A flexible and tunable route discovery mechanism for on-demand protocols

Carlos T. Calafate, Manuel P. Malumbres, Pietro Manzoni
Department of Computer Engineering
Polytechnic University of Valencia (UPV), Valencia, Spain
{calafate, mperez, pmanzoni}@disca.upv.es

Abstract

Mobile ad-hoc networks are characterized by multi-hop wireless links and absence of cellular infrastructure. For the last years, routing has been the area under most intensive research. In particular, on-demand routing protocols proposed are extremely attractive due to their low-overhead and efficiency.

Currently, wireless ad-hoc networks lack multi-path routing protocols. Multi-path routing enables increased QoS support, load balancing and enhanced route stability. In this paper we present a new route discovery solution that increases the number of routes found, while reducing the routing overhead when compared to previous proposals. Simulation results show that the additional routing overhead imposed is low, and also some improvements are achieved in terms of goodput for both TCP and UDP traffic at average node speeds.

1. Introduction

Mobile Ad-hoc networks, also known as MANETs, are composed by a set of independent mobile nodes which interact among themselves in an intelligent way, making any kind of infrastructure unnecessary for reliable communication.

The low cost and ease of deployment of this kind of networks, as well as the possibility of integration with wired networks makes them extremely attractive, justifying the increasing interest that these networks have been gaining in the last few years.

Currently, the 802.11b technology [1] is the most popular, achieving data rates of 11 Mbps. The new standard 802.11g provides higher bandwidths and, along with 802.11e (QoS) which is expected to be standardized soon, form a good base to support multimedia traffic.

In a previous work [2] we have done a performance analysis of H.264 [3] video streams in MANETs. From this study we observed that mobility alone can cause severe degradation on the performance of video streams. The degradation is associated with a very well defined loss pattern: long bursts of dropped packets. The size of these bursts is typically related with the re-routing time for each protocol, and can reach values of seconds. Our analysis as well as theoretical study show that protocols that use the information from the lower layer (level 2) are more effective performing re-routing tasks, since the detection of broken links is done earlier. However, even when this interaction is available, the time between route error advertising and the end of a route discovery cycle can be prohibitive for multimedia applications.

The purpose of this work is to find a solution that can serve as a basis for multi-path based routing and considers the requirements of multimedia traffic.

DSR [4] uses source routing, which allows traffic splitting through different routes in a straightforward way avoiding routing loops. We propose extensions to the DSR protocol in order to improve its route discovery procedures in terms of the total number of routes found. The technique proposed in this paper could be used with other reactive protocols such as AODV, since the route discovery mechanism is very similar. We focus on the route discovery technique itself and on the effect of having more available routes on node cache in typical scenarios.

Concerning the structure of this paper, in section 2 we present some related works, explaining their purpose and how they differ from our proposal. In section 3 we make a brief introduction to the DSR protocol and in section 4 we perform a detailed analysis of different route discovery techniques. Section 5 presents simulation results in a typical MANET scenario and finally in section 6 we do some concluding remarks, along with some references towards future work.

2. Related work

In the past there have been several approaches in the literature related to the discovery and use of multiple routes in MANETs.

In the work of Wang et al. [5] 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 to achieve load distribution as well as improved throughput, end-to-end delay and queue utilization.

In [6] 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.

Neither of this two works propose enhancements to the route discovery technique itself.

Nasipuri et al. [7] have proposed a strategy for quick route recovery through packet re-direction in intermediate nodes in order 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 unaware of any extra routes, which means that their solution does not aid in the task of splitting traffic through disjoint routes.

Wu [8] 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 the work of Lee and Gerla [9] the traffic is evenly split among the two first routes found in order to achieve load distribution; they analyze the options of starting a new route discovery process when one of the routes is lost or only when both are lost. The authors find that the standard route discovery mechanism of DSR not only returns few routes, but also that these routes are mainly overlapped (not disjoint). To solve this problem they enhance the route discovery mechanism of DSR to find more node disjoint paths; in this paper their route discovery solution is adopted under the name LG. In the LG route discovery solution, the route discovery process is altered 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". It is up to the source to analyze the disjointness of the paths found. In this work we tried to follow their route discovery proposal strictly in order to evaluate the performance of our own proposal.

