

Improving H.264 real-time streaming in MANETs through adaptive multipath routing techniques

Carlos T. Calafate, Manuel P. Malumbres, and Pietro Manzoni

Department of Computer Engineering

Polytechnic University of Valencia (UPV), Spain

Email: {calafate, mperez, pmanzoni}@disca.upv.es

Abstract—One of the main problems associated with MANETs is that mobility and the associated route discovery and maintenance procedures of reactive routing protocols cause interruptions on real-time video streams. Some of these interruptions are too large to be concealed using any sort of video technology, resulting in communications breaks unpleasant for the final end user. We present a solution for enhanced video transmission that increases route stability by using an improved route discovery process based on the DSR routing protocol, along with traffic splitting algorithms and a preventive route discovery mechanism. We also present a tuning methodology for an H.264 video codec in order to obtain optimal interaction with the underlying mechanisms proposed.

Combining our proposals, we achieve up to 97% less interruptions on communication with high mobility, and over 1.2 dB of improvements in terms of video distortion.¹

I. INTRODUCTION

Mobile Ad-hoc networks, also known as MANETs, are composed by a set of independent mobile nodes which “cooperate” without any type of infrastructure. The low cost and ease of deployment of this kind of networks makes them extremely attractive for applications ranging from military and disaster relief situations to small home environments. These networks have been under intensive research in the last few years, and more and more applications are being found for MANETs.

The JVT H.264 / MPEG-4 part 10 standard [1], jointly developed by the ITU-T VCEG and ISO MPEG-4 groups, is a new video technology able to achieve very good video quality levels at low bit-rates, as well as enhanced error resilience. Such properties make it adequate for MANET environments, which are bandwidth constrained and error-prone networks.

Current solutions in terms of routing protocols, access to the wireless medium, etc. are acceptable for best-effort traffic. However, the current performance of MANETs can hardly be accepted for real-time multimedia communication.

In [2] we found that even when a video flow does not have to face competition with other flows, and when the routing protocol operates in optimal conditions, video performance is still penalized by mobility.

In this work we show that there is an intimate relationship between video gaps and route disruptions. To alleviate this problem we propose a technique to reduce the number and

size of video gaps by enhancing the DSR route discovery procedure. We then describe an adaptive traffic splitting approach which uses a route selection mechanism that optimizes the use of multiple disjoint routes. The use of disjoint routes reduces video gaps occurrence generated by node mobility, thus improving the quality of the received video. Finally, in order to prevent the occurrence of route losses we provide a preventive route discovery mechanism activated when a video flow does not have at least two disjoint routes available.

To further improve the results, we make an analysis of the optimal tuning of the H.264 codec focusing on optimal integration with the packet splitting mechanism in terms of packetization, frame types, and packet replication procedures.

To measure the effect of video gaps over the final video quality delivered to the user, we propose a metric called “video annoyance” (VA). This metric helps in evaluating the behavior of the proposed route discovery and traffic splitting mechanisms.

Concerning the structure of this paper, in section II we refer some related works. In section III we make a brief introduction to the DSR protocol and describe the different route discovery techniques proposed. Section IV presents the effects of applying enhanced route discovery procedures over the delivered video quality, performing a detailed study of the video loss pattern. In section V we present a multipath routing algorithm that introduces traffic splitting as a mechanism to improve final video quality when node mobility is significant. In section VI we perform a global evaluation of all mechanisms presented in this paper. Finally, in section VII we make our concluding remarks.

II. RELATED WORK

The use of multiple routes in MANETs has recently become a promising solution for multimedia data transmission.

Wang et al. [3] use a probing technique is used in order to assess the quality of available routes, so that the traffic is forwarded based on the delay of each route. Their objective was also to achieve load distribution as well as improved throughput, end-to-end delay and queue utilization.

Nasipuri et al. [4] proposed a strategy for quick route recovery through packet re-direction in intermediate nodes to reduce the frequency of query floods. Their solution aims at reducing the number of *lost route* messages, as well as performing fewer route discoveries. However, the source is

¹This work was supported by the *Ministerio de Ciencia y Tecnología under grant TIC/2003/00339*, and the *Junta de Comunidades de Castilla La-Mancha under grant PBC/03/001*.

unaware of any extra routes, which means that their solution does not aid in the task of splitting traffic through disjoint routes.

Wu [5] proposes a more selective route discovery procedure to DSR to increase the degree of disjointness of routes found without introducing extra overhead. It allows the source to find a maximum of only two paths (node disjoint paths) per destination.

In [6] Lee and Gerla show that the paths found by DSR's route discovery mechanism are mainly overlapped, and so they propose an extension to find more paths.

In [7] the AODV protocol has been extended in order to provide multi-path capabilities, though no new route discovery mechanism was proposed. Both node disjoint and link disjoint approaches are presented. In their work there is no traffic splitting.

Finally, in [8] the authors use multiple node disjoint routes as a basis for reliable routing through the deployment of reliable nodes.

In this paper we will show how the availability of extra routes, along with new algorithms to optimize path disjointness, reduce the chances of losing all routes to a destination. Such methods prove to be very effective in reducing video gaps.

III. ROUTE DISCOVERY EXTENSIONS TO DSR

The Dynamic Source Routing (DSR) protocol is a high performance protocol for MANETs. Its route discovery process is on-demand, which means that routes are only built when needed; route maintenance also depends on the existence of traffic. Therefore, when there is no traffic flowing the routing traffic is effectively reduced to zero.

