A distributed envelope-based admission control for multihop IEEE 802.11 ad hoc networks

María del Pilar Salamanca and Néstor Peña
GEST - Electronics and Telecommunication Systems Group
Universidad de Los Andes
Bogotá, Colombia
Email: {marsalam, npena}@uniandes.edu.co

Nelson L. S. da Fonseca
Institute of Computing
Universidade Estadual de Campinas
Campinas, Brazil
Email: nfonseca@ic.unicamp.br

***Abstract*—Transmitting real-time traffic in ad hoc networks is such a complex process that even the packets of a traffic flow interfere among themselves. A large variety of mechanisms to provide Quality of Service guarantees to real time traffic have been proposed in the literature; admission control is one of them. This paper proposes a distributed, stateless, and routing protocol decoupled admission control scheme for ad hoc networks that guarantees average delay to more than one traffic class. During the admission process, probing packets are sent from the incoming node to the receiving node of the flow. Based on the traffic and service envelopes of the probing packets, the receiving node decides whether the new flow is accepted or rejected. The admission control scheme was tested in static networks, where it effectively controls the packet delay. In mobile networks, the algorithm was evaluated varying the amount of mobile nodes, which move with a pedestrian pattern. The operation limits of the admission control were determined, to guarantee maximum delay and to control the packet losses of each traffic class.**

***Index Terms*—ad hoc networks, admission control, envelopes**

I. Introduction

Ad hoc networks are formed by wireless nodes connected without a pre-built infrastructure. Neither access points nor base stations are required given that any node is able to route packets. In order to create an ad hoc network, only IEEE 802.11 ad hoc-mode enabled devices are required, allowing a rapid deployment at low implementation cost. These features turn ad hoc networks into a very attractive option for a wide range of applications, such as disaster recovery, animal tracking and battlefield operations, among others. Nevertheless, transmitting real-time traffic in this type of networks is a complex process due to the inherent difficulties of wireless communications, the interference among nodes and even the interference between packets of a traffic flow. In addition, given that nodes can freely move, traffic routes may change thus affecting other ongoing transmissions. A large variety of mechanisms to provide Quality of Service (QoS) guarantees have been proposed in the literature, such as QoS-enabled routing protocols [1], modifications to the MAC IEEE 802.11 standard [2] and admission control (AC) schemes [3]. This paper presents an AC scheme for ad hoc networks.

Several design trade-offs [4] must be considered to design AC schemes for ad hoc networks, such as whether or not the AC is coupled from the routing protocol, whether or not nodes along the route store state information (*stateful*) or if only source and destination nodes participate in the admission process (*stateless*). Stateless schemes are the simplest type of AC. They demand lower memory requirements given that only source and destination nodes run the algorithm that decides on the acceptance of incoming flows. Additionally, since knowledge of intermediate nodes is not necessary, stateless schemes may be decoupled from the network layer, allowing network nodes to use any standard routing protocol. Nevertheless, node mobility is a sensitive item given that a flow that is switched to a new route may unexpectedly interfere with other ongoing flows. Therefore, the ability of stateless AC schemes to support mobile nodes must be carefully assessed.

Some examples of stateless AC schemes are SWAN [5], DACME [6] and the proposal from Valaee [7]. These are distributed schemes that use probing packets to assess the state of the network. A distributed approach is chosen since ad hoc networks lack of centralized control. In SWAN (AC algorithm for Stateless Wireless Ad hoc Networks), the admission controller is located at the source node. The source node sends a probing packet where the intermediate nodes write the available bandwidth of the route. The incoming flow is accepted if the measured available bandwidth is greater than the bandwidth required by the flow. SWAN measures the available bandwidth of the route and supports node mobility because every node of the network receive information about packet delays from its MAC layer.

