

Dynamic cross-layer framework to provide QoS for video streaming services over ad hoc networks *

V. Carrascal
Telematic Engineering Dpt.
Polytechnic University of
Catalonia
Barcelona, Spain
vcarrascal@entel.upc.edu

A. Zavala
Telematic Engineering Dpt.
Polytechnic University of
Catalonia
Barcelona, Spain
azavala@entel.upc.edu

G. Díaz
University of Queretaro,
Mexico
Polytechnic University of
Catalonia, Spain
gdiaz@entel.upc.edu

M. Aguilar
Telematic Engineering Dpt.
Polytechnic University of
Catalonia
Barcelona, Spain
maguilar@entel.upc.edu

ABSTRACT

In recent years the growing proliferation of small wireless devices able to maintain wireless communications using IEEE 802.11 technologies has enabled the deployment of MANETs (Mobile Ad Hoc Networks). This fact has stimulated the demand of multimedia services over this type of networks, such as video-streaming services. Providing QoS in this environment is a challenging task due to the inherent dynamic characteristics of these networks. In this paper we propose a-MMDSR (adaptable-Multipath Multimedia Dynamic Source Routing), a dynamic framework for video streaming services which improves the user perceived quality. We have perfected our previous approach s-MMDSR (static-MMDSR), a multipath routing protocol based on DSR which works simultaneously with a cross-layer algorithm. Through simulations, we demonstrate that the performance of our proposal framework outperforms DSR and s-MMDSR.

Keywords

MANETs, Dynamic and Adaptable framework, Multipath QoS-aware Routing Protocol, Cross-layer design, End-to-end QoS provision.

*Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Qshine08 July 28-31, 2008, Hong Kong, China.

1. INTRODUCTION

A Mobile Ad Hoc NETWORK (MANET) is a communication network established spontaneously by a set of wireless mobile devices that are capable of communicating with each other in a peer-to-peer basis. MANETs are self-organized networks, without the need of any fixed network infrastructure or centralized administrative support. Since the transmission range of these wireless network devices is limited, intermediate hosts may be needed to transfer data across the network. Mobile nodes have short-life batteries and move freely, which may produce frequent changes in network topology. All these features have to be taken into account in the design of a proper framework for MANETs.

Lately, the growing proliferation of small wireless devices able to maintain wireless communications using IEEE 802.11 technologies has enabled the deployment of MANETs. This situation has made possible the development of multimedia services to be deployed over these networks, such as video-streaming services. Multimedia applications have specific requirements in terms of throughput guarantees, bounded delay and losses, etc. Thus, new techniques are needed to provide end-to-end QoS over MANETs, since the well-known techniques used in wired networks can not be applied directly. In this paper we propose a-MMDSR (adaptable-Multipath Multimedia Dynamic Source Routing), a dynamic framework for video-streaming services based on a cross-layer algorithm which aims at providing end-to-end QoS to the user. We have introduced dynamism in the computation of a set of design parameters as well as in the period of the routing algorithm. These design parameters are dynamically computed depending on the current state of the network. Our framework provides QoS to multiple sources of video transmitting simultaneously over the same ad hoc network, including situations with high traffic loads. Also, multipath routing techniques have been included to increase robustness of the ongoing communications as well as to balance the load. In addition, our framework treats packets with different priorities depending on their relevance within the video flow.

The remainder of this paper is organized as follows. Section 2 presents the general framework where our proposal lays. Section 3 describes the MMDSR protocol. The dynamism features of our proposal are presented in section 4. Simulation results to evaluate the benefits of the proposed scheme are shown in section 5. Finally, section 6 concludes the paper and discusses future work.

2. PROPOSED FRAMEWORK

Multimedia applications over MANETs have QoS constraints which need to be fulfilled. The particular features of MANETs (i.e. mobility, lack of fixed infrastructure, dynamic network topology, energy constraints) require proper techniques to be applied. Some protocols have been designed for ad hoc networks, focusing on different characteristics of the network, such as mobility [1]. Multipath techniques enable load balancing mechanisms, increase robustness in the transmissions and decrease the end-to-end delay. Different multipath schemes have been proposed in the literature [2, 3, 4, 5].

