Reducing Latency and Overhead of Route Repair with Controlled Flooding

Luís Henrique M. K. Costa* (luish@gta.ufrj.br)
GTA/COPPE/EE – Universidade Federal do Rio de Janeiro
P.O. Box 68504 – 21945-970 – Rio de Janeiro – RJ – Brasil
Phone: +55 21 2260-5010 (ext. 240)
Fax: +55 21 2290-6626

Marcelo Dias de Amorim (amorim@rp.lip6.fr)
Laboratoire LIP6 – Université Pierre et Marie Curie
8, rue du Capitaine Scott – 75015 – Paris – France
Phone: +33 1 44.27.87.72
Fax: +33 1 44.27.74.95

Serge Fdida (sf@rp.lip6.fr)
Laboratoire LIP6 – Université Pierre et Marie Curie
8, rue du Capitaine Scott – 75015 – Paris – France
Phone: +33 1 44.27.30.58
Fax: +33 1 44.27.53.53

Abstract. Ad hoc routing protocols that use broadcast for route discovery may be inefficient if the path between any source-destination pair is frequently broken. We propose and evaluate a simple mechanism that allows fast route repair in on demand ad hoc routing protocols. We apply our proposal to the Ad hoc On-demand Distance Vector (AODV) routing protocol. The proposed system is based on the Controlled Flooding (CF) framework, where alternative routes are established around the main original path between source-destination pairs. With alternative routing, data packets are forwarded through a secondary path without requiring the source to re-flood the whole network, as may be the case in AODV. We are interested in one-level alternative routing. We show that our proposal reduces the connection disruption probability as well as the frequency of broadcasts.

Keywords: Controlled flooding, alternative routing, connectivity maintenance, broadcast.

1. Introduction

A major challenge of multi-hop wireless networks is the design of efficient routing protocols that dynamically find routes between two communicating nodes [9, 23, 24, 27]. According to the routing strategy, ad hoc routing protocols fall into two categories: proactive and reactive. Proactive routing protocols share routing information even if there are no specific requests for a route [1, 5, 9, 10], just like a classical

* Contact author.

Internet routing protocol does. This strategy continuously produces control traffic (which should be avoided for wireless networks) but has desirable characteristics like low latency route access, QoS support, and monitoring for real-time applications. On the other hand, reactive (or on-demand) routing protocols operate only when there is an explicit request for a route [9, 12, 19, 21]. This strategy significantly reduces the memory consumption in the nodes and only generates control traffic when needed. Such proactive routing protocols have been extensively evaluated in the literature, and are currently subject of discussion at the Mobile Ad hoc NETworks (MANET) working group [12, 19] of the IETF. In this paper, we concentrate on the on-demand routing protocols, and more specifically the Ad hoc On-demand Distance Vector routing protocol (AODV) [19, 20].

On-demand ad hoc routing protocols typically use broadcast messages to discover routes between two communicating nodes [12, 19]. The Dynamic Source Routing protocol (DSR) [11, 12] uses source routing, where the source knows the entire path to the destination and the exact sequence of intermediate nodes is indicated in the data packet's header. In order to obtain a path to the destination, the source floods the network with Route REQuest (RREQ) packets until the destination is reached. As RREQ packets travel the network, they store the identifier of each node traversed. The destination responds with a Route REPly (RREP) packet to the source using the reverse path. AODV uses the same principle of route requests and replies, but differs from DSR because AODV uses traditional routing tables. Upon reception of an AODV RREQ packet, an intermediate node stores a route to the source (*backward learning*) and forwards the request. When the destination receives the RREQ packet, it responds using the path traced by the query.

In spite of providing fast route discovery, broadcast has several inconveniences. If broadcast is implemented by flooding, serious problems like redundancy, contention, and collision are frequently observed. In a typical mobile ad hoc network, the resource consumption of control packets has a significant impact because of the low-bandwidth links and the power-limited terminals. A number of protocol-specific techniques have been proposed to improve the performance of broadcast-based protocols [4, 13, 18, 22]. Nevertheless, little attention has been given to the broadcast mechanism itself. Although the proposed techniques improve the performance of the system in some situations, broadcast is always performed whenever a node requires a path to an unknown destination. In on-demand routing protocols, for instance, a node typically maintains a route to a specific destination until it receives a path error message from downstream nodes or timeout for a specific path occurs. If a message has no means to reach the destination, the source performs a new broadcast to discover a valid alternative route.

Broadcast is difficult to avoid if the source has no idea of the current location of the destination. Nevertheless, broadcasting the entire network for each route discovery is inefficient. We argue in this paper that even if broadcast has to be performed for the first time a route is searched, it can be avoided for subsequent ones [6]. In this paper we propose the Controlled Flooding (CF) technique to increase the probability that a source finds a path to a specific destination without requiring the network-wide broadcast of request messages. With CF, alternative routes are configured around the main path between a pair source-destination. In this paper, we consider only alternative routes that are one hop away from the main path.

CF can be used in two situations: route discovery and route repair. For the route discovery procedure, the search is "directed" toward the previous position of the destination, without flooding the entire network. Suppose that a source, s, has a route to a destination, d. If after some time this path is broken, it is likely that d is not (very) far from its original location, if node mobility is limited. Thus, the network-wide broadcast can be avoided since some nodes (at least the ones that were in the previous path) have a clue of d's current location.

We address in this paper the route repair mechanism of on-demand routing protocols, which is directly related to the connectivity maintenance. With mobility, the communication between two nodes may be frequently stopped, which is prejudicial for delay-sensitive applications since data packets must be stored, or dropped, until another path is constructed. The ideal configuration would be to promptly forward the messages through another path. In proactive protocols, this is straightforward. Nevertheless, proactive approaches are efficient for small population, but do not scale for larger ad hoc networks. In the case of reactive on-demand protocols, additional mechanisms have to be provided since we have, *a priori*, only one route per active destination. In our approach, besides the main path ($\mathcal{S}_{s,d}$) between the source s and the destination d, the network also establishes some alternative paths "around" $\mathcal{S}_{s,d}$. As soon as a node detects that its next hop toward the destination has failed, it promptly forwards data packets to one of the alternative routes.

We evaluate our approach in different network scenarios. We vary the number of hosts, the density of nodes, and the mobility pattern. The AODV-CF protocol is compared with the original AODV protocol in terms of connection disruption time and the frequency of network-wide broadcasts. The results show that AODV-CF performs better than pure AODV and that the one-level controlled flooding approach is an attractive technique to be combined with AODV.

This paper is organized as follows. In Section 2, we identify the main problems of flooding the network for route discovery. In Section 3 we describe the controlled flooding approach and its main advantages over traditional expanding ring search as well as the implementation issues. In Section 4 we present the results obtained from the simulation of our proposal applied to the AODV protocol. Finally, conclusions and future work directions are presented in Section 5.

2. Route discovery using broadcast

Network-wide broadcast (often implemented by flooding the network), in spite of its high overhead, typically obtains good results for reliable delivery and is simple to implement. Consequently, this technique has been extensively applied in different situations. In a mobile ad hoc network in particular, broadcast is a common operation in many applications, such as sending an alarm signal, paging a particular host, reliable multicast, and routing [17].

