

NICE-MRP: a Near-Optimal Radio-Interference Aware Multi-path Routing Protocol for MANETs

Wahabou Abdou*, Christelle Bloch[†], Damien Charlet[†], François Spies[†]

* Laboratory of Electronics, Computer Science and Image. University of Burgundy, France

Email: wahabou.abdou@u-bourgogne.fr

[†] FEMTO-ST Institute, Department of Computer Science. University of Franche-Comté, France

Email: {firstname.lastname}@univ-fcomte.fr

Abstract—Nowadays, with the miniaturization of communication devices, dense and highly mobile ad-hoc networks are very common. Besides, more and more devices have several radio interfaces. In such an environment, multi-path routing should be considered. Yet, only few multi-path ad-hoc routing protocols take into account the radio-interferences which might arise between two geographically close routes. Moreover, they generally only forbid the addition of interfering paths rather than looking for a different, more optimized, solution. This paper presents NICE-MRP, a novel multi-path routing protocol for mobile ad-hoc networks giving efficient solutions with respect to physical radio-interferences between paths. This protocol discovers and stores several combinations of non-interfering multi-path routes. The best multi-path route is used to transmit data, whereas alternative multi-path routes are kept to dynamically react to route breakages due to mobility and node failures. Compared to other well-known protocols, NICE-MRP presents good qualities in terms of latency, overhead and packet losses. Besides, these results end to be rather independent from the degree of mobility.

I. INTRODUCTION

Building routes in a wireless network is a complex task. In fact, there are various topologies for a wireless network, according to the network density (low or high) and the mobility of nodes (static or mobile). Most of the routing protocols build a single path from one source to one destination. Since the updating of routes can generate a lot of messages in the network, well-known protocols try to limit signalling traffic. Nevertheless, building more than one path from a source to a destination may use up roughly the same amount of power as for a one-path routing. Multi-path routing protocols can get new properties, such as fastest adaptation to mobility or node failure, bandwidth improvement or a better consumption balance between nodes. However, multi-paths in wireless networks differ from multi-paths in wired networks because of interferences. An efficient multi-path routing protocol must avoid interferences between neighboring nodes in case of the simultaneous use of two paths or more.

This paper proposes a new strategy called NICE-MRP to efficiently build multi-paths between a source and a destination in a wireless network. The ns-2 simulator was used to assess the performances of this protocol in realistic contexts including mobility and link loss between nodes.

The remainder of this paper is organized as follows. Section 2 is a survey on single and multi-path routing protocols. Section 3 describes the NICE-MRP protocol. The simulated

results of the comparison between NICE-MRP and SMR [1] is outlined in section 4.

II. STATE OF THE ART

Many ad-hoc routing protocols have been proposed in the literature to tackle the specificity of MANETs. Some of them only seek one route (single-path routing protocols) whereas others seek several routes. The following sections will briefly describe well-known single and multi-path routing protocols.

A. Single-path Routing Protocols for Ad-hoc Networks

There are two main classes of single-path routing protocols; they are classified according to how they build routing tables: proactive ones and reactive ones.

1) *Proactive protocols*: Proactive protocols discover and maintain pre-computed routes by regularly exchanging topological information. Each node fills up a routing table to reach all the hosts in the network. Several algorithms have been proposed to better compromise between network overhead and quality of routes (reliability, availability, latency). DSDV [2] and OLSR [3] are two of the best-known table-driven protocols. DSDV is a distance vector protocol. It uses the well-known distributed Bellman-Ford (DBF) algorithm [4] to compute the best paths between nodes. One-hop neighbors partially or totally exchange their routing tables to update information if necessary. This ensures the reliability of the computed distances. Besides, DSDV assigns a sequence number to each route in order to differentiate obsolete paths from updated ones. Frequent information exchanges permit to keep routes up-to-date in highly mobile networks, which prevents from data transfer failure. But they also increase the use of network resources and reduce the bandwidth available to transmit data packets. OLSR (*Optimized Link State Routing protocol*) lies on the optimized broadcasting of discovery and maintenance messages, using Multi-Point Relays (MPRs). Each node chooses its MPRs among its one-hop neighbors so that it can reach all its two-hop neighbors. Only MPRs broadcast link state messages. And each MPR only gives the state of its links to the nodes that chooses it. So, the number of MPRs must be minimized to limit overhead.