When initiating a route discovery, the source broadcasts a route request (RREQ) packet, which is successively broadcasted by other nodes until the destination is reached. When relying on IEEE 802.11 for transmission, packet broadcasting suffers from noise and collisions; this means that there is no assurance that the best route will be discovered, or that a request will arrive to the destination at all. Route replies, on the other hand, are unicast packets and, when using IEEE 802.11, this means that they are acknowledged so communication is more reliable.

DSR is a protocol that, by default, finds only a small number of routes. By extending the route discovery mechanism in DSR we are able to increase the average number of routes found per node. This extra information alone offers to each node extra possibilities when a route is lost, requiring on average fewer route discovery processes. As an enhancement, nodes can use the extra information for other useful purposes, like for packet splitting or replication over disjoint routes, route congestion analysis, etc. Although the enhancements proposed have a bandwidth cost on route discovery actions, we believe that this cost is acceptable, especially taking into consideration the trend towards increased bandwidth in recent standards.

With our route discovery proposal (see [9]) we decrease the routing overhead when compared to the Lee and Gerla's

route discovery technique. Their proposal consists of altering the route discovery process so that during the "RREQ" propagation phase packets with the same route request ID can be forwarded if they arrive "through a different incoming link than the link from which the first RREQ is received, and whose hop count is not larger than the first RREQ".

From now on we shall refer to our solution as "Super Restrictive" mode (SR). In SR mode, we add a list (*SRList*), to the already existing route discovery table structure in all nodes. This list is used to store the intermediate hops that forwarded the route request. The cost in terms of memory is very small - only 4 bytes per intermediate node per source. The main enhancement of the SR mode is that it discontinues the propagation of a route request if some of the previous hops (except the source) is the same. With this method we ensure that the discovered paths are node disjoint, increasing therefore the usefulness of the routes found.

The size of the *SRList* can be controlled and limited. When a route request arrives and the list is already full, it is not propagated. This means that only a pre-defined number of route requests are forwarded. If the size of the *SRList* is very high we obtain the basic SR mode; if it is equal to one the behavior is similar to the DSR's propagation mode. The size of the *SRList* is a new parameter and it will be referred to as *PNC* (Previous Node Count).

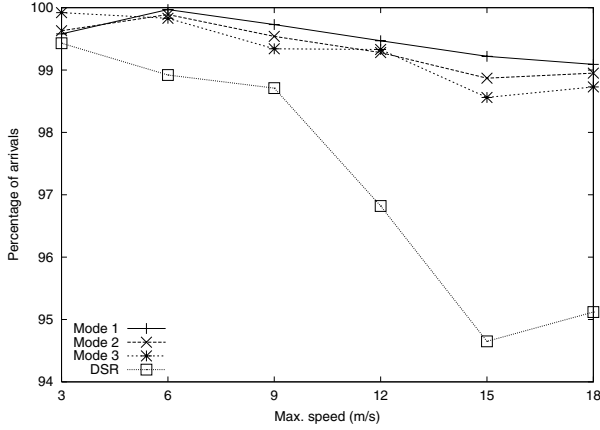
The SR solution restricts the route request forwarding process to route sizes not superior to the first one arriving. To increase the flexibility of the approach we accept routes with an extra size up to a certain value. We call this parameter *flexibility*. In [9] we found that only small values for this parameter are useful in terms of route size and stability.

In the following section we test three different *Flexibility / PNC* pairs again taking into account the results found in [9]. In mode 1 (*Flex=0, PNC=2*) the propagation using the SR technique is restricted to the maximum, so that only one extra route per node is allowed relative to the default DSR behavior. Modes 2 and 3 maintain one of the parameters of mode 1, but in mode 2 (*Flex=2, PNC=2*) we increase flexibility and on mode 3 (*Flex=0, PNC=4*) we increase the number of RREQs propagated per node. In all modes the use of cache on route propagation is turned off, maintaining the rest of DSR's behaviors unchanged.

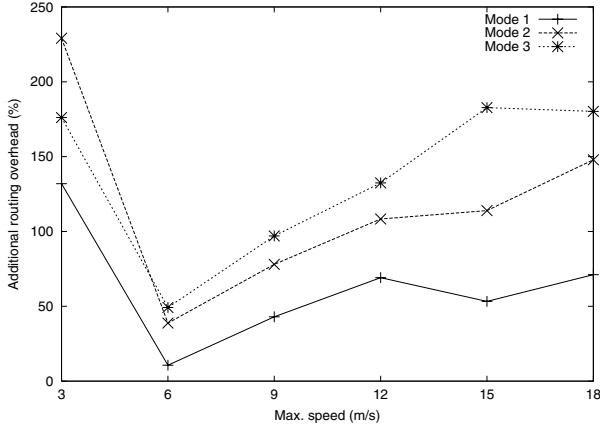
IV. EFFECTS OF ROUTE STABILITY ON REAL-TIME VIDEO STREAMS

In this section we put in evidence the presence of gaps in video streams, evaluating the performance of DSR and the three different SR modes when facing this problem. We show that video gaps are intimately related to route discovery procedures, and why the SR modes presented in the previous section can considerably alleviate the video gap problem.

The evaluations are done using the ns-2 simulator [10]. Each obtained value is an average over 5 simulation runs. Concerning node movement, it was generated using the random waypoint mobility model. A filter was applied to accept only scenarios without network partitioning (i.e., with no



a)



b)

Fig. 1. Packet arrivals (a) and routing overhead (b) for the H.264 video flow

unreachable destinations), in order to obtain a connected graph.

