

ROUTE STABILITY TECHNIQUES FOR ENHANCED VIDEO DELIVERY ON MANETS

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

Department of Computer Engineering

Polytechnic University of Valencia (UPV)

Camino de Vera S/N, 46022 Valencia, Spain

{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. In this work we argue that increased bandwidth and QoS strategies do not solve problems associated with mobility. 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 achieve improvements in terms of goodput, and more important, we reduce the video gap durations up to 97% for high mobility scenarios, improving the user viewing experience dramatically.

Keywords: Multipath routing, video streaming gaps

1. Introduction

Mobile Ad-hoc networks (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 disaster relief situations to small home environments. However, the current performance of MANETs can hardly be accepted for real-time multimedia.

The IEEE 802.11 [IEEE 802.11, 1999] technology is the most deployed and used. The IEEE 802.11a/g standards enable MANETs to use more bandwidth for multimedia applications, allowing various simultaneous video flows per

*This 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.

node to exist. The medium access issues concerning QoS, included in the IEEE 802.11e working group, are still not standardized but are of extreme importance to provide service differentiation.

The purpose of this work is to present the problems that still persist after applying QoS techniques at the MAC level. We found [Carlos T. Calafate et al., 2004b] 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 not optimal due to mobility.

We show that there is a close relationship between video gaps and route discovery events. We then describe a traffic splitting approach based on multipath, which uses a route selection mechanism that optimizes the use of 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 possible route losses we supply a preventive route discovery mechanism activated when a video flow has not at least two disjoint routes available.

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 2 we give a brief description of the related work in this area. In section 3 we propose enhancements to DSR’s route discovery technique. Section 4 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 5 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 6 we perform a global evaluation of all mechanisms presented in this paper. Finally, in section 7 we make some concluding remarks, along with some references to future work.

2. Related work

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

[L. Wang et al., 2001] use a probing technique to assess the quality of available routes, so that the traffic is forwarded based on the delay of each route. Their objective is also to achieve a fair load distribution as well as improved throughput, end-to-end delay and queue utilization. [Nasipuri et al., 2001] proposed a strategy for quick route recovery through packet re-direction in intermediate nodes to reduce the frequency of query floods.

[Wu, 2002] 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.

[Lee and Gerla, 2001] show that the paths found by DSR's route discovery mechanism are mainly overlapped. They therefore propose a route discovery technique that provides nodes disjoint paths.

In [Marina and Das, 2001] the AODV protocol [Perkins and Royer, 1999] 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.

3. Route discovery extensions to DSR

The Dynamic Source Routing (DSR) protocol is an efficient routing protocol for MANETs. DSR, by default, finds only a small number of routes. By extending its route discovery mechanism we 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 less route discovery processes. Lee and Gerla proposed in [Lee and Gerla, 2001] a route discovery technique based on altering the route discovery process. So, 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".

With our route discovery proposal we significantly reduce the routing overhead when compared to Lee and Gerla's proposal (see [Carlos T. Calafate et al., 2004a]). 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 some bytes 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 assure 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 [Carlos T. Calafate et al., 2004a] we demonstrated that only small values for this parameter, though, make sense in terms of route size and stability.

In the following sections we test three different *Flexibility / PNC* pairs. 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.

4. Effects of route stability on real-time video streams

In this section we show how the use of a standard routing protocol, such as DSR, can affect a real-time video stream in terms of video communication disruptions (video gaps). We will prove that video gaps are intimately related to route discovery procedures, and how the SR mode presented in Section 3 can considerably alleviate this problem. The evaluations are done using the ns-2 simulator [K. Fall and K. Varadhan, 2000]. Each obtained value is averaged 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 unreachable destinations), in order to obtain a connected graph.

We first evaluate a $1000 \times 1000 m$ squared scenario with 80 nodes. The traffic load consists of a single H.264 [H.264, 2003] 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.