2) *Reactive protocols*: Reactive routing protocols limit routing overhead by building and maintaining routes only when needed. DSR [5] and AODV [6] are two of the most

widely used reactive protocols. In DSR (*Dynamic Source Routing protocol*), the source node sends a route request (RREQ) for each route discovery. Every neighbor eventually forwards this RREQ until it reaches the destination or an intermediate node which has an entry for this destination in its routing cache. The source receives a route reply (RREP) which contains the identification of all the intermediate nodes. The list of intermediate identifiers is included in data packets. This list might be very long in large networks. The route discovery process of AODV, the Ad-hoc On-Demand Distance Vector protocol, is nearly the same as in DSR. But AODV does not use source routing. Each node on the path caches a pointer to the neighbor from which it received the RREP.

B. Multi-path Routing Protocols

In the last years many scientific works focused on multi-path routing protocols. Such protocols seek and maintain several routes between the source and destination nodes, either to split traffic between several paths or to keep alternative routes available to be used in case of failures. Generally, they lie either on DSR (source routing) or on AODV (distance vector routing), and they attempt to find disjoint paths in order to limit interference. The following sections present some well-known protocols belonging to these two classes.

1) *Source-based routing protocols*: SMR (*Split Multipath Routing*) [1] is an on-demand routing protocol which allows to split traffic between several routes. It is based on DSR, but intermediate nodes do not reply to RREQs. The destination selects several paths among the different copies of the RREQs it receives. The first selected route is the one with the shortest delay, then the second is maximally-disjoint to the first one. In case of equality, the number of hops is used as a secondary criterion to be minimized. Traffic allocation uses per-packet granularity. The multi-Path Dynamic Source Routing protocol (MP-DSR) [7] is a routing protocol with QoS support, based on a metric named *end-to-end reliability*. This metric represents the probability to successfully send packets from source to destination within a given time interval. During route discovery, the source node computes the number of routes nb_{routes} and the reliability level for each of them, which permits to satisfy a required value of this metric. nb_{routes} RREQs are used to build paths which comply with these reliability requirements. The destination node finally selects a set of disjoint paths through which it sends RREPs. IZM-DSR [8] builds several zone-disjoint paths. It uses the notion of ActiveNeighborCount (*ANC*) introduced in ZD-MPDSR [9]. This value estimates the risks of interference in each node of the computed paths. Thus the source node selects the paths corresponding to the minimal values of (*ANC*) in the various RREPs it receives.

2) *Distance vector-based protocols*: AOMDV (*Ad-hoc On-demand Multipath Distance Vector*) [7] provides fault tolerance by building, for each destination, a set of acyclic link disjoint (or node disjoint) paths. When it exchanges route information, a node only gives, for each destination, an advertised hop count (which is the maximal hop count for all

the paths). Each new information about a destination is an available alternative route. In order to ensure loop freedom, nodes only accept routes that have a lower hop count than the one advertised for this destination. Adaptive Ad-hoc On-demand Multipath Distance Vector (A²OMDV) [10] is an extension of AOMDV which dynamically adapts itself to network conditions such as congestion. Route discovery begins like in AOMDV, but A²OMDV further selects the first found path as a primary route. This selection is regularly updated using values of Round Trip Time (RTT). Zone-disjoint Ad-hoc On-demand Multi-path Distance Vector (ZD-AOMDV) [11] computes zone-disjoint routes. It adds a few fields in the RREQs and RREPs used in AODV, and uses additional control packets. This allows each node to estimate how many nodes among its neighbors received a RREQ, and to increment the corresponding parameter (called ActiveNeighbourCount) in the RREQ. The source node waits until it has received several RREPs before sending data packets. Then it selects the routes with the lowest ActiveNeighbourCount to split traffic between disjoint paths.