We first evaluate a 1000×1000 m squared scenario with 80 nodes. The traffic load consists of a single H.264 video stream obtained from the well known *Foreman* sequence at 10 frames-per-second. Each video frame is split into 7 RTP packets, resulting in a target bit-rate of 186 kbit/s. Our purpose is to observe the performance of the different routing protocols independently from other traffic flows.

In figure 1 we observe how modes 1 to 3 always perform slightly better than the original DSR implementation in terms of packet arrivals. The best performing mode is SR mode 1, where the improvements over DSR reach 4.5% in packet arrivals. Concerning routing overhead, it generates fewer routing control packets than the remaining two, with a small relative increase compared to DSR.

A. Loss pattern analysis

The H.264 standard offers a wide range of tools to reduce the effects of degradation in the presence of losses. Different types of intra macroblock updating strategies and error con-

cealment tools are available, which aims at estimating parts of frames which are not received. Thus, observing the results of figure 1, and taking into account these facts we could conclude that the difference in terms of PSNR (Peak Signal-to-Noise Ratio) between receiving 99% or 95% of the packets would be slight.

However, packets dropped in bursts long enough to cause video gaps affect the H.264 video decoder in a drastically different manner. When no information relative to consecutive frames arrives to the decoder, it will freeze the last decoded frame until communication is resumed. When communication is resumed the decoder's effort is also increased since it must resynchronize and recover from losses as quickly as possible. We therefore argue that the PSNR is not a representative factor and we propose a new metric which we called *VA (Video Annoyance)* parameter, defined as:

$$VA = \frac{\sum_{i=0}^N (G_i)^2}{NF^2}, \quad 0 \leq VA \leq 1,$$

where NF is the total number of frames in the sequence, N is the total number of independent video gaps occurring in a video sequence and G_i is the size of the i -th video gap. We define a video gap $G_i = \frac{B_i}{PPF}$ as the number of video frames lost sequentially; B_i is the number of consecutive packets lost for gap i ($B_i \geq 2$) and PPF is the number of packets per frame.

This number does not need to be an integer, since for example $4\frac{1}{2}$ frames can be lost in a single burst, being communication resumed with information relative to some position inside a frame. What must be defined, though, is the minimum number of consecutive lost packets to create a video gap. In this work we set that threshold to one entire frame, that is, 7 packets ($B_i \geq 7$).

Concerning VA limits, when $VA = 0$ there are no video frame gaps; when $VA = 1$ the entire sequence is lost. The quadratic relation takes into account the fact that many distributed 1-frame gaps are almost imperceptible to the viewer, while a single 50 frames gap (5 seconds interruption at 10Hz) is quite disturbing for the user.

This definition of VA requires comparisons to be made between sequences of equal length. If the sequence's length differs, we can split the sequences into smaller parts and find the average VA value for comparison. However, these intervals should be as big as possible to accurately penalize large video frame gaps. Another possibility is to concatenate the same video sequence over and over until a similar size is obtained.

Analyzing a typical packet drop pattern on a simulation using DSR and a single H.264 video flow, we can observe (see figure 2) that some of the packet loss events are bursty. This causes the video flow to be stopped, so that several entire frames are lost.

Table I presents a comparison between the different routing protocols concerning the VA parameter. As it can be seen, the VA associated with modes 1 to 3 is only a small fraction (1-2%) of the VA achieved with the original DSR implementation.

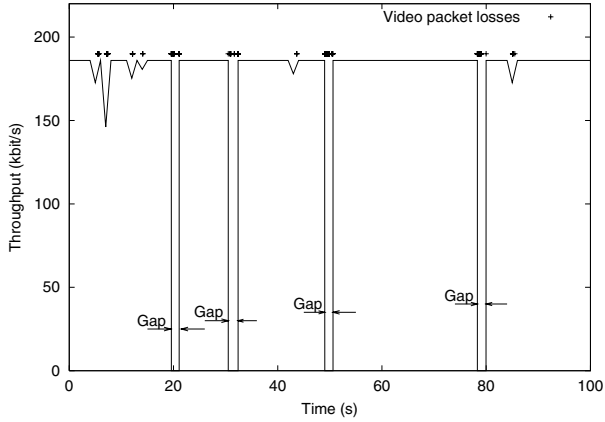


Fig. 2. Video throughput variation

TABLE I
VIDEO ANNOYANCE STATISTICS

Protocol	VA (10^{-6})	VA % relative to DSR
DSR	38.1	-
Mode 1	0.502	1.3169
Mode 2	0.708	1.8575
Mode 3	0.768	2.0167

This result alone proves how the different SR modes improve the video experience in terms of video disruptions.

B. Gap causes and solutions

In the scenarios where congestion is not a problem like the ones we use, packet losses are intimately associated with link breaks and subsequent route failures.