DACME (Distributed Admission Control for MANET Environments) uses sets of probing packets, sent back-to-back, to assess the available bandwidth. The destination node estimates the available bandwidth based on the number of probes that arrived and the time required to receive them. If the application is also delay constrained, a second step is performed. An *echo request* packet is sent by the source and the destination replies by sending an *echo reply* packet; the source node sends a new *echo request* packet as soon as the echo reply is received, and the process is repeated several times. After all the related measurements are finalized, the source node decides on the admission of the new flow. DACME was designed to work with IEEE 802.11e MAC, which already supports QoS by introducing different MAC Access Categories.

The AC proposed by Valaee also uses probing packets but they are employed to measure the *service curve* to infer the status of the network. The service curve is the amount of traffic served during an interval where packets are backlogged. Therefore, a network with light load will have a service curve close to the vertical axis, while a network with high load will have a service curve close to the horizontal axis. A new flow is accepted if the service curve, measured including the probing packets, is above a reference curve called the *universal service curve*. This approach demonstrates that the service curve certainly reflects the performance of the network. However, its ability to keep the network delay below the threshold was tested only in single-hop networks.

The contribution of this paper is an AC scheme designed for pedestrian multihop IEEE 802.11 networks that is able to guarantee maximum delay to more than one traffic class. The proposed AC scheme shares some features with the schemes described above: it is distributed, stateless, decoupled from the routing protocol and based on measurements. The admission process evaluates a previously discovered route by sending probing packets from the traffic generator node to the receiving node. The receiving node estimates the *envelopes* of the probing traffic and the service process and decides whether or not to admit the new flow. The envelopes are calculated according to the algorithm proposed by Cetinkaya *et al.* [8], which was already tested in chains of wireless nodes [9].

Two major requirements must be accomplished by the network nodes. The first is that all nodes must previously know the traffic classes that will be supported by the network and the amount of probing packets that should be expected during the admission process. The second requirement demands that both source and receiving nodes have synchronized clocks; otherwise, each node along the route should mark packet queueing time in the packets so that the destination node can compute the initial packet transmission time.

The structure of the paper is the following. Section II explains the algorithms that estimate traffic and service envelopes and that decide on the admission of a new flow. Section III describes the scenario and the traffic sources that were modeled. Section IV presents the results when the AC scheme is applied to static and mobile networks. Finally, section V summarizes the conclusions of this work.

II. Model Description

The *Envelope Process* concept is the foundation of the proposed AC scheme. Let $A(t)$ be the cumulative amount of traffic that arrived during the interval $(0, t)$. It is said that A has the envelope $\hat{A}$ if, for all t and τ, $0 \leq \tau \leq t$

$$A(t) - A(\tau) \leq \hat{A}(t - \tau) \tag{1}$$

Based on this concept, Cetinkaya *et al.* proposed a measurement-based envelope estimation algorithm that divides time into slots; this algorithm calculates the arrival envelope as the maximum traffic rate generated by the source, and the service envelope as the worst service provided by the network. Cetinkaya's algorithm was modified by Schlembach [10], who replaced the *time discretization* by a *data discretization* approach that calculates both envelopes as values in time units. In this way, the admission decision calculation is simpler since each traffic class has a QoS requirement given in terms of *delay*, thus eliminating additional conversions needed when the Cetinkaya's original algorithm is applied.

The admission process starts when the incoming node sends probing packets at a CBR rate equal to the peak rate of the flow. Each probe carries information regarding the peak rate of the incoming flow, the traffic class it belongs to and the time instant when the probe was sent, also known as *transmission time*. The receiving node stores the transmission and arrival times of each probing packet and, once the expected number of packets (or *window size*) is received, both arrival and service envelopes are estimated. Actually, this approach is called a data discretization algorithm because the envelopes are calculated only until a predefined amount of packets is received. The corresponding algorithms are explained in the following subsections and are based on the code available at [11].

A. Estimation of the arrival envelope

The arrival envelope characterizes the probing traffic by estimating the minimum time required by the source to generate a certain number of consecutive packets. Algorithm 1 illustrates the process to estimate the arrival envelopes.

