Computer Networks 53 (2009) 2586–2600

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Markovian-based traffic modeling for mobile ad hoc networks ☆

Carlos T. Calafate *, P. Manzoni, Juan-Carlos Cano, M.P. Malumbres

Universidad Politécnica de Valencia, Camino de Vera, S/N, 46022 Valencia, Spain

ARTICLE INFO

Article history:
Received 30 May 2008
Received in revised form 11 February 2009
Accepted 13 May 2009
Available online 25 May 2009

Responsible Editor: J. Zhang

Keywords:
Hidden Markov models
Mobile ad hoc networks
Video codec tuning

ABSTRACT

Mobile ad hoc networks (MANETs) show very significant difference with respect to other computer networks due to the presence of extremely large packet loss bursts. The development of protocols for mobile ad hoc networks, especially multimedia protocols, require extensive evaluation either through simulation or real-life tests. Such testing consumes a great amount of resources both in terms of time and trace file sizes. Therefore, finding efficient means of reducing the amount of data that is stored and processed is quite important to accelerate the evaluation of different audio/video streaming applications. If, moreover, we are able to model the loss pattern experienced, we can further accelerate the evaluation process.

In this work we propose two models based on hidden Markov chains that are able to grasp both packet arrivals and packet loss patterns in MANETs. A simpler two-state model is proposed to model losses when proactive routing protocols are used, while a more complex three-state model is proposed for reactive routing protocols. We also introduce a new set for packet loss pattern measurements that can be of interest for the evaluation of audio/video streaming applications.

Experimental results show that the proposed models can adequately reproduce extremely long packet loss patterns, typical of MANET environments, with a high degree of accuracy. Overall, we find that the proposed models are able to significantly reduce both the simulation time and the trace file sizes required.

1. Introduction

Mobile ad hoc networks (MANET) [1] are wireless networks where nodes can either be end-points of a data interchange or can act as routers when the two end-points are not directly within their radio range. Such networks are a solution to extend Internet connectivity to remote areas where no support infrastructure is available.

The most widely deployed technology to implement this kind of networks is based on the IEEE 802.11 [2] standard. Wireless ad hoc networks suffer from frequent topology changes and provide a poor QoS support. However, support for real-time communication in wireless networks is becoming more and more important due to the increasing demand for multimedia applications.

The issue of topology variability can only be handled through efficient routing mechanisms. A couple of years ago near to 60 proposals of routing protocols were being evaluated. Nowadays only four proposals, respectively the "Ad hoc On Demand Distance Vector" (AODV) [3], the "Dynamic Source Routing Protocol for Mobile Ad hoc Networks" (DSR) [4], the "Optimized Link State Routing Protocol" (OLSR) [5], and the "Dynamic On-demand Routing Protocol" (DYMO) [6], are being supported; AODV, DSR and OLSR have reached the *Request For Comments* (RFC) state.

Proactive routing protocols such as OLSR periodically send *"Hello"* messages for link state sensing. The delay

☆ A preliminary version of this article entitled "Speeding up the evaluation of multimedia streaming applications in MANETs using HMMs" was presented at the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'04).

* Corresponding author. Tel.: +34 96 387 7007x75727; fax: +34 96 387 7579.
E-mail address: calafate@disca.upv.es (C.T. Calafate).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.05.007

necessary to detect a broken link can be calculated as: $BLT = HI \times MHL$, where *HI* is the average *Hello* interval and *MHL* is the minimum number of consecutive *Hello*s lost that triggers a broken link event. Since $1 \leqslant HI \leqslant 2$ s and $2 \leqslant MHL \leqslant 3$, nodes will typically require between 2 and 6 s until the network topology updating task is activated. AODV and DSR are reactive routing protocols that typically use information from lower layers in order to detect broken links earlier. However, even though we achieve lower reaction times to link changes by enabling link awareness, the re-routing process can still introduce quite long disconnection periods.

Extremely long interruptions in communication is therefore a relevant problem that must be taken into consideration while evaluating different proposals at the higher layers of the communication model, like for example VoIP, video communication, or session management protocols. In the former cases, long loss bursts may cause the quality of an audio/video communication between two users to be unacceptable since the data flow is interrupted (silent audio/frozen video) during long periods of time (may be several seconds).In addition to mobility-related losses, other conditions such as channel fading, interference, noise and congestion also cause packet losses in wireless ad hoc networks.

By modeling the packet error bursts in an error prone network environment we achieve interesting benefits in terms of testing/simulation time and required resources for running and storing our simulation experiments, yet preserving the behavior in terms of packet loss bursts. A video codec is an example of a higher layer software which can benefit from this solution: distinct strategies can be tested in terms of error resilience and quick error recovery, obviating the need for several long simulation runs.

In this work we present two models for characterizing packet arrivals and packet loss patterns in MANETs based on hidden Markov models [7] (HMMs) theory. Though the models derived could be used in other packet networks, our focus on MANETs is due to the unusually large packet loss bursts that are prone to occur in these networks. We believe that such packet loss patterns impose great demands on the model, so that model validation is done in an extreme situation. We also propose new metrics aiming specifically at audio and video streams by extending some of the concepts presented in [8].

The article is organized as follows: in the next section we refer to some related works in the field. In Section 3 we describe our model and the methodology followed, and we also present the two proposed models to characterize packet loss bursts in multihop wireless paths. Section 4 presents a novel set of metrics for packet loss bursts, which are used to assess the degree of accuracy of the proposed modeling strategy. These metrics allow assessing how different routing protocols perform during a well-defined period in terms of loss bursts. Section 5 illustrates with an example the benefits obtained in terms of simulation time and resources saved when using the models derived. An extensive model validation work is presented in Section 6, including both congestion and mobility modeling. Finally, Section 7 presents the conclusions of our work.

2. Related work

Hidden Markov models have initially been developed to address the requirements of speech recognition. However their use has been spread to several other areas, like the computer networks area. Wei et al. [9] propose a solution based on modeling that uses periodic end-to-end probes to identify whether a "dominant congested link" exists along an end-to-end path. In [10], Liu et al. obtain an improved TCP version through end-to-end differentiation between wireless and congestion losses, providing effective operation in hybrid wired/wireless environments. Their approaches integrate HMMs with packet loss pairs (PLP).

In the literature we can also find uses of the simpler Markov chains in the Internet. Jiang and Schulzrinne [11] propose some extensions to the Gilbert model for Voice over IP (VoIP) applications taking into consideration playback delay adjustments and FEC. Their focus is on Internet behavior and the interactions between the model and the FEC mechanism. Sanneck et al. [12] also focus on Internet-related losses by describing model parameters using packet loss metrics existing in the literature.

Concerning the applications of Markov chains to MANETs, Lin and Midkiff [13] propose a link connectivity model using two-states Markov chains to estimate the link up/down times between a pair of nodes and compare it to the Random Waypoint Mobility Model. Each potential link for all pairs of nodes is modeled as a separate two-states Markov chain to derive a node connectivity matrix. In their work they focus solely on modeling mobility, and not on packet loss patterns.

In terms of loss burst metrics, IETF's RFC 3357 [8] defines two metrics, namely the "loss distance" and the "loss period", and the associated statistics that together capture loss patterns experienced by packet streams on the Internet through probes. However, the statistics proposed do not offer an in depth view of the packet loss phenomena.

To the best of our knowledge, our proposal is the first one to use hidden Markov models to model the extremely large packet loss bursts that are prone to occur in MANET environments at the application layer (end-to-end). Moreover, we introduce novel metrics that are able to adequately characterize the loss phenomena detected.

HMMs are well known for their effectiveness in modeling bursty behavior [14,15], relatively easy configuration, quick execution times achieved and general applicability. Therefore, we consider that they fit our purpose of accelerating the evaluation of multimedia streaming applications adequately, while offering similar results as with simulation or real-life testbeds.

3. Model description and proposed methodology

The IEEE 802.11 standard for wireless LANs [2] defines several mechanisms for reliable packet transmission in noisy wireless channels. Since all data is protected with a CRC field, it is unlikely that a corrupted packet gets to the destination, even if using an unacknowledged service, like with broadcast or multicast traffic. We can therefore

assume that either the packet is completely received, or it is dropped.

In a previous work [16] we found that routing related losses can provoke quite large packet loss bursts. Fig. 1 shows an example of the impact of mobility on a real-time H.264 video stream in a typical MANET scenario with no additional sources of traffic, and using DSR as the routing protocol. As it can be seen, even in these ideal conditions the data stream is affected by various communication gaps.

We assume that stations belonging to the MANET are found in different routing states (e.g. route available, route discovery, re-routing, etc.). Anyway, independently of the routing state, packet losses can occur for a variety of other reasons (collisions, channel noise, queue dropping, etc.). Therefore, an outside observer cannot relate a packet loss with a certain routing state. We deal with a situation where the observation is a probabilistic function that depends on the state. This means that only the output of the system, and not the state transitions, are visible to an observer. We will therefore try to solve the state assignment problem using a *hidden Markov model* (HMM) [7].