C. Current Limitations of Multi-path Routing Protocols

Most of the multi-path routing protocols designed for MANET try to establish paths as disjoint as possible at the node level, and only differ in their path selection criteria. Yet, most of the multi-path protocols derived from standard routing protocols optimize hop counts, as they postulate that the shorter the route is, the better, with respect to both latency (each hop increases costs) and fault tolerance (fewer nodes means smaller fault probability). Nevertheless, in a physically deployed MANET, a multi-path routing protocol should not build its solution upon the shortest path nor work at the node level, as this may lead to low quality solutions. As an example, in a given MANET environment with three routes, if the shortest path radio-interferes with the other two paths, traditional multi-path routing protocol algorithms will only give a single path (the shortest path), whereas the two other routes could have been used together and simultaneously, giving additional bandwidth, redundancy and availability.

Moreover, path combination can be computed on three levels: node, link and zone-disjoints. Node (respectively link) disjoint routes have no node (respectively link) in common. In case of zone-disjoint routes, there is no intersection between the global coverages of the selected paths. The first two types can be easily computed and thus are used in most routing protocols in literature. Yet, in physically deployed radio networks, link and node-disjoint algorithms are not sufficient, as two link/node disjoint paths may be geographically close, and generate radio interferences. In such a situation, the multi-path solution may perform worse than a single path routing protocol [12].

Therefore, an effective multi-path routing protocol for MANETs should not only rely on the shortest path. It should carefully select the most effective and radio-interference-free path combination.

III. A NEW INTERFERENCE-AWARE MULTI-PATH ROUTING PROTOCOL

This section describes NICE-MRP (Non-Interfering Coverage arEa Multi-path Routing Protocol), a new multi-path routing protocol for MANETs. It is a reactive routing protocol working on the zone-disjoint level. NICE-MRP computes several optimal and/or suboptimal sets of zone-disjoint paths to connect two nodes. Zone-disjoint routes reduce the probability of generating interference when multiple paths are used simultaneously. Thus, this approach allows load balancing and a potential increase of the throughput. In addition, NICE-MRP uses two routing tables, which allows the protocol to adapt to topology changes or node / link failure without necessarily sending routing packets.

A. Overview

The classical zone-disjoint algorithms in literature [13] use the knowledge of the neighborhood of each node to assess the risk of interference. Usually, the zone-disjoint multi-path routing protocols use Hello messages to acquire this knowledge. However, regularly sending such packets generates additional signaling traffic. To limit the consumption of network resources, NICE-MRP does not use Hello packets. The proposed protocol takes advantage of route discovery packets to establish a list of neighbors for each node.

Like any reactive protocol, NICE-MRP is run only when data have to be transferred from a source node (s) to a destination node (d). If s does not have a known route to reach d , it sends a *Route REQuest* (RREQ) packet. This packet will be relayed exactly once by each intermediate node and will propagate up to the target node (d). Then d will reply by sending one *Route REPLY* (RREP) packet or more. The RREPs will follow the reverse path of the ones used by the RREQs. Upon reception of the first RREP, s will begin sending data packets. When several RREPs are received, s can compute disjoint routes which may be used simultaneously. The computation of the disjoint paths will be detailed in section III-C. The best set of non-interfering routes is stored in the main routing table (called *active routing table*), while the other disjoint routes are stored in the secondary table (called *passive routing table*). The latter is not used for routing packets; it is used to cache potentially usable routes (e.g. in case of link failure).

