Reliable Routing in Wireless Ad Hoc Networks

Luiz Albini[1,2] Antonio Caruso[1] Stefano Chessa[1,2*] Piero Maestrini[1,2]

[1]Istituto di Scienza e Tecnologie dell'Informazione,
Area della Ricerca, CNR di Pisa-S.Cataldo, 56100 Pisa, Italy.

[2]Computer Science Department, University of Pisa,
Via Buonarroti 2, 56127 Pisa, Italy.

e-mails:{albini,caruso,chessa,maestrini}@isti.cnr.it

Abstract

A novel routing protocol for wireless, mobile ad hoc networks is presented. This protocol incorporates features that enhance routing reliability, defined as the ability to provide almost 100% packet delivery rate. The protocol is based on a virtual structure, unrelated to the physical network topology, where mobile nodes are connected by virtual links and are responsible for keeping physical routes to their neighbors in the virtual structure. Routes between pairs of mobiles are set up by using information to translate virtual paths discovered in the virtual structure. Route discovery and maintenance are based on unicast messages travelling across virtual paths, with sporadic use of flooding protocol. Most floodings are executed in background using low priority messages. The routing protocol has been evaluated and compared with the Dynamic Source Routing protocol (DSR) by means of simulation.

1 Introduction

Mobile, wireless ad hoc networks are among the fastest growing areas in networking research. They are best suited for applications in environments where fixed infrastructures are unavailable or unfeasible. Examples of such applications are communication in remote or hostile environments, management of emergencies, and disaster recovery.

Mobile ad hoc networks are an heterogeneous mix of different wireless and mobile devices, ranging from little hand-held devices to laptops. Such devices rely on on-board batteries for energy supply, hence energy efficiency of mobiles is an important issue [1]. The effect of battery depletion is similar to a crash fault, from which the mobile[1] may or may not recover depending on the availability of battery replacement/recharge.

Ad hoc networks implement a distributed cooperation environment, based on a peer-to-peer paradigm. Given the limited range of wireless communication, the network is generally multihop, since direct communication between mobiles is generally not available. For this reason, a distributed routing protocol is required in order to provide communication between arbitrary pairs of nodes. A major problem arises from the mobility of nodes causing the network topology to be variable and to some extent unpredictable. In fact, communication links between nodes may be broken, nodes may fail and possibly recover from failures and new links may appear. The routing protocol must react promptly to recover from link and node failures and to take advantage of new links. For this reasons, existing routing protocols designed for fixed networks are unsuitable, and routing in ad hoc networks is a major issue.

Routing protocols for wireless ad hoc networks can be classified into the main categories of *table-driven* (or *proactive*) [9, 13, 15] and *on-demand* (or *reactive*) [7, 5, 10, 11, 14].

Reactive routing protocols, which appear to be more suitable for ad hoc networks, do not maintain up-to-date information about the network topology as it is done by the proactive ones, but they create routes on demand. Among reactive routing protocols, the *Dynamic Source Routing* (DSR) [5, 4] and the *Ad hoc On Demand Distance Vector Routing* (AODV) [11] are the most established developments.

In DSR, when a mobile (source) needs a route to another mobile (destination), it initiates a *route discovery* process which is based on flooding. The source originates a *route request* (RREQ) packet, that is flooded over the network. The RREQ packet contains a list of hops which is collected by the route request packet as it is propagated through the network. Once the RREQ reaches either the destination or a node that knows a route to the destination, it responds with a *route reply* (*RREP*) along the reverse of the route collected by the RREQ. This means that the source may receive several

*Corresponding author

[1]in the rest of the paper the terms mobile and node will be used as synonyms.

RREP messages corresponding, in general, to different routes to the destination. DSR selects one of these routes (for example the shortest), and it maintains the other routes in a cache. The routes in the cache can be used as substitutes to speed up the route discovery if the selected route gets disconnected. To avoid that RREQ packets travel forever in the network, nodes that have already processed a RREQ discard any further RREQ bearing the same identifier.