3.1. General methodology

We start by selecting a data stream (e.g. audio, video, etc.) for analysis, as well as the criteria for considering a packet good or unusable by the application. We can take into account factors such as which packet arrives to destination within a maximum delay, the delay jitter limits, the dependency among packets, etc. We then label each packet sequence number with value 1 – considering the packet good – or value 0 if the packet does not arrive to destination, or does not meet any of the chosen criteria. Starting from this sequence of observations we obtain the distributions of *consecutive packets arriving* (CPA) and *consecutive packets lost* (CPL). The latter two will be used to tune the proposed HMMs for a particular network profile (area, number of nodes, mobility pattern, traffic load, etc.).

In a HMM the number of states is not defined by the possible output events. To choose an adequate HMM configuration we propose starting from a very simple 2-state model as presented in the next section. We consider that one of the states models a currently broken path, where the probability for a packet to reach destination is zero. The other state models packets lost mostly due to collisions, but also due to channel noise, packet fragmentation, buffers overflow, and the type of MAC used; the probability for a packet to reach destination is given by function $h(s)$, where s is the packet size.

Starting from the 2-state model, we can compare the model's output with the distributions used for its tuning, and assess if the desired degree of accuracy is achieved. If the results are not accurate, we add one more state to the model and repeat the process until the results are satisfactory.

We estimate the different parameters of the HMM taking as input the results of experiments from the ns-2 simulator [17]. The characteristics of the routing protocol employed can be useful to provide an insight on how to enhance the model (see Section 3.3 for an example). In our experiments we did not have to use more than three states, showing that the model complexity can be kept low and still provide the desired results.

In the two following sections we show how to model the transmission of data streams on MANETs using routing protocols such as OLSR or DSR using 2-states and 3-states HMMs. In order to speed up the determination of the optimum values for the model parameters, we also present, for each case and for each parameter, a set of heuristics that offers good estimates.

3.2. Two-states packet loss burst model

In this section we present the simplest HMM that is able to model large lost bursts. The idea is to focus on two distinct situations: when a path towards the destination is lost and no packet can arrive successfully, and when a path to the destination exists but some of the packets are dropped due to congestion, transmission errors, buffer overflow, etc. It consists of a two-states HMM based on the Markov chain shown in Fig. 2 (also known as the Gilbert model).

State B models the situation where a path towards the destination has been lost; the probability for a packet to reach the destination is zero. In state F packets are lost according to a probability defined by function $h(s)$, where s is the packet size. Mapping state B with 0 and state F with 1 we obtain the following transition probability matrix:

$$A_2 = \begin{bmatrix} a_{00} & a_{01} \\ a_{10} & a_{11} \end{bmatrix}. \quad (1)$$

For our experiments based on ns-2 [17] we have tested several different scenarios with different mobility and traffic patterns, and we have chosen one that was particularly representative in terms of large packet loss bursts. This

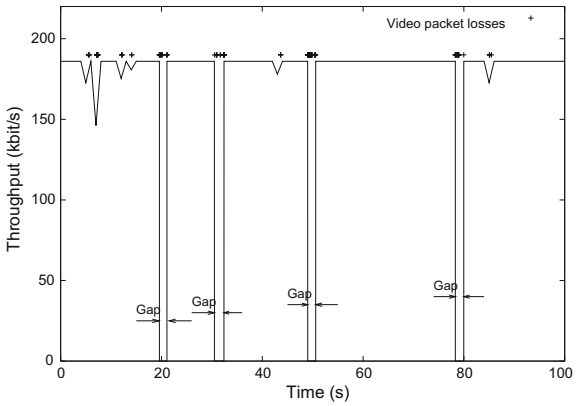

Fig. 1. Impact of mobility on a real-time video stream.

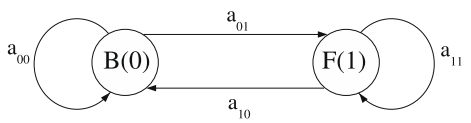

Fig. 2. Two-state Markov chain for the multi-hop wireless path model.

choice aimed at stressing the model using a very demanding example.

Our setup consists of a 1000 m × 1000 m scenario with 80 nodes. The wireless interfaces are based on the IEEE 802.11b standard with radio range limited to 250 m. The medium access used is the distributed coordination function (DCF). The node mobility is generated using the random waypoint model with node speed between 0 and 12 m/s. The source of the reference flow sends packets with random sizes ranging between 64 and 2300 bytes at a rate of 50 pkt/s. The background traffic consists of 4 UDP sources generating 512 bytes packets at a rate of 4 pkt/s. We evaluate both a reactive (DSR) and a proactive (OLSR) routing protocol. Applying a filter to the simulation's output we obtain a trace file (*ST*) where incrementing packet sequence numbers are tagged with either a 1 or a 0 (for packets received and packets lost respectively). Our criteria is that all packets that arrive to destination in less than 300 ms are considered good packets (tagged 1); the remaining were tagged as lost (0). From this sequence of observations we calculate the distribution of *consecutive packets lost* (CPL) and of *consecutive packet arrivals* (CPA). These distributions will be used as training sequences when tuning our model.

Using trace *ST* we first analyzed the correlation between packets size and the event of losing, or not, a packet. The correlation coefficient found is $r^2 = 6.03 \times 10^{-6}$ which indicates that, within our simulation framework, the event of losing a packet is basically independent from packet size. Therefore, in our experiments, the probability function associated to state F will be fixed at a constant value $h(s) = \bar{h}$.

From the first distribution obtained (CPL) we calculate the ratio between the total number of packets lost and the sum of the lengths of CPL sequences bigger than one; we do the same with the packets received. Table 1 shows the obtained values. The ratios for the packets received is high, as expected. The interesting result is that the ratios for packets dropped are also high, indicating that packet loss bursts are the dominant cause of losses, contrarily to a random-loss situation. Notice that in mobile ad hoc networks frequent route losses due to mobility are the main cause for the bursty loss phenomenon. Thus, since the main reason for sequential packet losses is a route failure, these events shall take place mostly in state B.

From these results, and since parameter $\bar{h}$ accounts mainly for non-consecutive packet losses, we consider that $\bar{h} = 1 - \varepsilon \approx 1$. This allows us to propose an heuristic to find the vector of estimated parameter values $\hat{v} = (\hat{a}_{10}, \hat{a}_{01}, \hat{h})$ that offers a high degree of accuracy (near-optimal solution). Notice that if the condition $\bar{h} \approx 1$ no longer holds, the accuracy of the heuristic we now present would only be slightly reduced.

Table 1
Ratios relative to the total number of packets lost/received with respect to the sum of the sequences bigger than 1 of packets lost/received.

	DSR RB (%)	OLSR RB (%)
Packets dropped	97.9	99.77
Packets arriving	99.74	99.98

We estimate $\hat{a}_{10}$ and $\hat{a}_{01}$ taking into account that the runs at each state of a Markov chain are memoryless, having by definition a geometric distribution. Using this information we find that run lengths for B and F states have an average size of $\frac{1}{a_{01}}$ and $\frac{1}{a_{10}}$ respectively. Therefore we have:

$$\hat{a}_{10} = \frac{1}{\mu_{\text{CPA}}} \quad \text{and} \quad \hat{a}_{01} = \frac{1}{\mu_b},$$

where μ_{CPA} is the average length of the sequences of consecutive packets arriving, and μ_b is the average length of the consecutive packets lost (CPL) after removing all isolated packet losses (CPL > 1).

The value $\hat{h}$ is estimated using the transition probability matrix A_2. We can find the steady-state probability π for all states by evaluating $\pi = \pi A_2$. After finding π we can define the exact probability for a packet to arrive to destination, $p_{arrival}$, using the following expression:

$$p_{arrival} = \hat{h} \cdot \pi_1 = \hat{h} \cdot \frac{\hat{a}_{01}}{\hat{a}_{01} + \hat{a}_{10}}. \tag{2}$$

Since we have already estimated values for a_{10} and a_{10}, and since $p_{arrival}$ can be found using the simulation results, we can obtain from Eq. (2) the value for $\hat{h}$.

Starting from the vector of estimated parameter values $\hat{v} = (\hat{a}_{10}, \hat{a}_{01}, \hat{h})$, we proceeded to find a more precise solution through an iterative process, which can be any of the many available in the literature [7]. We consider that our estimates $\hat{v}$ are close to the definitive ones, and so the method we use is a hybrid iterative/brute force technique. Starting from the estimated parameter values we select a search interval for each parameter testing several points in this interval and choosing the one that minimizes error function *f*. In the next iteration we reduce the search interval around the point that minimizes *f* in the previous iteration. We proceed with this algorithm until the output from function *f* is smaller than a pre-defined error value (ξ). This value defines the desired degree of accuracy of the model.

The minimization function used for the iterative process was:

$$f = \left|\frac{\mu'_{\text{CPA}} - \mu_{\text{CPA}}}{\mu_{\text{CPA}}}\right| + \left|\frac{\mu'_{\text{CPL}} - \mu_{\text{CPL}}}{\mu_{\text{CPL}}}\right|, \tag{3}$$

where μ_{CPA} and μ'_{CPA} refer to the mean values of the consecutive packet arrival distribution for the simulator and the model output respectively, and μ_{CPL} and μ'_{CPL} refer to the mean values of the consecutive packets loss distributions. We have chosen this function for minimization since it also allows to set bounds on the probability of packet arrivals. If we impose that $f < \xi$, and since $p_{arrival}$ can also be defined as:

$$p_{arrival} = \frac{\mu_{\text{CPA}}}{\mu_{\text{CPA}} + \mu_{\text{CPL}}}, \tag{4}$$

we find that the relative error for $p_{arrival}(e)$ is bounded by $\frac{1-\xi}{1+\xi} < e < \frac{1+\xi}{1-\xi}$. We consider *f* a good choice because similar values for $p_{arrival}$ obtained from the simulator and the model will allow us to perform consistent comparisons when evaluating multimedia applications. In fact, if we achieve similar distributions for CPL and CPA but do not achieve

Table 2
Estimated parameters values ($\hat{v}$) vs. the values obtained through the iterative process (v_i).