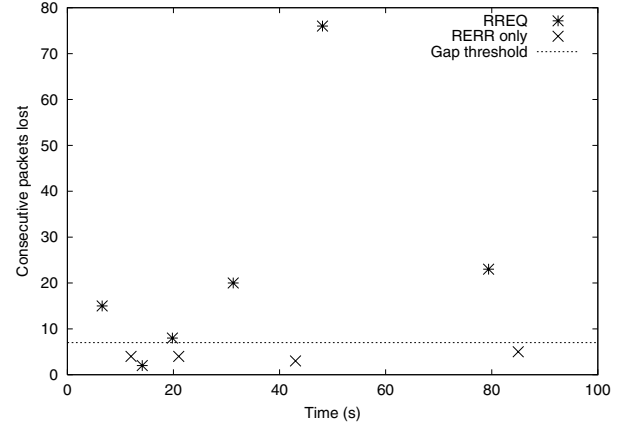
However, DSR uses link layer information to detect broken links, so the interval between the detection of a broken link and the reception of associated notification by the source is, in terms of video streaming, not excessively long. In fact, we find that when a route breaks the number of packets lost can be estimated by:

$$N = 2 \cdot S_{rate} \cdot T_{RERR},$$

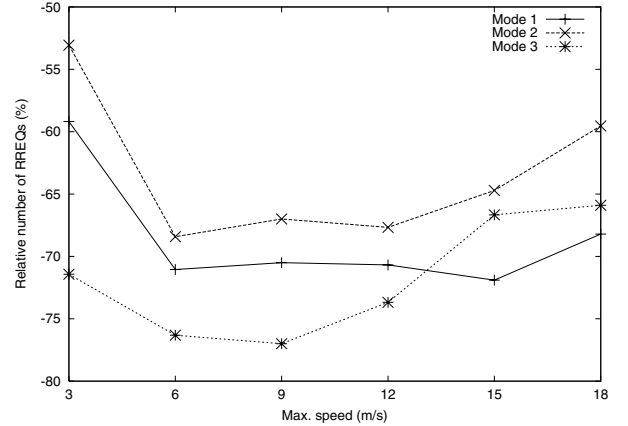
where S_{rate} is the source's packet generation rate in packets per second and T_{RERR} is the time that the "Route error" packet takes to arrive at the source. This phenomena can sometimes be alleviated by DSR's packet salvaging mechanism.

We therefore conclude that the cause for such long video gaps is related to route discovery procedures, and not just route failures. Figure 3 (left) aids at proving this statement. As it can be seen, only RREQ procedures force the number of consecutive packets dropped to be above the gap threshold, while route failures (RERR) events are usually not associated with video gaps at all. We also verified that in this scenario all video gaps are associated with a route request (RREQ) event since there is no competing traffic. Relative to the Gap threshold, it is equal to 7 packets as defined previously.

Proven this direct relationship between video gaps and route request events, we studied the behavior of the different SR



a)



b)

Fig. 3. Relationship between RREQs, RERRs and video gaps (a) and relative number of RREQs generated (b)

modes relative to DSR in terms of RREQs in order to provide a clear explanation of the results of table I and figure 2. We calculated the number of route requests generated by the video source at different speeds and found that all SR modes offer a reduction over 50% in generated RREQs (see figure 3). The mode with best overall performance (mode 1) shows an average reduction of 68% on the number of RREQs generated in relation to DSR, and so we will use it as a basis for the improvements performed in the following sections.

V. MULTIPATH ROUTING

In the previous section we have shown that video streams flowing in MANETs are prone to be degraded through packet loss bursts which cause gaps on communication. We presented some techniques to measure those gaps such as the VA parameter and the video gaps histogram and proved that using any of the SR modes proposed we can achieve improvements to this problem. We also showed that video gaps are intimately related to route discovery processes.

In this section we analyze how the use of simultaneous paths on data transmission effectively reduces the down-time

of multimedia flows, making the communication experience smoother and more pleasant to the user. The use of multipath routing can be extended to any type of data. In this proposal we concentrate on multimedia streams alone.

We consider that out-of-order delivery provoked by multipath routing is not a primary issue. In fact, in the packetization of the raw video stream into RTP packets - functionality integrated in the H.264 codec - sequence numbers are included in RTP header, which allows the receiver to reconstruct the sending order. Also, the video decoder can use sequence numbers to determine the proper location of a packet without necessarily decoding packets in sequence.

A. Traffic splitting strategy

Traffic splitting in the context of multipath routing refers to the technique of distributing the packets of a certain stream through different paths. Concerning the paths themselves, we can talk about their degree of disjointness and also make a distinction between link disjoint and node disjoint paths [7]. Node disjoint paths are those where none of the intermediate nodes are in common. Link disjoint paths are those where all links differ.

The optimal strategy in terms of traffic splitting would be using the shortest disjoint paths. In general, node disjoint paths are preferable since these achieve the best use in terms of both bandwidth and node resources. There are some cases where no node disjoint paths are available and, therefore, link disjoint paths are recommendable. In fact, the link disjointness condition is enough to reduce the effect of mobility on ad-hoc networks. The algorithm we propose for maximum disjointness is algorithm 1.

With algorithm 1 we are able to completely detach the route discovery and maintenance process from the packet splitting process. From now on, we shall refer to *Disjoint* solution as the one which makes use of this algorithm.

The action of finding the disjointness of one route is always done relatively to the previously used route. This technique allows to easily adapt to extra routes found through the forwarding or interception of routing packets, as well as to routes which were considered lost. To provide a good basis of comparison we define a metric that clearly evidences the exact gains in terms of path dispersion, that is, an average degree of path disjointness. So, the dispersion between two consecutive paths for a single stream would be:

$$Dispersion = 1 - \frac{CL}{NL},$$

Algorithm 1 Adaptive route disjointness algorithm

```

if (no path has been chosen previously) then
    choose the first shortest path;
else { find the shortest node disjoint path;
    if (not found) then
        find the shortest link disjoint path;
    if (not found) then
        find the shortest path with least common links;
    if (not found) then choose first shortest path; }

```

where CL is the number of common links relative to the previous path and NL is the number of links of the current path.