Nevertheless, serious problems are observed when flooding is performed without rigid control, such as redundancy, contention, and collision [14, 25]. In [25], the authors identify three main causes of these problems. First, a single transmission is received by an arbitrary number of nodes because of the omni-directional characteristic of the antennas. This leads to redundant rebroadcasts. Second, if rebroadcasting nodes are close to each other they may experience significant contention. Third, the timing of rebroadcasts is highly correlated, which leads to collision. The authors refer to these problems as the broadcast storm problem. They also propose some possible solutions for these problems, such as the reduction of the probability of redundant rebroadcasts and the differentiation of the timing of rebroadcasts.

Mobile ad hoc routing protocols often use flooding for route discovery. In this section, we briefly describe how flooding is performed and focus only on the basic functioning of the technique. Further details on improvements and variations can be found in the literature [25, 16, 26].

Suppose wireless links with omni-directional antennas. In this case, each transmitted packet is received by every node within the range of the sender. Each node rebroadcasts the message to its neighbors unless it is the destination or it already has a route to the destination. In order to avoid loops, each message is assigned an identifier (or sequence number), and nodes do not re-broadcast messages more than once. Thus, for a network with n nodes, the number of transmissions is $O(n)$.

To reduce the portion of the network flooded with broadcast messages, different routing protocols (e.g. AODV) use the Expanding Ring Search (ERS) [3] technique during the route discovery phase. The idea is to limit the scope of the broadcast message with the utilization of the packet's TTL field. ERS starts with a fixed value for the TTL, which is progressively incremented each time the search fails, until the destination is found or a maximum TTL is reached. An inherent problem of expanding ring search is the choice of the TTL values. Since the source does not know the exact location of the destination, it is difficult to determine the range of the search. On the one hand, if the range is short, further searches will be required until the destination is found. In this case, a potentially large number of links will be used unnecessarily and the route discovery latency will be higher. On the other hand, if the TTL is too large, network resources will be wasted in case of closely located destinations.

AODV may perform ERS in two situations. The first one is when a new connection must be established and the source does not have a route to the destination. The second one is a consequence of route loss, due to node mobility or failure. When the route is broken, a new broadcast is performed in order to find a new route. This frequent use of broadcast may degrade the overall system performance. Nevertheless, it is inherently difficult to determine the first route between two communicating nodes without performing a network-wide search. Thus, we argue that broadcast might be performed for the first search, but can be avoided for subsequent ones.

Suppose that a source, s, has a route to a destination, d. If, after some time, this path is broken, it is likely that d is not (very) far from the original location if node mobility is limited. Performing broadcast to discover another path might not be the best solution because some nodes (at least the ones that were in the previous path) have a clue of the current location of d. In this case, the search can be "directed" toward the previous position of the destination, without flooding the entire network.

Recent proposals have investigated the overhead introduced by broadcast-based route discovery for AODV. In [26], Yi and Gerla propose a variant of the AODV protocol where RREQ packets are forwarded by only a subset of nodes (called dominating nodes). This selective flooding is based on the Passive Clustering (PC) scheme and leads to performance improvements when compared to the pure AODV protocol, but supposes a hierarchy that has to be maintained.

The Ad hoc On-demand Multipath Distance Vector (AOMDV) routing proposed by Marina and Das [16] builds multiples paths using duplicate RREQ (RouteREQuest) copies during AODV route discovery. These paths are redundant and therefore a new route discovery is only performed when all paths fail. Thus, the goal of AOMDV is, as for AODV-CF, to avoid the flooding of the entire network for route discovery. Nevertheless, in AOMDV, when all alternative paths fail the source performs a new network wide broadcast. The controlled flooding approach on the other hand aims the overhead reduction when a new route discovery is needed. In this paper we present both uses of controlled flooding, but focus on the alternative routing.

3. Controlled Flooding

Controlled Flooding (CF), the technique we propose in this section, aims at reducing the amount of control traffic generated by on-demand ad hoc routing protocols during the route repair procedure. CF increases the probability of finding a path to a destination in case of node failure or mobility. Additionally, CF can be used to improve the robustness of on-demand routing protocols by the construction of alternative (or backup) routes. The CF approach is particularly interesting when nodes move regularly and paths are long. In such cases, the probability of route failures increases and the routing mechanism must react to find another valid path. This may lead to high overhead if the route discovery mechanism does not scale well (e.g. flooding). Using CF, nodes keep few additional states but the global communication overhead is reduced.

CF borrows its basic idea from the Oriented Multicast (OM) framework proposed by Magoni and Pansiot [15], where modified multicast trees are used to direct service requests. Nevertheless, CF has a different objective as we explain in the remaining of this section. The Oriented Multicast [15] protocol is used as a network-layer agent search service for locating agents that can be everywhere in the Internet. Agents are specific nodes in the network that have to be found before data transmission, such as graft nodes for multicast trees or retransmission nodes for reliable multicast communication. OM is close to the Reverse Path Forwarding (RPF) [7], but implements a much more efficient control over flooding. OM supposes that services can be found in nodes which are closer to the client (multicast initiator) than the target node (destination). Indeed, in many applications, specific nodes (or services) can be found around some known entity in the network. The objective is to reduce the delay between the client and the service provider. In OM, a message is multicast to nodes inside a limited area around the shortest path between the source and the destination. This avoids resource wasting introduced by algorithms like IP-broadcast or

Expanding Ring Search (ERS) [3]. A message is then sent to this entity and to some nodes around this path. The authors show that OM leads to good performance improvements when compared to IP-broadcast or ERS.

The main difference between OM and CF is that OM supposes some known entity in the network that serves as the target point of the oriented multicast message. The source must know the shortest-path to the destination and depends on some underlying routing algorithm. Instead, CF is used to discover and maintain paths based on the previous state of the network. Once a path has been determined, subsequent routes are obtained without re-flooding the entire network.

The CF algorithm supposes the existence of a first path from the source to the destination. Once this path is established, the CF algorithm determines alternative routes that will serve in case the main route fails. The next section describes the implementation details of AODV-CF, the protocol obtained from the application of the controlled flooding approach to AODV.

3.1. AODV-CF Implementation Details

This section gives the main implementation details of AODV-CF. The algorithm described here is responsible by the construction of the alternative (or backup) routes used for local route repair by AODV-CF. Alternative route maintenance is done through AODV's neighbor sensing.

First, we consider the alternative route construction. An AODV-CF route entry has two additional variables compared to AODV's. The first one is an array that stores the set of alternative routes (next hops). The other variable is a "primary flag". Given a route to d, the primary flag indicates if the route is a main route, an alternative route, or a "neutral" route to d. A neutral route does not carry data, as the main route, neither is a backup route. An example is a reverse route installed by AODV after the reception of a RREQ packet.

AODV-CF uses two additional messages to construct alternative routes: `Controlled` and `Controlled-Ack`. Both messages are sent with TTL equal to 1. Nodes in the main route are the only ones that produce `Controlled` messages, whereas nodes in the backup route produce `Controlled-Ack` messages. Each message is uniquely identified by a sequence number (just as for AODV RREQs). Thus, a node can control which messages were already treated and eventually forwarded. Other implementation details, including loop avoidance tests, are omitted to simplify the description.