DSR	$\hat{v}$	v_i	OLSR	$\hat{v}$	v_i
P	10.786×10^{-3}	11.043×10^{-3}	P	5.440×10^{-3}	5.011×10^{-3}
S	1.357×10^{-3}	1.312×10^{-3}	S	2.868×10^{-3}	1.850×10^{-3}
$\bar{h}$	0.99560	0.99998	$\bar{h}$	0.96902	0.99904

very similar values of $p_{arrival}$, it would not be possible to validate the model against the simulator correctly. It would mean that different goodput values are achieved with the simulator and with the model, making any kind of comparison unfair.

Table 2 presents both the $\hat{v}$ values and the final values obtained through the iterative process (v_i). In addition, Table 3 presents a comparison, in terms of consecutive packets arriving (CPA) and consecutive packets lost (CPL), of the two-state model using either the values of vector $\hat{v}$ or vector v_i, taking the simulation results as reference.

Notice that, as shown in Table 2 the heuristic proposed to calculate $\hat{v}$ offers values that are very close to those of v_i. Yet, as shown in Table 3 we must further refine these values iteratively because very small differences in v provoke large variations in terms of burst behavior, which explains why the error varies so greatly.

Figs. 3 and 4 show a comparison of the consecutive packet arrivals patterns and the consecutive packet loss patterns respectively. Using vector v_i we compare the probability density function and the cumulative distribution function for the simulation and for the model outputs. From Fig. 3 we can observe that the statistical distribution provided by the model has a close resemblance with the simulator output.

Table 3
Statistical average matching for the estimated and iterated model values.

	Simulator	Model			
		$\hat{v}$	Error (%)	v_i	Error (%)
DSR					
μ_{CPA}	737.04	554.65	24.75	746.58	1.29
μ_{CPL}	86.91	71.53	17.70	88.99	2.39
OLSR					
μ_{CPA}	348.69	29.58	91.54	346.96	0.50
μ_{CPL}	129.99	17.49	86.55	129.92	0.05

Concerning the distribution of consecutive packet losses, Fig. 4 shows that the two-state model fails at accurately modeling the desired consecutive packets loss pattern for DSR. Concerning OLSR, we consider that the HMM is able to approximate the consecutive packet loss distribution satisfactorily.

The different precision of the results for the two routing protocols is due to their different routing nature. DSR belongs to the reactive family of protocols. These protocols are able to reestablish a path very quickly until there are no more available routes on the source node's cache. Afterward they have to proceed with the possibly high time-consuming process of route discovery until communication is resumed. Proactive protocols such as OLSR rely on frequent "Hello" and topology update messages to manage the routing tables. Therefore, these are not prone to present the asymmetry encountered with DSR, being more closely modeled with the two-states HMM presented before. Modeling more accurately DSR's distribution for consecutive packet losses can be done at the cost of introducing more complexity to the model. In the next section we show how this can be achieved using a three-state Markov model.

3.3. Three-states packet loss burst model

In this section we present an enhancement of the model described in the previous section which obtains a much

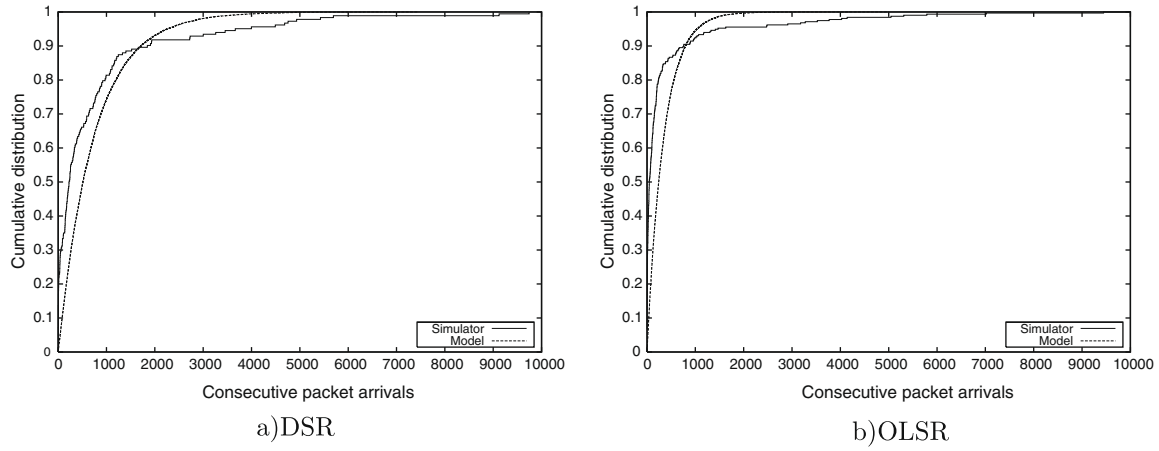

Fig. 3. Cumulative distribution function of consecutive packet arrivals (CPA) for DSR and OLSR.

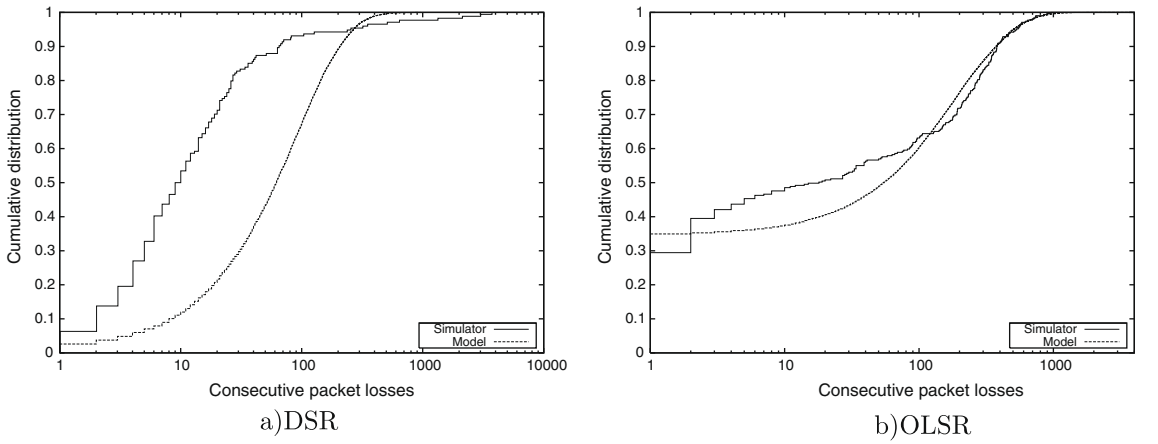

Fig. 4. Cumulative distribution function of consecutive lost packets (CPL) for DSR and OLSR.

better approximation of DSR's packet loss bursts distribution. Analyzing DSR's behavior we find that path breaks can be either short if breakage is handled by a quick rerouting process using the node's cache, or long if a route discovery process is required. Taking into account this different behavior, we replace state B from the two-states model with states L and R, where state L models short path breakages and state R models route discovery processes (R). The resulting three-states HMM is shown in Fig. 5.

As in the two-states model, packets arrive to destination in state F only, with probability $\bar{h}$. In states L and R all packets are lost. Mapping state L as 0, state F as 1, and state R as 2, we obtain the following transition probability matrix:

$$A_3 = \begin{bmatrix} a_{00} & a_{01} & 0 \\ a_{10} & a_{11} & a_{12} \\ 0 & a_{21} & a_{22} \end{bmatrix}. \quad (5)$$

Similarly to the previous section, we obtain a trace file *ST* with the mapping between packet sequence number and packet received/lost events. From this sequence of observations we obtain the distribution of consecutive packets lost and consecutive packets received. These distributions will be used as training sequences to find the optimum values for matrix A_3.

As in the previous section, we maintain $\bar{h} = 1 - \varepsilon \approx 1$. We classify consecutive packets lost (CPL) events into two groups by picking a threshold t. The value for t can be chosen by determining the inflection point of the cumulative distribution for consecutive packets lost, or using any other criteria. Notice that the optimum a_{ij} values obtained through iteration are independent of the threshold t, but better guesses for t allow finding such values with fewer iterations. In this example, the value we have chosen for t is 200 (see Fig. 6), which is slightly above the inflection point and corresponds to 4 s at a source rate of 50 pkt/s.