We aim at dispersion values near to 1, which is the optimal solution; dispersion values near 0 reflect a bad traffic dispersion policy.

B. Preventive route discovery

In order to improve the traffic splitting *Disjoint* strategy presented in the previous section, we propose a mechanism to perform preventive route discovery processes. Its objective is to minimize the video gap effects on the video quality delivered to the user.

By default, the DSR protocol initiates a route discovery process when it has no available routes to the destination. So, until a new valid route is discovered, our video flow transmission will lose a burst of packets, producing a video gap. We propose to perform preventive route discovery processes in order to avoid that situation. So, we think that it is reasonable to look for new routes when there are no disjoint paths available.

We also have to consider the possibility that, after completing a preventive route discovery cycle, no disjoint routes are found. In this case, we have to start another route discovery process to avoid video flow stall if the current route is lost. The rate at which we generate preventive route discovery processes must be evaluated in order to be useful and not to overload the network. By varying the preventive route discovery period among values: 0.5, 1, 2, 4, 8, 16 and never (default) we calculated the routing overhead differences among the various inter-request values, see figure 4. Remind that when all routes to the destination are lost, a new route discovery process is started and the probability of producing a video gap is high.

As expected, the routing overhead is higher than the default solution in all cases. The lower the inter-request interval, the higher the routing overhead becomes. We consider that for values under 4 seconds the routing overhead becomes prohibitive. In terms of packets arrivals, around 99% of packets arrive for all solutions under test at all speeds.

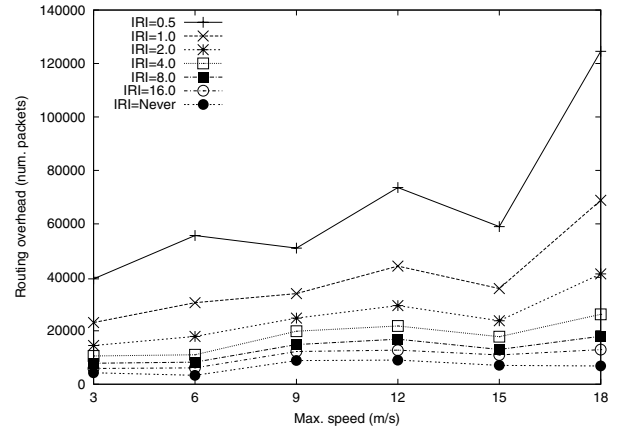


Fig. 4. Effect of inter-request interval on routing overhead

Considering the VA parameter, we achieve the best average results for an inter-request interval of 8 seconds (about 1/3 compared with the default solution), which also offers very good results in terms of packet arrivals and routing overhead. We will use this setting from now on.

C. Enhancements achieved using multipath routing

In this section we are going to perform a global evaluation of DSR, SR (mode 1) and Disjoint solutions, adjusting the best options found for them. The simulation setup is configured with a 1000x1000 meters scenario and 80 nodes. The mobility pattern is generated using the random waypoint mobility model bundled in ns-2 and an extra script was created in order to accept only scenarios where node topology forms a connected graph.

Only one video flow conforms the injected traffic with the same characteristics described in section IV. The Disjoint solution uses a multipath routing algorithm with a preventive route discovery mechanism set to minimum period of 8 seconds.

We can clearly see on the left side of figure 5 how DSR performs worse than the remaining modes for moderate/high speeds. The Disjoint mode is the best for all speeds, but the SR mode alone already provides very good enhancements.

In terms of routing overhead, we can see in figure 5 that the SR mode does not generate an excessive number of control packets. The Disjoint mode causes more routing overhead since this protocol is performing preventive route discoveries frequently. Also notice that the rate of growth between the three solutions is quite different. Comparing the routing overhead for the minimum and maximum speeds we can see that while DSR control packets have increased by a factor of 9, SR and Disjoint mode have increased by factors of 5 and 2 respectively. This shows the better adaptation and appropriateness of these two to high mobility scenarios. We also show the routing overhead achieved with the Disjoint mode when turning off the Preventive Route Discovery mechanism (No PRD in figure 5). We can see that the routing overhead is highly reduced when that mechanism is turned off, showing an overhead similar to the SR mode.

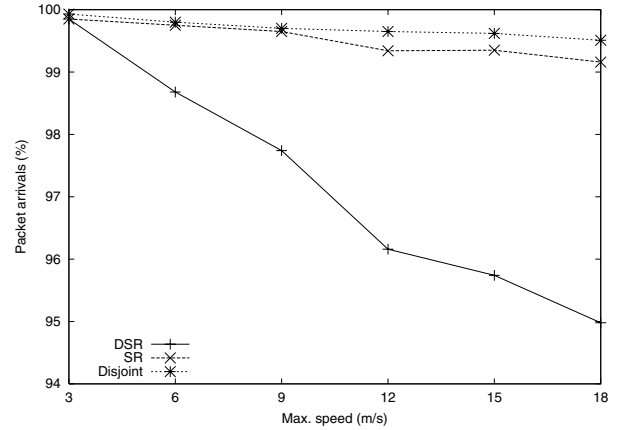
If we now focus on our main goal - reducing video streaming gaps - we find that there has been a gradual improvement as shown in figure 6. As it can be seen, after introducing the SR mode the video gap occurrence has been reduced significantly. With the Disjoint mode the improvements are even higher, keeping the size of most video gaps below 1.5 frames.