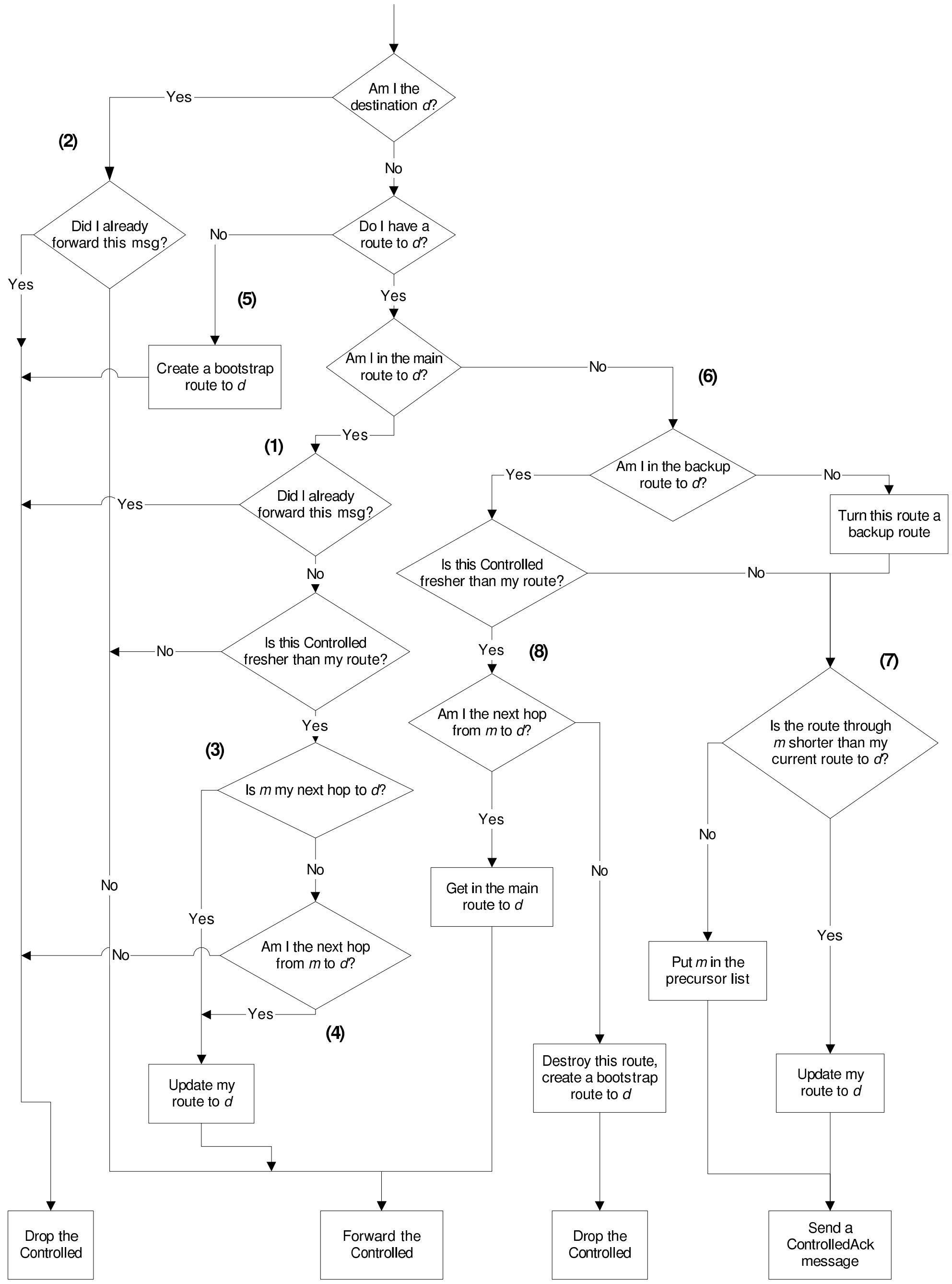


Figure 1. Node n received a `Controlled` message from node m (route to d with next hop h).

Fig. 1 summarizes the `Controlled` message processing rules. The alternative route construction is triggered by the node that requested a route. When this node receives an AODV Reply message, it sends the first `Controlled` message. Suppose that the node n received a `Controlled` message from node m (m is in the main route). This message advertises that m has a route to d, with $l_{m,d}$ hops, and that node h is the next hop from m to d. Additionally, the `Controlled` message has the sequence number (age) of the advertised route.

First, suppose that node n is in the main route. Each node in the main route, including the destination d, forwards the `Controlled` message once ((1) and (2) in Fig. 1). A node in the main route that receives a `Controlled` message with a sequence number larger than its own updates the route to d if the received `Controlled` was sent by the next node, or by the previous node in the route to d ((3) and (4) in Fig. 1). When a node performs a route repair, it uses one of the stored alternative routes, increases the route sequence number, and sends a `Controlled` message. With rules (3) and (4), nodes that are still in the main route will update its sequence number, but not the nodes that are no more in the main route to d. These rules contribute to loop avoidance.

Suppose now that node n is not in the main route and receives a `Controlled` message ((5) in Fig. 1). If n has no route to d, it creates a "bootstrap" route pointing to m (who issued the `Controlled`). The bootstrap route is marked DOWN (AODV standard flag) and BACKUP (AODV-CF primary flag). Suppose now that n has a non-primary route to d ((6) in Fig. 1). If the `Controlled` sequence number is the same as n's route, n checks if m provides a shorter route to d. If yes, n updates the backup route to d. If not, n puts m in its AODV precursor list, because m potentially uses n as next hop to d ((7) in Fig. 1). If the `Controlled` is fresher than n's route ((8) in Fig. 1), n checks if it is the next hop used by m (the node in the main route). If yes, it means that n is now in the main route to d. If not, n destroys its route to d and behaves as in the beginning of the alternative route construction process, creating a bootstrap route.

Suppose now that node n received a `Controlled-Ack` message from node m (Fig. 2). Nodes that are not in the main route to d ignore the `Controlled-Ack` messages (Fig. 2 (1)). (The destination d also ignores these messages.) If n is in the main route with the same sequence number as the `Controlled-Ack` message (the announced route has the same age as n's route - (2) in Fig. 2), there are two possibilities. If n is the next hop used by m to reach d, then n inserts m in its precursor list ((3) in Fig. 2). If not, n can possibly add an alternative route to d through m ((4) in Fig. 2). This will be possible if the route to d

through m is shorter than, equal to, or exactly one-hop longer than, n's main route to d. This is because alternative routes are one-hop away from the main route. If n has a route to d which is older than the route announced in the `Controlled-Ack` message ((5) in Fig. 2), then n removes its alternative routes to d. In this case, a route fix probably occurred. Thus, n will eventually update the route to d after receiving a `Controlled` message with a greater sequence number, and reconstruct the alternative routes. Anyway, the alternative route list is currently invalid.