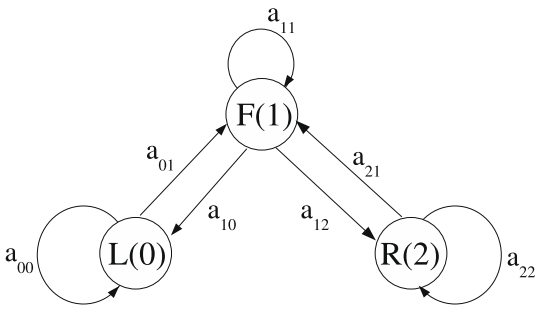

Fig. 5. Three-states Markov chain for the multi-hop wireless path model.

We now proceed to determine vector $\hat{v} = (\hat{a}_{10}, \hat{a}_{12}, \hat{a}_{01}, \hat{a}_{21}, \hat{h})$ with estimated values for the model. We consider that μ_{CPA} is the average length of the sequences of consecutive packets arriving (CPA), μ_b is the average length of the consecutive packets lost (CPL) when their length is greater than 1 and less or equal than $t-1$, and μ_B is the average length of the CPL when their length is equal to or greater than t. We can then calculate the values for $\hat{a}_{ij}$ using the following equations:

$$\hat{a}_{10} = \frac{1}{\mu_{\text{CPA}}} \cdot P(\text{CPL} < t \,|\, \text{CPL} > 1), \quad (6)$$

$$\hat{a}_{12} = \frac{1}{\mu_{\text{CPA}}} \cdot P(\text{CPL} \geqslant t \,|\, \text{CPL} > 1), \quad (7)$$

$$\hat{a}_{01} = \frac{1}{\mu_b} \quad \text{and} \quad \hat{a}_{21} = \frac{1}{\mu_B}. \quad (8)$$

After determining these values a_{00}, a_{11} and a_{22} are also defined, thus completely defining transition probability matrix A_3; hence, we may proceed to determine the steady-state probability for all states, π, obtaining:

$$p(F) = \pi_1 = \left(1 + \frac{\hat{a}_{10}}{\hat{a}_{01}} + \frac{\hat{a}_{12}}{\hat{a}_{21}}\right)^{-1}. \quad (9)$$

The expression $p_{arrival} = \hat{h} \cdot \pi_1$ gives us the exact probability that a packet arrives to destination, and it is used to calculate the value for $\hat{h}$, thus completely defining vector $\hat{v}$. We then find the final parameters values using the same methods exposed in the previous section. Table 4 presents both the vector of estimated values ($\hat{v}$) and the vector of values obtained through the iterative process (v_i).

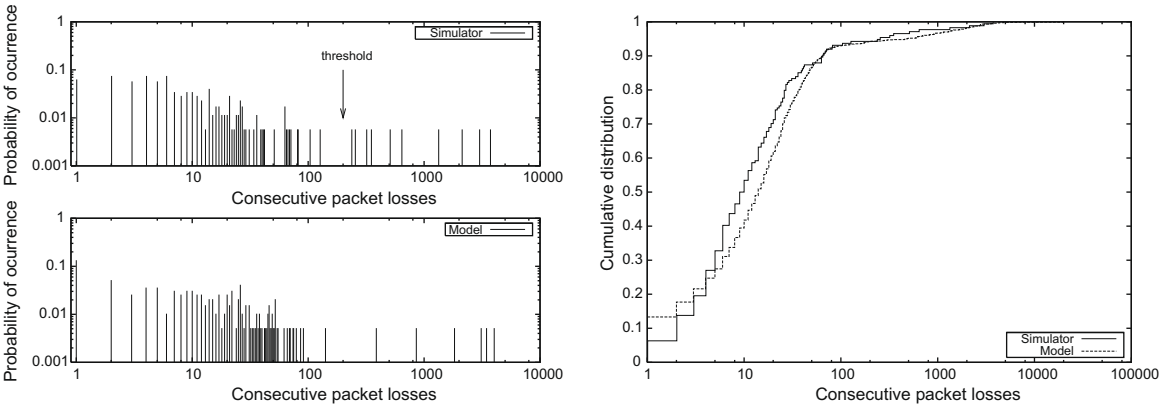

Fig. 6. Probability density function and cumulative distribution function for packet loss bursts.

Table 4
Estimated parameters values ($\hat{\nu}$) vs. the values obtained through the iterative process (ν_i).

	$\hat{\nu}$	ν_i
a_{10}	1.274×10^{-3}	1.173×10^{-3}
a_{12}	0.08324×10^{-3}	0.07669×10^{-3}
a_{01}	59.21×10^{-3}	59.21×10^{-3}
a_{21}	0.79821×10^{-3}	0.79820×10^{-3}
$\bar{h}$	0.99916	0.9999

Table 5
Statistical average matching for the estimated and iterated model values.

	Simulator	Model			
		$\hat{\nu}$	Error (%)	ν_i	Error (%)
μ_{CPL}	86.91	28.59	67.10	85.82	1.25
μ_{CPA}	737.04	268.15	63.62	737.12	0.01

Table 5 presents a comparison of the error values when comparing the estimated ($\hat{\nu}$) and iterated (ν_i) vectors for the three-state model. The comparison is made in terms of consecutive packets arriving (CPA) and consecutive packets lost (CPL). We observe that now the probability density function and cumulative distribution function obtained with the model fit the desired distribution with a much higher degree of accuracy, as shown in Fig. 6. It is evident that the introduction of two loss states instead of one improves the accuracy of the model's cumulative distribution function. We also observe from the probability density function that our model can reproduce very large bursts.

Concerning the consecutive packet arrivals distribution shown in Fig. 7, both density and distribution functions are very similar to the ones obtained with the two-states model, as expected. Though the model could be further extended in order to achieve small values of consecutive packet arrivals, thus offering a better fit to the cumulative distribution curve of the simulator, we consider that it is an irrelevant issue to our purposes.

In the sections that follow, all the results related with the models proposed in this section are obtained using the optimum solution (ν_i), that is, the values for the different model parameters are refined through the iterative process.

4. Novel metrics for model accuracy testing

In this section we validate the models proposed in Sections 3.2 and 3.3, verifying their correctness and adequateness for the purpose of evaluating multimedia streaming applications.

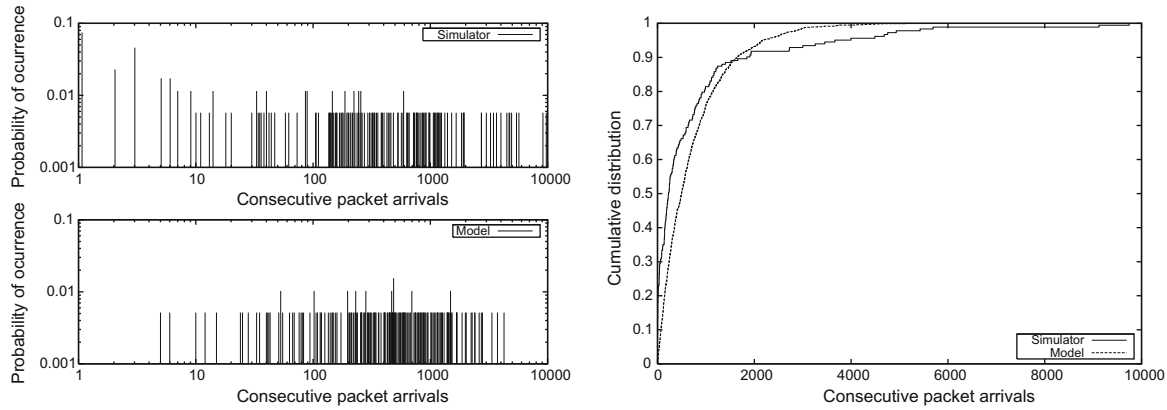

Fig. 7. Probability density function and cumulative distribution function for consecutive packet arrivals.

With that purpose we define a set of metrics that measure packet loss bursts. Then, we use these metric to compare both the simulator and model outputs in order to verify the effectiveness of the models proposed.

4.1. Metrics for packet loss burst measuring

Before detailing the different metrics proposed to characterize packet loss bursts, we provide a definition of the boundaries of a packet loss burst specifically designed for video and audio data flows. We consider that data flows belonging to different applications will not be affected by packet loss bursts in the same way. It is also important to point out that loss burst measurements are always done focusing on a single traffic flow, and not for all the traffic in the network simultaneously, even if there are other similar flows.

To delimit a burst we follow an approach based on hysteresis where we transit into the burst state whenever n_s consecutive packets are lost. Similarly, we transit out of the burst state whenever n_e consecutive packets arrive successfully. The values of both n_s and n_e shall depend on the type of information sent and the packetization granularity. For example, if we consider that at least one entire video frame has to be lost for a burst to be meaningful and that one entire frame has to arrive for communication to be resumed, n_s and n_e will have the same value, and it will be equal to the number of packets per frame defined in the video codec.

We will now proceed with the definition of some indicators to describe packet loss burst occurrences. The most simple indicator is *burst percentage* (PBP), defined as:

$$\mathrm{PBP}(\%) = \frac{\sum_{i=1}^{K} B_i}{N}, \tag{10}$$

where B_i is the size of loss burst i in number of packets, K is the total number of loss bursts and N the total number of packets sent. The PBP gives a measure of the relative burst incidence. To measure the relative impact of bursts over the total number of packets lost L, we define the *Relative Burstiness* (RB) metric as:

$$\mathrm{RB} = \frac{\sum_{i=1}^{K} B_i}{L}, \quad 0 \leqslant RB \leqslant 1. \tag{11}$$

In a situation where most packets are lost in a random manner the RB parameter approaches 0, while when packet loss bursts dominate, RB will be greater than 0.5.