When a path in the active routing table is no longer usable, NICE-MRP looks up alternative non-interfering paths in its passive routing table. If alternative paths are found, they are moved to the active routing table and they replace the previous ones used to reach the same destination. If there is no path to the destination, NICE-MRP tries to discover new routes.

In short, NICE-MRP aims to:

- Determine several sets of non-interfering routes;
- Limit the routing overheads;
- Reduce the end-to-end delay when transferring data;
- Swiftly and efficiently respond to the node / link failures.

B. Route Discovery

The route discovery is the first step of the NICE-MRP protocol. It is made up of two main processes: route request and route reply.

1) *Route Request*: When s has a packet to send to d and it does not have a route to reach the destination, s sends an RREQ. Each RREQ is uniquely identified by a pair of identifiers: the sequence number of the RREQ and the address of the node originating the RREQ. The header of an RREQ is depicted in Figure 1. The *path* field contains the list of intermediate nodes between s and d . This list is empty when s issues the RREQ. Then the *path* field will be updated during the propagation of the RREQ. Each relay node will add its own address in the list of intermediate nodes. This list allows to prevent routing loops and helps to build a return path for sending the reply to this RREQ¹. Since NICE-MRP is a source routing, the list of intermediate nodes is also used by the source node to specify the whole path for data packets (the path will be encapsulated in data packets).

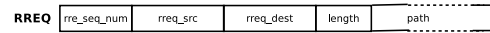


Fig. 1: NICE-MRP RREQ header

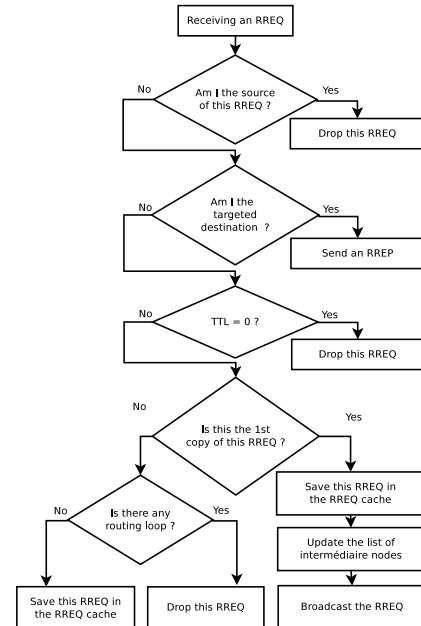


Fig. 2: NICE-MRP RREQ processing algorithm

To collect as much information as possible regarding the paths (including information about the neighboring nodes) and preserve their reliability, the intermediate nodes are not allowed to reply to RREQs. All this information will be used for computing zone-disjoint paths. However, to limit the number of redundant RREQs and to avoid the broadcast

¹It is assumed in this paper that the links are symmetric

storm problem [14], the intermediate nodes are only allowed to relay the first copy of each RREQ. When an intermediate node receives the first copy of a RREQ, it saves this request in an RREQ cache. Then the node adds its address in the list of intermediates and forwards the RREQ. Upon reception of a redundant copy of an RREQ, the intermediate node ensures that its own address is not already part of the list of intermediates nodes (the *path* field) to avoid routing loops before saving the RREQ in the RREQ cache.

At the same time as the RREQs are processed, NICE-MRP allows each node to acquire a knowledge of its neighborhood. To reduce the routing overhead, Hello packets are not used. Upon reception of every copy of an RREQ, each node updates its neighbor list by adding the address of the neighbor that transmitted the packet. In order to avoid obsolete entries, a removal timeout mechanism is involved, where a counter decreases over time, and is reinitialized upon reception of a new packet from the same source.

When an RREQ is received by the final destination node, the latter replies by sending an RREP.

2) *Route Reply*: The intermediate nodes are not allowed to reply to the RREQ packets. Thus, only d is allowed to send an RREP (route reply) packet described in Figure 3. RREP headers are similar to RREQ ones, with two added fields: a route reply sequence number (distinct from the RREQ sequence number) and a list of path-neighboring nodes.

