Contrarily to what occurs in simulations, lots of problems exist in real testbeds, not only at the transmission level (e.g. interferences, packet loss), but also at the application, kernel, and hardware levels.

After receiving each BM_i ***do*** *{*

$range \leftarrow max(\{BM\}) - min(\{BM\})$

$\mu_i \leftarrow \frac{(i-1)\cdot\mu_{i-1}+BM_i}{i}$, $\sigma_i \leftarrow \sqrt{\frac{(i-2)\cdot(\sigma_{i-1})^2+(BM_i-\mu_i)^2}{i-1}}$

find the unbiased bandwidth estimator $\upsilon_{p,i}$

if $(\frac{range}{2} > t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}})$ ***then***

$(B\hat{W}_{low}, B\hat{W}_{high}) \leftarrow (\mu_i - t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}},\ \mu_i + t_{i-1,0.95}\frac{\sigma_i}{\sqrt{i}})$

else $(B\hat{W}_{low}, B\hat{W}_{high}) \leftarrow (\mu_i - \frac{range}{2},\ \mu_i + \frac{range}{2})$

if $(B\hat{W}_{low} > B_R)$ ***then*** *Flag(BW)* $\leftarrow$ *1*

else if $(B\hat{W}_{high} < B_R)$ ***then*** *Flag(BW)* $\leftarrow$ *0*

else if *(i < 5)* ***then*** *send a new probe }*

Fig. 3. The HRCI-DA decision algorithm.

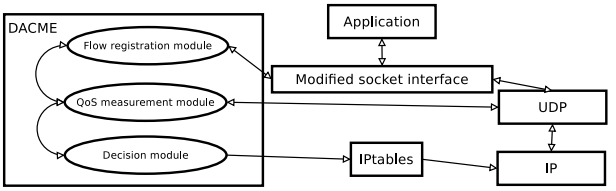

Fig. 4. DACME diagram illustrating the dependencies among the different elements involved.

Fig. 5. Illustration of the chain topology used for testing.

Examples of effects occurring at these levels include (but are not limited to) CPU Usage, RAM paging, time measurement delays, and loss of synchronization.

Focusing on the interferences problem, one of the main differences between simulation and real testbed experiments has to do with wireless channel. While in simulation experiments wireless channels are free from external interferences, in testbeds this is rarely true. In fact, in our laboratory, we experienced significant interferences when carrying out the tests, which caused estimated bandwidth to experience frequent and drastic fluctuations. This impeded obtaining low confidence intervals in most situations since we used a T-Student function with only a few degrees of freedom for calculating those intervals. Such large confidence intervals provoked that most flows were not accepted, or decisions could not be made even when the minimum bandwidth values measured were higher than the demanded ones.

To avoid this problem we formulated HRCI-DA, an improvement to the CIB-DA measurement algorithm. The main goal of HRCI-DA is to reduce the interval used to make flow acceptance/denial decisions, never allowing it to become higher than the half range, that is, half the difference between the maximum and minimum measurements. Algorithm 3 shows the pseudo-code for HRCI-DA.

In this algorithm two different intervals are used to determine the value of the bandwidth flag - *Flag(BW)* -, being one based on the range of the values, and the other based on confidence intervals. The former is used whenever the values produced by the latter strategy are too high.

B. Implementation details

To evaluate the effectiveness of DACME in a real environment, we developed an application level library[1] that interacts with both the applications and the kernel to achieve all the required functionality. Figure 4 shows the interaction between the different DACME elements.

Initially the application must register with DACME by using a modified socket interface that also accepts flow QoS specifications (bandwidth required, maximum delay, maximum jitter) as input. This interface, which is part of the developed library, exchanges information with both the operating system (creating a regular socket for communications) and DACME's core. The flow is registered with DACME by including not only the QoS specifications, but also the connection details (source port, destination port, destination IP). In a second step, DACME's QoS measurement module will probe the end-to-end path using the techniques described above. By interacting with the DACME agent at the destination, the DACME agent at the source will gather information that will allow it to make admission control decisions using any of the bandwidth-based decision algorithms presented in section III-A. According to the decision made, this module relies on the *iptables* tool [9] to dynamically block flows during periods of congestion, unblocking them when QoS conditions are again met. The IP ToS header field of accepted flows is also modified in order to take advantage of the IEEE 802.11e Video and Voice medium access categories. Notice that the IEEE 802.11e MAC driver will automatically map the IP ToS values to the four different medium access categories available.