This parameter allows us to detect where the network needs more improvements: if on the routing protocol side ($RB > 0.5$) or on the MAC support for traffic flows ($RB < 0.5$).

Both these indicators are burst size independent. They penalize equally very small bursts occurring in a distributed fashion and very large bursts, as long as the total number of packets lost is the same. From the user's point of view, however, long communication breaks may be unacceptable. To take into account such discrepancies, we introduce the *Burstiness Factor* (BF):

$$\mathrm{BF} = \frac{\sqrt{\sum_{i=1}^{K} B_i^2}}{N}, \quad 0 \leqslant \mathrm{BF} \leqslant 1. \tag{12}$$

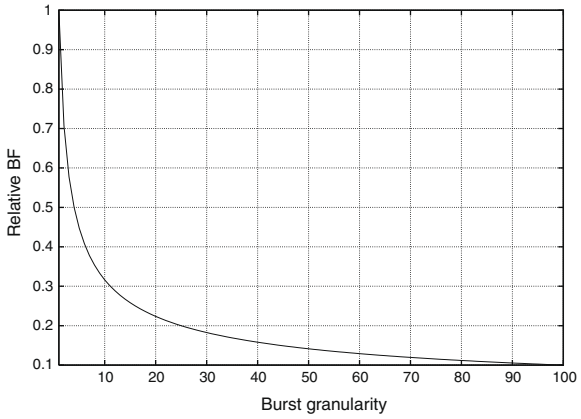

Fig. 8. Normalized BF variation with increasing burst granularity.

In Fig. 8, we show how BF decays when increasing *burst granularity*. The burst granularity indicates that the original (single) burst was split into g smaller independent bursts. The BF is a metric of the impact of the re-routing time of different routing protocols over a given flow; smaller values indicate that interruptions caused by routing protocols are either fewer or smaller.

Though BF is a good indicator to measure improvements on routing protocols, it does not take into account the relative position of the bursts, which can have different impact on multimedia streams from a codec point of view. We therefore introduce the *media smoothness factor* (MSF):

$$\mathrm{MSF} = \frac{\sqrt{\sum_{i=1}^{T} F_i^2}}{N}, \quad 0 \leqslant \mathrm{MSF} \leqslant 1, \tag{13}$$

where T is the total number of *inter-bursts* or *burst delimited* periods, identified as F_i, and N is the total number of packets. Fig. 9 shows an example of plotting the values of F_i and B_i. In this example n_s and n_e are set to 3 packets, thus obtaining $K = 2$ and $T = 3$. Applying this threshold we have two well defined loss bursts, shown on the left side of the figure; as it can be seen, these are complementary.

The MSF measures the *fluidity* experienced by a multimedia data stream; obviously, MSF $\gg$ BF must hold for communication to be sustainable. To better illustrate the different properties of BF and MSF, we propose a case study scenario, depicted in Fig. 10, where we have a train of K bursts of length G, separated by exactly X packets. The burst sequence is centered so that Y packets separate the first and last bursts from the beginning and end of the observation period, where $Y = (N - K \cdot G - (K-1) \cdot X)/2$.

In this scenario $\mathrm{BF} = \sqrt{K} \times G/N$, which is independent from the bursts separation value (X), while:

$$\mathrm{MSF} = \frac{\sqrt{(K-1) \cdot X^2 + 2 \cdot Y^2}}{N}, \tag{14}$$

which depends not only on the size and number of bursts, but also on the distance between them. Considering that the upper limit for X when $Y = 0$ is given by:

$$X_{max} = \frac{N - K \cdot G}{K - 1}, \tag{15}$$

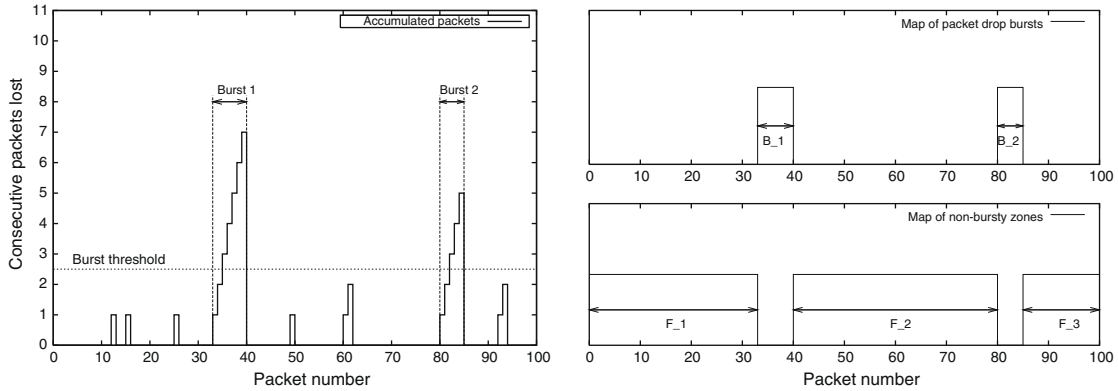

Fig. 9. Example of plotting the values of F_i and B_i.

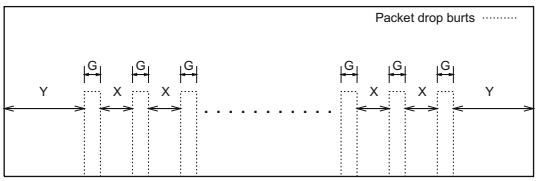

Fig. 10. The BF vs. MSF case study scenario.

we can normalize Eq. (14) using $z = X/X_{max}$, obtaining:

$$\mathrm{MSF} = \frac{N - K \cdot G}{N} \cdot \sqrt{\frac{z^2}{K-1} + \frac{1}{2} \cdot (1-z)^2} \tag{16}$$

Fig. 11 shows the behavior of Eq. (16) as a function of K, taking $\frac{G}{N} = 0.02$.

Eq. (14) reaches its minimum when $x_m = y_m = \frac{N-K \cdot G}{K+1}$. This indicates that the minimum value of MSF is reached when interruptions on communication are evenly separated, that is, when distance between loss bursts is equal to the distance to the extremes.

The normalized expression for z_{min} is:

$$z_{min} = \frac{x_m}{X_{max}} = \frac{K-1}{K+1}, \quad K \geqslant 2, \tag{17}$$

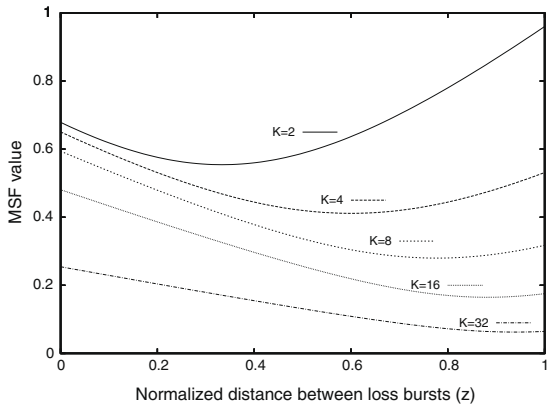

Fig. 11. MSF variation with distance between bursts.

which depends solely on the number of loss bursts present on the sequence.

We can directly check the results from Fig. 11, and also check that it approaches 1 for large values of K. Since typically we will have a large number of gaps ($K \gg 1$), the MSF will be monotonically decreasing. This result allows us to conclude that MSF offers a measure of burst concentration for similar values of BF, increasing as the concentration of bursts increases.

In summary, we have defined four metrics for analyzing packet loss bursts: the PBP (burst percentage), the RB (relative burstiness), the BF (burstiness factor), and the MSF (media smoothness factor). These metrics give us different information about loss burst patterns, and they will help us in the model validation process.

4.2. Model accuracy testing

We now apply the previously defined metrics to compare the two-state and three-state HMM results with the simulator's output when using the DSR protocol. We set n_s equal to n_e for the sake of simplicity in the presentation of results.

The bursts percentage over the total number of packets sent (PBP) and the RB parameter vary with increasing thresholds for burst start/end values, see Fig. 12. The RB parameter represents clearly the relation between packet losses that pertain to bursts and those that do not. As it can be seen, the three-states model approaches the reference curve from the simulator with greater accuracy than the two-states model.

In Fig. 13 we compare the output from the HMMs and the simulator in terms of the Burstiness Factor (BF) and the Media Smoothness Factor (MSF). We observe that the results for the three-states HMM are much closer to the reference values. We consider that the degree of accuracy achieved is acceptable for applications such as video codec enhancing and tuning. In terms of the MSF, which takes into account consecutive packet arrivals instead, we observe that the three-states HMM approaches the reference MSF value with increasing thresholds. The slight difference is expected since the accuracy of the consecutive packet arrivals distribution was not the main focus of ours model.