We proved in a previous work [6] that it is not worthwhile to arrange more than three paths simultaneously in a multipath scheme, due to excessive overhead increase and little improvement. We use a MPEG-2 codification for the video flow. The most important frames of the coded video flow (I frames) are sent through the best path available. P frames are sent through the second best path and B frames through the third. Also, a four-priorities queuing system in the access of the MAC level is used, so that the more relevant frames in the video decodification scheme have higher priority than both the best effort data and less relevant frames. These are the four queues (Q0 to Q3) and the type of traffic served in each one:

- Q0: MMDSR signaling.
- Q1: high priority packets (I frames).
- Q2: medium priority packets (P frames).
- Q3: low priority packets (B frames + BE).

3. MMDSR PROTOCOL

In this section we describe the elements of our proposal: the routing protocol MMDSR, several algorithms, special packets and parameters that have been designed. We have started from the plain DSR [7] as the routing engine, since it is suitable to be easily extended for multipath operation. As in other multimedia applications, there are the customer requirements established by means of a Service Level Agreement (SLA). Such SLA specifies network QoS parameters and their necessary values to provide the committed image quality. These QoS parameters are the minimum expected bandwidth (BW_{min}), the maximum percentage of data losses (p_{max}), the maximum delay (d_{max}) and the maximum delay jitter (j_{max}):

$$customer_req \equiv \{BW_{min}, p_{max}, d_{max}, j_{max}\} \quad (1)$$

MMDSR has been designed to support multiple sources of video in the same ad hoc network. All the decisions (i.e. path selection) and operations (e.g. tuning of configuration parameters) are managed from the source and they depend on the state of the network, thus the framework operation is dynamic and adapts to the environment. Video is distributed using RTP/RTCP (Real Time Protocol/Real Time Control Protocol) over UDP as the transport protocols.

3.1 Operation

Assuming that a video-streaming communication takes place between a pair of nodes, the proposed system works basically from the source node. MMDSR periodically discovers the available paths between source and destination nodes using the DSR engine. Sending periodic monitoring packets MMDSR perceives the global state of the available paths between source and destination. We propose to use two new QoS parameters created to achieve this goal: *Reliability Metric* and *Mobility Metric*.

Previous to starting the video transmission, a *Probe Message* (PM) packet is sent from source to destination through each one of the D paths discovered by the DSR routing engine. Then, at destination a reply packet is generated to carry the collected information regarding the quality of the available paths. This information will be analyzed at the source. A score is assigned to each one of the paths and they are classified accordingly. After that, the source selects as many paths as needed by the multipath scheme. The parameters computed for each one of the available paths are collected in a vector, *path-state*:

$$path - state_k^i \equiv \{BWe, p, d, j, h, RM, MM\}_k^i \quad (2)$$

where i is the number of iteration of the algorithm and k refers to each one of the K selected paths (with $K \leq D$). These parameters are the *end-to-end* available bandwidth BWe_k^i , the percentage of losses p_k^i , the delay d_k^i , the delay jitter j_k^i , the hop distance h_k^i and the two new proposed QoS parameters designed specially for ad hoc networks: Reliability Metric RM_k^i and Mobility Metric MM_k^i .

This process is repeated over time with a certain frequency in order to refresh the paths, as the dynamic nature of MANETs may produce frequent link breakages and thus the topology can vary throughout time. In previous versions of our protocol [6] this period of operation was fixed to 10 seconds, as it was shown to be a proper value from numerous simulations performed for a typical scenario, as the ones presented in section 5. In the updated version we present in this article, the period of operation of our protocol changes dynamically depending on the state of the network, as we describe in section 4. In the next subsections we describe the control packets used in MMDSR, each one of the parameters present in the path-state and how they are computed.

3.2 Control packets

3.2.1 Probe Messages

At the beginning of each iteration of the algorithm, the source sends a PM packet through each one of the D discovered paths. Each one of the intermediate nodes of each path updates the information carried in the PM payload contents regarding link quality, and includes its current values.

At destination, a time-out is triggered upon the arrival of the first PM packet. Possible PM packets received after expiration are considered either as lost packets or as a signal of a path that produces too high delay for the service. After time-out expiration, a *Probe Message Reply* (PMR) is generated with all the information collected from all the PM packets that have arrived in time, and it is sent back to the

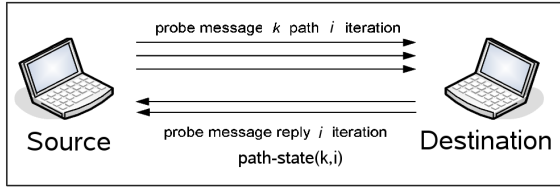


Figure 1: PM and PMR packets.

source through each one of the paths through which a PM arrived.