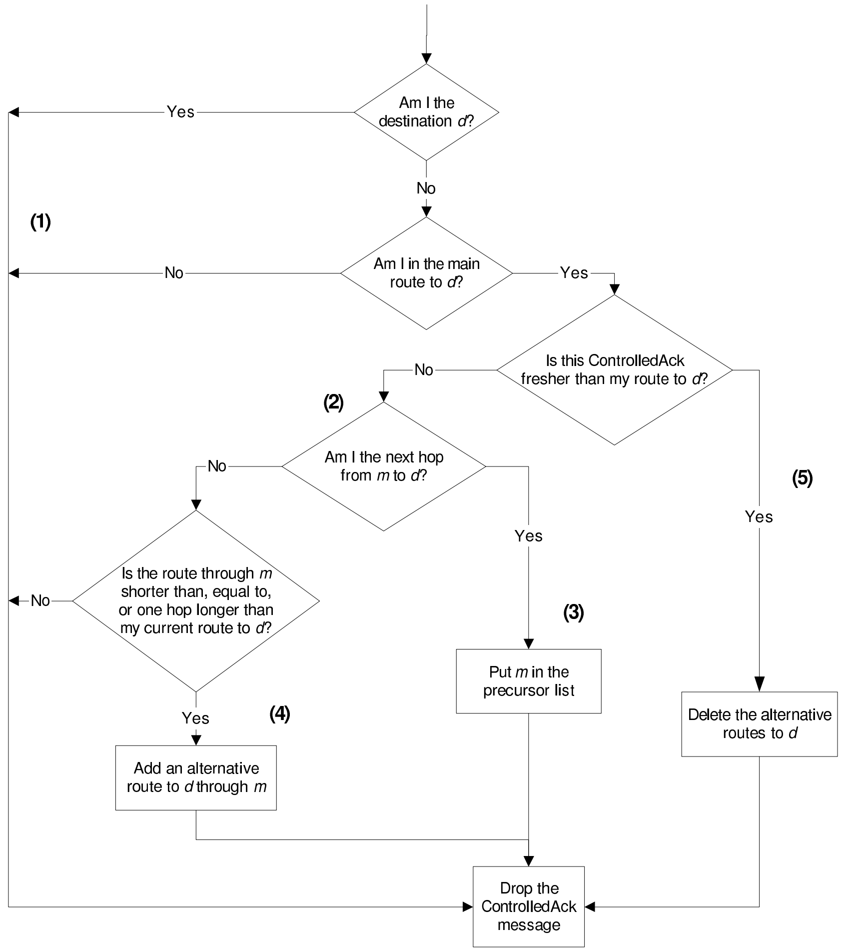


Figure 2. Node n received a `Controlled-Ack` message from node m (route to d with next hop h).

We consider now the maintenance of alternative routes. The problem of alternative route maintenance is that in AODV data traffic is used to keep routes active (to refresh routes). In AODV-CF, an alternative route does not carry data (not until a route repair occurs). For alternative routes that are one-hop away from the main route, there is a simple solution. The idea is then that a backup node keeps backup routes alive as long as its neighbor, used as next hop in the backup route, is alive. This is done through AODV's neighbor sensing, i.e., through AODV `Hello` messages or link layer detection, depending on the MAC layer used. In both cases, there is no additional signaling involved. Obviously, the mechanism is simple because alternative routes are one hop away from the main route. Alternative routes with other patterns are subject of future work.

3.2. Applications of CF

As previously mentioned, the controlled flooding technique can be used for two purposes: route discovery and route repair. This section presents the two applications.

3.2.1. *Route discovery*

The route discovery phase is performed each time the source looses its route to the destination and there is no alternative routes. In this case, the source must start a new search procedure to obtain a valid route. If the protocol uses ERS (e.g. AODV), the search is performed "omni-directionally", i.e. every node within TTL hops away from the source receives the RREQ message. The protocol does not take into account the previous "position" of the destination.

The Controlled Flooding mechanism intends to overcome this particular memoryless behavior of traditional search algorithms. Consider the scenario depicted in Fig. 3(a). The main route is represented by the solid line and alternative routes by dashed lines. In the same way, nodes in the main route are represented in black and backup nodes in gray. This means that these nodes have some information about the destination's location. In Fig. 3(b), the destination moves and looses the route to the source. The latter initiates a new route discovery using the information contained in the nodes around the previous main path and the backup nodes. This is shown in Fig. 3(c). Finally, the destination node has been located and a new route can be established (Fig. 3(d)).

Note that the search phase does not produce messages in regions where the destination is unlikely to be found. If the source had applied a naive flooding to discover a new route to the destination, we would have had the situation shown in Fig 4. In Fig 4(a), the search message is broadcast through the entire network. This is clearly inefficient. We can also observe by comparing Figs. 3(d) and 4(b) that the resulting paths may be different. In AODV, the route converges to the shortest

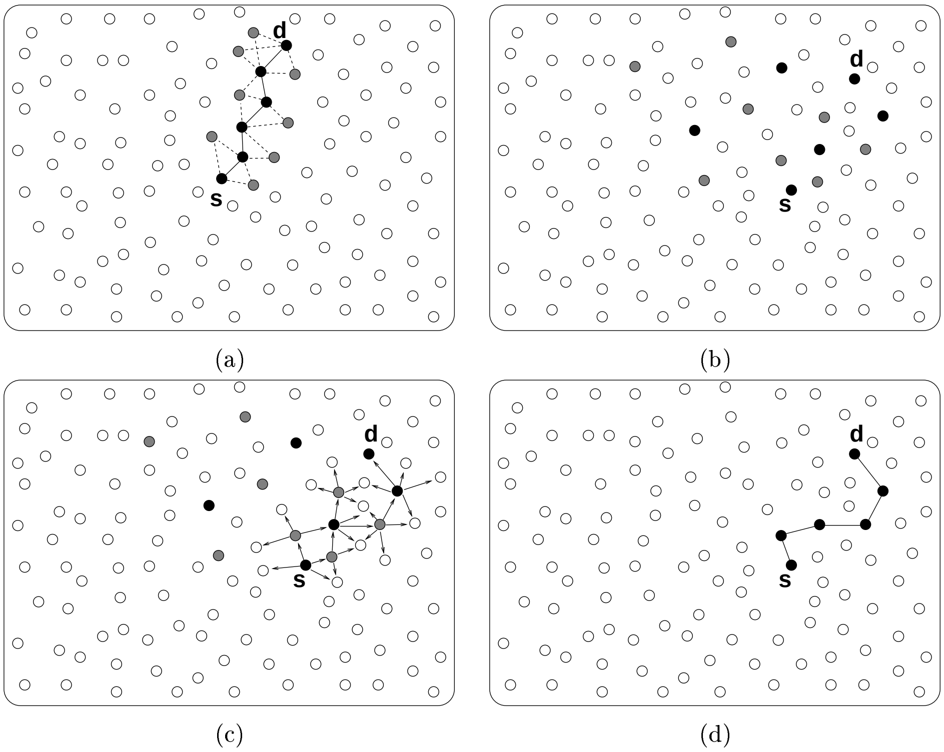


Figure 3. The CF mechanism for searching nodes.

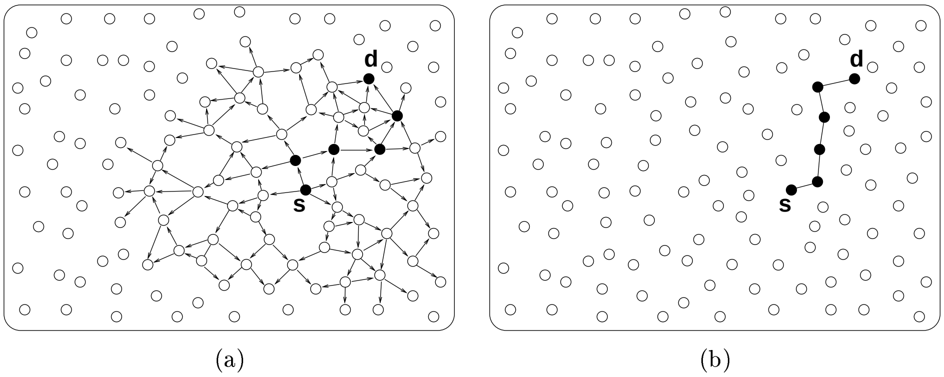


Figure 4. The naive flooding searching nodes.

path in the entire network, while using CF the resulting route converges to the shortest path inside the scope region. Nevertheless, the cost to obtain a global shortest path is a much higher overhead.

