NARD: Neighbor-Assisted Route Discovery in Wireless Ad Hoc Networks

J. Gomez, J. M. Cervantes, V. Rangel and R. Atahualpa
Department of Electrical Engineering
National Autonomous University of Mexico
Mexico City, Mexico
{javierg, juan, victor, rodrigo}@fi-b.unam.mx

Miguel Lopez-Guerrero
Department of Electrical Engineering
Metropolitan Autonomous University
Mexico City, Mexico
milo@xanum.uam.mx

Abstract

Routing protocols for mobile ad hoc networks usually discover routes by flooding the entire network with control packets; this technique is known as blind flooding. This paper presents ***NARD****, a Neighbor-Assisted Route Discovery protocol for wireless ad hoc networks. In NARD, a source node floods a limited portion of the network looking not only for the destination node, but also for routing information of other nodes (called neighbors) that were known to be near the destination node recently. Neighbor nodes can be used as anchor points where a second limited flooding takes place in search for the destination node. Because only a limited portion of the network is flooded by control packets near the source and destination nodes, NARD can significantly reduce the signaling overhead of route discovery compared with blind flooding techniques. Simulations with NS2 were undertaken to verify the validity of our approach.*

1. Introduction

An ad hoc network is a collection of nodes equipped with a wireless interface located in an area where there is not a significant fixed infrastructure in place. Because nodes have a limited range, it is usually necessary that some nodes participate in the routing process relaying packets among source-destination pairs. This type of routing is also known as multi-hop routing. Finding routes among nodes has been the main research challenge since the origins of ad hoc networks in the 70's. Even today, there is a significant amount of research going on in this area [4] [11].

Routing protocols for ad hoc networks are typically divided as either proactive or reactive. Proactive protocols discover routes from any node to all the other nodes in the network in advance, and these routes are periodically updated as route changes occur. Proactive protocols have the main advantage that whenever a node needs to send a packet to other node, there is already a route available, however more signaling overhead is needed. Reactive protocols, on the other hand, discover routes on demand only when they are needed. This operation adds a delay while a route is found, however less signaling overhead is needed in general compared with proactive protocols.

Many routing protocols recommended by the Mobile Ad hoc Network Working Group of the IETF (MANET) are reactive. Routing protocols in this category discover routes (the route-discovery phase) only after there is a need for it. Once a route is found, data packets can then be forwarded from source to destination.

In this paper we focus on the route-discovery phase of reactive routing protocols for ad hoc networks. Although there are several improved proposals for route-discovery (proposals that are discussed later in this paper), the blind-flooding technique remains as the most widely used protocol in practice. Blind flooding operates as follows. A source node transmits a broadcast control packet to announce its intention to communicate with a certain destination node. Nodes overhearing this request retransmit this control packet, thus increasing the coverage area of the search. This simple mechanism has the effect of flooding the network with control packets from the source node to the entire network. Blind flooding is adequate for small, low-density and slow-mobility networks, otherwise blind flooding could generate a prohibiting number of control packets.

This paper presents Neighbor-Assisted Route Discovery (NARD), an efficient route-discovery protocol for wireless ad-hoc networks. NARD is intended for medium to large ad hoc networks where traditional flooding is not a practical solution. In NARD, a source node floods a limited portion of the network looking not only for the destination node, but also for information related to other nodes (called neighbors) that were known to be near the destination node recently. Neighbor nodes can be used as anchor points where a second limited flooding takes place in search for the destination node. Because only two limited portions of the network near the source and destination nodes are

1-4244-1455-5/07/$25.00 ©2007 IEEE

flooded by control packets, NARD can significantly reduce the signaling overhead of route discovery compared with blind flooding techniques.

The structure of this paper is as follows: Section 2 discusses previous work in this area. In Section 3, we present the motivation and a detailed description of NARD while Section 4 provides some guidelines to select the appropriate scope of the first and second flooding that guarantee with a high probability that the destination will be found while at the same time will minimize the signaling overhead. Section 5 presents a performance evaluation of NARD in a network simulator and a performance comparison with blind flooding and FRESH [4]. Finally, in Section 6, we present some final remarks.

2. Related Work

The work developed in the MANET group of the IETF in the past years represents the baseline work in this area of research. As we mentioned before, most MANET routing protocols rely on blind-flooding to discover routes. There are however, variations of this technique that are worth mentioning here. In DSR [1], for example, there is a provision for reducing the scope of the route discovery to N hops only. The rationale here is that if a node manages to find the destination node within this limited search, then a big reduction in signaling traffic can be achieved. However, in case the destination node is not found within this limited search, a full search becomes necessary thus generating even more signaling, plus the increased delay involved in finding the destination node with two searches.

FRESH [4] is an algorithm for efficient route discovery in mobile ad hoc networks. In FRESH, nodes keep a record of their most recent encounter times with all nodes. Instead of searching for the destination node directly, the source node searches in a limited portion of the network for any node that has encountered the destination node more recently than the source node did itself. This procedure continues until the destination is finally reached. Because FRESH represents the state of the art and possibly the closest competitor we compare its performance with NARD in Section 5.1.