3. DSR

The Dynamic Source Routing (DSR) protocol [4] is a high performance protocol for MANETs. It is currently under standardization by the IETF working group MANet. 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 network traffic the network 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. Each node forwards only the first packet it receives for a certain route request ID. Therefore, the number of routes found is typically small. When relying on 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; all unicast packets transmitted using 802.11 technology are acknowledged, so communication is more reliable. If no route reply arrives to the source before the established timeout then a new route discovery procedure is launched. DSR allows a packet to be queued for transmission up to 30 seconds.

The destination of a route request can send a reply only to the first RREQ packet or to all of them. Replying only to the first one allows to reduce the routing overhead, but when that route is invalidated a new route request cycle has to be initiated.

Once a route is found, packets are sent with the entire route on their IP header. This method, though provoking a small increase in bandwidth, has other advantages such as avoiding routing loops in a simple and efficient manner.

DSR uses the information from lower levels in order to detect broken links. This method allows it to react very quickly to link failures, so that packets on transit using that link are salvaged, being forwarded through other paths. The node which detects the broken link sends a route error message (RERR) back to the source. The source, as well as the rest of the nodes through which the RERR packet passes, removes from its cache all routes using that link.

DSR can also be set for promiscuous reception, so that nodes are able to learn new routes by listening to their neighbor’s packets.

4. Multiple route discovery extensions

DSR is a protocol that, by default, finds only a small number of routes; it does so with a very small routing overhead, and also rather quickly. By extending the route discovery mechanism in DSR we are able to increase the average number of routes found per node. This extra information increase the route choices at nodes when a route is lost, so fewer route discovery processes would be required. As an enhancement, nodes can use the extra information for other useful purposes, like packet splitting or replication over disjoint routes, route congestion analysis, QoS routing, etc.

Obviously, the success of these techniques depends on having good routes to forward packets and, at the same time, keeping a low routing overhead. When referring to good routes we mean disjoint routes, either node or link disjoint, so that when mobility causes a link break between two nodes, an alternate route that does not rely on that link is available.

Node disjoint paths are those which have only the first and last nodes in common; link disjoint paths are those where all links differ, though nodes in common may exist.

In this paper we present a strategy called "Super Restrictive mode" which, starting from the basic DSR approach, enhances it in order to allow more routes to be found. We compare it with the LG solution in order to assess its effectiveness. In addition to the SR mode, we also propose the RLG (Relaxed LG) solution in order compare less restrictive approaches than LG so as to find more useful routes between two nodes.

4.1. Relaxed LG (RLG) route discovery solution

The RLG solution we propose here is similar to the LG, except that the restriction concerning the last hop through which the first message arrived is no longer used. This means that nodes will propagate RREQs packets with the same route request ID even if they arrive through the same input link. The route size restriction, though, is maintained.

To better explain the difference between DSR, LG and RLG route discovery techniques we will now expose what would happen in a scenario such as the one presented in figure 1.

During a route discovery process using the standard DSR a RREQ packet would arrive to node 3 through both sub-paths *a* and *b*. Only the first to arrive would propagate to the receiver through both sub-paths *c* and *d*. As can be seen, these routes are not link disjoint, though link disjoint routes do exist.

The propagation technique used in LG allows node 3 to propagate both RREQ packets arriving through sub-paths *a* and *b*. However, only the first one to be forwarded will arrive to D through both c and d sub-paths; the second one is

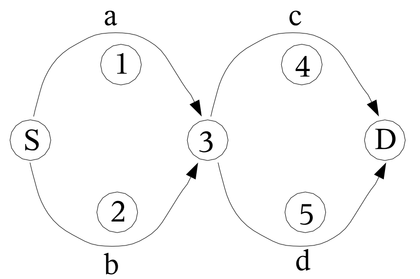

Figure 1. Multiple paths applying the RLG solution.

retained at nodes 4 and 5 because the previous node on the path (node 3) is the same.

Moreover, the weakening of restrictions in the RLG solution allows up to 4 possible routes to be found, which are {S-1-3-4-D}, {S-2-3-4-D}, {S-1-3-5-D} and {S-2-3-5-D}. As it can be observed, this solution is less restrictive that the LG solution, which means that routing overhead shall be increased accordingly.