Table II shows the improvements in terms of global gap percentage and the VA parameter. The reason why the difference in terms of VA between SR and Disjoint modes is not greater is due to the inevitable occurrence of some long video gaps

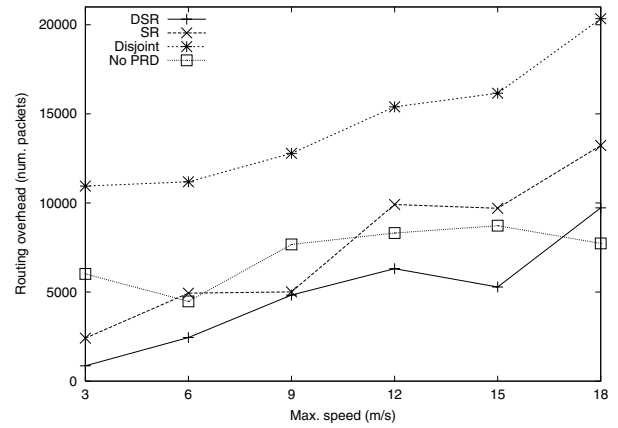
TABLE II

VA PARAMETER COMPARISON

Protocol version	DSR	SR	Disjoint
Gap (%)	2.41	0.303	0.0776
Avg. VA (10^{-7})	32.3	2.14	0.619



a)



b)

Fig. 5. Comparison in terms of video packet arrivals (a) and routing overhead (b)

which have more weight in the VA computation. For video gaps of less than 3 frames, though, the disjoint mode shows its effectiveness without doubt.

Finally, relative to the benefit of including or not the preventive route discovery mechanism, we achieve a slight increase in routing overhead and a reduction of 60% in terms of VA and of 50% in gap percentage by turning it on. The main reason for this difference are the few situations when preventive routing saves us of video gaps.

VI. VIDEO ENHANCEMENTS FOR MULTIPATH SCENARIOS

The techniques presented in the previous sections focus on enhancing the routing protocol to provide a better video service. In this section we present new enhancements to the Disjoint solution by focusing on an optimal adaptation of video flows. We start with a study on the impact of video-aware packet replication and proceed analyzing the need for good data distribution through different paths.

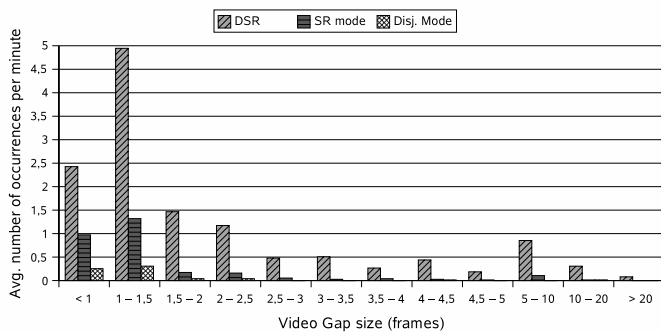


Fig. 6. Video gaps distributions for the 3 protocols under test at high mobility

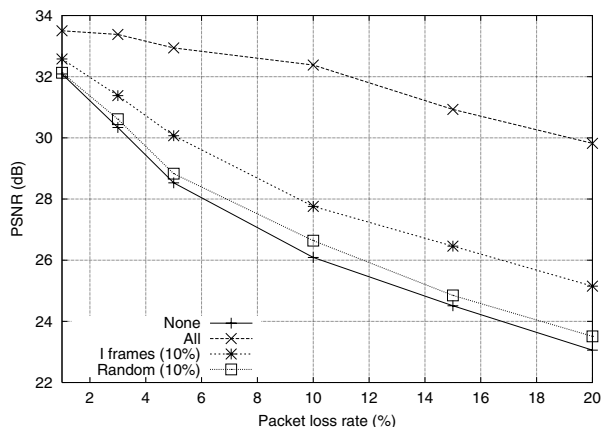


Fig. 7. Distortion values for different packet replication strategies.

A. Video-aware packet replication

Splitting traffic through two different routes towards a destination opens new possibilities for the enhancement of video streaming. A simple solution is to replicate all the information through both paths, so that if we have a packet loss probability p on each path the receiver sees it as if losses were occurring with a probability of p^2 . This way we achieve good improvements at twice the cost. If we analyze the video data organization we observe that not all the information is equally important. Intra coded frames, which are an essential source of reference for predicted frames, contain information which is more relevant than the rest. What we wish to achieve is, therefore, to improve the video distortion performance, but at the same time avoid an increase of 100% in the transmitted video data.

In our simulation analysis we created an H.264 video file of the well-known Foreman video sequence. We have configured the codec by tuning the error-resilience options according to the results we obtained in previous works such as [11] and [2]. So, in our video stream 10% of the frames are intra-coded, which means that an I frame is followed by nine P frames. The packetization of the video sequence and the encapsulation into RTP packets is performed by the H.264 software package. The initial video distortion under no loss is of 33,51 dB.

By simulating a statistically random loss pattern on both paths we analyzed the evolution of distortion by varying the packet loss ratio from 1 to 20%. The results presented in figure 7 are average values over 10 simulation runs.

The tag “None” indicates that no packet replication is being performed, which means that packets are just evenly split among the two paths. The “All” tag indicates a 100% packet replication. The remaining tags refer to the replication of 10% of the packets. In the experiment referred to as “Random” the replication process selects the 10% of packets in a random manner, while in the experiment tagged “I frames”, the replication mechanism replicates only those packets related to I frames.