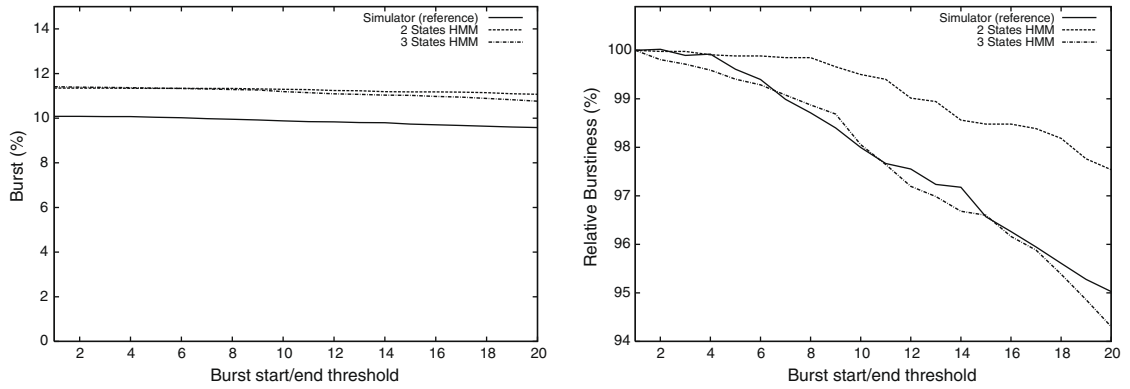

Fig. 12. PBP (left) and Relative Burstiness (right) comparison.

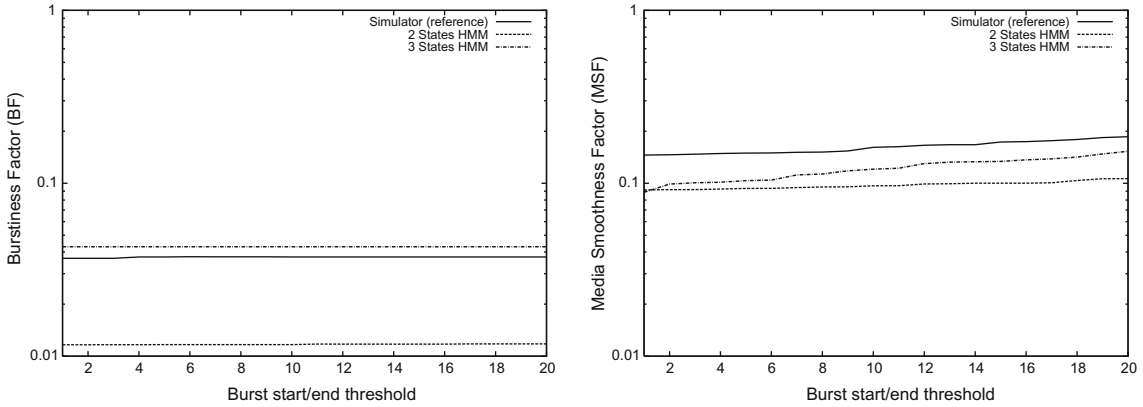

Fig. 13. BF (left) and MSF (right) comparison.

It could be improved by increasing the number of states in the HMM, similarly to was done for DSR's consecutive packet losses distribution.

5. Benefits of the proposed models

In this section, to illustrate the applicability of our models, we use them as a tool to speed up the evaluation and tuning of a video codec (in this example, an H.264 video codec). A video codec is one of the most representative applications for our model since, when optimizing a video codec for a certain network environment, we must test several options related to video resilience, error recovery, packetization, etc. to maximize performance. Hence, instead of having to run several simulations for each parameter combination being tested (which would take a lot of time), we propose using the model developed.

We measure the impact of the different steps required for simulation and data extraction using either the ns-2 simulator or HMMs for both DSR and OLSR. We simulate the streaming of a typical movie 1 h and 30 min long in a MANET scenario. Our setup is similar to that of Section 3, consisting of a 1000 m $\times$ 1000 m scenario with 80 nodes. The node mobility is generated using the random waypoint model with node speed between 0 and 12 m/s. The wireless interfaces are based on the IEEE 802.11b standard with radio range limited to 250 m, and background traffic consists of 4 UDP sources generating 512 bytes packets at a rate of 4 pkt/s.

Results presented in Table 6a and b allow comparing the time consumed at each step using the ns-2 simulator only or using the models. The values presented are achieved on a dual 2.6 GHz Pentium-IV server with 2 Gbytes of RAM running GNU/Linux version 2.6.22.

Table 6
Duration of different simulation steps using (a) ns-2 simulator and (b) HMMs.

	DSR	OLSR
(a)		
Mobility generation time (s)	840	840
Single simulation time (s)	1320	9720
Extraction of packet loss details (s)	60	60
100 simulations total time	61 h 40 m	295 h
(b)		
Mobility generation (s) – once	840	840
Single simulation time (s) – once	1320	9720
Extraction of packet loss details (s) – once	60	60
Determination of model parameters (s) – once	3600	3600
Single simulation time using model (s)	0.40	0.39
100 simulations total time	1 h 38 m	3 h 58 m

Table 6a shows that simulating with OLSR takes much longer than with DSR. We can also see that mobility generation takes a considerably long time.

On Table 6b we show that almost all the time is consumed in simulation and in the determination of model parameters. Once that is done, though, the execution of the model is very fast. Relatively to the entry named "Determination of model parameters", we wish to point out that this time takes into account not only the time to determine the initial estimates for the different parameters ($\hat{v}$), but also to find the final iterated values (v_i). In the bottom of both tables we present the estimated time to run 100 simulations, a value required to extract statistically significant results.

Relatively to the improvements achieved, we find out that our algorithm allows executing the same set of 100 simulations up to 38 times faster with DSR, and up to 74 times faster with OLSR. In fact, we find that applying our model we obtain time efficiency if we wish to run more than 3 simulations when using DSR, or more than 2 simulations when using OLSR. Such low values justify the need for models when the desired number of simulation runs is higher, which typically happens when strict confidence intervals for a certain parameter are desired. We should point that the desired confidence intervals are for codec level performance metrics, and not simulation metrics.

In terms of trace file output, we can see (Table 7) that, comparing trace file sizes, the model's output is 300–12,000 times smaller than the simulator's output, though the output from the last can be reduced. Concerning real-life experiments, the trace file size can be reduced to the size of the HMM's trace file.

Optimal tuning of the video codec using both the simulator's output and the HMM's output was also performed. We find that the most error-resilient parameter choices for the codec are the same with both solutions, which allows us to conclude that the methods and techniques exposed in this paper fit our purpose adequately.

6. Validation through large scale experimentation

In this section we will compare the results obtained from a large set of experiments in MANETs to those offered by the model. The purpose is not only to check if our Hidden Markov Model consistently models losses and the respective loss patterns, but also of analyzing if the probability of being in each of the model's states is related to the events they represent (e.g., Route Discovery, Route Error).

To accomplish this we devise a MANET scenario where 50 nodes are moving in an area sized 870×870 meters. Node mobility is based on the random way-point model, and speed is fixed at 4 m/s. The routing protocol used is DSR, and we use the 3-state model developed since it offers the best accuracy in terms of loss pattern modeling. Our routing protocol choice allows us to stress the model to the limit, forcing it to replicate very asymmetric burst conditions such as those generated by DSR, something that did not occur with OLSR, as we have shown in Section 3.2. This asymmetry in DSR is mainly due the existence of both small loss bursts – caused by path loss and quick re-routing using cached routes – and more sporadic losses related to route discovery, which conform significantly longer bursts. This is in contrast with OLSR, were we typically have very large loss bursts whenever routes are updated (only a single route updating strategy is available). In fact, the high latency associated with OLSR re-routing tasks is one of the main factors hindering performance when supporting real-time multimedia applications, as we show in [16].

Table 7
Trace file sizes using ns-2 and the proposed models.

	DSR	OLSR
ns-2 output (Mb)	149	5700
HMM output (Mb)	0.5	0.5

We used IEEE 802.11g/e enabled interfaces in all the experiments performed. Stations transmit at the maximum rate of 54 Mbit/s up to a range of 250 m and, moreover, benefit from QoS as offered by annex E of the IEEE 802.11 standard [18]. Concerning traffic, we have six sources of background traffic transmitting FTP/TCP traffic in the Best Effort MAC Access Category throughout the entire simulation time.

The total simulation time is of 450 s, and results are extracted from a 300 s period where all traffic sources are active. All the values related to simulation in the graphs shown below are obtained from 10 distinct experiments under the same conditions (only the scenario differs). Concerning the model, it was tuned using the aggregated output of the same set of 10 simulation experiments.

6.1. Congestion modeling

In our tests we start by modeling losses due to increasing degrees of congestion as experienced by an H.264 video flow transmitted with real-time constraints. With this purpose we obtained real traces of a H.264 video encoded at a data rate of 1 Mbit/s. Then, we increase the number of simultaneous H.264 connections in the MANET to increase congestion, using one of them as reference for measurement purposes. Both the reference and the additional H.264 flows are mapped to the Video MAC Access Category, almost eliminating completely possible congestion losses that typically take place in MANETs when competing with TCP traffic.

In Fig. 14 (left) we present confidence intervals for the packet loss results obtained from simulation, along with the results obtained from the model. As can be seen, the model is able to reproduce similar packet loss values; the same holds for video PSNR values (see right picture of Fig. 14).

Since we are more interested in packet loss patterns than the packet loss rate itself, in Fig. 15 we show the cumulative histogram for bursty losses obtained via simulation and with our model for comparison. We see that there is a resemblance between both, especially in the region of interest (large burst sizes).