[1]Freely available at: http://dacme.sourceforge.net/

IV. Testbed setup

For testing the effectiveness of DACME in a real testbed we used a set of 8 PCs, where 6 of them are low-end Asus EeePC netbooks, one is a regular laptop, and another one is a desktop computer. All the netbooks have a Ralink RT2860 wireless card, and the laptop and desktop systems use a Linksys WUSB600N USB wireless card which employs the Ralink RT2870 chipset. All wireless cards support the IEEE 802.11n draft3 standard, which includes IEEE 802.11e QoS extensions by default. The drivers employed were available in Linux kernel version 2.6.32, allowing us to build a realistic system with all its inherent characteristics and problems.

The eight stations are wirelessly connected according to a chain topology (see figure 5) to achieve a seven-hop ad-hoc network, which allows us testing with different hop number combinations per flow. The different stations involved in the tests are also interconnected via Ethernet for remote experiment control and, as a return channel, to measure the delay of the UDP packets injected. To avoid high CPU usage, we select different source/destination pairs for the different traffic flows. To introduce variable degrees of congestion in our tests, we also relied on Best Effort traffic flows. Notice that all the video

TABLE I
FLOW ENDPOINT DEFINITION FOR BOTH VIDEO AND BEST EFFORT TRAFFIC FLOWS.

Video	End points	BestEffort	End points
#1	C → F (3 hops)	#1	A → H (7 hops)
#2	E → H (3 hops)	#2	C → F (3 hops)
#3	G → D (3 hops)	#3	G → B (5 hops)
#4	F → C (3 hops)	#4	E → A (4 hops)

flows share a same link (between terminals E and F) which becomes the network's bottleneck.

To manage the repeatability of the experiments we used Castadiva [10], a tool previously developed by our research group which we extended to achieve DACME compatibility, as well as an experiment planner which allows automating large sets of experiments and collect all the statistics required.

In our experiments we vary both the number of best effort traffic flows and QoS (video) traffic flows. To introduce variable degrees of congestion, each best effort traffic flow consists of a 1.5 Mbit/s UDP stream; by varying the number of best effort flows we were able to achieve different channel congestion conditions. With respect to video flows, they consist of synthetic traffic at a rate of 1 Mbit/s (unidirectional). Table I summarizes the stations acting as sources and destinations of the different video and best effort traffic flows. Notice that hop count values are representative of typical MANET studies.

V. EXPERIMENTAL RESULTS

In this section we perform a detailed evaluation to assess the improvements introduced by both DACME AC algorithms (CIB-DA and HRCI-DA). With this goal we created two scenarios: in the first one we varied congestion by increasing the number of best-effort flows (background traffic), and in the second one we increased the number of competing QoS flows. Our goal was to study the QoS stability of the DACME flows in a real environment. Thus, for each test, the performance parameters under analysis were: throughput, delay, total activity time per-flow, and mean number of DACME on/off state transitions.

A. Varying the number of Best Effort flows

In this first set of experiments we study how best effort traffic affects the stability of video flows. In our experiments we have three concurrent video flows which start at random times during the first 20 seconds of the experiment, and then they last for 80 additional seconds. Additionally, for each test, we increase the number of best effort flows from zero to four to assess the impact of congestion on QoS performance. Notice that, for all the results presented, each measurement corresponds to the mean of 25 independent tests, each lasting 100 seconds. Throughput and delay values include 95% confidence intervals.

Figure 6 (left) shows the mean time each QoS flow is active. Notice that, if we do not use DACME, the activity time is always the maximum. The usage of DACME implies that, during certain periods, the flow can not be accepted since QoS requirements are not met; however, we find that, on average, the activity periods are maintained high, as desired. Focusing on the differences between CIB-DA and HRCI-DA, we find that the former is more restrictive, typically introducing more flow blockage in order to meet the QoS requirements.