In AODV, routes are also set up by flooding the network with RREQ packets which, however, do not collect the list of the traversed hops. Rather, as a RREQ traverses the network, the traversed mobiles store information about the source, the destination, and the mobile from which they received the RREQ. The latter information is used to set up the reverse path back to the source. When the RREQ reaches a mobile that knows a route to the destination or the destination itself, the mobile responds to the source with a *route reply* packet which is routed through the reverse path set up by the RREQ, also setting the forward route from the source to the destination. To avoid overburdening the mobiles with information about routes which are no longer (if ever) used, nodes discard this information after a timeout. A comparison between DSR and AODV can be found in [12].

The main difference between DSR and AODV is in the way they keep the information about the routes: in DSR it is stored in the source while in AODV it is stored in the intermediate nodes. However, the route discovery phase of both is based on flooding. This means that all nodes in the network must participate in every discovery process, regardless of their potential in actually contributing to set up the route or not, thus increasing the network load.

To reduce this burden, we propose an alternate routing strategy (called *Virtual Routing Protocol*, or, briefly *VRP*) which manages node mobility and link and node failures with a limited dependency on flooding for route discovery. VRP also implements different priority levels for messages: higher priorities are mainly used by unicast messages, while almost all floodings are performed on lower priorities.

The rest of the document is organized as follows: Section 2 discusses the reliability issues of routing in ad hoc networks, Section 3 introduces the system model and preliminary definitions, and Section 4 presents the Virtual Routing protocol. Simulation studies are presented and discussed in Section 5 and Section 6 draws the conclusions.

2 Reliability Issues

In a wireless ad hoc network where pairs of mobiles communicate by exchanging a variable number of data packets along routes set up by a routing algorithm, reliability may be defined as the ability to provide high *delivery rate*, that is, to deliver most of the data packets in spite of faults breaking the routes or buffer overflows caused by overloaded nodes. Given the intrinsic nature of wireless, ad hoc networks, reliability is a major issue.

Links failures may occur due to interferences on the wireless medium, or, most probably, to nodes' mobility, when pairs of nodes move out of the reciprocal transmission range or are shadowed by obstacles. The situation where a node is disconnected from the rest of the network is equivalent to a recoverable crash fault of the node. Node failures may be caused by battery depletion, hardware faults, or by software crashes.

Failures may be recoverable; for example a broken link may become available when the two nodes move again within the transmission range of each other or across an obstacle, or a node fault due to a depleted battery may be recovered after the battery replacement.

Link and node failures may rise at any time, and may be revealed during the route discovery phase or during communication along a route already established.

The common approach to deal with faults during the route discovery phase is to exploit flooding-based protocols. If a combination of faults prevents the source from building a route to the destination, the destination is declared unreachable, and the route discovery can be tried later in the hope that the disconnection has been recovered.

Faults affecting a communication between two mobiles along a route that was successfully established are managed by means of a *route maintenance* protocol, which, however, may not avoid substantial packet losses. Once a route R has been established, the source starts sending packets through R. If a link or a node of R fails, the node preceding the failed link or node detects the failure of R. Typically, the latter node sends a *route error* message (*RERR*) to the source. Once the source receives the RERR it starts again a route discovery to establish a new route and resume communication. In the time elapsed between the notification of the RERR and the setup of a new route the source cannot send further data packets generated by the application layer for that destination. Although the packets can be buffered by the source, packets may be dropped if the buffer size is exceeded. Furthermore packets sent in the time elapsed between the occurrence of the fault and its notification to the source may also be lost. For this reason the management of data packet losses is generally left to the application layer, and packet losses should be kept as low as possible.

The overhead of the routing protocol may also contribute to packet losses. In fact, both the route discovery and the route maintenance protocols rely on a considerable number of packets travelling in the network. This is specially true if the above protocols rely on floodings. These packets contribute to network congestion, and may contribute to longer buffering of data packets, and, ultimately, to data packet losses if the mobiles buffer capacities are exceeded.

3 System Model

We consider a mobile ad hoc network composed by n mobiles, each of which is assigned a unique integer ID ranging from 0 to $n-1$. Mobiles communicate with each other via radio transceivers, and they use a MAC protocol to solve contentions over the wireless channel.

Since the transmission range of the mobiles is limited, the network is multihop and its topology changes with time due to nodes mobility or faults. In the real world the wireless links are not necessarily symmetric, i.e., mobile i might send messages to mobile j directly, but j may not be able to send messages to i directly. However, we focus on ad hoc networks with bidirectional links, thus meeting the specifications of the IEEE 802.11 RTS/CTS protocol [2] which works only with bidirectional links.