Algorithm 1 Estimation of the arrival envelope

1: **if** $probesReceived = windowSize$ **then**
2:     $p, q, j, i = 0$
3:     Set n
4:     **while** $p < n$ **do**
5:         $j = 2^p$
6:         $minTime = 0$
7:         **for** $i = 0, i < (windowSize - j)$ **do**
8:             $temp = txTime[i + j] - txTime[i]$
9:             **if** $minTime > temp$ **then**
10:                 $minTime = temp$
11:             **end if**
12:         **end for**
13:         $R(q)_t = minTime$
14:         $p = p + 1; q = q + 1$
15:     **end while**
16: **end if**

The algorithm searches the *minimum time* elapsed at the source between the generation of packet i and packet $i + j$, where the value of j is set at line 5. A sliding window mechanism, represented by the *for* block, examines all the existing blocks of j consecutive packets within the window and stores the minimum time in the variable $minTime$. At the end of the process, the arrival envelope is a vector with *n* rows that stores the minimum time found for each value of j.

In order to obtain statistical envelope of the arrival process, the source sends M windows of probing packets. The mean value of the arrival envelope is also a vector with n rows, where each value $R(q)$ is given by

$$R(q) = \sum_{t=1}^{t=M} R(q)_t / M \tag{2}$$

and $R(q)_t$ is the arrival envelope of window t at row q and $q = 1, 2, \ldots, n$. The variance of the arrival envelope is expressed as

$$\sigma_q^2 = \frac{1}{M-1} \sum_{t=1}^{t=M} (R(q)_t - R(q))^2 \tag{3}$$

B. Estimation of the service envelope

The service envelope measures the maximum time required to service a certain amount of probing packets when they are backlogged; when the packets are not queued, only their individual delays are considered. Algorithm 2 shows the details of the estimation process.

Algorithm 2 Estimation of the service envelope

```
1: if probesReceived = windowSize then
2:     p, q, j, i = 0
3:     while p < n do
4:         j = 2^p − 1
5:         for i = 0, i < (windowSize − j) do
6:             flag = checkBackloggingCond()
7:             if flag = 1 then
8:                 temp = rxTime[i + j] − txTime[i]
9:                 if temp < maxTime then
10:                    maxTime = temp
11:                end if
12:            end if
13:        end for
14:        S(q)_t = maxTime
15:        p = p + 1; q = q + 1
16:    end while
17: end if
```

The algorithm evaluates if a predefined amount of consecutive packets, given by $j+1$, was continuously backlogged. A sliding window, represented by the *for* block, checks this condition for all the possible groups of $j+1$ consecutive packets within the window. The backlogging condition is true for two packets if the first packet has not arrived to the destination node when the second packet is generated at the source. In a group of $j+1$ packets, the algorithm initially verifies whether this backlogging condition is true for packets $j+1$ and j, then for packets j and $j-1$, and so on. If the algorithm confirms that this condition is true for all packets of the chosen group, the service time is calculated as the time elapsed since the first packet of the group was generated until the last one was received (line 8). The algorithm chooses the maximum service time for each group size and this value is stored in a vector with n rows, similarly to the arrival envelope.

The mean and the variance of the service envelope are calculated in a manner analogous to the mean and the variance of the arrival envelope.

C. Admission Control

Consider a set of probing packets that belongs to a certain traffic class, with a mean arrival envelope $R(q)$ and variance $\sigma^2(q)$. The mean and the variance of the service envelope are respectively $\overline{S}(q)$ and $\psi^2(q)$, and r is the peak-rate of the incoming flow . The flow is admissible with delay bound D and confidence level $\Phi(\alpha)$ if

$$\overline{S(q)} - \overline{R(q)} + \alpha\sqrt{\sigma^2(q) + \psi^2(q)} < D \tag{4}$$

for any $q = 1, 2, \ldots, n$. $\overline{R(q)}$ is the mean value of the arrival envelope including the incoming traffic rate, given by

$$\overline{R(q)} = \frac{2^q \cdot s \cdot 8}{\frac{2^q \cdot s \cdot 8}{R(q)} + r} \tag{5}$$

where s is the packet size in bytes. The confidence level α characterizes the variations of the measurements and the uncertainty of the prediction of future service and arrivals, and is determined by using the Extreme Value Theory [8]. Once the admission test is performed, the egress node notifies its decision to the source node.