Figure 6 (right) shows the mean number of state transitions per flow, which refers to transitions from active to blocked state, or the opposite. Since DACME performs periodic bandwidth measurements, it can block flows during ongoing communication as soon as QoS loss is detected. Although these dynamic decisions are mandatory to handle the effects of mobility, it is important to maintain this value as low as possible. Results show that the HRCI-DA offers significant improvements compared to CIB-DA, reducing the mean number of state transitions by up to 40%, and making communication more fluid.

Figure 7 (left) shows the mean throughput per video flow during periods of activity. We can see that bitrate values are maintained close to the maximum if DACME is used (for both admission control algorithms). When DACME is not used, we can clearly observe the negative impact of background traffic, drastically affecting the QoS of the video flows. The confidence intervals presented further evidence the goodness of both DACME decision algorithms, showing that bitrate variability associated with these algorithms is much lower compared to the "Without DACME" situation. In particular, we found that the standard deviation when using DACME was never higher than 19% while, when not using DACME, it is never lower than 30%, surpassing 100% in the worst case.

Concerning delay, figure 7 (right) shows that the differences between using DACME and not using it are again quite noticeable. In particular, we can see that, when the video flows are managed by DACME, the delay is typically lower than 100ms, being adequate for real-time communication. On the contrary, if we do not use DACME, the mean delay rises up to 700ms, being the lowest value of about 250ms. Similarly to what occurs for throughput, the confidence intervals for the delay are very low if DACME is used, becoming quite high otherwise.

Overall, the results presented in this section show that the usage of DACME is of utmost importance to achieve good QoS levels in real environments, which agrees with previous simulation results. Additionally, we find that the novel HRCI-DA algorithm, which addresses the specificities of real testbeds, improves the original CIB-DA strategy by increasing the overall activity time and reducing the number of transitions, while maintaining the good QoS values for both throughput and delay.

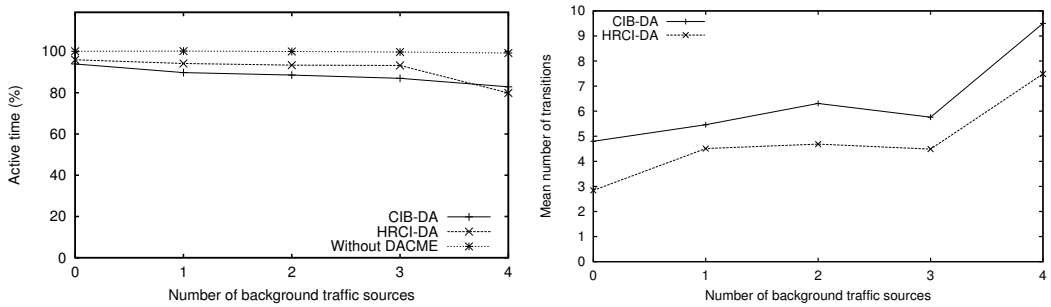

Fig. 6. Mean active time per flow (left) and mean number of state transitions for DACME flows (right) when varying the number of background traffic sources.

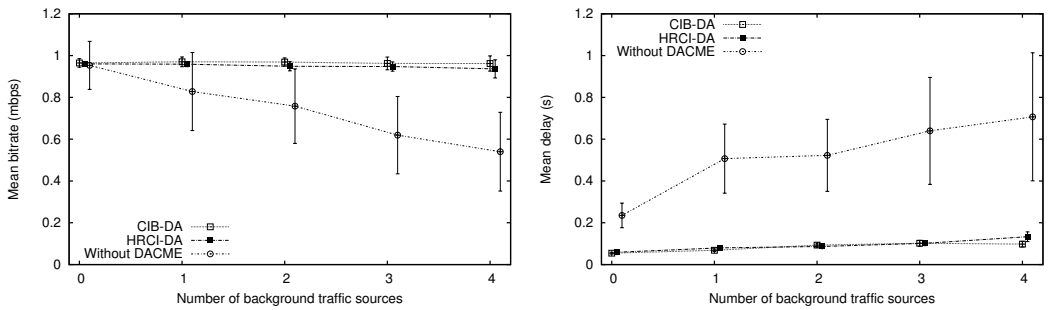

Fig. 7. Mean values for throughput (left) and end-to-end delay (right) when varying the number of background traffic sources.