The information contained in the PMR packet consists of the list of possible paths available to route data to the destination, and the collected values of the parameters of the path-state for each path. This information is processed by the source to evaluate the network state. The source gives qualifications to each one of the paths in order to classify them. The qualifications are obtained by comparing those values with certain thresholds for each one of the QoS parameters. In previous versions of MMDSR [6] these thresholds were fixed, but in this improved version they are dynamically computed as a function of the network state. This computation is described in section 4.

3.2.2 Hello Messages

Periodic Hello Messages (HM) help to compute two parameters that are used in the algorithm: *Reliability Metric* (RM) and *Mobility Metric* (MM). HM are sent once a second by nodes that belong to any path involved in a video transmission, in order to achieve continuous monitoring of the perceived signal strength of the neighbouring nodes. Once a HM is received, the reception node computes the SNR (Signal to Noise Ratio) regarding the received packet and attaches this value to a Hello Message Reply (HMR), which is sent back to the source of that HM packet.

3.3 QoS parameters computation

3.3.1 Reliability Metric, RM_k^i

We propose to compute a performance measure of the whole quality of a path by taking SINR (Signal-to-Interference plus Noise Ratio) measures between consecutive nodes within path k . For each one of the i iterations of the algorithm and each k path, we obtain the $SINR_j^{k,i}$ values of each j node within respect the $j-1$ node on the downstream path from source to destination, and we assign qualifications $x_j^{k,i}$ as follows:

According to simulation results in ns-2 [8] under several packet loss scenarios in MANET, we have set the SINR thresholds to classify links in (3) by measuring the fraction lost parameter obtained from the RR-RTCP feedback packet for different SINR values. Besides, we have used the results of [9] that relate the MOS (Mean Opinion Score), a well-known subjective QoS parameter, with the fraction lost parameter. Finally, the SINR thresholds have been set as shown in (3).

We compute the mean qualification of the whole path k , RM_{mean}^k , applying the geometric average of all the partial

$x_j^{k,i}$ qualifications of each node j of the path k . L_k^i corresponds to the number of links (i.e. hops) in each path k . This way, RM_{mean}^k is computed as follows:

$$\begin{aligned}
 & \text{If } SINR_j^{k,i} \geq 25dB, \text{ then } x_j^{k,i} = 3 \\
 & \text{If } 15dB \leq SINR_j^{k,i} < 25dB, \text{ then } x_j^{k,i} = 2 \\
 & \text{If } 10dB \leq SINR_j^{k,i} < 15dB, \text{ then } x_j^{k,i} = 1 \\
 & \text{If } SINR_j^{k,i} < 10dB, \text{ then } x_j^{k,i} = 0
 \end{aligned} \tag{3}$$

$$RM_{mean}^k = \sqrt[L_k^i]{\prod_{j=1}^{L_k^i-1} x_j^{k,i}} \tag{4}$$

$$\begin{aligned}
 & \text{If } (RM_{mean}^k > 2 \cdot f_{RM}^{i-1}) \text{ then } RM_k^i = 3 \\
 & \text{If } (1.5 \cdot f_{RM}^{i-1}) < RM_{mean}^k \leq 2 \cdot f_{RM}^{i-1} \text{ then } RM_k^i = 2 \\
 & \text{If } (1 \cdot f_{RM}^{i-1} < RM_{mean}^k \leq 1.5 \cdot f_{RM}^{i-1}) \text{ then } RM_k^i = 1 \\
 & \text{If } (RM_{mean}^k \leq 1 \cdot f_{RM}^{i-1}) \text{ then } RM_k^i = 0
 \end{aligned} \tag{5}$$

The final qualifications for each one of the available paths k during iteration i , are assigned as in (5), where f_{RM}^i is a function that changes dynamically throughout time depending on the state of the network. It is detailed in section 4.

3.3.2 Mobility Metric, MM_k^i

Each node X detects the received signal power $RxPr_{Y \rightarrow X}$ from its neighbour Y (which belongs to one of the available paths from source to destination) from successive packet transmissions (periodic “Hello” messages).