The route discovery functionality is one of the advantages of using Controlled Flooding. Nevertheless, in this paper, we will focus on the route repair property of the CF approach presented in the following.

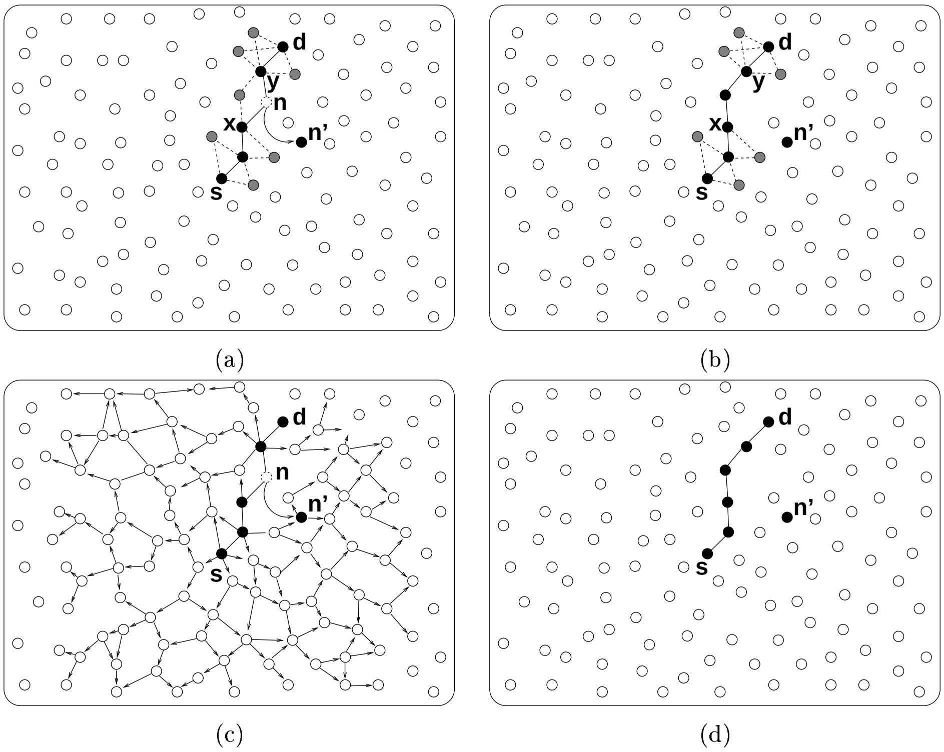


Figure 5. Network-wide broadcast versus Controlled Flooding.

3.2.2. *Route repair*

AODV-CF can apply different strategies for determining alternative routes. These strategies differ on the scope of the controlled flooding messages around the main path between the source s and the destination d, $\mathcal{S}_{s,d}$. Fig. 5 shows how a path can be promptly found after a node failure. We suppose that the first path has already been established using network-wide broadcast. The first scheme (Fig. 5(a)) represents the main route between the source s and the destination d. When a node in $\mathcal{S}_{s,d}$ fails or moves (node n in the figure), the main route between nodes x and y is lost. Since there is an alternative path between these two nodes, a new path between them can be dynamically established (Fig. 5(b)). Thus, no extra delay and communication overhead are generated. Fig. 5(c) shows the same scenario of Fig. 5(a) but without alternative paths. When node n moves and does not further belong to the main path, the source performs a new flooding (Fig. 5(d)).

With low to moderate mobility, one-hop alternative routes can be effective. Suppose the scenario of Fig. 6(a). Let i and j be consecutive nodes in the main path between the source and the destination and

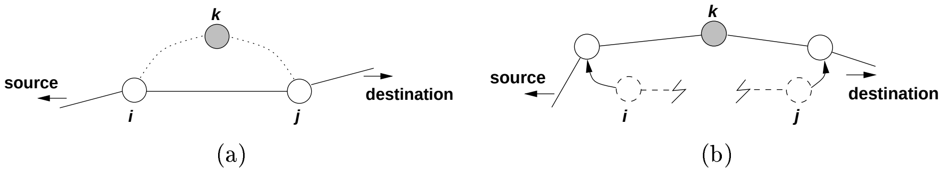


Figure 6. Route repair using one-level alternative route.

k be a node in the neighborhood of both i and j, as shows Fig. 6(a). Since j is closer to the destination than k, i chooses j to be its next hop in the main path. At the same time, node k is chosen as an alternative route, since k can be seen by both i and j. A few moments later, with node mobility, nodes i and j can no longer see each other – the main path between the source and the destination is broken – but node k can still be seen by i and j. Then, node i sets k as the next hop and the communication can resume (see Fig. 6(b)). The route has been promptly repaired at the cost of some additional state without extra delay.

This type of route repair is likely to happen because broadcasting request messages typically leads to a path which is equal (or close to) the shortest path between the source and the destination. In the shortest path, the next hop of each node is, in average, the farthest possible neighbor. In many situations, consecutive hops work in an extreme configuration, close to the limit of the nodes' power range. With node mobility, this path can be easily lost even under light network dynamics.

3.3. Open features

As we will see through a number of simulation analysis, using one-level alternative routing does reduce the latency and the overhead in AODV. Nevertheless, one can argue that the controlled flooding approach could be improved if the alternative paths were established in a wider area. In this case, the distance between an alternative path and the main path would be greater than one. This could be implemented through a π-level policy, where π is the maximum distance between the alternative and main routes. In this paper, we analyze the behavior of the controlled flooding for $\pi = 1$. Fig. 7(a) illustrates the mechanism with $\pi = 2$ and Fig. 7(b) shows the possible alternative paths for this scenario. By increasing π, we can also increase the robustness of the system. The larger the scope size π, the larger will be the number of nodes which have a clue of the destination's position. Thus, the probability of success when searching for a node increases. Also, more alternative routes can be established around the main path.

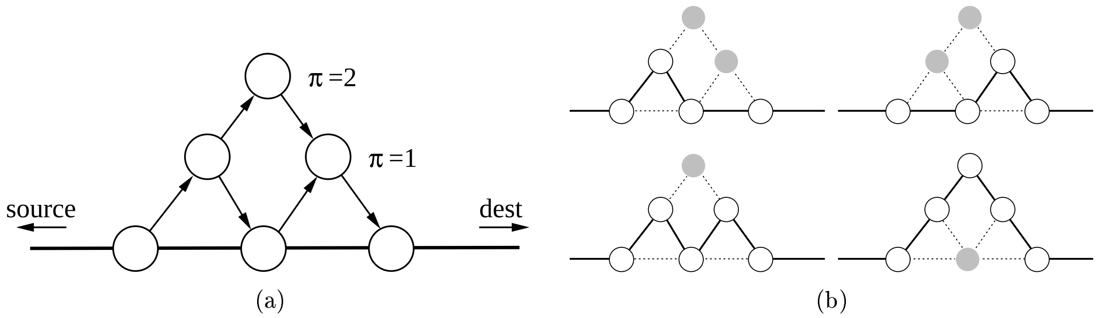


Figure 7. Alternative routes constructed with CF scope size $\pi = 2$.

Therefore, in order to increase the number of alternative routes in our scheme, one must increment the scope size. Nevertheless, increasing the scope size implies that other nodes are involved in the routing procedure. On the other hand, using low π reduces the number of involved nodes in alternative routes but increases the probability of connection disruption. These tradeoffs can be chosen by taking into account for example the density of nodes in a region or the node capacity and the mobility pattern.