From that figure we find that by replicating 10% of the packets randomly, the distortion increases by 0,5 dB relative to no-replication, so it almost does not justify the 10% increase in bandwidth. We do achieve considerable gains by performing an intelligent packet replication, though still maintaining the same bandwidth increase. In fact, the introduction of video awareness in the packet replication process allows to achieve up to 2,14 additional dBs for packet loss rates of 20%.

B. Video data distribution in multipath scenarios

We now focus on some aspects of video distribution in multipath scenarios. As exposed previously, packet splitting attenuates the effects of mobility by assuring that even when one path is lost, the other one remains available for data transmission. As soon as the loss is detected, the source stops sending packets through that invalid route. However, from the moment a broken link is detected to the moment the source receives the correspondent notification, a considerable amount of video packets could have been transmitted. Contrarily to random error scenarios, now the losses will affect well defined data - packets assigned to the invalid path. For example, if we split each frame in two packets and we route data through two disjoint paths, the loss of one path will always result in the loss of the upper or lower part of the frame, depending on the affected path. This situation will go on until the failure is detected.

To further examine this problem we use the same video sequence as before (Foreman), and simulate the loss of one of the routes. Our evaluation envisaged scenarios where the source splits packets through 2 and 3 disjoint paths for comparison.

The results presented in figure 8 refer to the average distortion perceived in the first 10 affected frames. Also, when using 2 disjoint paths we average the result by losing one path or another; when using 3 paths we also average the 3 distinct possibilities.

We also consider that the affected frames may be intra coded (I) or predicted (P).

From that figure we can see that the number of packets per frame affects the sequence’s distortion, with differences of up to 2 dB. Here we have two distinct factors acting simultaneously: (1) a higher granularity in packetization will cause information loss to be spread in a more uniform manner

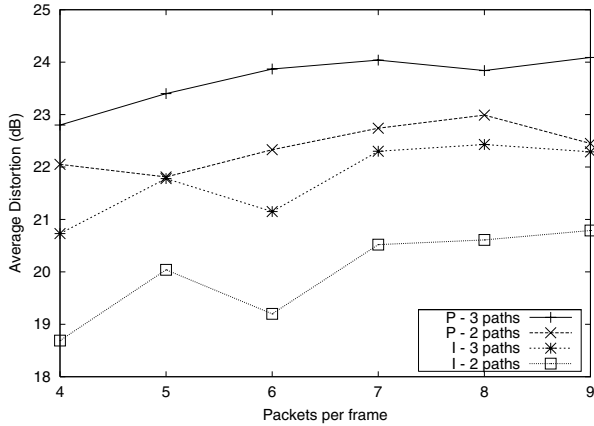


Fig. 8. Average distortion value for varying packetization granularity in the first 10 affected frames

throughout the frames. This effect helps in increasing the quality, but has the drawback that it does also increase the network traffic. And (2), there is a relation between the number of packets per frame and the number of routes used. So, when the number of packets per frame is a multiple of the number of paths, the lost information will always refer to the same frame areas.

As a consequence of these interactions, we can see that in terms of video distortion using 7/8 packets per frame is the best solution for both predicted and intra-coded frames.

Another aspect shown in figure 8 and that should also be taken into consideration is that I frames are more severely affected from losses than P frames. However, once the route is established an intra-coded sequence will recover to maximum quality instantly, while sequences that rely only on prediction will recover slower. Therefore, a good balance between intra coded and predicted frames on a sequence should be found in order to optimize simultaneously the usage of bandwidth and error propagation.

We tuned the sequence for a GOP size of 10 frames, that is, groups formed by an I frame followed by 9 P frames.

C. Video enhancements achieved

To conclude our analysis we apply the different tools and methods developed until now to determine global improvements. The scenario is the same one used in section IV, but now we add background traffic. 10% of the nodes are either sources or destinations. Each background source sends UDP traffic at a rate of four 512 byte packets per second.

Table III shows the results achieved with different protocol/replication combinations. In terms of packet arrivals for video data we see that the difference is minimum, except for DSR plus replication where the arrival percentage is slightly reduced. Concerning the use of the *Disjoint* mode with or without I frames replication, it causes a 2% increase in the background traffic, along with a considerable increase on the routing overhead as expected, though it can be reduced by

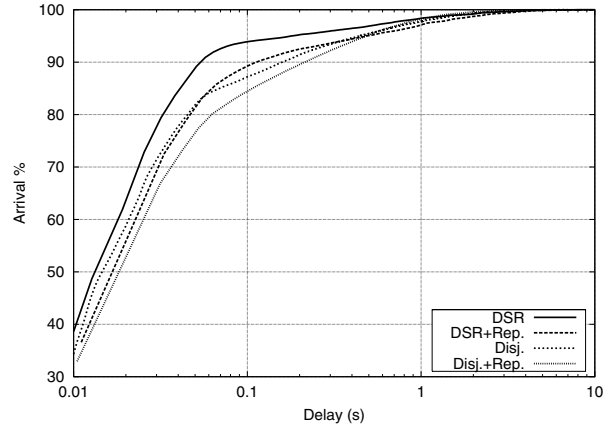


Fig. 9. Distribution function for video packet delays

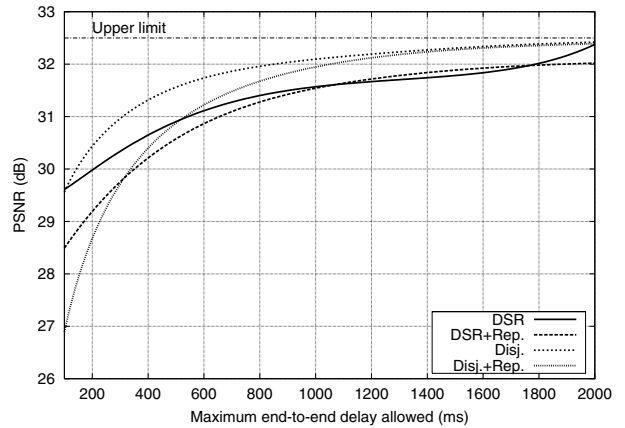


Fig. 10. PSNR variation for varying end-to-end delay thresholds

about 50% if the preventive route discovery mechanism is turned off.