Once this value has been returned to the original node of the *Hello Message*, each local qualification for the relative mobility metric is computed as follows:

$$M_X^i = E \left[\left(10 \log_{10} \frac{RxPr_{Y \rightarrow X}^i}{RxPr_{Y \rightarrow X}^{i-1}} \right)^2 \right] \tag{6}$$

A low value for M_X^i means that the X node is relatively less mobile than its neighbours, while a high value shows that X moves more than their neighbours. A path whose nodes have a lower aggregate relative mobility will be preferred in front of other paths whose nodes have a high global value of mobility. Then, next time that a *Probe Message* arrives at node X , the node will append the current value of its mobility metric qualification, M_X^i .

$$\begin{aligned}
& \text{If } (M_X^i < 0.02 \cdot f_{MM}^{i-1}) \text{ then } MM_j^{k,i} = 3 \\
& \text{If } (0.02 \cdot f_{MM}^{i-1} \leq M_X^i < 0.08 \cdot f_{MM}^{i-1}) \text{ then } MM_j^{k,i} = 2 \\
& \text{If } (0.08 \cdot f_{MM}^{i-1} \leq M_X^i < 0.5 \cdot f_{MM}^{i-1}) \text{ then } MM_j^{k,i} = 1 \\
& \text{If } (M_X^i \geq 0.5 \cdot f_{MM}^{i-1}) \text{ then } MM_j^{k,i} = 0
\end{aligned} \tag{7}$$

After this, the destination computes the global qualification of the mobility metric for that path k . We propose to compute a mobility measure of each path this way: for each path k we obtain the M_X^i values of each node X within that path using (6). After that, we assign qualifications to each node j within path k as in (7), where f_{MM}^i is a function that changes dynamically as a function of the state of the network (see section 4).

Finally, we compute the arithmetic mean from the $MM_j^{k,i}$ values for each node j to obtain a measure of the mobility of path k , during iteration i of the algorithm, MM_k^i .

$$MM_k^i = \left(\sum_{j=1}^{L_k^i-1} MM_j^{k,i} \right) / L_k^i \tag{8}$$

3.3.3 Hop Metric, h_k^i

We obtain a measure regarding the metric of the distance, from h_{max}^i (the number of hops of the longest path), and h_{min}^i (the number of hops of the shortest path) in the current algorithm iteration i . This way, qualifications regarding the lengths of the paths (i.e. Hop Metric Mh_k^i) are obtained as in (9). This way, we assign higher scores to shorter paths. Shorter paths are preferred, as fewer losses will take place due to contention for the medium.

$$\begin{aligned}
A_i &= h_{max}^i - \frac{h_{max}^i - h_{min}^i}{3} \\
B_i &= h_{max}^i - 2 \cdot \frac{h_{max}^i - h_{min}^i}{3} \\
\text{If } A_i < h_k^i \leq h_{max}^i, & \text{ then } Mh_k^i = 0 \\
\text{If } B_i \leq h_k^i \leq A_i, & \text{ then } Mh_k^i = 1 \\
\text{If } h_{min}^i \leq h_k^i < B_i, & \text{ then } Mh_k^i = 2
\end{aligned} \tag{9}$$

3.3.4 p_k^i, d_k^i y j_k^i computation

Once the PMR packet arrives at the source node, this node gets the values for losses, delay and jitter which have been sampled in each path. With these sampled values, the source node computes the losses p_k^i , delay d_k^i , delay jitter j_k^i for each k path in the current i iteration. We apply EWMA (Exponentially Weighted Moving Average) filters, with a low value for the α coefficient (e.g. 0.125) to give more relevance to the historical values in front of the instantaneous ones and to let parameters evolve in a smooth way:

$$v_k^i = (1 - \alpha) \cdot v_k^{i-1} + \alpha \cdot v_k^{i_sample}$$

where v_k^{i-1} (standing for p_k^{i-1}, d_k^{i-1} y j_k^{i-1} respectively) are the values computed in the previous iteration.

Finally, qualifications will be applied to each path depending on the values of these parameters, as it is shown in equation (10), which is a shared equation to compute v_X^i corresponding to losses Mp_k^i , delay Md_k^i and delay jitter Mj_k^i for each path k . In (10) v_{MAX} corresponds to the user's requirements expressed in (1), i.e. v_{MAX} equals $p_{MAX}, d_{MAX}, j_{MAX}$, respectively. f_{pdj}^i is a function that adapts to the network state (see section 4).