4. Simulation Results

To evaluate the effectiveness of the Controlled Flooding approach, we used *ns* (Network Simulator) version 2.1b9 [8], which includes the Monarch/CMU implementation of AODV [2]. This implementation supports multi-hop wireless networks complete with physical, data link, and MAC layer models. We developed a new module to implement AODV-CF which inherits from AODV's original implementation.

In the following, we analyze the control traffic generated by AODV and AODV-CF, as well as the service disruption time. We show that AODV-CF results in lower overhead in terms of control messages and that the disruption periods are shorter (less packet drops).

4.1. Simulation scenario

In our simulations, we used the IEEE 802.11 MAC layer with the free-space radio propagation model. We used the mobility scenario generator provided by CMU in the *ns-2* distribution. In our scenarios every node is mobile.

We varied the network size from 50 to 150 nodes using a 25-node step. For the 50-node network, we used a radio range of 225m. This range was varied for the other network sizes in order to keep the average number of neighbors of a node constant. In that way, larger network sizes mean longer routes. Another possibility would be to keep the radio range constant and vary the size of the simulated physical area, but this choice leads to much longer running times for the mobility scenario generator. For each network size, we made 250 simulation runs. Nodes move in a 1000×1000 m square, using the random way-point movement model, with maximum speed of 5 m/s. We have two sets of scenarios. In the first one, which simulates low mobility, nodes have a 50 s pause time and each simulation run lasts for 500 simulated seconds. In the second set, to simulate higher mobility, nodes have a 10 s pause time and each simulation lasts for 200 seconds.

The scenarios have one source-destination pair where the source generates one 512-byte packet per second. We use only one traffic source because the goal of the simulations is to evaluate the effectiveness of AODV-CF in reducing service disruption (by the use of backup routes) and in route reconstruction. In the case of various source-destination pairs, both protocols can benefit from the possible existence of previously constructed routes in case the same destination is searched by a different source node.

4.2. Metrics

In our measurements, the control traffic includes every packet generated by the routing protocol. For AODV, there are three message types: `Route-Request`, `Route-Reply`, and `Route-Error`. AODV-CF uses two additional message types, `Controlled` and `Controlled-Ack`. `Hello` messages were used in our simulations for neighbor sensing. Nevertheless, the `Hello` messages were not taken into account in the control traffic because AODV and AODV-CF generate exactly the same amount of such messages.

The service disruption time was not directly measured. Instead, we measured the number of data packet losses. Service disruption time is proportional to the number of losses of the CBR traffic.

4.3. Results

Fig. 8 shows the total number of control packets generated by both AODV and AODV-CF using the simulation scenarios detailed above. In all cases, the CBR source sends a request for a route at $t = 30$ s. The communication lasts for the whole period of the simulation. Note that the results are integrated over a period of 10 seconds, i.e. the number of packets shown at time t is the total number of packets generated in the interval $](t-10), t]$.

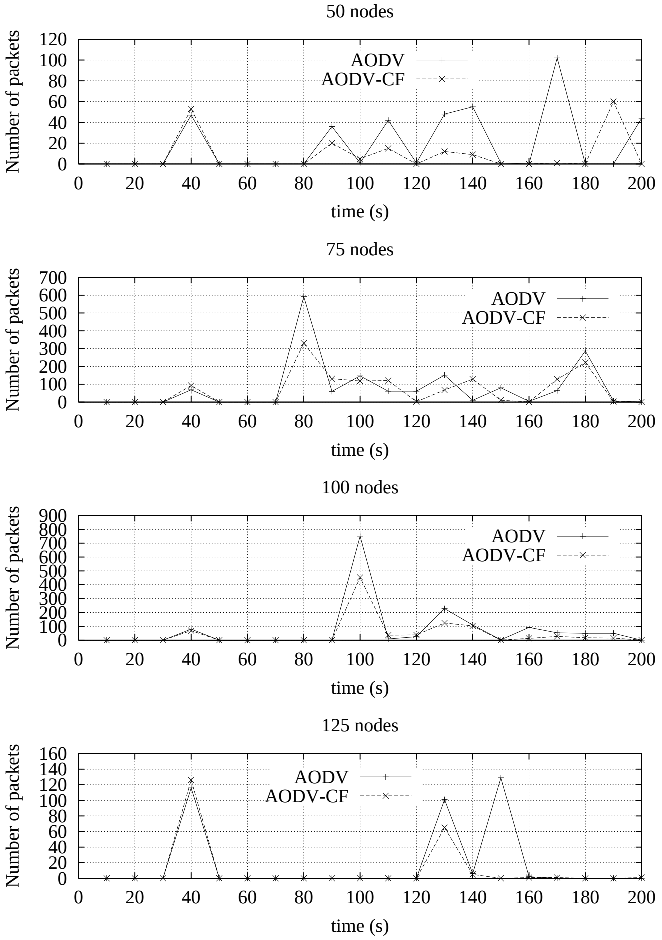


Figure 8. Control traffic.

These results reflect the operation of AODV-CF. At $t = 40$ s, the AODV-CF mechanism leads to more control messages than AODV. The difference, as expected, is due to the `Controlled` and `Controlled-Ack` messages generated by the Controlled Flooding algorithm. Nevertheless, this extra overhead is compensated by future gains, because the alternative routes established during this phase result in route repairs in all scenarios. For instance, in the 75-node topology this can be observed at 80 seconds. While the AODV protocol restarts a route discovery by flooding the network, AODV-CF fixes the lost path and forwards data through an alternative route. The control traffic generated by AODV-CF corresponds to the Controlled Flooding procedure performed after the path failure.

In the simulations, AODV-CF operates in promiscuous mode, which means that every node that receives two or more `Controlled` messages responds with a `Controlled-Ack`. If the network is dense, the number of acknowledgments may be arbitrarily large. This explains why sometimes in the simulations the number of control messages is large, even after a route repair (most of these messages are `Controlled-Ack` messages). Rules for limiting the number of alternative routes (and consequently, the number of `Controlled-Ack` messages) will be subject of future work.