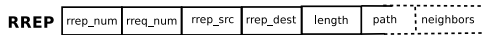


Fig. 3: NICE-MRP RREP header

d replies to all the RREQs by sending a unicast RREP packet with an empty neighbor list. Indeed, the neighbors of the source node (which originates the RREQ) and those of the destination node (which originates the RREP) are not taken into account for the computation of zone-disjoint paths (see Section III-C). d sends an RREP to all its neighbors that have transmitted the RREQ. As shown on Figure 4, each intermediate node which receives the RREP ensures that there is no routing loop in the path (if there is a loop, the RREP is dropped). If there is no loop, the intermediate node updates the message by adding (i) its own address to the *path* field and (ii) its list of neighbors to the *neighbors* field. The updated RREP is then forwarded by the intermediate node to its neighbors from which it has received the route request (this list is available in the RREQ cache). The RREP propagates until it is received by s .

C. Computation of Non-Interfering Paths

1) *Collecting single routes*: Upon reception of the first RREP for a given destination, s saves the path in both its *active routing table* and its RREP cache. Then s starts sending data. The data are transmitted immediately to reduce the end-to-end delay. s also starts a countdown. During that time, s might receive other RREPs (each of them representing a different path) which will also be recorded in the local RREP cache.

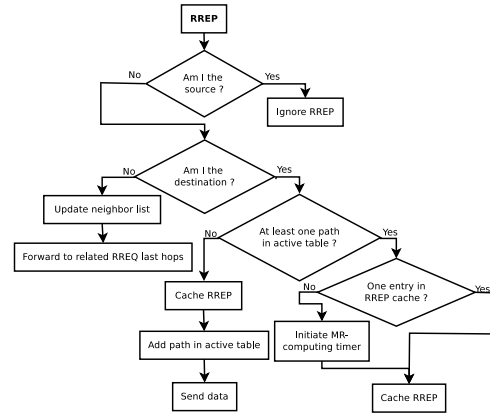


Fig. 4: NICE-MRP RREP processing algorithm

Thus, at the end of countdown, s has the local knowledge of all the available routes between s and d . Moreover, RREP packets also give information on neighboring nodes via the *neighbors* field. When the countdown expires, s computes the multi-path routes as described in the next section.

2) *Computing multi-path routes*: Using the RREP cache, s is able to compute zone-disjoint multi-path routes. For each p path to the destination d in the RREP cache, the algorithm used by NICE-MRP for computing multi-path routes selects only mutually non-interfering paths. This set of paths built from p should have no common intermediate node or neighborhood. Indeed, single routes may be combined into a multi-path route if their paths have no common node with (i) the other paths and (ii) the union of the neighboring nodes from all the paths.

Once all sets of non-interfering multi-path are computed, the best set is chosen to replace the route recorded in the *active routing table*. The other sets are recorded in the *passive routing table* for later use (e.g. in case of path failure). A common sorting strategy for multi-path routing is the size of the set of paths (i.e., the more paths in the route, the better). In the basic version of NICE MRP, the multi-path sets are classified according to the number of paths they contain. Some other criteria may be used, including the available bandwidth or an energy-based criterion. Another efficient strategy makes use of the expected transmission count (ETX) metric [15]. The ETX metric is a measure of the quality of a path between two nodes in a wireless network (i.e., the number of expected transmissions necessary to send a data packet to a destination without error). In a multi-hop wireless network, the ETX is the sum of all the ETX on the path. Thus, path sorting may be refined with respect to the ETX. Whatever the criteria used, this has no significant impact on the non-interfering multi-path selection algorithm. But some particular criteria could be used to ensure the fulfillment of quality of service. The results presented in this paper are obtained using the basic criterion, i.e. the number of non-interfering paths.