$$\begin{aligned}
& \text{If } (v_X^i < 0.4 \cdot f_{pdj}^{i-1} \cdot v_{MAX}) \text{ } Mv^{k,i} = 2 \\
& \text{If } (0.4 \cdot f_{pdj}^{i-1} \cdot v_{MAX} \leq v_X^i < 0.8 \cdot f_{pdj}^{i-1} \cdot v_{MAX}) \text{ } Mv^{k,i} = 1 \\
& \text{If } (v_X^i > 0.8 \cdot f_{pdj}^{i-1} \cdot v_{MAX}) \text{ } Mv^{k,i} = 0
\end{aligned} \tag{10}$$

3.3.5 End-to-end bandwidth BW_e^i

The PM packet also collects BW_e^i , which is the end-to-end available bandwidth for each one of the k paths. It corresponds to the bandwidth of the bottleneck link (i.e. the link with lower bandwidth) within the path. Qualifications are applied to each path as follows:

$$\begin{aligned}
& \text{If } (BW_e^i > 1.2 \cdot f_{BW}^{i-1} \cdot BW_{min}) \text{ } MBW_j^{k,i} = 2 \\
& \text{If } (f_{BW}^{i-1} \cdot BW_{min} \leq BW_e^i \leq 1.2 \cdot f_{BW}^{i-1} \cdot BW_{min}) \text{ } MBW_j^{k,i} = 1 \\
& \text{If } (BW_e^i < f_{BW}^{i-1} \cdot BW_{min}) \text{ } MBW_j^{k,i} = 0
\end{aligned} \tag{11}$$

being BW_{min} the minimum bandwidth required to transmit the video stream. This way, we assign a higher score when the available bandwidth is higher. Again, f_{BW}^i is a dynamic function that reflects the changes in the network state (see section 4).

3.4 Path classification

Once the available paths which fulfils the user's requirements set in (2) have been selected, the algorithm arranges these paths by checking sequentially the qualifications of the parameters as follows in the next list. Finally, the source selects the number of paths required by the multipath routing scheme.

1. $RM_k^i + MM_k^i$.
2. Mh_k^i .
3. MBW_k^i .
4. $Mp_k^i + Mj_k^i$.
5. Md_k^i .

Among several possibilities to configure this list, we have chosen the RM and MM to be the most important parameters to classify paths. The reason is that for video-streaming services the most reliable and stable paths are preferred. Delay, jitter and losses are not so determinant metrics.

4. DYNAMISM IN THE OPERATION TO IMPROVE THE PERFORMANCE

So far, our framework was operating with a fixed period of 10 seconds per iteration of the algorithm. The thresholds to classify paths according to the QoS parameters were also fixed and heuristically set after carrying out several representative simulations. Nevertheless, due to the inherent characteristics of MANETs, the network topology is highly dynamic. Hence, the designed framework should be dynamic as well. Only a few works have been done following this line, i.e. [10]. We foresee that a dynamic design will considerably improve the service performance. To achieve this dynamism, our system has to be able to adapt its configuration to the different network situations it can face. As the state of an ad hoc network can vary considerably throughout time, the framework has to adjust itself in order to offer always the best available paths for the transmission. Besides, it has to be able to classify paths properly. We realized that with a static configuration in some scenarios the protocol is not able to act correctly in front of congested situations. If no adaptation is done, in high load traffic situations all the paths would obtain bad values in their qualifications. The reason is the lack of sufficient granularity that makes the system unable to distinguish paths. In addition, if the period of the algorithm is fixed, the amount of traffic overhead generated is the same either under low traffic and low mobility situations than in saturated conditions. To tackle these issues, in this paper we have designed a-MMDSR, an improvement of our previous proposal s-MMDSR [6]. The main novelty of this work is to transform the fixed network parameters in dynamic ones, so that our framework is able to adapt itself according to the network dynamism. The adaptation is fulfilled applying a correction factor to adjust dynamically the thresholds to assign qualifications. This correction factor varies as a function of the collected values of the quality parameters of the paths. With this dynamic adjustment of the thresholds the granularity and resolution to classify paths increases and thus, the system is able to classify paths better. Besides, depending on the qualifications obtained for the paths (i.e. depending on the network state) the iteration period of the algorithm will be modified (lower period for bad global qualifications of the paths and higher period for good qualifications). This way, lower overhead is produced under stable and favourable situations, while under high mobility situations new paths are needed sooner as the topology varies frequently.