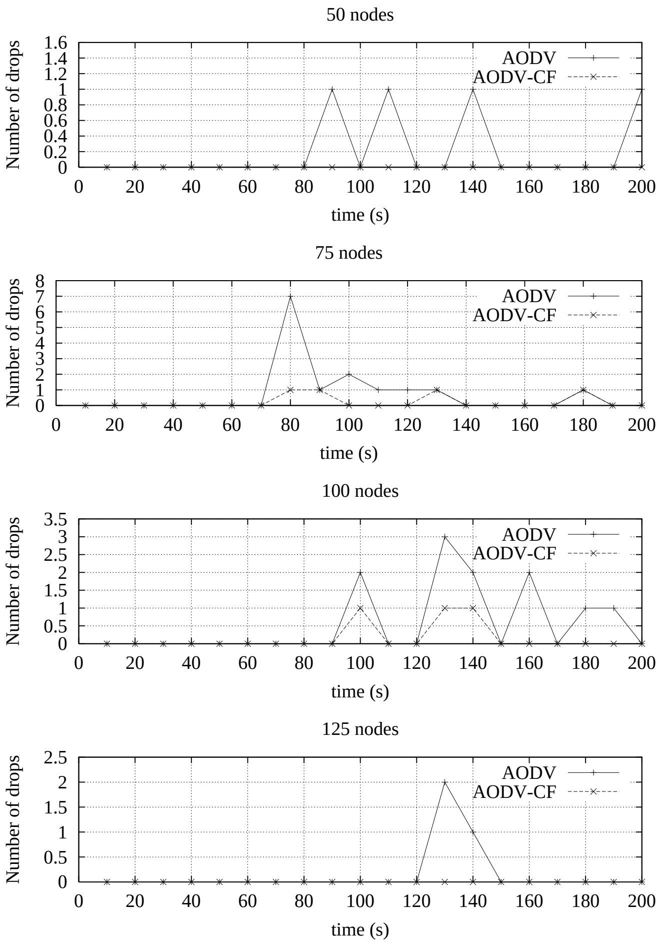
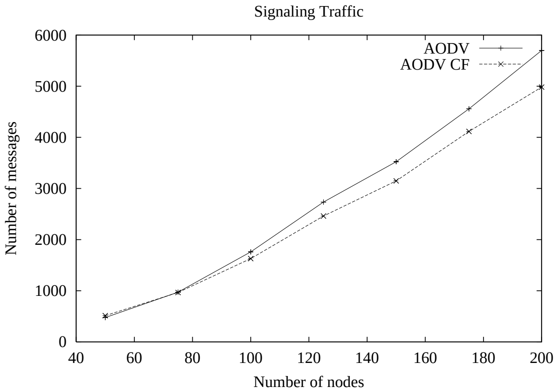
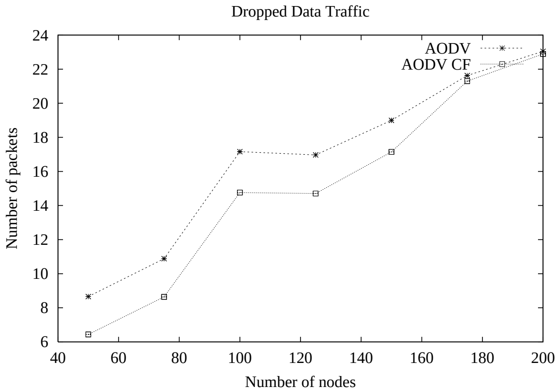


Figure 9. Data packet losses.

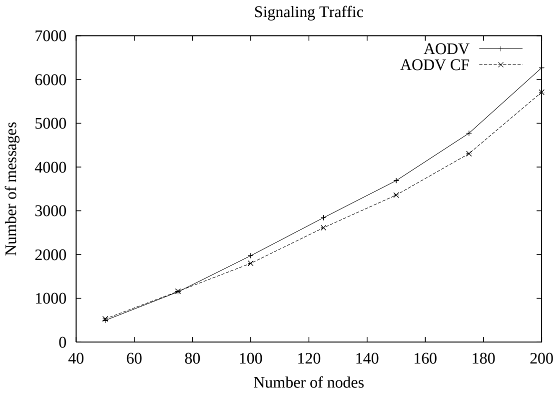
Fig. 9 shows the number of packet drops for the same scenarios of Fig. 8. The results reflect the disruption time when another route must be found after a path failure or when an alternative route is being set up. In all situations, AODV-CF outperforms AODV. One



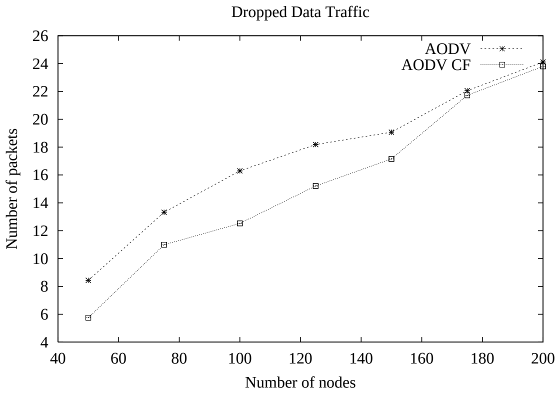
(a) low mobility (p = 50s d = 500s).



(b) low mobility (p = 50s d = 500s).



(c) medium mobility (p = 10s d = 200s).



(d) medium mobility (p = 10s d = 200s).

Figure 10. Average comparison of AODV and AODV-CF.

interesting observation is that sometimes a node forwards data packets to alternative routes that are no more valid. This provokes some losses until the node finds a valid alternative path. This can be observed for instance at $t = 100$ s in the scenario with 100 nodes.

We also compared the average results obtained from the 250 runs of each scenario. Figs. 10(a) and 10(b) show the results for the low mobility scenario, where nodes move at 5 m/s with 50 s pause time, and each run simulates 500 seconds (therefore, each node moves 10 times in average during the simulation, or 0.02 times/s). Figs. 10(c) and 10(d) correspond to higher mobility scenarios, where nodes move at the same speed but with 10 s pause time and simulation runs of 200 s (each node moves 20 times, or 0.1 times/s).

Figs. 10(a) and 10(c) show that the average control traffic produced by AODV is larger than that of AODV-CF, and that the difference increases with the network size. This result is expected because with larger networks, paths are longer, and therefore the cost of flooding the entire network is larger. On the other hand, the difference between AODV-CF and AODV is smaller for the higher mobility scenario, because in this case the probability that the alternative routes constructed by AODV-CF are still useful is smaller.

Figs. 10(b) and 10(d) show the average number of packet drops obtained in the two mobility scenarios. We note that for both protocols, the number of packet drops increase with the network size, since longer paths have a higher probability of failure. AODV-CF performs better than AODV for most network sizes, but the difference is smaller for the larger networks. For such networks, the number of alternative routes is likely to be larger. Therefore, in case of an unsuccessful route repair by AODV-CF, a larger number of alternative routes were tried, generating longer route repair times and greater number of losses. To reduce this side effect, we are investigating the limitation of the number of alternative routes stored by AODV-CF.

5. Conclusions

In this paper, we proposed the Controlled Flooding (CF) approach to provide efficient connectivity maintenance and routing discovery in ad hoc routing protocols that use network-wide broadcast for route discovery. With CF, alternative routes are built once the first shortest path between the source and the destination has been established. This mechanism does not replace the original routing protocol, since this latter must be used to determine the first route. Instead, CF is to be applied to the original protocol.

In AODV, a new route discovery is performed each time the path between the source and the destination is broken. One of the causes of route disruption is node mobility or failure. When the network population is sparse, the new route may be completely different from the original path. In this case, re-broadcasting a request message may be a good approach. Nevertheless, if the network is relatively dense, a portion of the new route may contain a subset of the links of the original path. Thus, prior to call the search procedure, the source has a clue of the current location of the destination. Applying a broadcast strategy in such a scenario is an inefficient strategy.

In the proposed AODV-CF, nodes in the main path between the source and the destination keep a pointer to the next hop in the main route and to some other nodes in its neighborhood that keep backup routes. With a simple protocol for exchanging control messages between the nodes in the main path and the nodes in the scope region, our proposal leads to performance improvements even for small ad hoc networks.

Acknowledgements

The authors would like to thank CNPq, CAPES/COFECUB, UFRJ, CNRS, and Euronetlab for the financial support of this work. The authors would also like to thank the anonymous reviewers for their helpful comments.