Unlike the classical methods for computing disjoint routes which define a single set of disjoint paths (often based on the first path which is discovered), NICE-MRP takes into account

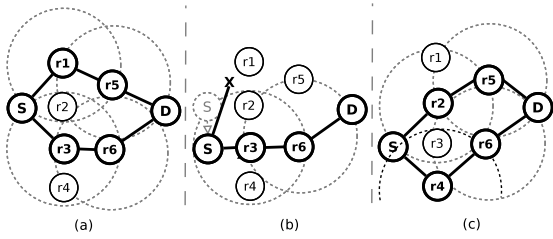


Fig. 5: Protocol reconfiguration scenario

all the route replies (RREPs). This increases the probability of getting non-interfering multi-path. Since the computation of non-interfering paths is done after the beginning of data transfer, it does not increase the end-to-end delay. For the purpose of optimization, and when the node architecture allows it, the algorithm of non-interfering paths can be parallelized (RREP cache entries are mutually independent).

In short, the algorithm of disjoint paths used by NICE-MRP does not depend on the first path discovered, but all the received RREPs.

D. Maintenance Strategies

Over time, routing protocols need periodic maintenance as several factors may influence the quality of the active route, or even render it useless. Node failure is a common cause, even if it will not happen too often. Nowadays, the miniaturization of the mobile devices used in MANETs increases the frequency of node mobility, which further generates frequent link disruptions. Therefore, a routing protocol suitable for ad-hoc networks should critically take care of mobility tolerance and reconfiguration after link loss.

To illustrate the mechanism used in NICE-MRP, an example scenario of link failure caused by source mobility is depicted in figure 5. In this environment, routing paths must be chosen between two similar zone-disjoint multi-path solutions: “ $r1 \rightarrow r5$ ” + “ $r3 \rightarrow r6$ ” and “ $r2 \rightarrow r5$ ” + “ $r4 \rightarrow r6$ ”. Figure 5(a) shows that “ $r1 \rightarrow r5$ ” + “ $r3 \rightarrow r6$ ” where chosen, and communication is balanced over these two paths. In Figure 5(b), the source node s moves south, going out of $r1$'s range. However, as it is a multi-path route and $r3$ is still in the range of s , the communication doesn't cease, but its quality decreases. Before declaring $r1$ as *lost*, s will try to initiate three re-acknowledgments. If necessary, the active routing table is updated accordingly. This is the same mechanism as the one used in the case of hardware failure.

As long as one path is still available, the communication still progresses in a degraded mode. NICE-MRP will maintain the remaining paths and try to add further paths to the active route. That is, s parses its passive routing cache to find previously recorded multi-path routes containing $r3$, while not containing $r1$. Simultaneously, all the records containing $r1$ are removed. In our example, there is no other solution containing $r3$ without $r1$. Later, the passive routing table is browsed again, to try to find a better solution than the currently active one, even if a more important change in

topology has to occur. This is depicted in Figure 5(c), where the communication via $r3$ is stopped to initiate the new multi-path solution: “ $r2 \rightarrow r5$ ” + “ $r4 \rightarrow r6$ ”. The active routing table must be updated accordingly.

IV. PERFORMANCE EVALUATION OF NICE-MRP

This section describes the performance evaluation of NICE-MRP, and a comparison with another multi-path routing protocol (SMR). These evaluations were conducted with the ns-2 network simulator [16]. NICE-MRP is compared with two different versions of SMR : SMR with link disjoint algorithm (SMR_ld) and SMR with node disjoint algorithm (SMR_nd) [17].

A. Experimental procedure

The simulated network contains 30 nodes with a radio range limited to 250 meters. The nodes move in a 1000 m x 300 m area during 900 seconds. The mobility model is random waypoint with a pause time varying from 0 to 900 seconds: 0 means permanent mobility while 900 means no mobility. All nodes have the same pause time value during a simulation run. The nodes are first randomly placed on the grid, which is the studied area. They remain in these positions until the end of the pause time, then each node chooses a random destination point and tries to reach it with uniformly distributed speed chosen between 0 and 10 m.s⁻¹. Once the destination is reached, the node remains stationary for a period of *pause_time* seconds before choosing a new destination. This is repeated until the end of the simulation (duration 900 seconds). Among the 30 nodes, 10 pairs are randomly chosen to exchange data packets. The traffic is modeled with a CBR (Constant Bit Rate) stream, sending data at a rate of 4 packets (512 bytes) persecond.