B. Varying the number of Video Flows

The goal of this second set of experiments is to assess the performance of the different DACME AC algorithms when handling a variable number of QoS flows, as well as analyzing the interactions between these flows. With this purpose our experimental settings are similar to those of the previous section, but we now fix the number of best effort flows to three, and we increase the number of video flows from one to four.

Figure 8 shows the results obtained in terms of mean active time per flow (left) and mean number of state transitions (right).

Focusing on activity time, we find that increasing the number of video sources does not cause a proportional decrease in terms of activity time. This is mostly due to the distributed nature of wireless channel access in MANETs. Comparing both decision algorithms, we find that HRCI-DA offers better results in terms of both activity time and mean number of state transitions. In fact, the latter experiences a reduction of up to 30%, which shows the effectiveness of HRCI-DA at improving video streaming stability in real environments compared to its predecessor (CIB-DA).

In terms of throughput, figure 9 (left) shows that both DACME AC algorithms allow achieving a similar throughput (nearly 1 Mbit/s) while, when DACME is not used, this value drops to about 0.6 Mbit/s. As in the previous set of tests, the confidence interval obtained is much lower when we use DACME, meaning that variability is strongly reduced compared to the "Without DACME" case. This set of experiments allows concluding that, even when using IEEE 802.11e to achieve traffic differentiation at the MAC layer, the Video traffic category at the MAC layer is still highly affected by Best Effort traffic interferences; thus, although IEEE 802.11e allows some differentiation between the different traffic categories, it is not powerful enough by itself to provide full QoS guarantees in MANET scenarios.

Focusing on the delay (see figure 9, right), the behavior is similar to that of previous tests. Again, the delay experienced by QoS traffic becomes excessive when DACME is not used, and delay variability significantly increases. Additionally, we find that, when using either DACME decision algorithm, the delay experienced by the video flows is maintained low, and mostly immune to the increase of video sources.

The results presented in this section validate the effectiveness of DACME in real testbeds, and evidence the improvements introduced by the HRCI-DA decision algorithm compared to its predecessor (CIB-DA) in such environments. Hence, we can conclude that our DACME implementation mostly retains the QoS properties inferred based on simulation results, although some adjustments can help at further boosting performance in real environments.

VI. Conclusions and future work

In this paper we presented the implementation of DACME, a distributed admission control system for MANET environments that was earlier validated through simulation. We test its effectiveness in a real testbed using different performance indexes such as throughput, delay, total time of activity, and number of on/off state transitions for QoS flows.

To cope with bandwidth estimation accuracy problems occurring in the testbed, but not in simulation, we proposed a novel decision algorithm (HRCI-DA) for the admission control module that offers significant performance improvements compared to the

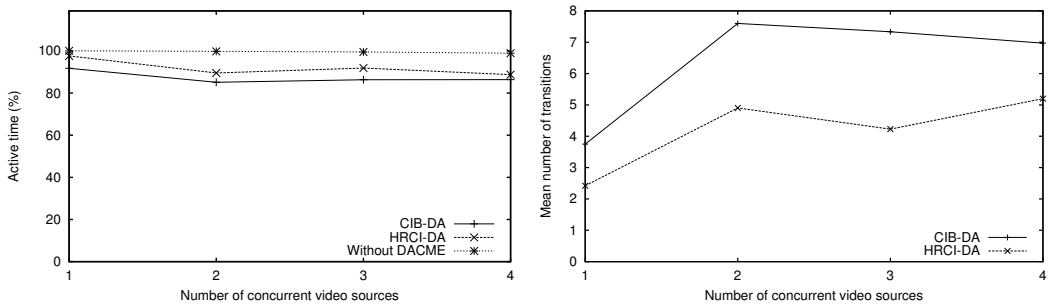

Fig. 8. Mean active time per flow (left) and mean number of state transitions for DACME flows (right) when varying the number of concurrent video sources.