In terms of end-to-end delay, we found that except for very high packet arrival values, DSR alone performs better than the remaining modes as shown in figure 9. This result is expected since both packet splitting and packet replication mechanisms are prone to cause additional delays to a video stream. It is also interesting to notice that using multipath, 96% of the packets arrive faster without replication, but the remaining 4% arrive faster with replication on.

In order to assess the effect of the different strategies under study on the video stream we evaluate the results in terms

TABLE III
SIMULATION RESULTS

Mode	Video arrivals (%)	Routing overhead
DSR	97.2	7011
DSR+Rep.	95.9	6438
Disj.	97.4	18380
Disj.+Rep.	97.6	18953

of average distortion for varying end-to-end delay thresholds. Such results are presented on figure 10.

Contrarily to what would be expected by observing figure 9, DSR alone never performs better than its multipath counterpart. Concerning *Disjoint* mode plus replication, we also notice that it behaves better than DSR for end-to-end delay thresholds above 500 ms. We should point out that the PSNR differences found are due to packet drops that are related to routing and delay thresholding, but are not related to congestion.

In this scenario it is not possible to observe the goodness of I frames replication obtained theoretically in section VI-A since the main cause for packet losses is the loss of paths, which are not frequent, randomly distributed packet losses. It is clear, though, how the 10% increase in video traffic affects the overall performance, which leads us to think that this option shall be optimal only in situations where packet loss is more severe and more randomly distributed.

VII. CONCLUSIONS AND FUTURE WORK

We presented several enhancements to the DSR protocol in order to provide a better support to real-time multimedia streaming, taking an H.264 video stream as a reference. We started by presenting the SR mode, an extension to DSR's route discovery method which is part of a previous work. Afterwards we evidenced the existence of video transmission gaps in wireless ad-hoc networks. We studied their causes, as well as the effect of route stability on reducing this problem. Concerning video streams, we also proposed an alternative metric to PSNR, called video annoyance, in order to measure video gaps in a clear and straightforward manner.

We presented additional solutions to the video gap problem through packet splitting procedures and performing preventive route discoveries. The first of these mechanisms aims at reducing small but frequent video interruptions, while the second aims at eliminating very large video interruptions caused by the loss of all routes to destination. We achieved reductions in video gaps of up to 97%.

After the multipath framework was set, we analyzed the interactions between the video codec and the packet splitting mechanism, tuning the former so as to obtain the best performance.

We concluded our work with an overall evaluation of the improvements achieved, showing that there are improvements in terms of video distortion values for different delay thresholds, where the gains can surpass 1.2 dB.

As future work we intend to analyze the behavior of the mechanisms proposed in this paper using group mobility for different group mobility patterns, as well as their integration in IEEE 802.11e enabled MANETs.

REFERENCES

- [1] "Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264 — ISO/IEC 14496-10 AVC)," March 2003.
- [2] Carlos T. Calafate, M. P. Malumbres, and 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.

- [3] Lei Wang, Yantai Shu, Miao Dong, Lianfang Zhang, and O.W.W. Yang, "Adaptive multipath source routing in ad hoc networks," ICC 2001. IEEE International Conference on Communications; Page(s): 867-871 vol.3, 2001.
- [4] Asis Nasipuri, Robert Castaneda, and Samir R. Das, "Performance of multipath routing for on-demand protocols in mobile ad hoc networks," *ACM/Baltzer Mobile Networks and Applications (MONET) Journal*, vol. 6, pp. 339-349, 2001.
- [5] J. Wu, "An Extended Dynamic Source Routing Scheme in Ad Hoc Wireless Networks," 35th Annual Hawaii International Conference on System Sciences (HICSS'02)-Volume 9, Big Island, Hawaii, January 2002.
- [6] S. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," In Proceedings of the IEEE ICC, pages 3201-3205, 2001.
- [7] M. Marina and S. Das, "On demand multipath distance vector routing in ad hoc networks," In Proceedings of IEEE International Conference on Network Protocols (ICNP), pages 14-23, 2001.
- [8] Zhenqiang Ye, Srikanth V. Krishnamurthy, and Satish K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," in *IEEE INFOCOM 2003*, San Francisco, USA, April 2003.
- [9] Carlos T. Calafate, M. P. Malumbres, and Pietro Manzoni, "A flexible and tunable route discovery mechanism for on-demand protocols," in *12-th Euromicro Conference on Parallel, Distributed and Network based Processing*, La Coruña, Spain, February 2004.
- [10] K. Fall and K. Varadhan, "ns notes and documents." The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC, February 2000.
- [11] Carlos T. Calafate and M. P. Malumbres, "Testing the H.264 Error-Resilience on Wireless Ad-hoc Networks," in *4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications*, Zagreb, Croatia, July 2003.