B. Performance metrics

Three widely used metrics were chosen to assess the performance of each protocol: end-to-end delay, packet loss rate and routing overhead.

- **End-to-end delay** is the packet transmission time. It begins once a packet is sent by upper layers and ends when the packet is received by the destination's corresponding layers. So, it includes route discovery time, the waiting time due to the WiFi MAC layer standard (including the backoff algorithm), the packet propagation time on the radio channel, the packet buffering and reordering (if required by the routing protocol) time.
- **Packet loss rate**. A packet is considered to be lost when a data packet sent on the radio channel earlier has not been received by the final destination. For each pair "source - destination", packet loss rate is the ratio of lost packets to the total number of packets sent.
- **Routing overhead** is the number of packets generated and transmitted by the routing protocol to fulfill its own purposes. Any routing information is taken into account, without discriminating sent packets from relayed packets.

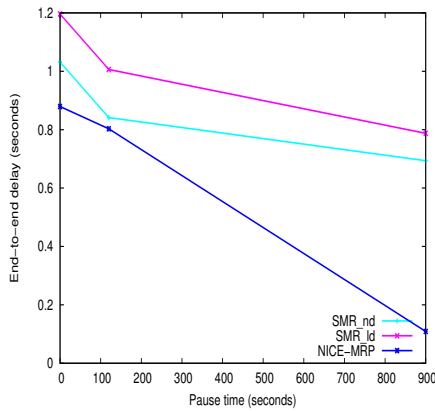


Fig. 6: End-to-end delay

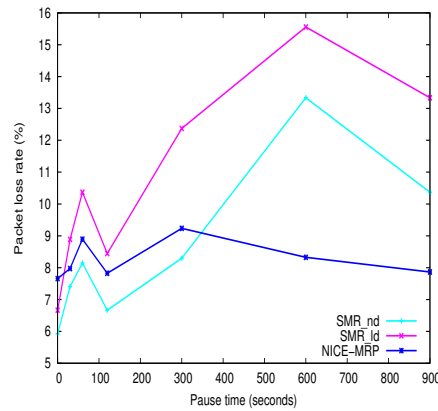


Fig. 7: Packet loss rate

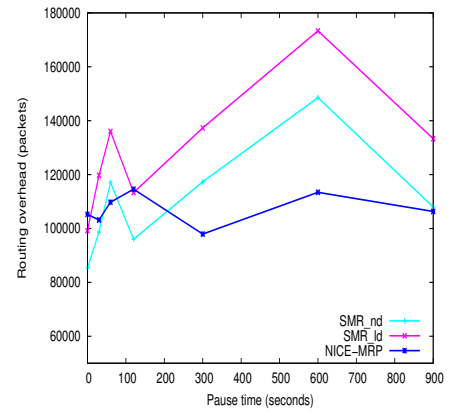


Fig. 8: Routing overhead

C. Results

NICE-MRP was compared with SMR, as regards the simulation results published by Parissidis et al. [17], so the simulations were carried out in the same experimental conditions.

Overall, NICE-MRP has lower latency than the two versions of SMR (Figure 6). This can be explained by the fast dissemination of requests for NICE-MRP. Since each node retransmits each RREQ only once, the radio channel availability is increased and the dissemination of route discovery packets is faster. One can also note that higher mobility (i.e., low pause time) increases latency. This is because of the fact that when nodes move, some of the links break. Routing protocols require a slight reaction time to rediscover new routes or switch to the backup routes.