4.2. Super Restrictive mode (SR)

The solution we propose in this work shall be referred as "Super Restrictive" mode. The required enhancement to DSR is based on a new data structure added to the already existing route discovery table structure on all nodes. So each node is able to keep track of all the last hops whose route request was forwarded through. The LG solution presented previously is simpler, keeping track of just the first node through which a RREQ packet arrived.

The main enhancement of the SR mode is that it discontinues the propagation of a route request arriving through a repeated last hop. The reason behind this choice is that when the last hop is the same than that of a previous request, there will be at least one link in common between the routes. Therefore, the usefulness of these extra routes will be reduced. With the SR mode we expect to decrease the routing overhead significantly when compared to the LG solution.

The propagation process of the "Super Restrictive" mode will be further enhanced and tuned in section 5.

4.3. Simulation setup and results

In this work we used the ns-2 simulator [10] since it models collisions occurring at the physical level, which we believe to be very important. Otherwise, the route request process would not suffer performance degradation from this, and would find all the best routes to a certain destination without problems, which is not accurate. Most analytical models also fail to model this reality, which explains why analytical results often differ from simulation results for MANETs.

In order to perform an initial evaluation of the different propagation strategies we generated a no-movement scenario with dimensions 1500x300; 40 nodes are positioned randomly. The no-movement situation aims at measuring the total number of routing packets generated in a controlled environment. Section 5.4 will present further results varying mobility and traffic conditions.

Between 0 and 20 seconds, half of the nodes start a route discovery process towards a destination node belonging to the other half. This is achieved by creating a client application that generates only a single small udp packet, which as a consequence starts a route discovery process.

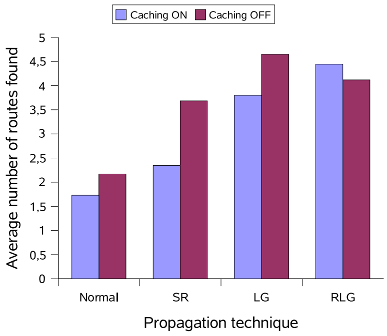

Figure 2. Average number of routes found

Concerning the use of cache in DSR, we performed tests with both cache ON and OFF. Turning the cache ON means that intermediate nodes can reply to RREQ packets instead of the destination; turning it OFF means that no node except the destination itself can reply, so that RREQ packets are propagated all the way.

In figure 2 we present the average number of routes found with the different propagation techniques under test. As can be seen, our purpose was met correctly since the SR mode allows a node to obtain a number of routes which is higher than that achieved with the standard DSR implementation, though lower that that achieved with the LG and RLG techniques.

Contrarily to what was expected through theory, the RLG solution does not always result in a superior number of routes found. To understand the cause of this phenomena we show figure 3, where the RLG solution, specially with route caching OFF, generates too much routing overhead. Consequently, the packet loss rate for routing traffic is also quite high.

Concerning the LG solution, we see that it does not perform very well compared to the original and SR mode versions. In fact, it generates more that twice the overhead with cache ON and more than five times with cache OFF.

On tables 1 and 2 we detail the reasons of packet losses. As it can be seen, the RLG solution clearly generates too many control packets, and so its use is not recommended. This fact can be explained by looking at the number of packets lost on the interface queue. While the normal DSR and DSR in SR mode practically do not appreciate losses on the queue, the LG and RLG solutions do so.

This fact again points to a network overloading, especially when the cache is OFF. The high number of packets lost at physical level (MAC Collision) gives a measure of the magnitude of the broadcast storm generated.

To clearly notice the duration of the broadcast storm generated during a route request cycle, we present at figure 4 the average time it takes to complete.

As it can be seen, the normal DSR and the SR solution present acceptable packet loss rate values, while the LG and RLG solutions present much higher values. In fact, except for the LG technique with caching ON, the rest of the values are too high, being therefore prohibitive.