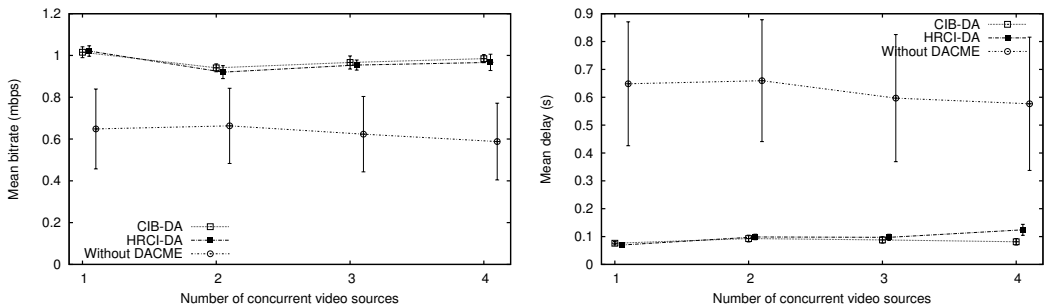

Fig. 9. Mean values for throughput (left) and end-to-end delay (right) when varying the number of concurrent video sources.

previous version (CIB-DA). HRCI-DA reduces the number of on/off state transitions for QoS flows, and improves the total activity times, while also maintaining good throughput and delay values.

Overall, the results presented in this paper clearly show that: (i) traffic differentiation provided by IEEE 802.11e is not enough in real multi-hop ad-hoc networks, and so a distributed admission control like DACME becomes essential; (ii) the proposed HRCI-DA algorithm improves the previous version by providing greater stability to QoS flows, increasing their total activity time and reducing the total number of on/off state transitions; (iii) despite the greater number of active QoS-flows, bandwidth and delay values are maintained or even improved by HRCI-DA compared to CIB-DA; and (iv) the DACME QoS architecture was fully effective in a real testbed, successfully validating the previous simulation results obtained.

As future work we plan to test with several routing algorithms and mobility patterns as well as with scalable video streams, adapting DACME decision algorithms to the multi-level quality characteristics inherent to such streams.

Acknowledgments

This work was partially supported by the Ministerio de Ciencia e Innovación, Spain, under Grant TIN2011-27543-C03-01, and also by the Ministerio de Educación FPU (Formación de Profesorado Universitario) grant program - AP2009-2415.

References

[1] J. Cano, J.-C. Cano, Chai-Keong Toh, C.T. Calafate, and P. Manzoni. EasyMANET: an extensible and configurable platform for service provisioning in MANET environments. *IEEE Communications Magazine*, 48(12):159–167, 2010.

[2] IEEE 802.11 WG. International Standard for Information Technology - Telecom. and Information exchange between systems - Local and Metropolitan Area Networks - Specific Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, ISO/IEC 8802-11:1999(E) IEEE Std. 802.11, 1999.

[3] IEEE 802.11 WG. 802.11e IEEE Standard for Information technology- Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 8: Medium Access Control (MAC) Quality of Service Enhancements, 2005.

[4] C.T. Calafate, M.P. Malumbres, J. Oliver, J.C. Cano, and P. Manzoni. QoS Support in MANETs: a Modular Architecture Based on the IEEE 802.11e Technology. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(5):678 –692, May 2009.

[5] Hanzo II and Rahim Tafazolli. Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey. *Communications Surveys & Tutorials, IEEE*, 11(4):78–108, December 2009.

[6] A. Derhab and A. Bouabdallah. Admission control scheme and bandwidth management protocol for 802.11 ad hoc networks. In *Innovations in Information Technology, 2007. IIT '07. 4th International Conference on*, pages 362 –366, nov. 2007.

[7] Anders Lindgren and Elizabeth M. Belding-royer. Multipath admission control for mobile ad hoc networks. In *in: Proceedings of Mobiquitous*, 2005.

[8] Seoung-bum Lee, Gahng-Seop Ahn, Xiaowei Zhang, and Andrew T. Campbell. INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks. *Journal of Parallel and Distributed Computing*, 60:374–406, 2000.

[9] The netfilter.org iptables project. Available at: http://www.netfilter.org/. Accessed: July 28, 2011.

[10] Jorge Hortelano, Juan-Carlos Cano, Carlos T. Calafate, and Pietro Manzoni. Testing applications in manet environments through emulation. *EURASIP Journal on Wireless Communications and Networking, vol. 2009, Article ID 406979, 20 pages, 2009. doi:10.1155/2009/406979.*