Regarding packet loss (Figure 7), NICE-MRP has an almost constant rate because it dynamically updates its main route table (active route table) before all routes break, as described in Section III-D.

For SMR_nd and SMR_ld, when loss rate increases, a peak of routing packets is observed (Figure 8). Packet losses trigger the sending of new route discovery packets. NICE-MRP, using two route tables, more rarely executes its route discovery process. Indeed, when the routes of the active table are broken, the protocol looks for alternative routes in the passive table.

V. CONCLUSION

This paper assessed a new efficient strategy to identify multi-paths in wireless network. The results show that NICE-MRP reacts faster than SMR in case of link loss. Our contribution will allow to identify multi-paths without radio interference between nodes of different paths with common source and destination. This property is necessary to potentially improve overall bandwidth between two nodes that are using more than one path simultaneously.

REFERENCES

[1] T. Shu, M. Krunz, and S. Liu, "Secure data collection in wireless sensor networks using randomized dispersive routes," *IEEE Transactions on Mobile Computing*, vol. 9, pp. 941–954, 2010.

[2] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *SIGCOMM Comput. Commun. Rev.*, vol. 24, pp. 234–244, October 1994.

[3] P. Jacquet, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," 2001, pp. 62–68.

[4] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, Inc., 1987, pp. 297 – 333.

[5] D. B. Johnson, D. A. Maltz, and Y. C. Hu, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for ipv4," *RFC 1058*, 2007.

[6] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," *RFC 3561*, 2003.

[7] V. C. Patil, R. V. Biradar, R. R. Mudholkar, and S. R. Sawant, "On-demand multipath routing protocols for mobile ad hoc networks issues and comparison," *International Journal of Wireless Communication and Simulation*, vol. 2, no. 1, pp. 21–38, 2010.

[8] N. T. Javan, A. Dareshoorzade, S. Soltanali, and Y. G. Birgani, "IZM-DSR: A new zone-disjoint multi-path routing algorithm for mobile ad-hoc networks," *IEEE Symp. on Comp. Model&Sim*, pp. 511–516, 2009.

[9] N. T. Javan and M. Dehghan, "Reducing end-to-end delay in multi-path routing algorithms for mobile ad hoc networks," in *Procs. of MSN'07*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 715–724.

[10] D. Shin, J. Lee, J. Kim, and J. Song, "A2OMDV: An adaptive ad hoc on-demand multipath distance vector routing protocol using dynamic route switching," *Journal of Engineering Science and Technology*, vol. 4, no. 2, pp. 171–183, 2009.

[11] N. T. Javan, R. Kiaefifar, B. Hakhamaneshi, and M. Dehghan, "ZD-AOMDV: A new routing algorithm for mobile ad-hoc networks," in *Procs of 8th IEEE/ACIS Intl Conf. on Comp. and Inf. Science*, 2009, pp. 852–857.

[12] S. Waharte and R. Boutaba, "Totally disjoint multipath routing in multi-hop wireless networks," in *Procs. of the IEEE International Conference on Communications (ICC)*, vol. 12, Jun 2006, pp. 5576 – 5581.

[13] M. Natarajan, "Performance comparison of link, node and zone disjoint multi-path routing strategies and minimum hop single path routing for mobile ad hoc networks," *International Journal of Wireless and Mobile Networks (IJWMN)*, vol. Vol.2 No.4, 2010.

[14] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *MobiCom '99: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, 1999, pp. 151–162.

[15] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wirel. Netw.*, vol. 11, no. 4, pp. 419–434, Jul. 2005.

[16] "The Network Simulator Project - ns-2." [Online]. Available: <http://www.isi.edu/nsnam/ns>

[17] G. Parissidis, V. Lenders, M. May, and B. Plattner, "Multi-path routing protocols in wireless mobile ad hoc networks: A quantitative comparison," in *NEW2AN*, 2006, pp. 313–326.