With respect to the relationship between routing events and the probability of being in each of the model's states, Fig. 16 shows that DSR's route discovery and route loss events have a strong relationship with the model's states

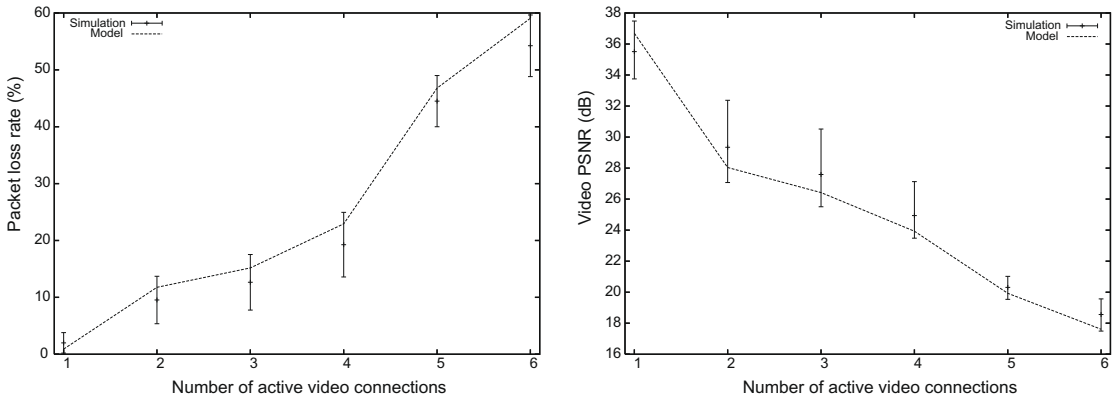

Fig. 14. Packet loss rate accuracy (left) and video PSNR accuracy (right) at different degrees of congestion.

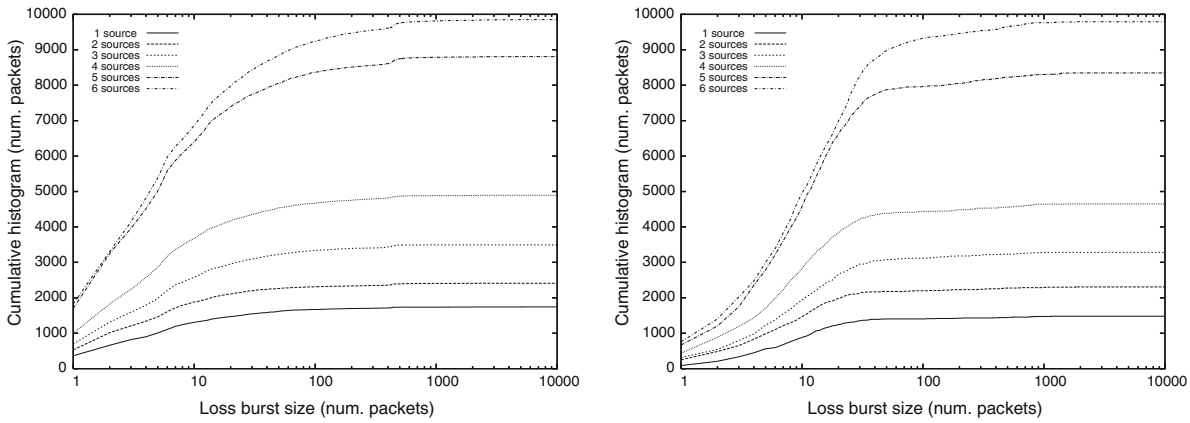

Fig. 15. Cumulative histogram for loss bursts obtained from simulation (left) and with our model (right) at different degrees of congestion.

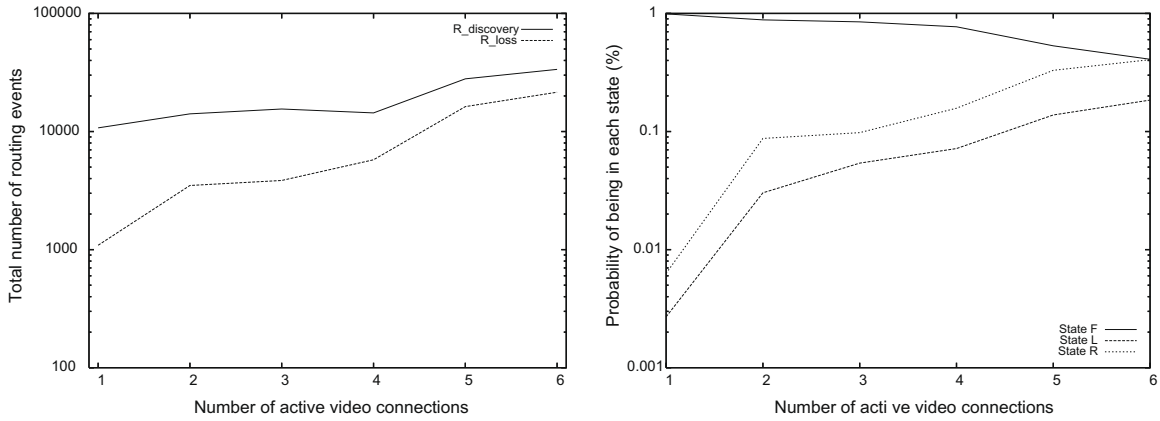

Fig. 16. DSR's route discovery/link loss events (left) and probability of being in each of the model's states (right) at different degrees of congestion.

representing them (states R and L, respectively). Since in this case video losses occur due to both mobility and congestion, there is not a perfect match between simulation events and model states. In the next section, where the congestion is minimal, we show that the relationship is much more evident.

6.2. Mobility modeling

We now proceed by modeling losses due to increasing degrees of mobility. In our tests we increase node speed from 1 m/s up to 10 m/s. In terms of QoS traffic, we have a single H.264 video stream with real-time constraints as

reference. Since the congestion encountered by the video flow is minimal, mobility alone is responsible for most of the losses encountered.

In Fig. 17 (left) we show confidence intervals for the packet loss rate obtained via simulation, along with the values obtained through the Hidden Markov model proposed. As occurred for variable congestion, the model offers an acceptable degree of accuracy, and the same is true in terms of video PSNR (see right picture in Fig. 17).

Concerning the packet loss patterns obtained from the model, they closely resemble the ones obtained from simulation at all speeds (see Fig. 18), demonstrating the effectiveness of the proposed technique.

Regarding the relationship between DSR's events (route discovery and link loss) and the model's states (R and L), and keeping in mind that results are not directly comparable, Fig. 19 shows that there is a very high degree of similitude among the curve shapes of related parameters (that

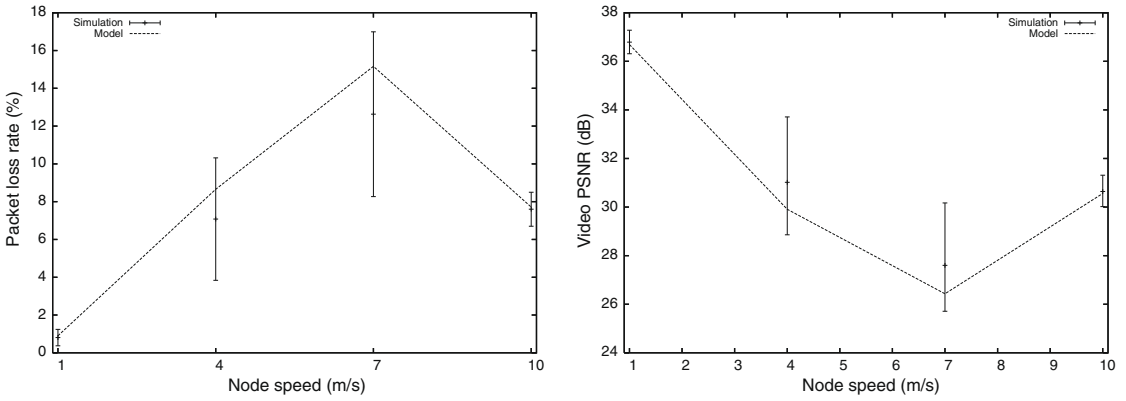

Fig. 17. Packet loss rate accuracy (left) and video PSNR accuracy (right) at different degrees of mobility.

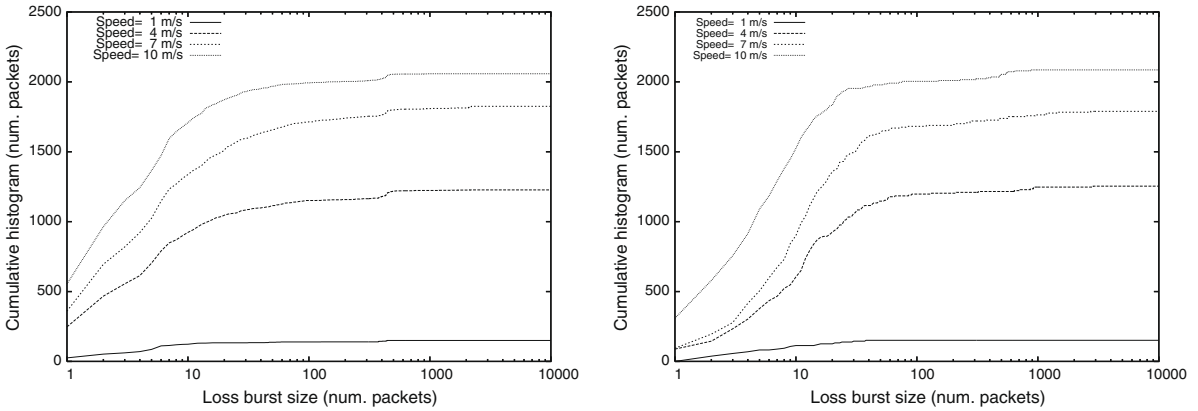

Fig. 18. Cumulative histogram for loss bursts obtained from simulation (left) and with our model (right) at different degrees of congestion.