To achieve this goal, we have analyzed the static version s-MMDSR working under different network situations. We have defined a new parameter called $NState$, which gives information about the global network state. This value is defined as follows:

$$NState^i = w_{RM} \cdot \overline{RM}^i + w_{MM} \cdot \overline{MM}^i + w_{BWe} \cdot \overline{BWe}^i + w_p \cdot \overline{Mp}^i + w_d \cdot \overline{Md}^i + w_j \cdot \overline{Mj}^i + w_h \cdot \overline{Mh}^i \quad (12)$$

$$w_{RM} + w_{MM} + w_{BWe} + w_p + w_d + w_j + w_h = 1 \quad (13)$$

The average values of each parameter in (12) are the average qualifications obtained for all the paths in iteration i of the algorithm. The w_x values correspond to the weighting values for each parameter, and they have to add the unit (13). Depending on the relevance we want to give to each parameter in the algorithm, weights will have a higher or lower value. Once the source has received the PMR packets and has processed all the information, it computes the $NState$ as defined in (12). The variation of this parameter throughout time depends directly on the qualifications obtained for the paths and therefore on the variations of the network. We have modified the thresholds in the algorithm parameters, so that they vary depending on $NState$ following a set of functions which are described in next section.

4.1 Dynamic thresholds to qualify paths

We make our framework adapts dynamically to the network conditions by means of a set of linear functions which will modify the thresholds in the algorithms. These functions depend on $NState$, which tracks the global state of the network as (12) shows. Equation (14) shows the function for each QoS parameter: Reliability Metric, Mobility Metric, losses, delay, jitter and available bandwidth.

$$\begin{aligned} f_{RM}^{i+1} &= 0.4 \cdot NState^i + 0.6, \\ f_{MM}^{i+1} &= -0.6 \cdot NState^i + 2 \\ f_{pdj}^{i+1} &= -0.4 \cdot NState^i + 1.5 \\ f_{BW}^{i+1} &= 0.12 \cdot NState^i + 0.9 \end{aligned} \quad (14)$$

We have designed these parameters after carrying out a high number of simulations under a wide range of situations where the network performance was good, normal and bad. These functions have an initial value of 1 and they evolve throughout time iteration by iteration. The parameter h regarding the hop counter is not involved in the dynamism of the algorithm, as no changes are produced in this parameter due to fluctuations in $NState$. The dynamism behaviour has been introduced in the period of the algorithm as well. In a similar way as with the parameters, the period has been established in dynamic mode by using a linear function that depends on $NState$:

$$MMDSR_{period} = 10 \cdot NState + 3 \quad (15)$$

Notice since the range of values for RM and MM is $[0,3]$ and for BW , Mp , Md , Mj and Mh are $[0,2]$, the $MMDSR_{period}$

Table 1: Simulation settings.

Area	400x400 m
Number of nodes	80
Max. nodes speed	10m/s
Transmission range	70m
Movement pattern	Random Waypoint
MAC specification	IEEE 802.11b
Bandwidth	11 Mbps
Simulation time	250s
Video codification	MPEG-2 VBR
Video rate	150 Kbps
Transport protocol	RTP/RTCP/UDP
Maximum packet size	1500 bytes
Multipath scheme	N=3
Queue size	80 packets
Video sources	1
w_x	1/7
Interferent CBR traffic rate	150 Kbps
Channel noise	-92 dBm

is a value somewhere between [3, 25.85] seconds. Once all the dynamic thresholds in the algorithm have been brought up to date, their values are stored and they will be applied in the next iteration of the algorithm.

5. SIMULATION RESULTS

We have carried out simulations with the well known open source network simulator, ns-2 v2.27 [8]. The whole system has been implemented in it and different ad hoc network scenarios have been analyzed emulating real environments where video-streaming services are used. The main simulation settings are shown in Table 1. We have compared the performance of our new framework a-MMDSR with respect to our previous version s-MMDSR and with respect to the well known routing protocol DSR.

In the simulations there is a single flow of video between two nodes. The communication is disturbed by an interferent CBR traffic. As we can see in Table 1 we have given the same relevance to all the QoS parameters involved in the computation of $NState$, i.e. $w_x = 1/7$ in (12).

In Figure 2 the evolution of the packet losses is shown. Now we can perceive again that both MMDSR protocols outperform DSR, but also that a-MMDSR performs better than s-MMDSR. Here, the dynamic adaptation of the thresholds and the period of the algorithm has been the key to achieve these good results.

Regarding the quality of the video received, Figure 3 depicts the PSNR obtained at different moments over the simulation using [11], to have an objective measure of the subjective performance in the user-level. As we can see, the higher PSNR is obtained with a-MMDSR.