Under this assumption it is possible to model the network at any time t as an undirected graph $G(V, E)$, where V is the set of nodes, and E is the set of links at time t. Link $(i, j) \in E$ (that is, i and j are *adjacent*) iff both nodes are within each other's transmission range at time t and they are not shadowed by obstacles. The set of all nodes that are adjacent to a node i is called the *Neighbor Set* of i, denoted by N_i.

VRP assumes the existence of a link-layer protocol, which ensures that each mobile is always aware of its neighbors. This assumption is feasible with the IEEE 802.11, as it can provide such information to the routing protocol [2]. This assumption is also used by other routing protocols such as [10] and [6]. If the link layer cannot provide such service, this information can be constructed by the routing protocol by exchanging "hello" messages between the mobiles.

VRP uses different priority levels on messages: e.g. *low*, *medium* and *high*. Priority levels are used internally by the mobiles to decide the order in which the messages should be sent, i.e., when mobile i has messages of different priorities to send, it sends one with the highest priority. If the node has multiple messages with the same priority, it can choose any one of these messages. A newly proposed supplement of the 802.11, the 802.11e [3, 8], introduces the concept of priority levels at the link-layer.

4 The Virtual Routing Protocol

The *Virtual Routing Protocol* (VRP) has been developed with the goal of providing high delivery rate in mobile, ad hoc networks. To this purpose, VRP incorporates the following strategies to reduce the use of floodings in route discovery and maintenance, thus reducing the network congestion:

- a *virtual structure*, where each mobile, called a *scout*, is responsible for maintaining routes to a small subset of other mobiles (called *peers*). A *virtual path* from a source to a destination is a path in the virtual structure. A route from source to destination is constructed by translating the virtual path; i.e. by chaining the routes from scouts to peered mobiles in the virtual path. The virtual structure is redundant; i.e. there are many virtual paths for any source-destination pair.

- route discovery by means of unicast messages which translate the virtual path travelling from source to destination along the route. If discovery fails, a different virtual path is attempted. Flooding-based discovery is the last resource after repeated failures of the unicast-based discovery.

- reactive scout update when the route translation reveals that the route to a peered mobile stored in a scout is invalid. Scout updates require a flooding which, however, is performed in low priority, since most probably route setup can be successfully achieved using an alternate virtual path. This greatly reduces the network congestion due to flooding. In order to fully exploit the power of flooding protocols, reactive update of scout-peer routes is combined with the proactive update of all routes from the given scout to all its peers.

- random distribution of routes: VRP does not attempt to set up minimum length routes, however routes are essentially constructed at random by translating some virtual path. While this feature somehow increases the time to set up routes and the number of data packets to be dispatched in communication, it distributes over the network the burden of packet dispatching, thus contributing to a reduced congestion.

4.1 The Virtual Structure

VRP defines a directed graph $L(V, A)$ (called *virtual structure*), which is unrelated to the actual network topology. Node set V represents the mobiles and set A represents the virtual links. Given a virtual link $(i, j) \in A$, mobile i is called a *scout* to j and j is said to be *peered* by i. Given mobile i, the *peer set* P_i is defined as $P_i = \{j : (i, j) \in A\}$, and the *scout set* is $S_i = \{j : (j, i) \in A\}$.

It should be noticed that VRP is, to some extent, independent of the graph implementing the virtual structure. However, it appears that the graph should be regular (i.e. the degree is the same for all nodes) to ensure that the number of peers is the same for all scouts, thus contributing to an even distribution of the load in the Route Discovery. The most suitable graph structure should be selected considering properties such as diameter, bisection width and the scalability, i.e., the capacity of tolerating addition or removal of mobiles without disrupting

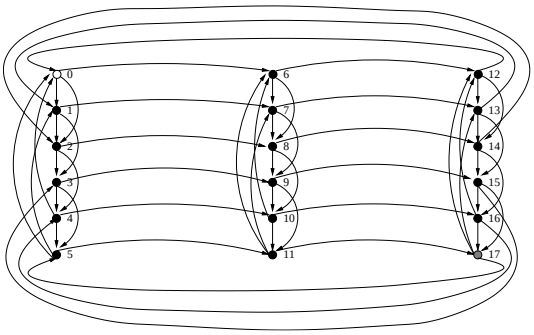

Figure 1: A Rings of Rings with $x = 3$ rings of $y = 6$ nodes each and 3 scouts per node ($s = 3$).

the structure itself. VRP was simulated using different virtual structures, like: RoR, hypercube, CCC and torus, however results reported in this paper are obtained using the *Rings of Rings* (RoR) structure which is introduced below.