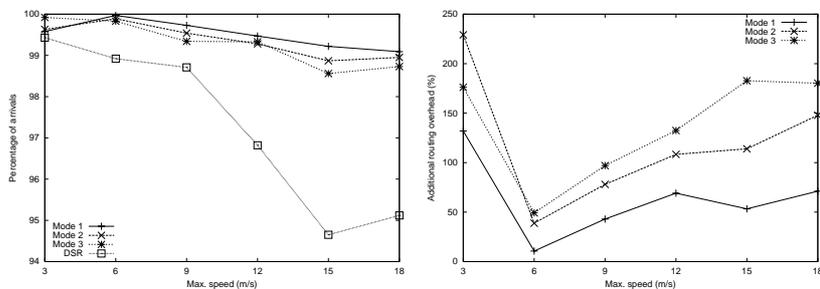


Figure 1. Packet arrivals and routing overhead when streaming an H.264 video flow

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 mode 1, where the improvements over DSR reach 4.5% in packet arrivals. Mode 1 also generates less routing control packets than the other two modes, with a small relative increase compared to DSR.

Loss pattern analysis

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 concealment 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 peak signal-to-noise ratio (PSNR) between receiving 99% or 95% of the packets is very slight.

However, packets dropped in bursts long enough to cause video gaps affect H.264 video decoders in a drastically different manner. When no information relative to consecutive frames arrives to the decoder, this 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:

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

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 ($B_i \geq 7$). When $VA = 0$, there are no video frame gaps, though losses can occur. When $VA = 1$ the entire sequence was 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, a 5 seconds interruption at 10Hz, is quite disturbing for the user.

Analyzing a typical packet drop pattern on a simulation using DSR and a single H.264 video flow, we can observe that some of the packet loss events

are bursty. Bursts of packets lost cause the video flow to be stopped, so that several entire frames are lost.

Table 1 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. This result proves how the different SR modes improve the video experience in terms of video disruptions.

Table 1. Video annoyance statistics

Protocol	VA (10^{-6})	VA % towards DSR
DSR	38,1	-
Mode 1	0,502	1,3169
Mode 2	0,708	1,8575
Mode 3	0,768	2,0167

To further analyze and validate the improvements shown with the VA parameter, we now consider the video gap histogram for all protocols at maximum node speed (see figure 2). We observe that DSR performs much worse than any of the SR modes for all gap intervals, being mode 1 the one that achieves the best results.

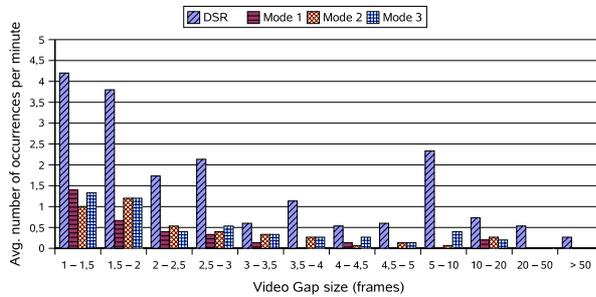


Figure 2. Video frame gaps histogram at a maximum node speed of 18 m/s

Another important factor shown in figure 2 is that SR modes 1 to 3 present gap sizes of no more than 20 frames, contrarily to DSR. In fact, DSR is prone to lose as much as 217 consecutive frames, equivalent to more than 20 seconds of video interruption at 10Hz, while in mode 1 the maximum gap size experienced is of only 13 frames (1.3 seconds of video interruption at 10Hz).

Gap causes and solutions

In scenarios like the one under evaluation, where congestion is not a problem, packet losses directly depend on link breaks and subsequent route fail-

ures. Since DSR uses link layer information to detect broken links, the interval between the detection of a broken link and the reception of the 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:

$$N = 2 \cdot S_{rate} \cdot T_{REERR}, \quad (2)$$

where S_{rate} is the source's packet generation rate and T_{REERR} 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.

To clarify the results of table 1 and figure 2 we compared the behavior of the different SR modes with the behavior of DSR in terms of RREQs. We calculated the number of route requests generated by the video source at different speeds (see figure 3.) The results are presented using the values obtained for the DSR as a reference. We observe that modes 1 to 3 present less video gaps for all speeds as expected due to the higher number of routes found. The relationship between the number of RREQs and video gaps also explains the improvements in terms of VA. 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.