In Figure 4 we can see how the iteration period of a-MMDSR is higher than the iteration period of s-MMDSR, which was fixed to 10 seconds. As long as a-MMDSR is able to distinguish the paths correctly, there is no need to decrease the period of the algorithm iteration. Using a higher period, the traffic due to overhead is lower in a-MMDSR than in

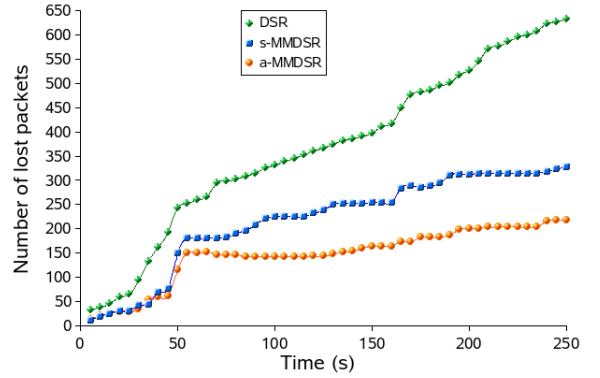


Figure 2: Packet losses evolution for DSR, s-MMDSR and a-MMDSR.

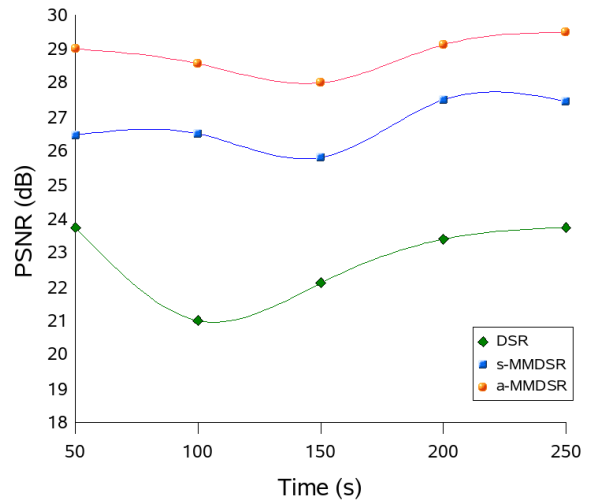


Figure 3: PSNR evolution for DSR, s-MMDSR and a-MMDSR.

s-MMDSR, as Table 2 shows. Although in both MMDSR cases the overhead is higher than with standard DSR, it is worthwhile as the operation is better. The overhead traffic in MMDSR includes standard DSR engine control packets and the MMDSR monitorization packets.

In the simulations, there is a video source node (#0) which sends a video flow to a destination node (#20) using a multi-path scheme with 3 paths. In Table 3 we can see all the available routes discovered in an iteration between source and destination using MMDSR, and the path qualifications obtained for both protocols s-MMDSR and a-MMDSR. Those path qualifications simply are the addition of all the partial qualifications, i.e. RM , MM , MBW , Md , Mp , Mj and Mh . Notice that a-MMDSR has finer granularity to classify paths than s-MMDSR. This way we are able to distinguish the best three paths from all the available paths. With s-MMDSR there is a virtual draw between three paths that have obtained the highest qualification, i.e. 15. On the other hand, as a-MMDSR adapts dynamically to the network state, there is a higher resolution to qualify paths, and the marks obtained are different enough so that the algorithm is able to distinguish those paths. Finally, paths are

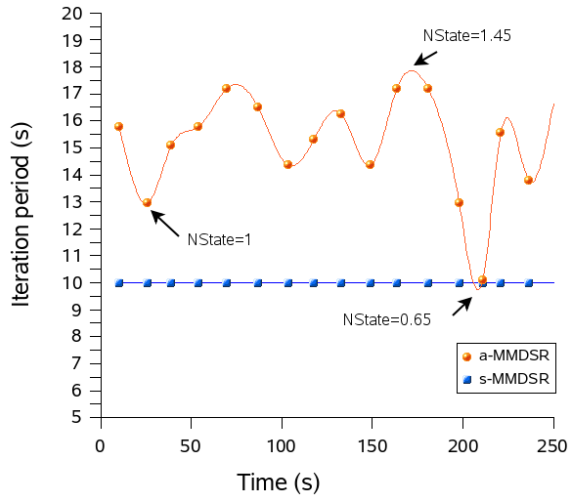


Figure 4: MMDSR period algorithm evolution for s-MMDSR and a-MMDSR.

Table 2: Signaling overhead.