III. Scenario Description

The envelope-based AC scheme was developed and tested in Qualnet® Simulator. The chosen scenario consists of 50 identical nodes, randomly placed in an area of 1500m x 400m. This area size was chosen to avoid network partitioning when nodes are moving. Nodes communicate with IEEE 802.11b using the DCF at a speed of 2 Mbps. The transmitter power was adjusted to obtain a transmission range of 250m. The routing protocol is AODV.

TABLE I
Traffic Parameters

Parameter	Voice	Data
Distribution	Exponential	Pareto
Mean ON Time	1.49s	250ms
Mean OFF Time	1.722s	250ms
Packet Size (Bytes)	48	1024
Delay Bound	100ms	500ms
Max. Bit Rate (bps)	7875	400000

Two traffic classes share the network resources: voice and data sources. Voice calls are modeled as two sources, one at each end, while data sources are modeled as a single source. The duration of each flow is exponentially distributed with a mean value of 5 minutes. Source and destination nodes of each flow are randomly selected among the 50 nodes. The parameters and delay bounds of each traffic class are shown in Table I. The chosen codec for voice calls was G.723.1 and its configuration is based on the voice traffic characterization presented in [12]. Traffic flows are generated alternately, i.e. one voice flow is followed by a data flow. Flow generation rate varies from 1 flow per minute (fpm) up to 6 fpm.