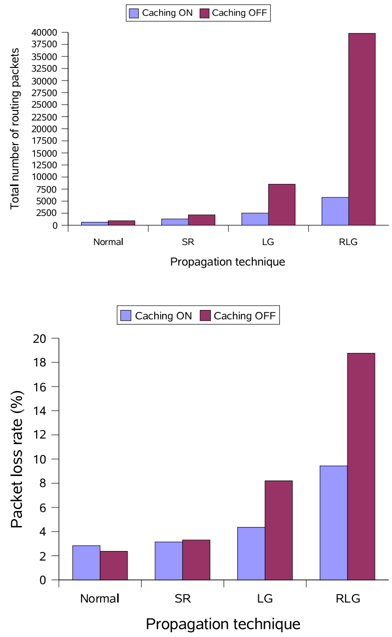

Figure 3. Routing overhead and packet loss rate for different propagation techniques

5. Tuning of the Super Restrictive mode

In the previous section we presented a new proposal for route discovery enhancements which we called Super Restrictive (SR) mode.

In this section we propose new enhancements to the SR mode by restricting even more the propagation process or by increasing its flexibility. Both proposals are independent and can be used together.

Packet loss reason	Normal	SR	LG	RLG
Address resolution	13,3	36,8	92,2	297,6
MAC Collision	2,0	3,0	10,7	19,9
Queue full	0,0	0,0	5,9	225,7
Other	2,8	0,9	0,8	1,2
Total	18,1	40,7	109,6	544,4

Table 1. Packet loss reason (Cache ON)

Packet loss reason	Normal	SR	LG	RLG
Address resolution	17,7	64,0	214,4	510,5
MAC Collision	2,6	6,2	30,7	157,5
Queue full	0,0	0,3	452,1	6751,3
Other	1,6	0,6	1,0	46,9
Total	21,9	71,1	698,2	7466,2

Table 2. Packet loss reason (Cache OFF)

5.1. Increasing restrictions

As described previously, the super restrictive mode maintains a list with all the last nodes through which a route request arrived. However, as long as the last hops differ, there is no limit to the size of this list.

We now propose an extension to this method by restricting the list to a certain size. Now, when a route request arrives it is not propagated if the list is already full; this means that only a pre-defined number of route requests are forwarded. If the list size is very high we obtain the original super restrictive mode; if it is reduced to one the behavior is similar to the default DSR propagation.

From now on this parameter will be be referred as PNC (Previous Node Count).

5.2. Increasing flexibility

All the solutions presented until now restrict the route request forwarding process to route sizes not superior to the first one arriving. We propose an increase of flexibility by allowing routes with an additional number of hops up the a certain value (the flexibility parameter) to be also forwarded. This technique allows alternate routes to be found in scenarios similar to the one presented on figure 5. As it can be seen, if flexibility is increased to 1 the source is able to find two link disjoint routes to the destination, which are: {S,1,4,5,R} and {S,2,3,4,6,R} or also {S,1,4,6,R} and {S,2,3,4,5,R}. Otherwise, it would just find two non-disjoint routes: {S,1,4,5,R} and {S,1,4,6,R}.

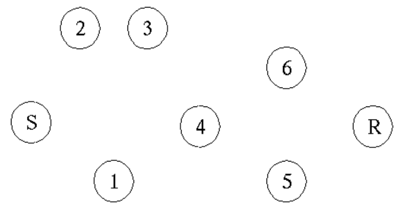

Figure 5. Scenario that benefits from increased flexibility on route discovery

In general, the increase of flexibility can have two different effects in terms of route propagation. On one hand it allows more routes to be found, so the routing overhead should increase. However, if the number of requests that each node is going to propagate is small (as with PNC=2), then it can result in a reduction of the route request cycle time, especially for dense scenarios.

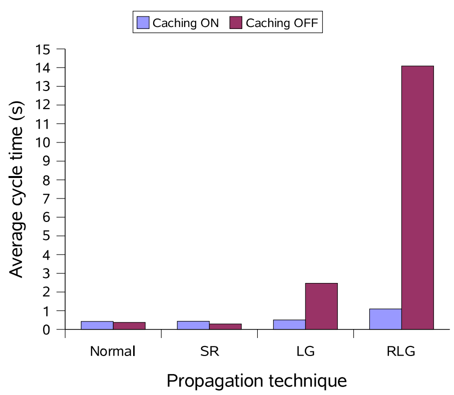

Figure 4. Average time for route discovery cycle

5.3. Simulation results

We now present the results relative to the new enhancements on the super restrictive mode. The simulation used the same 1500x300 no-mobility scenario with 40 nodes used in section 4.3.