The *Rings of Rings* (RoR) structure is based on congruences [16] and it is defined as follows. Assume that there are two integers, x and y, such that, $x * y = n$ (if needed, this assumption can be met by introducing dummy nodes), and let s be an integer such that $1 < s \leq y$. The nodes in the virtual graph L are arranged into x rings, each of which is composed by y nodes; each node is connected to $s - 1$ nodes inside its ring and to 1 node in the next ring.

Formally, set V is partitioned into x rings, called $V_0, V_1, ..., V_{x-1}$, where, for each $a \in [0, x)$, $V_a = \{i : a * y \leq i < (a + 1) * y\}$. Link (i, j) belongs to A iff either $j \ mod \ y = (i + d) \ mod \ y$ for some $1 \leq d < s$ or $j = (i + y) \ mod \ n$. A RoR with $x = 3$, $y = 6$ and $s = 3$ is shown in Figure 1.

A notable feature of RoR structure is the redundancy of virtual paths, whose degree is determined by parameters x, y, and s. Even in the occurrence of multiple node and/or link failures, it is most likely that there exists a virtual path of available nodes from the source to the destination, and this path can be translated in a route chaining valid links.

Another advantage of RoR is the ability to be "reshuffled" when the average length of routes constructed over the virtual structure is unsatisfactory. Although the independence of virtual and physical structures prevents from guessing how the virtual structure should be reshuffled, random reshuffling may also provide positive results. Reshuffling may be implemented by congruence-based [16] implementation of mobiles identifiers, leading to an isomorphic virtual structure. This matter is beyond the scope of this paper and is subject of ongoing research.

4.2 Route Discovery

Route Discovery is used by a mobile i to set up a route from itself to mobile j. If j is a neighbor of i, then the route is trivial. Otherwise the route acquisition is divided in two steps: the *virtual path setup* and the *physical route setup*. All messages of this phases are high priority messages.

In the virtual path setup the source i establishes a loop-free *virtual path* to the destination j on graph L: the virtual path is a path from the source to the destination, where every intermediate node (if any) is a scout to its successor node and a peer of its predecessor node in the virtual path. Since L is known in advance by all mobiles, this step does not require communication. The virtual path is constructed by the source node i by selecting a shortest path of available nodes in L connecting destination j to any node in $\{i\} \cup N_i$. Mobiles belonging to the selected virtual path are called *virtual hops*.

Given the virtual path $VP = (vp_0, vp_1, \ldots, vp_t)$ with $vp_0 = i$ and $vp_t = j$, the physical route setup consists in chaining the routes stored in the scouts $vp_0, vp_1, \ldots, vp_{t-1}$. To this purpose the source i sends a *route translation* (*RTRANS*) message which traverses all the virtual hops. The route translation message carries the destination ID j, the virtual path VP, and the list of (physical) hops taken by the RTRANS message. This list is updated as the RTRANS is propagated through the network.

The RTRANS message is first routed from $vp_0 = i$ to vp_1. If vp_1 is a neighbor of i, then i simply puts the ID of vp_1 in the physical route of the RTRANS and forwards it to vp_1. Otherwise, as i is a scout to vp_1, it attaches to the RTRANS message its own route to vp_1. The RTRANS message is forwarded along mobiles in this route until reaching scout vp_1.

When a node which is not part of the virtual path receives a RTRANS, it simply forwards the message to the next node in the physical route.

When an intermediate node, vp_k, of the virtual path receives the RTRANS, it appends to the RTRANS its own route to node vp_{k+1}, and it forwards the message through this route. This process is repeated until RTRANS reaches the destination $j = vp_t$. As an optimization, when mobile h receives an RTRANS, it delivers this message directly to destination if the destination is in N_h.

As it is propagated from i to j, the RTRANS collects the sequence R of the traversed mobiles. If the routes stored by the scouts are up-to-date, RTRANS eventually reaches j and R is a physical route from i to j.