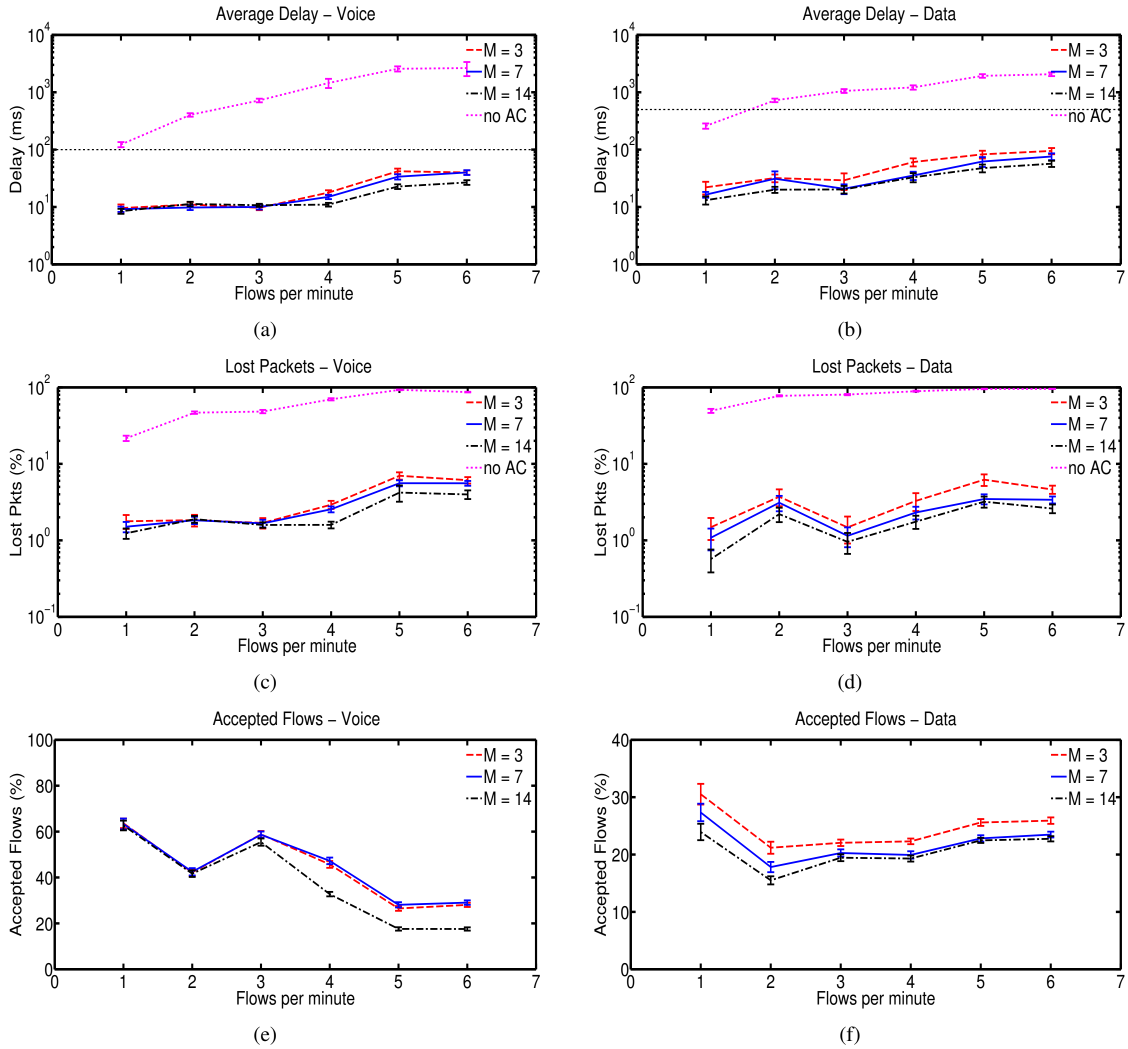


Fig. 1. Statistics of the envelope-based AC scheme with static nodes, for $M = 3$, $M = 7$, $M = 14$ and without AC. Graphs at the left side correspond to voice traffic statistics while right side graphs show data traffic statistics. (a) and (b) illustrate average packet delay; (c) and (d) depict lost packets; (e) and (f) show accepted data flows.

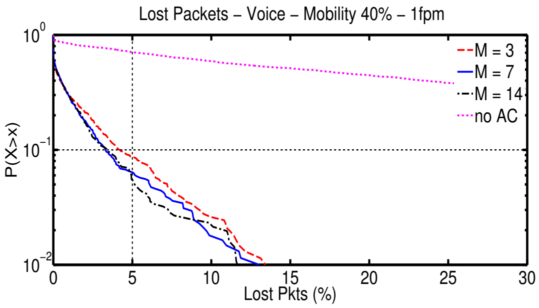
IV. Numerical Results

This section summarizes the results of the application of the envelope-based AC scheme to the scenario described above. The impact of the amount of probing packets on the network performance is assessed by using different values of M, i.e. number of windows of probing packets sent during the admission process. Each window has 16 packets.

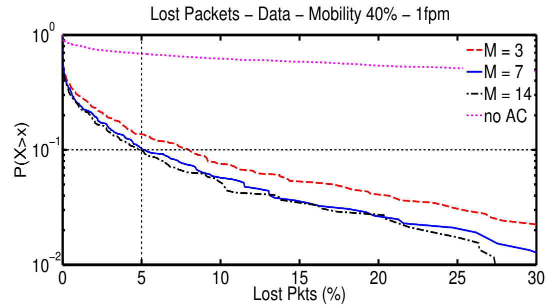
The duration of each simulation is 1800s. All figures show confidence intervals with confidence level of 95%, which were derived with at least 60 replications.