We tested several *<Flexibility, PNC>* pairs in order to obtain a general view on the effects of each of these parameters.

Figure 6 shows the average number of routes found varying flexibility for different values of Previous Node Count (PNC). As it can be seen, all solutions allow more routes to be found compared with the original DSR. When the cache is OFF the average number of routes is also increased. Fig. 6 also shows that though increasing PNC does usually translate into more routes found, increasing the flexibility parameter may reduce the number of routes found; explanations to this behavior have been presented on subsection 5.2.

In terms of routing overhead, we can see from figure 7 that the extra routes found have a cost as expected. That figure evidences that when increasing flexibility from 0 to 1 there is a considerable increase on the routing overhead, which remains more or less stable afterwards.

Concerning packet losses, the packet loss rate is under 4% in all simulations that we executed, which is an acceptable value taking into account that there is no way to avoid collisions for broadcast packets with the 802.11b technology.

To end this evaluation of the SR mode we also analyzed the average cycle time, that is, the time from the moment a

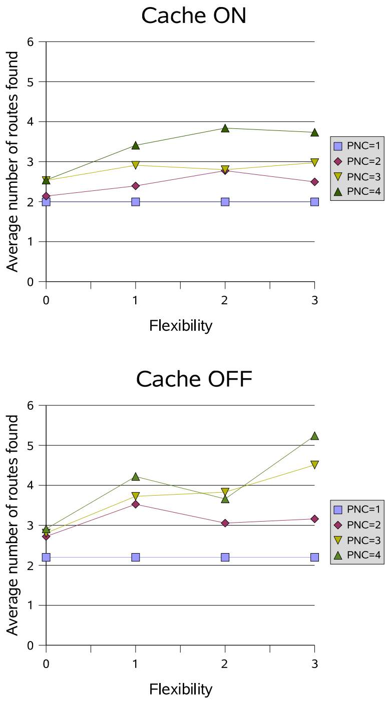

Figure 6. Average number of routes found

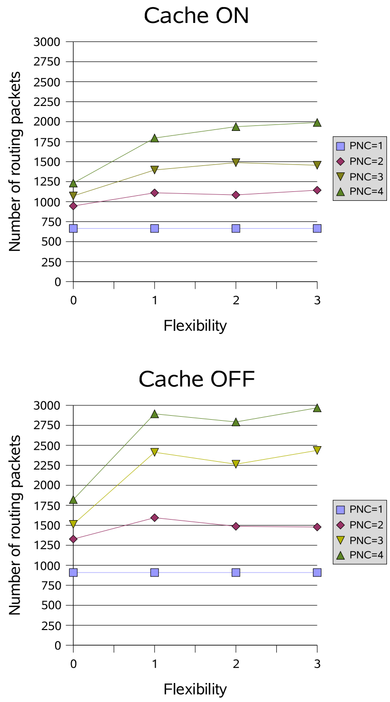

Figure 7. Routing overhead

route request is generated to the time that the last message associated with that request is received. The results are presented on figure 8. In general, turning the cache OFF results in a reduced cycle time, which is rather unexpected. When the maximum propagation count is 2 we can see clearly a minimum when Flexibility equals 1 for cache ON and 2 for cache OFF. This phenomena has been referred to in section 5.2.

5.4. SR validation in typical scenarios

To evaluate the effectiveness of the SR solution in a MANET with typical load, we now proceed with another set of simulations taking into account node mobility. In these simulations packets are sent through only one route until it fails, which is the default behavior in DSR. Our purpose is to measure the effects of the extra routes on cache and additional routing overhead caused by our modified route discovery technique.

In our framework we used the pairs of *Flexibility / PNC* presented on table 3. Each pair is assigned a mode.

On mode 1 the propagation using the SR technique is restricted to the maximum, so that only one extra route per node is allowed in relation to the default DSR behavior. Modes 2 and 3 maintain one of the parameters of mode 1, but in mode 2 we increase flexibility and on mode 3 we increase the number of RREQs propagated per node. Concerning the cache, it is turned off.