It should be observed that virtual paths are loop-free and the RTRANS message travels along routes from vp_k to vp_{k+1} which do not contain loops. Thus the RTRANS message always reaches the destination, if all scouts have valid routes to their peers. However, routes stored by different scouts in the virtual paths may not be dis-

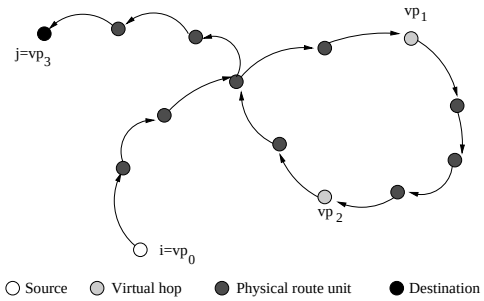

Figure 2: Virtual path translation.

joint, implying that route R may contain loops. For this reason, upon receiving the RTRANS message, the destination cuts the loops in R to obtain a loop-free route R' from i to j. This process is illustrated in Figure 2 showing the route of an RTRANS message from the source $i = vp_0$ to $j = vp_3$.

After cutting the loops, the destination sends to the source a *route reply* message containing R'. This message is routed through the reverse of route R'. As it receives the route reply, the source may start sending data packets to the destination through R'.

The Route Discovery may fail due to node or link failures, disrupting the route stored in some scout vp_k in the virtual path. This means that some node h along the route going from scout vp_k to vp_{k+1} is unable to forward the message to the next node h' in the route.

When node h detects the failure, it attempts to find a new route (a *detour*) from itself to the destination j of the RTRANS message. To this purpose it executes a Route Discovery by selecting a virtual path from h to j and by translating this virtual path into a physical route. If the detour is also broken, the node detecting the detour failure may attempt another detour. This process is repeated up to a maximum number of iterations, according to a parameter set by the source and stored in the RTRANS message itself.

After forwarding the RTRANS through the detour, node h sends a *route error* (RERR) to the last scout through which the RTRANS has passed, node vp_k. The RERR is used to warn vp_k that the route it has archived to reach vp_{k+1} is no longer valid. When vp_k receives the RERR it starts a Scout Update, which will be detailed later.

If the maximum allowable number of detours is reached without success, a RERR is sent back to the source along the reverse of the route accumulated in the RTRANS message. The route error will first reach the scout vp_k before reaching the source (vp_k is the last scout through which the RTRANS has passed). As vp_k receives the route error, it determines that its route to vp_{k+1} is no longer valid and it starts a low-priority Scout Update.

Once the source receives the route error, it retries the route discovery using different virtual paths up to a maximum allowable number of attempts. If this maximum is reached without success, the source attempts to set up a route to the destination by flooding the network with an high-priority *route request* (*RREQ*) as it is done in DSR. If this ultimate attempt also fails, the upper application layer is notified that the destination is unreachable.

It should be observed that route discovery based on unicast messages travelling along virtual paths performs well only if the scouts have valid routes to their peered mobiles most of the time. Under this hypothesis, the overhead of the flooding-based protocols can be avoided in most of the cases and the cost of the route discovery is proportional to the length of route R along which runs the RTRANS message.

4.3 Scout Update

The *Scout Update* is executed by a scout i when it receives a route error notifying that the route to some peer $k \in P_i$ is no longer valid. The Scout Update phase of the protocol is similar to the reactive route discovery of DSR. However in VRP the Scout Update executed by mobile i, beside looking reactively for peer $k \in P_i$ also looks proactively for routes to all nodes in P_i.

Another feature of Scout Update in VRP is the forwarding of messages in low priority which greatly reduces the interferences with the packets used in route discovery and in data communication, which are forwarded with higher priorities.

The Scout Update initiated by mobile i uses a *multiple destination route request* (*MD-RREQ*) message, which contains the ID of scout i and the list of hops collected as it is propagated through the network. The MD-RREQ message is flooded in the network as a low priority message.

When MD-RREQ reaches node $h \in P_i$, that is, i is a scout to h, then h responds with a *route reply* (*RREP*) sent as a low priority message along the reverse of the path accumulated in the MD-RREQ. In any case h also forwards the MD-RREQ to its neighbors to guarantee that the MD-RREQ reaches all the mobiles in the network. To avoid the MD-RREQ packets travel forever in the network, nodes that have already processed an MD-RREQ discard any further MD-RREQs belonging to the same Scout Update.