A. Performance without node mobility

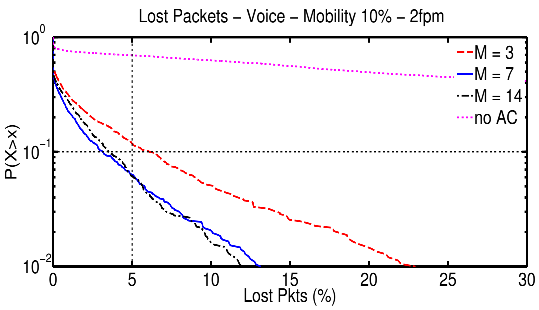
Fig. 1 shows results as a function of the number of fpm when all nodes are static, for $M = 3$, $M = 7$, $M = 14$ and without AC scheme. Graphs (a) and (b) correspond to the average delay of voice and data packets, respectively. Notice that delays have an increasing trend as the number of fpm grows, however the AC scheme is able to keep the average packet delay of both traffic classes below the bounds, even when 6 flows are arriving per minute. Without AC, the delay of voice packets exceeds the bound even with only 1 fpm. Packet losses of voice and data traffic are depicted in graphs (c) and (d), respectively. Observe that packet losses are higher for $M = 3$. This fact is related to the percentage of accepted flows, shown in graphs (e) and (f). With $M = 3$, the network accepts more data flows than with either $M = 7$ or $M = 14$, and data flows have the greatest bit rate. More flows competing for network resources increase packet losses. Therefore, regarding packet losses, a larger M improves the network performance. Nevertheless, it is important to keep in mind that larger values of M mean longer admission times and higher packet overhead. Finally, notice that without the AC scheme, packet losses can be greater than 90% and, with only 1 fpm, packet losses of data traffic are close to 50%.



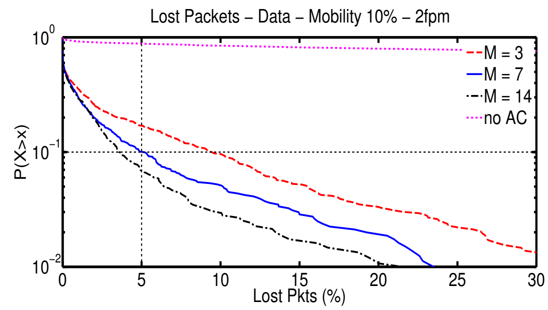
(a)



(b)



(c)



(d)

Fig. 2. Empirical complementary distribution of the packet losses for each traffic class in the operation limits, for $M = 3$, $M = 7$, $M = 14$ and without AC.

B. *Performance with node mobility*

In this section, the effect of node mobility on the performance of the AC scheme is assessed by letting a certain percentage of nodes move while the rest remain static. The envelope-based AC scheme is decoupled from the routing protocol, therefore it must be tested under low mobility conditions. Mobile nodes have a pedestrian mobility pattern that follows the random waypoint model, where the speed ranges between 0.5 and 1 m/s and the pause time is equal to 180 s. The analysis is focused on finding the value of M where the network operates as expected, not only in terms of average delay but also in terms of packet losses. As will be seen, the value of M also determines the operation limits of the AC. From now on, the percentage of mobile nodes will be called *mobility factor*.

Fig. 2 shows the empirical complementary distribution of the packet losses for voice (left side) and data (right side) traffic. The parameter M is chosen assuming the following condition: the maximum value allowed for $P(X > 5\%)$ is 0.1 for both traffic classes, where the random variable X corresponds to the percentage of packet losses. The dotted horizontal line shows the value $P(X > 5\%) = 0.1$, the dotted vertical line corresponds to $X = 5\%$, and both lines are used as reference. Besides determining the value of M, this condition also bounds the mobility factor supported by the network for a given value of fpm. In the studied case, the condition is true for 1 fpm when the mobility factor is less than or equal to 40%, and for 2 fpm if the mobility factor is equal to 10%. Graphs depicted in Fig. 2 correspond to these operation limits. Notice that either $M = 7$ or $M = 14$ could be chosen for both 1 and 2 fpm, although $M = 7$ means lower packet overhead for the network and a shorter admission time. $M = 3$ would not be a good choice given that cannot guarantee that the condition is accomplished in the operation limits evaluated; if it were used, the network would operate as expected only for 1 fpm and mobility equal to 10%. Notice that without the AC scheme, $P(X > 5\%) = 0.7$ in the best case.