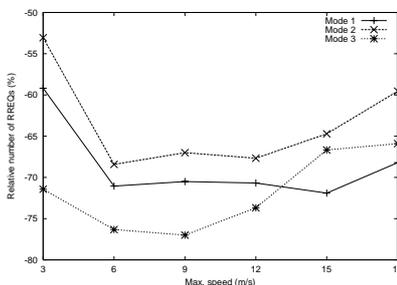


Figure 3. The behavior of the different SR modes towards DSR in terms of RREQs

5. Multipath routing

In this section we will present further improvements by analyzing how the use of simultaneous paths on data transmission effectively reduces the downtime of H.264 video flows, making the communication experience smoother and more pleasant to the user. We should point out that the sequence numbers included in RTP headers allow 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. Therefore, we consider that out-of-order delivery provoked by multipath routing is not a primary issue.

Traffic splitting strategies

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 [Marina and Das, 2001]. Node disjoint paths are those where none of the intermediate nodes are in common. Link disjoint paths are those where all links differ, though common nodes may exist.

An optimal strategy in terms of traffic splitting would be one where the shortest disjoint path is used. In general, node disjoint paths are preferable since they achieve the best trade-off 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 used. In fact, the link disjointness condition is enough to reduce the effect of mobility on ad-hoc networks.

We define a metric, see Equation 3, that demonstrates the exact gains in terms of path dispersion using the average degree of path disjointness.

$$Dispersion = 1 - \frac{CL}{NL}, \quad (3)$$

where CL is the number of common links relative to the previous path, and NL is the number of links of the current path. The objective is to obtain values close to 1, which is the optimal solution; dispersion values close to 0 reflect a bad traffic dispersion policy.

We propose an algorithm (see Algorithm 1), which describes how to get the maximum disjointness path. We shall use the term “*Disjoint solution*” to the mechanism that makes use of this algorithm.