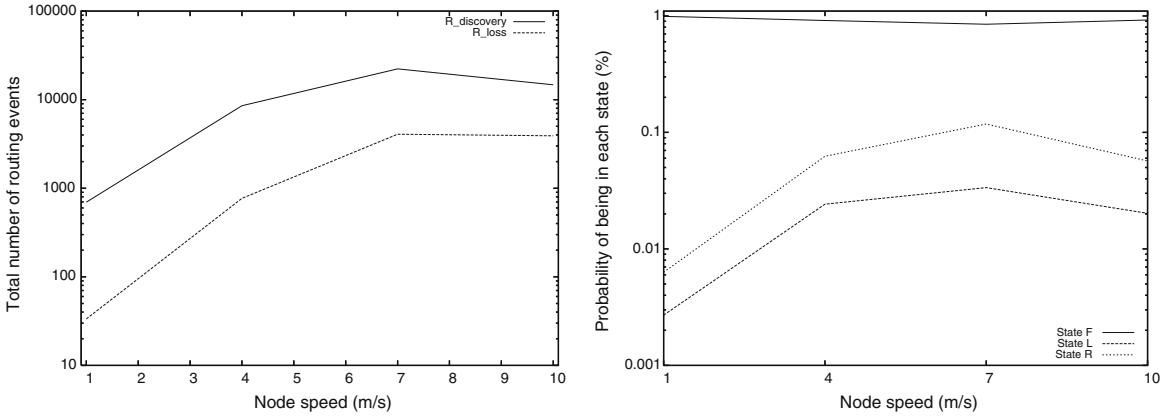

Fig. 19. DSR's route discovery/link loss events (left) and probability of being in each of the model's states (right) at different degrees of congestion.

is, route discovery and state R, route loss and state F). Notice that, despite the input of our model lacks any information about routing data or routing states (only arrival and loss patterns are used), MANET environments are characterized by an intrinsic relationship between routing and bursty losses, which explains the success of the proposed model in representing both.

7. Conclusions

Modeling end-to-end loss patterns in mobile ad hoc networks in an important challenge due to the large loss bursts that are prone to occur in these networks.

Developing and validating novel application-layer protocols and applications for these networks is an effort that consumes too much time and too many CPU/disk resources. To solve this problem, in this work we developed a Hidden Markov representation of end-to-end path behavior in MANETs that is quite effective in modeling bursty behavior. The proposed technique allows to evaluate the effects of packet loss and arrival patterns when sending a compressed audio/video stream using different routing protocols. We proposed a simpler solution that uses a two-state HMM, and a more elaborate solution with a three-state HMM. For both solutions we presented a heuristic that allows estimating the model parameters with a good degree of accuracy, greatly reducing the number of iterations necessary to find accurate parameter values.

We validated our models showing that the proposed HMMs provide similar results in terms of the loss burst metrics we defined. In terms of benefits, we compared time and resource consumption when using either a simulator or our model, showing dramatic improvements when the number of simulation runs is moderate/high.

Finally, we have employed our model in the scope of a large scale MANET test environment where the purpose is to confirm model compliance at the user level. Experimental results have showed that the model's output in terms of both packet losses and video Peak Signal-to-Noise Ratio (PSNR) are within the confidence intervals obtained from simulation. In terms of loss patterns we confirmed that the model's losses, similarly to what occurs in simulation, are essentially due to large loss bursts. We also pointed out the strong relationship between actual routing events and the probability of being in each of the burst-related states in the HMM.

Overall, we verified that the model is able to retain its effectiveness for a wide range of loss burst patterns. Hence, we consider that our proposal is an adequate alternative to the developers of multimedia streaming applications for MANETs, showing excellent results in terms of both accuracy achieved and speedup, and avoiding repetitive, time-consuming simulations.

References

[1] IETF, MANET Working Group Charter. <http://www.ietf.org/html.charters/manet-charter.html>.

[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] Charles E. Perkins, Elizabeth M. Belding-Royer, Samir R. Das. Ad hoc on-demand distance vector (AODV) routing. Request for Comments 3561, MANET Working Group, July 2003. Available from: <http://www.ietf.org/rfc/rfc3561.txt>.

[4] David B. Johnson, Y. Hu, David A. Maltz, The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4. Request for Comments: 4728, MANET Working Group, February 2007. Available from: <http://www.ietf.org/rfc/rfc4728.txt>.

[5] T. Clausen, P. Jacquet, Optimized link state routing protocol (OLSR). Request for Comments 3626, MANET Working Group, October 2003. Available from: <http://www.ietf.org/rfc/rfc3626.txt>.

[6] I. Chakeres, C. Perkins, Dynamic MANET On-demand (DYMO) Routing, Internet Draft, MANET Working Group, draft-ietf-manet-dymo-17.txt, September 2009, in preparation.

[7] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (1989) 257–286.

[8] R. Koodli, R. Ravikanth, One-way loss pattern sample metrics, IETF RFC 3357 (August) (2002).

[9] Wei Wei, Bing Wang, Don Towsley, Jim Kurose, Model-based identification of dominant congested links, in: Internet Measurement Workshop Proceedings of the Conference on Internet Measurement (IMC'03), Miami Beach, Florida, October 2003.

[10] J. Liu, I. Matta, M. Crovella, End-to-end inference of loss nature in a hybrid wired/wireless environment, in: Proceedings of Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'03), Sophia-Antipolis, France, March 2003.

[11] W. Jiang, H. Schulzrinne, Modeling of packet loss and delay and their effect on real-time multimedia service quality, In: Proc. NOSSDAV, Chapel Hill, North Carolina, USA, June 2000.

[12] H. Sanneck, G. Carle, R. Koodli, A framework model for packet loss metrics based on loss runlengths, in: Proceedings of the SPIE/ACM SIGMM Multimedia Computing and Networking Conference 2000 (MMCN 2000), San Jose, CA, January 2000, pp. 177–187.

[13] T. Lin, S.F. Midkiff, Mobility versus link stability in simulation of mobile ad hoc networks, in: Proceedings of Communication Networks and Distributed Systems Modeling and Simulation Conference, Orlando, FL, USA, January 2003, pp. 3–8.

[14] Weiling Zhu, J. Garcia-Frias, A new hidden Markov model for very bursty channels, in: 57th IEEE Semiannual Vehicular Technology Conference, 2003, pp. 524–528.

[15] S. Srinivas, K. Shanmugan, Characterization of bursty channels using Markov models, in: IEEE International Conference on Communications, 1993, pp. 1615–1619.

[16] Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni, Performance of H.264 compressed video streams over 802.11b based MANETs, in: International Conference on Distributed Computing Systems Workshops (ICDCSW'04), Hachioji – Tokyo, Japan, March 2004.

[17] K. Fall, K. Varadhan, Ns notes and documents, The VINT Project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000.

[18] 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.

Carlos T. Calafate graduated with honors in Electrical and Computer Engineering at the University of Oporto (Portugal) in 2001. He received his Ph.D. degree in Computer Engineering from the Technical University of Valencia in 2006, where he works as an assistant professor since 2005. He is a member of the Computer Networks Group (GRC). His research interests include mobile and pervasive computing, security and QoS on wireless networks, as well as video coding and streaming.

Pietro Manzoni is an associate professor of computer science at the Technical University of Valencia (SPAIN). He received the M.S. degree in Computer Science from the "Università degli Studi" of Milan (ITALY) in 1989 and the Ph.D. in Computer Science from the Polytechnic University of Milan (Italy) in 1995. His research activity is related to wireless networks protocol design, modeling and implementation.

Juan-Carlos Cano is an associate professor in the Department of Computer Engineering at the Technical University of Valencia (UPV) in Spain. He received the MSc and a Ph.D. degrees in Computer Science from the UPV in 1994 and 2002, respectively. Between 1995 and 1997 he worked as a programming analyst at IBM's manufacturing division in Valencia. His current research interests include power aware routing protocols for mobile ad hoc networks and pervasive computing.

Manuel Perez Malumbres received his B.S. in Computer Science from the University of Oviedo (Spain) in 1986. In 1989 he joined to the Computer Engineering Departament (DISCA) at Technical University of Valencia (UPV), Spain, as an assistant professor. Then, he received the M.S. and Ph.D. degrees in Computer Science from UPV, at 1991 and 1996 respectively. In september 2005, he moved to Miguel Hernandez University holding the same position. His research and teaching activities are related to multimedia networking (audio/video coding and network delivery) and wireless technologies (MANETs and sensor networks).