Finally, Fig. 3 illustrates some traffic statistics as a function of the mobility factor for both 1 and 2 fpm. When the AC is applied within the operation limits, with either $M = 7$ or $M = 14$, the AC scheme keeps the average delay of voice packets below the bound, as shown in Fig. 3(a) and 3(b). The average delay of data packets for both 1 and 2 fpm is not shown, but is always below the threshold, even without AC. However, when none AC scheme is applied, the amount of lost data packets is really high. Fig. 3(c) and 3(d) depict the packet losses for data traffic with 1 and 2 fpm, respectively. Observe that with 1 fpm, the AC scheme keeps packet losses below 3%, while without AC are around 40%. With 2 fpm, packet losses with the AC scheme are below 4%, but are greater than 60% without AC scheme.

V. Conclusions and future work

In this paper, a distributed and stateless AC scheme for IEEE 802.11 wireless ad hoc networks was presented. The AC scheme is based on an adapted version of the envelope estimation algorithm proposed by Cetinkaya and is able to guarantee maximum delay to more than one traffic class. During the admission process, the incoming node sends probing packets

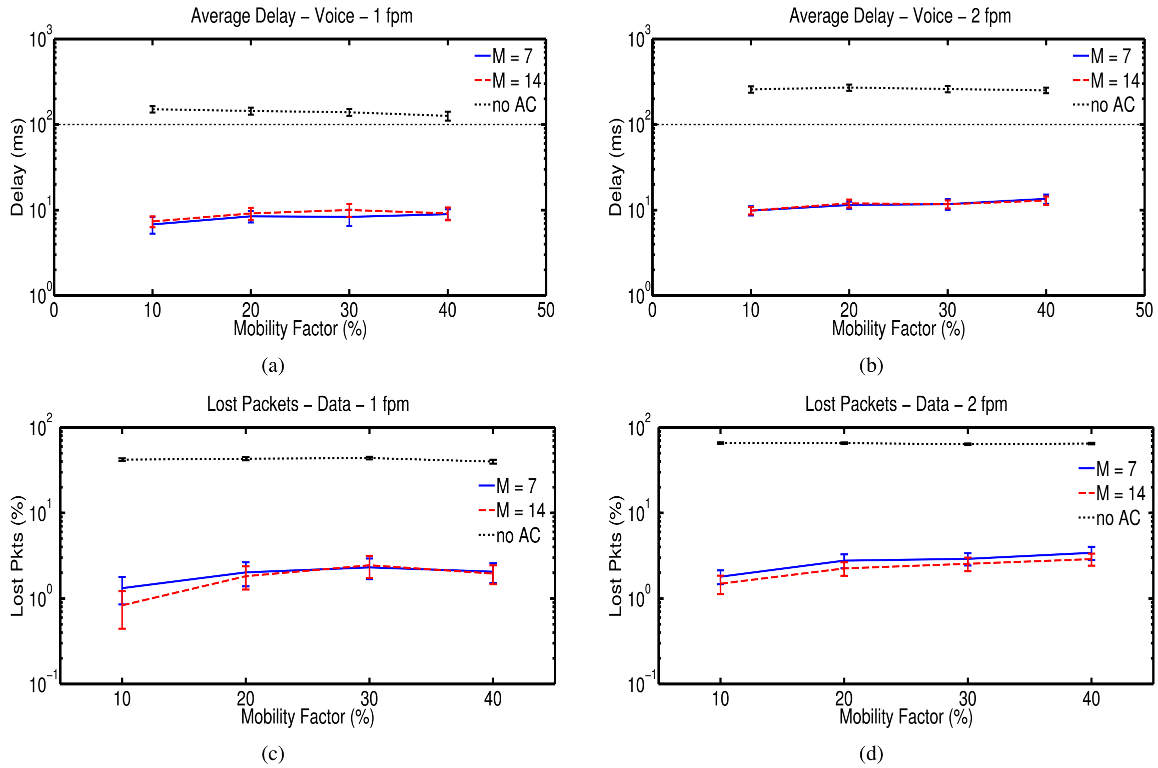


Fig. 3. Traffic statistics when the network is working in the operation limits.