The scenario used in our simulations has 40 nodes in a 1500x300 area, and the simulation runs for 310 seconds. The scenario generation process uses the random waypoint mobility generator that come bundled with the ns-2 simulator; the output of the generator has been restricted with the help of a script that assures that no node partitioning phenomena occurs during the simulation. This is done in order to assure that all packet drops are due only to congestion or routing problems.

We used 3 TCP traffic sources, 3 UDP sources and 2 H.264 video sources starting at a random time between 0 and 10 seconds and generating traffic until the end of the simulation. One second before the beginning of each H.264 video flow, one UDP packet is sent to the destination to initiate the session, assuring that a route is available for video streaming. The average bit rate for each H.264 video flow is 186 kbit/s. H.264 packets are identified in the simulator as video packets, so that they have a higher priority on the interface queue. Therefore, only routing packets have higher

Mode	Flexibility	PNC
1	0	2
2	2	2
3	0	4

Table 3. Different test modes using SR

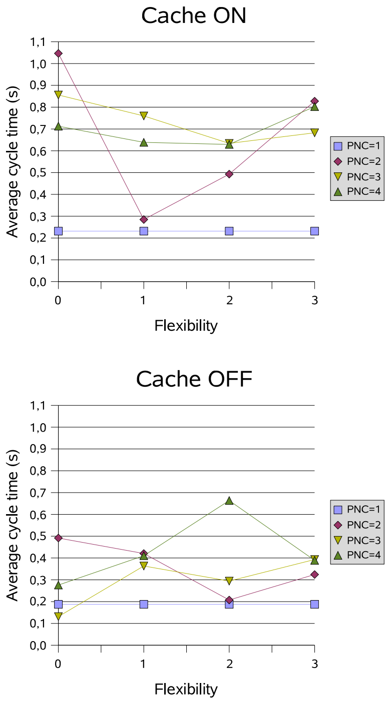

Figure 8. Average time for the route request propagation cycle

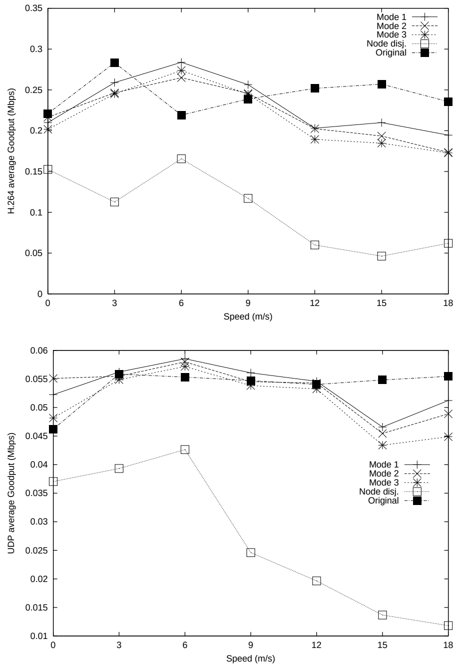

Figure 9. H.264 Goodput and UDP Goodput for varying mobility

priority that video packets. Concerning UDP traffic, each source sends 512 byte packets at a constant bit rate of 20.5 kbit/s.

For each speed we generated 10 different scenarios where all DSR modes have been evaluated. All results are the average values of these 10 distinct scenarios.

We will now present the simulation results achieved by using the simulation framework described above. We compare each of the 3 modes proposed with the LG solution, as well as with the original DSR implementation.

Figure 9 shows the results achieved in terms of Goodput at different speeds. As it can be seen, even though nodes are not using the extra routes available to perform traffic splitting or replication through them, the effect of having more routes in the cache is quite visible, being the results superior to the original DSR implementation for moderate mobility (H.264 streams) and for low mobility (UDP traffic). Relative to the LG solution, we can see that its performance is quite poor, as we expected from previous results.

Figure 10 shows the results achieved in terms of TCP performance and total number of data packets received on the network. The TCP improvements at high speeds show clearly the increased route stability caused by having more routes on cache. We therefore believe that such improvements are the main cause of the worse video and UDP results at high speeds. In terms of global performance, mode 1 surpasses DSR for speeds over 5 m/s. We can also see from that figure that when the speed reaches 18 m/s, the advantage by having more routes is lessened; this can be explained by the fact that having many routes is not useful if topology changes too quickly.