Algorithm 1 Maximum 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; }
```

The action of finding the disjointness of one route is always done relatively to the previously used route. This technique easily adapts to extra routes found through the forwarding or interception of routing packets, as well as to routes which were considered lost. Though this algorithm aims at finding the best choices in each situation, it could be considered computationally expensive for small embedded systems. Therefore we also propose a much simpler solution which consists of randomly choosing routes which are not larger than a certain size (s) relative to the first one. This alternative solution, referred as R_s , is used as a reference for the Disjoint solution.

Using the same simulation setup that in section 4, we evaluated the Disjoint algorithm comparing it with the R_s solution. Table 2 shows the average results by setting $s = 2$ and setting the maximum allowed node speed to 18 m/s. As it can be seen, the maximally disjoint solution always achieves the best results.

Table 2. Comparison of traffic splitting strategies

Mode	Video arrivals(%)	Routing overhead	End-to-end delay (ms)	Dispersion
Disjoint	99,70	6759	39,54	0,71
R_2	97,60	11346	51,19	0,32

In terms of end-to-end delay, the Disjoint solution always performs quite better than the R_2 one, which means that the paths used are shorter. In what refers to the routing overhead, again the Disjoint solution performs much better.

If we observe the results concerning the dispersion achieved with both methods, we verify that the Disjoint solution presents dispersion values that more than double the ones from solution R_2 . We also verify that the dispersion value almost does not vary with speed.

This analysis allows us to conclude that the results achieved justify the extra computational complexity required by the Disjoint solution, being the R_2 a possible solution for environments with few resources, though the performance suffers some degradation. From now on we will always use the Disjoint solution when performing traffic splitting.

Preventive route discovery

In order to improve the traffic splitting *Disjoint* strategy presented in 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.

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 seconds 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

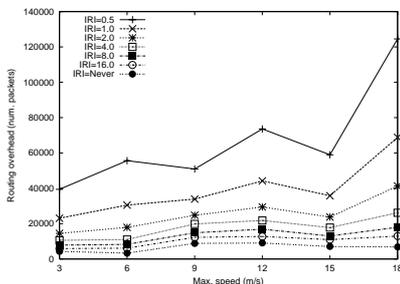


Figure 4. Effect of inter-request interval on routing overhead

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. Considering the VA parameter, we achieve the best average results for an inter-request interval of 8 seconds, which also offers very good results in terms of packet arrivals and routing overhead.

6. Overall evaluation

In this section we perform a global evaluation of the SR and Disjoint solutions. The simulation setup is configured with a 1000x1000 meters scenario and 80 nodes. As before, the mobility pattern is generated using the random waypoint mobility model and we consider only scenarios where node topology forms a connected graph. Only one video flow conforms the injected traffic with the same characteristics described in section 4. 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 can provide very good enhancements.

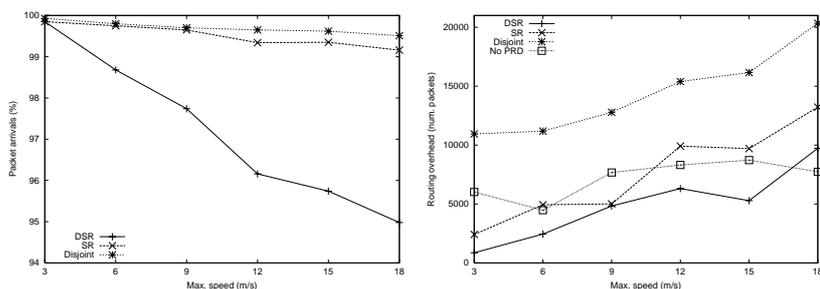


Figure 5. Comparison in terms of video packet arrivals (left) and routing overhead (right)

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 dis-

coveries 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, and that the SR mode plus the multipath Disjoint solution are able to significantly reduce the video gap occurrence. Table 3 shows the improvements in terms of global gap percentage and the VA parameter. The difference in terms of VA between SR and Disjoint modes is not greater due to the fact that both approaches avoid large video gaps, being large video gaps those that provoke significant differences in terms of VA metric. For video gaps of less than 3 frames, though, we find that the disjoint mode is very effective.

Table 3. 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

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 slight difference are the few situations when preventive routing saves us of video gaps.

7. Summary

We presented several enhancements to the DSR protocol in order to provide a better support to H.264 video stream delivery. The proposals focused on the route discovery process, the packet splitting strategy, and the preventive route discovery process.

We showed that video gaps are intimately related to route discovery procedures and that this problem can not be solved through conventional QoS mechanisms. We also proposed an alternative metric to PSNR, called video annoyance, in order to measure video gaps in a clear and straightforward manner. We extended DSR's route discovery mechanism 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 less route discovery processes. We introduced a dynamic algorithm for maximizing the degree of disjointness of consecutive paths for a same stream, and evidenced the goodness of the algorithm against a more relaxed solution.

By comparing the standard DSR protocol against our proposals we showed that the enhancements lead to a very significant reduction of video gaps. We also verified that the routing overhead is maintained low, even when applying packet splitting and performing preventive route discovery, showing that the effectiveness of the presented strategy does not come at the cost of too many additional control packets.

References

- (1999). International Standard for Information Technology - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE 802.11 WG, ref. no. ISO/IEC 8802-11:1999(E) IEEE Std. 802.11.
- (2003). Draft ITU-T Rec. and Final Draft Int. Standard of Joint Video Specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC).
- Nasipuri, A., Castaneda, R., and Das, S. R. (2001). Performance of multipath routing for on-demand protocols in mobile ad hoc networks. *ACM/Baltzer Mobile Networks and Applications (MONET) Journal*, vol. 6, pages 339–349.
- Carlos T. Calafate, M. P. Malumbres, and P. Manzoni (2004a). A flexible and tunable route discovery mechanism for on-demand protocols. *12-th Euromicro Conf. on Parallel, Dist. and Network based Proc.*, La Coruña, Spain.
- Carlos T. Calafate, M. P. Malumbres, and Pietro Manzoni (2004b). 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.
- Johnson, David B and Maltz, David A (1996). Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers.
- K. Fall and K. Varadhan (2000). ns notes and documents. The VINT Project. UC Berkeley, LBL, USC/ISI, and Xerox PARC.
- Lee, S. and Gerla, M. (2001). Split multipath routing with maximally disjoint paths in ad hoc networks. *IEEE ICC*, pages 3201–3205.
- L. Wang, Y. Shu, M. Dong, L. Zhang, and O.W.W. Yang (2001). Adaptive multipath source routing in ad hoc networks. *ICC 2001*. Page(s): 867-871
- Marina, M. and Das, S. (2001). On demand multipath distance vector routing in ad hoc networks. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pages 14–23.
- Perkins, Charles E. and Royer, Elizabeth M. (1999). Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA*, pages 90–100.
- Wu, Jie (2002). 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.