It is implicitly stated that the destinations of the MD-RREQ are all peered mobiles of i. This means that i updates the routes to all of its peered mobiles, not just to the one that was notified as broken. If the mobiles have s peers, the proactive update of routes to all peers in P_i add little overhead to the reactive update of the route from i to k since the additional cost consists in $s - 1$ unicast RREP messages travelling in the reverse directions along the newly discovered routes from i to every $h \in P_i$.

4.4 Route Maintenance

Once a route has been established and communication along this route has been initiated, the route can get disconnected. Disconnection occurs when two consecutive mobiles r_p and r_{p+1} in $R = \{r_0, r_1, \ldots, r_m\}$ lose their connection because of failures, obstacles or exceeded transmission range. In this case mobile r_p notifies the source r_0 that route R is disconnected, by sending a *route error* (*RERR*) message.

Contrary to the technique used in the Route Discovery protocol, r_p cannot notify the last upstream scout because data messages do not carry information to identify the virtual hops.

As the RERR message inherit the priority from messages suffering the route error, in this case RERR inherits the medium priority of the data message. When the source receives the route error, it must set another route to the destination, if a route to that destination is still desired. The new route is set up by the source by means of the (high priority) Route Discovery phase of the protocol.

5 Simulation and Analysis

The simulative evaluation compares VRP with DSR, which is also a source routing algorithm. The DSR implementation was based on [4]. Comparison between the VRP and AODV is part of future work.

We developed a simulator[2] for wireless ad hoc networks modelling events above the physical layer. The simulator proceeds in *rounds*. In the link-layer implementation a random subset of not-interfering mobiles is selected for each round, and these mobiles transmit their packets. Note that only mobiles that have some packets to transmit in this round are considered.

This model is a simplification of the 802.11 DCF that use Request-to-Send (RTS) and Clear-to-Send (CTS) control packets for "unicast" data transmission to a neighboring node. The RTS/CTS is used to implement a form of *virtual carrier sensing* and channel reservation to reduce the *hidden terminal problem* [2].

Each round of the simulator lasts for the time required to send one packet. In the simulations the link bandwidth was set to $1\text{Mb}/\text{s}$ and the packet size to 512 byte. Hence, the round duration was set to $1/256$ seconds.

Traffic is modelled as a number of randomly selected source/destination nodes, where each sender generates a Constant Bit Rate (CBR) traffic. The number of senders is constant during the whole simulation experiment, while the duration of each connection is randomly selected in the interval $(0, CBRLen)$. Simulation experiments were performed under different traffic loads.

[2]The simulator can be downloaded from http://www.di.unipi.it/∼ste/AdHocSimulator.tgz

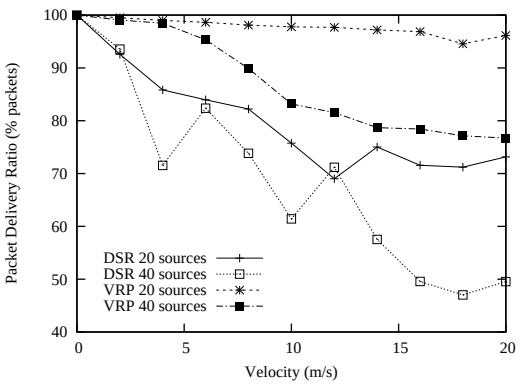

Figure 3: Comparison of the average packet delivery ratio of DSR and VRP with 20 or 40 CBR sources, each of duration of 15 seconds.

In particular we considered experiments with 20 and 40 sources, combined with $CBRLen$ of 15sec and 60sec.

The mobility model used is the *random waypoint* model [12] in a square field. In this model each mobile starts in a random location paused for *pause time*. When the pause time expires the mobile moves to a randomly chosen destination inside the area with a random speed. The speed is uniformly distributed between $(0, vmax)$. When the mobile reaches the destination, it pauses again for *pause time* then it restarts moving. We set *pause time* to 1sec and varied $vmax$ from 0 (static networks) to 20 m/s.