References

1. Bellur, B. and R. G. Ogier: 1999, 'A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks'. In: *IEEE INFOCOM'99*.
2. Broch, J.: 1998, 'A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols'. In: *ACM/IEEE Mobicom*.
3. C., K. and J. Crowcroft: 1997, 'Building shared trees using a one-to-many joining mechanism'. *ACM Computer Communication Review* **27**(1), 5–11.
4. Castaneda, R. and S. R. Das: 1999, 'Query localization techniques for on-demand routing protocols in ad hoc networks'. In: *ACM/IEEE Mobicom*.
5. Clausen, T., P. Jacquet, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot: 2002, 'Optimized Link State Routing Protocol'. Work in progress, <draft-ietf-manet-olsr-07.txt>.
6. Costa, L. H. M. K., M. D. de Amorim, and S. Fdida: 2002, 'Avoiding network-wide broadcasting with controlled flooding for on-demand ad hoc routing protocols'. In: *IFIP Med-Hoc-Net*.
7. Dalal, Y. and R. Metcalfe: 1978, 'Reverse Path Forwarding of Broadcast Packets'. *Communications of the ACM* **21**(12), 1040–1048.
8. Fall, K. and K. Varadhan: 2001, 'The ns Manual'. UC Berkeley, LBL, USC/ISI, and Xerox PARC. Available at http://www.isi.edu/nsnam/ns/ns-documentation.html.
9. Hong, X., K. Xu, and M. Gerla: 2002, 'Scalable Routing Protocols for Mobile Ad Hoc Networks'. *IEEE Network* **16**(4), 11–21.
10. Iwata, A., C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen: 1999, 'Scalable Routing Strategies for Ad Hoc Wireless Networks'. *IEEE Journal on Selected Areas in Communications* **17**(8), 1369–1378.
11. Johnson, D. B. and D. A. Maltz: 1996, *Dynamic source routing in ad hoc wireless networks*, Chapt. 5, pp. 153–181. Kluwer Publishing Company.
12. Johnson, D. B., D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva: 2002, 'The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)'. Work in progress, <draft-ietf-manet-dsr-07.txt>.
13. Ko, Y.-B. and N. H. Vaidya: 2000, 'Location-aided routing (LAR) in mobile ad hoc networks'. *ACM/Baltzer Wireless Networks Journal (WINET)* **6**(4), 307–321s.

14. Lou, W. and J. Wu: 2002, 'On Reducing Broadcast Redundancy in Ad Hoc Wireless Networks'. *IEEE Transactions on Mobile Computing* **1**(2), 111–122.
15. Magoni, D. and J.-J. Pansiot: 2002, 'Network Layer Search Service Using Oriented Multicasting'. In: *IEEE INFOCOM'02*.
16. Marina, M. K. and S. R. Das: 2001, 'On-Demand Multipath Distance Vector Routing in Ad Hoc Networks'. In: *International Conference on Network Protocols*.
17. Obraczka, K., K. Viswanath, and G. Tsudik: 2001, 'Flooding for Reliable Multicast in Multi-Hop Ad Hoc Networks'. *ACM/Baltzer Wireless Networks Journal (WINET)* **7**(6), 627–634.
18. Park, V. and M. Corson: 1997, 'A highly adaptive distributed routing algorithm for mobile wireless networks'. In: *IEEE INFOCOM'97*.
19. Perkins, C. E., E. M. Belding-Royer, and S. R. Das: 2002, 'Ad hoc On-Demand Distance Vector (AODV) Routing'. Work in progress, <draft-ietf-manet-aodv-11.txt>.
20. Perkins, C. E. and E. Royer: 1999, 'Ad-hoc On Demand Distance Vector Routing'. In: *IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*.
21. Perkins, C. E., E. M. Royer, S. R. Das, and M. K. Marina: 2001, 'Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks'. *IEEE Personal Communications* **8**(1), 16–28.
22. Royer, E. and C.-K. Toh: 1999, 'A review of current protocols for ad hoc wireless networks'. *IEEE Personal Communications* pp. 45–55.
23. Santivanez, C. A., B. McDonald, I. Stavrakakis, and R. Ramanathan: 2002, 'On the Scalability of Ad Hoc Routing Protocols'. In: *IEEE INFOCOM'02*.
24. Sucec, J. and I. Marsic: 2002, 'Clustering Overhead for hierarchical Routing in Mobile Ad hoc Networks'. In: *IEEE INFOCOM'02*.
25. Tseng, Y.-C., S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu: 2002, 'The Broadcast Storm Problem in a Mobile Ad Hoc Network'. *ACM/Baltzer Wireless Networks Journal (WINET)* **8**(2/3), 153–167.
26. Yi, Y. and M. Gerla: 2002, 'Scalable AODV with Efficient Flooding based on On-Demand (Passive) Clustering'. In: *AODV Next Generation (AODVng) Workshop*.
27. Zhu, C. and M. S. Corson: 2002, 'QoS Routing for Mobile Ad Hoc Networks'. In: *IEEE INFOCOM'02*.

Authors' Vitae

Luís Henrique M. K. Costa

Luis Henrique M. K. Costa received the Electronic Engineer degree and the M. Sc. degree in Electrical Engineering from the Federal University of Rio de Janeiro (UFRJ), Brazil, in 1997 and 1998, respectively. In december 2001, he defended his D. Sc. thesis at the University Pierre et Marie Curie (Paris 6), France. During 2002, he has been working as a post-doc researcher at LIP6 (Laboratoire d'Informatique de Paris 6), France. At the end of 2002, Luis received a research grant from the Brazilian Ministry of Education, and is currently working at COPPE/UFRJ. His major research interests are in the area of routing, especially on group communication, quality of service, multicast, and large scale routing. Luis is associate member of IEEE and ACM.

Marcelo Dias de Amorim
Marcelo Dias de Amorim received the "Cum Laude" Degree in Electronic Engineering from the Polytechnique School of the Federal University of Rio de Janeiro (UFRJ), Brazil, in 1996, and the M.Sc. Degree in Electrical Engineering from COPPE/UFRJ, Brazil, in 1998. He received his Ph.D. degree with honors in computer science from the University of Versailles, France, in 2001. He spent one year at the LRI Laboratory of the University of Paris Sud. He is currently a research scientist in the Network and Performance Group of the LIP6 Laboratory (CNRS – University of Paris 6) and in EuronetLab, the research laboratory dedicated to next-generation Internet, France. His major research interests are in QoS guarantees, self-organizing networks, mobility peer-to-peer networks, and large-scale routing.

Serge Fdida
Serge Fdida is currently a full professor at the University Pierre et Marie Curie (Paris 6). He has been an assistant professor from 1983 to 1989 and a professor with the university René Descartes (Paris) from 1989 to 1995. He also spent a sabbatical year in 1995 with IBM RTP (Raleigh, USA). Serge Fdida is heading the Network and Performance group of the Laboratoire dInformatique de Paris 6 (LIP6), a research laboratory associated with CNRS (National Scientific Research Center). He is on the editorial boards of Computer Communication and Computer Networks Journals. He served as the program chair and program committee member of numerous international events. He has extensively published in the field of performance evaluation and networking and led several research grants. He was for 8 years, the chair of the french national research group on high-speed networking (RHDM) and is currently the chairman of the European COST264 Action "Enabling Multimedia Group Communication", and the director of Euronetlab, a joint academic-industrial laboratory. Serge Fdida is a senior member of IEEE, member of ACM and IFIP TC6 (WG6.3 and WG6.4). Since early 2000, Serge Fdida is working part-time with CNRS-STIC (French National Scientific Research Center).