DSR	0.4 %
s-MMDSR	6.5 %
a-MMDSR	5.2 %

arranged according to section 3.4 and the results show that the final performance is improved.

6. CONCLUSIONS AND FUTURE WORK

In this article we have proved that the dynamic version of our protocol a-MMDSR outperforms the static version s-MMDSR of our proposal. Simulation results demonstrate the effectiveness of our approach by providing less percentage of losses suffered by the video flows whereas less traffic due to overhead is introduced and a better PSNR is achieved.

In a situation where practically all the paths obtained very similar qualifications with s-MMDSR, a-MMDSR is able to distinguish the good paths from the bad ones. This is achieved because a-MMDSR auto-modifies the thresholds of its parameters properly. Also, the period between iterations of the algorithm varies depending on the state of the net-

Table 3: Path qualifications.

Route	s-MMDSR	a-MMDSR
[0 30 4 20]	15	18
[0 62 77 11 20]	15	17
[0 35 34 12 20]	15	14
[0 39 14 20]	14	12
[0 31 9 15 20]	14	11
[0 23 7 17 20]	13	9
[0 23 19 20]	13	8
[0 33 8 6 20]	11	4

work, seeking to decrease the amount of overhead traffic in stable and low loaded networks, whereas improving the performance under saturated networks.

As immediate work, it would be interesting to evaluate the system performance using the MAC IEEE 802.11e specification, which is focused to offer certain QoS levels over ad hoc networks. This MAC specification uses four different priority queues with different values for MAC access parameters, which we could make adapt to the network state.

7. ACKNOWLEDGEMENTS

This research article is supported by the spanish projects SECONNET (CICYT-TSI2005-07293-C02-01), ITACA (CICYT TSI2007-65393-C02-02) and by these grants: CONACYT, Carolina Foundation, PROMEP-UAQ, Alban E05D052 898MX and UPC Recerca.

8. REFERENCES

- [1] A. Bamis, A. Boukerche, I. Chatzigiannakis, S. Nikolettseas, "A mobility aware protocol synthesis for efficient routing in ad hoc mobile networks", *Computer Networks* 52 (2008) pp. 130-154.
- [2] Xuefei Li, Laurie Cuthbert, "Multipath QoS Routing of supporting Diffserv in Mobile Ad Hoc Networks", *Proceedings of SNPD/SAWN'05*.
- [3] S. Mao, Y. Thomas Hou, X. Cheng, H. D. Sherali, S. F. Midkiff, "Multipath Routing for Multiple Description Video in Wireless Ad Hoc Networks", *Proceedings of IEEE INFOCOM*, March 2005.
- [4] W. Rong, M. Wu, T. Yu, "LSMR: A Label Switching Multipath Routing Protocol for Ad Hoc Networks", *8th ACIS International Conference*, Vol. 2, 2007, pp. 546-551. DOI 10.1109/SNPD.2007.347.
- [5] L. Zhao, J.G. Delgado-Frias, "Performance Analysis of Multipath Data Transmission in Multihop Ad Hoc Networks", *SECON 2006. 3rd Annual IEEE Communications Society on Volume 3*, pp. 927-932. DOI 10.1109/SAHCN.2006.288584.
- [6] V. Carrascal Frías, G. Díaz Delgado, M. Aguilar Igartua, "Multipath Routing with Layered Coded Video to Provide QoS for Video-streaming applications over MANETS". *14th IEEE (ICON 2006)*. ISSN: 1556-6463, 2006. ISBN 0-7803-9746-0.
- [7] RFC 4728: The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4, February 2007, <ftp://ftp.rfc-editor.org/in-notes/rfc4728.txt>
- [8] The Network Simulator, ns-2, <http://nsnam.isi.edu/nsnam/>
- [9] J. Shin, J.W. Kim, J. Kuo. "Quality-of-Service Mapping Mechanism for Packet Video in Differentiated Services Network", *IEEE Transactions on Multimedia*, Vol. 3, Num. 2, pp. 219-231, 2001.
- [10] K. Zita Haigh, S. Varadarajan, Ch. Yik Tang, "Automatic learning-based MANET Cross-Layer Parameter Configuration", *WWASN 2006*.
- [11] vidprofile 0.80. <http://vidprofile.berlios.de/setup.html>
- [12] C.T. Calafate, P. Manzoni, M.P. Malumbres, "On the interaction between IEEE 802.11e and routing protocols in Mobile Ad-hoc Networks", *13th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, 2005.