We performed experiments with $n = 75$ mobiles, each with a transmission range set to 250m, in a field of side 1000m. The virtual structure chosen for the simulations is the Ring of Rings with $x = 3$ rings and $y = 25$ mobiles per ring in the 75 mobiles network. The number of scouts for VRP has been set to $s = 5$.

In the comparison between VRP and DSR we considered the *Packet delivery ratio*, the *Average Route Discovery delay* and the *Normalized routing load*, where the Packet delivery ratio is defined as the ratio between the delivered data packets over the number of packets generated by each CBR source, and the Normalized routing load is defined as the ratio between the total number of routing packets transmissions over the number of delivered data packets.

Figure 3 (CBR of duration of 15 sec.), and Figure 4 (CBR of duration of 60 sec.) compare the packet delivery ratio of VRP with DSR. It is seen that VRP outperforms DSR in the packet delivery ratio. Also, the high delivery ratio of VRP is substantially unaffected by the mobiles velocity. The major cause of the lower delivery ratio of DSR is high percentage of messages dropped due to buffer overflows. This percentage is depicted in Figure 5, with buffer sizes ranging from 300 to 1000, and with a data traffic generated by 40 simultaneous CBR connections, where each connection has a duration of 15 seconds. Since the route discovery in DSR

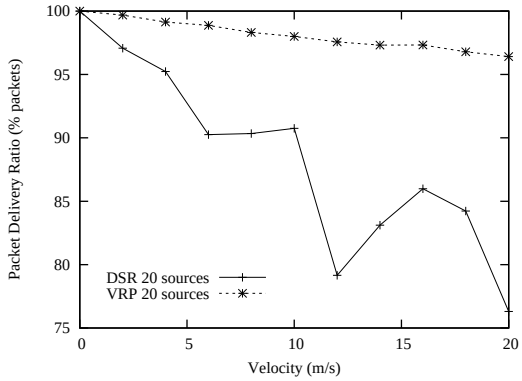

Figure 4: Comparison of the average packet delivery ratio of DSR and VRP with 20 CBR sources, each of duration of 60 seconds.

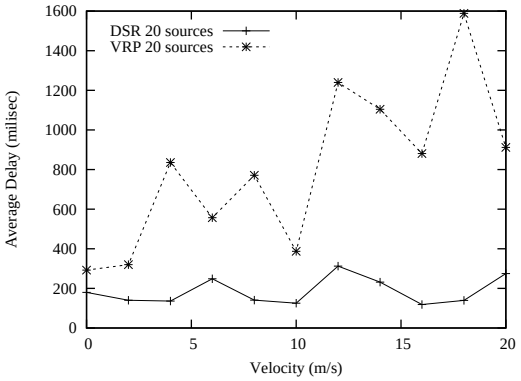

Figure 6: Comparison of the average delay, between DSR and VRP with 20 CBR connections, each of duration of 60 seconds.

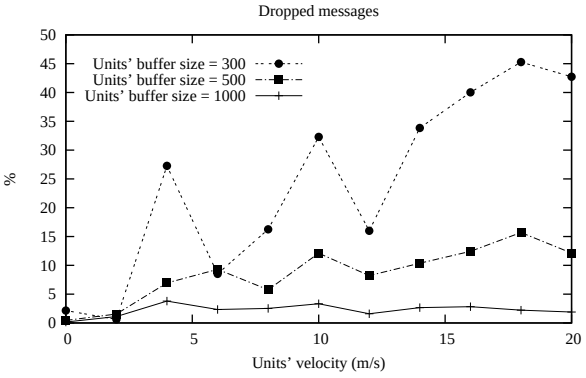

Figure 5: Percentage of dropped data messages due to buffer overflows in DSR with 40 CBR connections, each of duration of 15 seconds

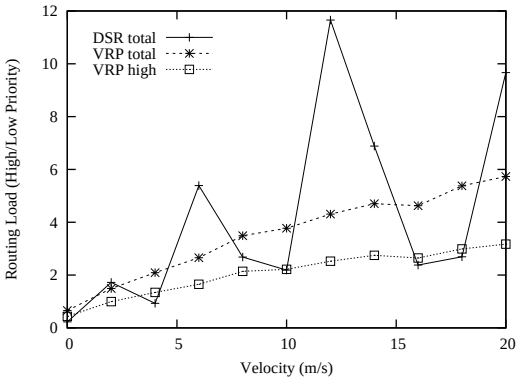

Figure 7: Comparison of normalized routing load between VRP and DSR, evidencing high priority overhead of VRP. Simulation with 20 CBR connections of duration 60 seconds.

is based on flooding and all messages have the same priority, the number of simultaneous flooding, with a high traffic load, consume all the available bandwidth. This induces the overflow of data message buffers in most of the nodes.