at a CBR rate equal to the maximum rate of the flow and, once the destination node receives the expected amount of packets, the service and traffic envelopes of the probes are calculated. Based on the envelopes, the receiving node decides on the admission of the flow. The algorithm was tested on a scenario with 50 randomly placed nodes and two traffic types, voice and data, modeled by ON-OFF sources. Initially, the AC scheme was evaluated with static nodes, and it was shown that the algorithm effectively guarantees the delay requirements to both traffic classes. Then, the AC scheme was assessed in networks with pedestrian mobile nodes. The operation limits of the AC were determined so that not only the delay requirements but also the probability of packet loss exceedance were satisfied.

As future work, the envelope-based AC scheme will be improved with a cross-layer design to support higher mobility levels.

ACKNOWLEDGEMENT

This work was supported by LACCIR Federation Award grant S1110LAC003, by the Department of Electrical and Electronic Engineering, Universidad de Los Andes, and by CNPq.

REFERENCES

[1] L. Chen and W. Heinzelman, "A survey of routing protocols that support QoS in mobile ad hoc networks," *IEEE Network*, vol. 21, no. 6, pp. 30 –38, November-December 2007.

[2] S. Nazhad, J. Jabari, and R. Alguliev, "A new MAC algorithm for QoS support in ad hoc networks," in *Application of Information and Communication Technologies (AICT), 2011 5th International Conference on*, Oct. 2011, pp. 1–6.

[3] C. Calafate, J. Oliver, J. Cano, P. Manzoni, and M. Malumbres, "A distributed admission control system for MANET environments supporting multipath routing protocols," *Microprocessors & Microsystems*, vol. 31, pp. 236–251, 2007.

[4] L. Hanzo and R. Tafazolli, "Admission control schemes for 802.11-based multi-hop mobile ad hoc networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 78 –108, 2009.

[5] G. S. Ahn, A. Campbell, A. Veres, and L. H. Sun, "Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (SWAN)," *IEEE Trans. Mobile Computing*, vol. 1, pp. 192–207, July 2002.

[6] C. Calafate, J. C. Cano, P. Manzoni, and M. Malumbres, "A QoS architecture for MANETs supporting real-time peer-to-peer multimedia applications," in *Seventh IEEE International Symposium on Multimedia*, Dec. 2005.

[7] S. Valaee and B. Li, "Distributed call admission control for ad hoc networks," in *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, vol. 2, 2002, pp. 1244 – 1248.

[8] C. Cetinkaya, V. Kanodia, and E. W. Knightly, "Scalable services via egress admission control," *IEEE Transactions On Multimedia*, vol. 3, no. 1, pp. 69–81, 2001.

[9] M. del Pilar Salamanca, N. Pena, and N. da Fonseca, "Evaluation of an envelope-based access control scheme in multihop IEEE 802.11 ad hoc networks," in *Communications (LATINCOM), 2011 IEEE Latin-American Conference on*, Oct. 2011.

[10] J. Schlembach, A. Skoe, P. Yuan, and E. Knightly, "Design and implementation of scalable admission control," *Lecture Notes in Computer Science*, 2001.

[11] ——. Software for scalable admission control. Rice Networks Group. [Online]. Available: http://networks.rice.edu/ext/egress/index.html

[12] R. Estepa, J. Vozmediano, and A. Estepa, "Background noise influence on VoIP traffic profile," in *Lecture Notes in Computer Science*, V. Roca and F. Rousseau, Eds. Springer Berlin / Heidelberg, 2004, vol. 3311, pp. 13–24.