In terms of routing overhead, we can see that the difference between different SR modes and DSR is minimal, being even lower than DSR's for mode 1 at average speeds. Obviously, when the speed is zero there is no interest in obtaining further routes, since no route will ever be lost. The lack of efficiency of the LG solution is clearly put in evidence on this figure and, as it can be seen, consumes most of network bandwidth with routing packets. The use of the SR strategy is, therefore, a more acceptable and recommendable technique.

Concerning the different SR modes, in these simulations Mode 1 always has advantages for non-zero speeds when compared to Modes 2 and 3. Mode 1 is the most restrictive propagation method and, therefore, can find extra routes without much extra routing overhead. We believe that this is the main cause for the superior results achieved when compared to the other two modes.

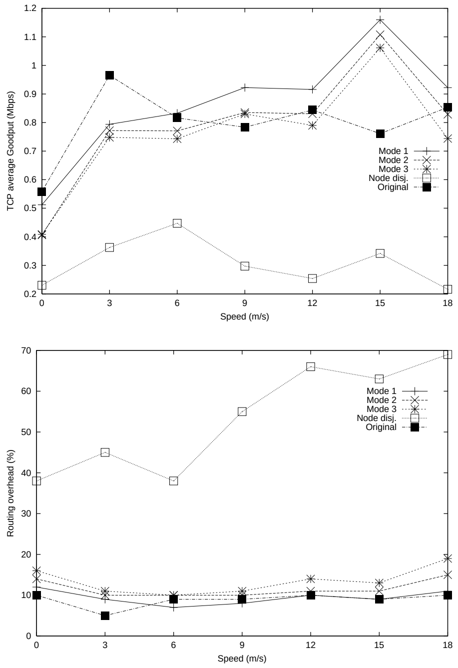

Figure 10. TCP Goodput and Relative Routing Overhead for varying mobility

6. Conclusions

In this work we propose an enhanced route discovery technique for DSR which we named Super Restrictive mode. In order to assess its effectiveness we simulate a no-movement scenario and obtain values for several quality parameters. We also compare it with the LG solution proposed by Lee and Gerla and observe that it achieves excellent improvements in terms of routing overhead, lost packets and route discovery time.

We also propose enhancements to the SR mode through two solutions which aim at both the increase and decrease the route discovery overhead, varying the number of routes found accordingly. The PNC parameter successfully restricts the propagation overhead, while increasing the Flexibility parameter allows more routes to be found. We make an evaluation of the behavior of these parameters through simulation and analysis of the results.

We proceeded with an evaluation in a typical scenario at different speeds and with background traffic load, observing the improvements relative to the LG solution and the default DSR implementation. In fact, we conclude that the LG solution, though allowing nodes to find more routes, generates too much overhead. We therefore consider it not appropriate for networks under average/high load. The SR solution, on the other hand, allows nodes to find extra routes without introducing too much routing overhead. Results show that, for some speeds, the routing overhead achieved can be even lower that DSR's.

Concerning data packets, the extra routes provided by the SR mode enhance route stability for average speeds (6 to 15 m/s), which is clearly evidenced by the improvements on TCP goodput and also reflected on the total number of data packets received.

As an extension to this work we pretend to study traffic splitting by using multi-path routing techniques, in order improve route stability for critical applications such as multimedia applications, as well as to reduce congestion by load balancing network traffic.

Acknowledgments

This work was partially supported by the Spanish CICYT under Grant TIC2003-00339 and by the *Junta de Comunidades de Castilla la Mancha*, Spain, under Grant PBC-03-001.

References

[1] IEEE/IEC Std 802.11. *Wireless LAN Medium Access Control(MAC) and Physical Layer (PHY) specifications*. The Institute of Electrical and Electronics Engineers, Inc., August 1999.

[2] Carlos T. Calafate and M. P. Malumbres. Evaluation of the H.264 codec (internal report). DISCA/60-2003, UPV, Spain, 2003.

[3] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC. http://bs.hhi.de/ wiegand/JVT.html, March 2003.

[4] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[5] 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.

[6] 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., 2001.

[7] 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*, pages 339–349, 2001.

[8] Jie 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.

[9] 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.

[10] K. Fall and K. Varadhan. NS notes and documentation. The VINT project, UC Berkeley, LBL USC/ISI, and Xerox PARC, November 1997.