The Average Route Discovery delay to establish routes in VRP and in DSR is compared in the plot of Figure 6. It is noticed that the average delay of DSR is always smaller than the average delay of VRP. This is because DSR always perform a flooding to find a route, while VRP forwards unicast RTRANS messages along routes translating virtual paths, and such routes may be relatively long and may contain loops. Also, the source may send several RTRANS messages to establish a single route, and may eventually performs a flooding in the case that all RTRANS fail. However, when VRP succeeds in establishing the route to the destination with the first RTRANS message, the delay of the VRP is close to the delay of the DSR.

The plot in Figure 7 compare the normalized routing load. It is noticed that VRP performs better than DSR. Moreover, it should be observed, that a significant part of the traffic generated by VRP is due to low priority messages, and the amount of traffic due to high priority messages is lower than the amount of traffic of DSR.

6 Conclusions

A new Routing Protocol for wireless, ad hoc networks, named VRP has been presented. VRP has been developed with the goal of providing reliable routing, defined as the ability to provide almost 100% packet delivery rate. To this purpose VRP incorporates several strategies aimed at reducing the network congestion due to route discovery and maintenance, usually based on flooding-based protocols.

VRP has been evaluated and compared with the Dynamic Source Routing protocol (DSR) by means of simulation. It is seen that the delivery rate of VRP is always much above the delivery rate of DSR, although at the expense of increased delay in route discovery. Furthermore, the routing overhead due to high priority messages of VRP is lower than the overhead of DSR under different network loads, while the total overhead of VRP is lower than the overhead of DSR only under high network loads.

Future work includes the further development and optimization of VRP and extended comparison of VRP with other reactive and hybrid routing protocols.

References

[1] L.M. Feeney and M. Nilsson. Investigating the energy consumption of a wireless network interface in an ad hoc networking environment. In *IEEE 20th INFOCOM, vol. 3, Anchorage AK*, pages 1548 –1557, April 2001.

[2] M. S. Gast. 802.11 *Wireless Networks*. O'Really, 2002.

[3] IEEE. *Draft Supplement to Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*, November 2002.

[4] D. B. Johnson, D.A. Maltz, and Y. C. Hu. The dynamic source routing protocol for mobile ad hoc networks (DSR). IETF Internet Draft, version 9.

[5] 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.

[6] S.J. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. *ACM Mobile Networks and Applications*, 7:441–453, 2002.

[7] D. A. Maltz, J. Broch, J. Jetcheva, , and D. B. Johnson. The effect of on-demand behavior in routing protocols for multihop wireless ad hoc networks. *IEEE JSAC - Special Issue on Wireless Ad Hoc Networks*, 17(8):1439–1453, August 1999.

[8] S. Mangold, S. Choi, P. May, O. Klein, G. Hietz, and L. Stibor. IEEE 802.11e wireless LAN for quality of service. In *Proc. European Wireless Symposium*, 2002.

[9] Shree Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, 1996.

[10] Vincent D. Park and M. Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM (3)*, pages 1405–1413, 1997.

[11] C. Perkins and E. M. Royer. Ad-hoc on-demand distance vector (AODV) routing. In *IEEE WMCSA 99*, pages 90–100, 1999.

[12] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications*, pages 16–28, February 2001.

[13] Charles Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, 1994.

[14] J.P. Singh S. Agarwal, A. Ahuja and R. Shorey. Route-lifetime assessment based routing (RABR) protocol for mobile ad hoc networks. In *IEEE International Conference on Communications (ICC), New Orleans, LA*, pages 1697–1701, 2000.

[15] V.R. Syrotiuk S. Basagni, I. Chlamtac and B.A. Woodward. A distance routing effect algorithm for mobility (DREAM). In *ACM/IEEE MOBICOM 98*, pages 76–84, 1998.

[16] I. M. Vinogradov. *An Introduction to the Theory of Numbers*. Pergamon Press, London & New York, 1955.