Fireworks [11] is a protocol for managing multicast groups in mobile ad hoc networks. In [11], the authors assume that members of a multicast group are grouped locally in an ad hoc network, and therefore, it is more efficient (i.e., it generates less signaling) to simply broadcast multicast packets to all members rather than sending independent unicast packets to each member. As we will show later, Fireworks is similar to the last operational phase of NARD. In both protocols a unicast packet is broadcasted in a limited area, however, while in Fireworks this process is related to multicast group communications, in NARD, this process is related to node encounter times and node mobility.

A different way of discovering routes in ad hoc networks is by taking advantage of location information or Cartesian Routing [5], which assigns each node with a unique identifier and geographic location to send the packets through the closest neighbor to the destination node. A similar approach to Cartesian Routing is GLS [2], which gets location information by means of a service such as GPS in terms of latitude and longitude. GPSR [3] is a geographic routing system that uses a planar sub-graph of the wireless network graph to route around holes. Both, GLS and GPSR are designed for large metropolitan area networks, but they need a high node density and expensive location devices that do not always work well within urban areas.

3. NARD

Opposite to blind-flooding where the entire network is flooded with control packets, NARD floods only two arbitrarily small regions in the network. One region is centered at the source node while the other region is located in the vicinity of the destination node. In this way NARD is capable of reducing significantly the number of control packets used for route-discovery, thus freeing precious bandwidth for data packets.

The main innovation of our approach is that the source node performs a limited search looking not only for the destination node, but also for routing information about nodes that were neighbors of the destination node recently (see Figure 1). The source node sends unicast packets to those neighbors which then perform an additional limited search looking for the destination node. NARD is composed of three phases called *neighbor discovery*, *neighbor search* and *target search* which are explained below.

$Neighbor\ discovery$: In NARD, all nodes collect neighbor information by means of overhearing packets from other nodes. A node can overhear packets by putting its transceiver into "promiscuous mode". These overheard packets can be either data or control packets. When a node has not transmitted any data packets for a while, it may transmit an explicit control packet called "Hello" so its current neighbors can learn about its presence. Hello packets are quite small (IP headers basically) and are not retransmitted by other nodes in order to limit the signaling overhead. Similarly, Hello packets may not be always necessary. Anytime a node transmits its own data packets or forwards packets from other nodes, its one-hop neighbors can learn about its presence. Similarly, the rate of Hello packets transmitted by each node is constant and independent of size. A node can also control the rate of Hello packets according to its own needs (e.g., power reserves, network congestion etc.). A node can even stop transmitting Hello packets if

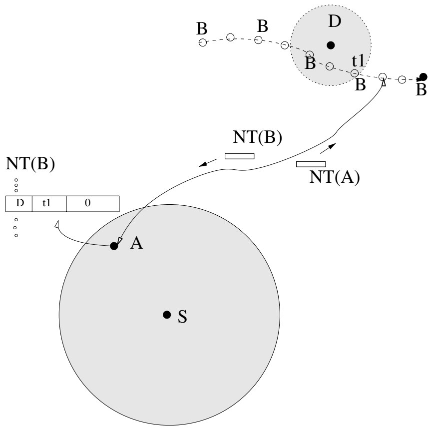


Figure 1. NARD concept: Nodes A and B are end points of a connection and, as a result, node A has a copy of the list of nodes that node B has recently overheard, NT(B). When node S looks for routing information about node D within a limited portion of the network (shaded area), node A knows that node B was in contact with node D at time t1 and node A knows a route towards node B. Node S can therefore send a request to node B to look for node D on its behalf.

it is not expecting any connection with other node in the near future. For these reasons we won't present quantitative overhead measurements of Hello packets in the evaluation section.

Nodes collect and store neighbor information in a *neighbor table* (NT) and every time a connection is established, source and destination nodes exchange their corresponding neighbor tables (see Figure 2a). All nodes keep in memory their own neighbor tables, plus other neighbor tables acquired while communicating with other nodes that function as either source or destination.

The structure of the neighbor table includes the fields [IP address, time stamp, EP] where the first field is the IP address of the overheard node, time stamp is the time when the entry was created/updated, and EP is a flag that indicates whether the overheard packet was transmitted by an end point of a connection (either source or destination). A node learns that an overhead packet comes from an end point by relating the IP and MAC addresses of the packet. The EP flag will be used later in Section 4 in order to compute the initial scope of the first flooding (setting the n parameter).

$Neighbor\ search$: Whenever a node intents to transmit a packet to other node, it checks whether it has a valid route to it, in case a route is available the packet is relayed to the next hop immediately. When no route is available, the source node creates and transmits a Route Request packet (RREQ) including source and destination IP addresses in it. This initial search is limited to only n hops away from the source node (see Figure 2b). In this RREQ the source node looks not only for the destination node, but also for routing information about recent neighbors of the destination node. Upon receiving a RREQ, a node queries its routing table first, in case there is no route to the destination node, it queries its neighbor table and other neighbor tables from other nodes obtained during neighbor discovery. In case a node has a route to either the destination node directly or to past neighbors of the destination node, it answers via a RREP unicast packet back to the source node.

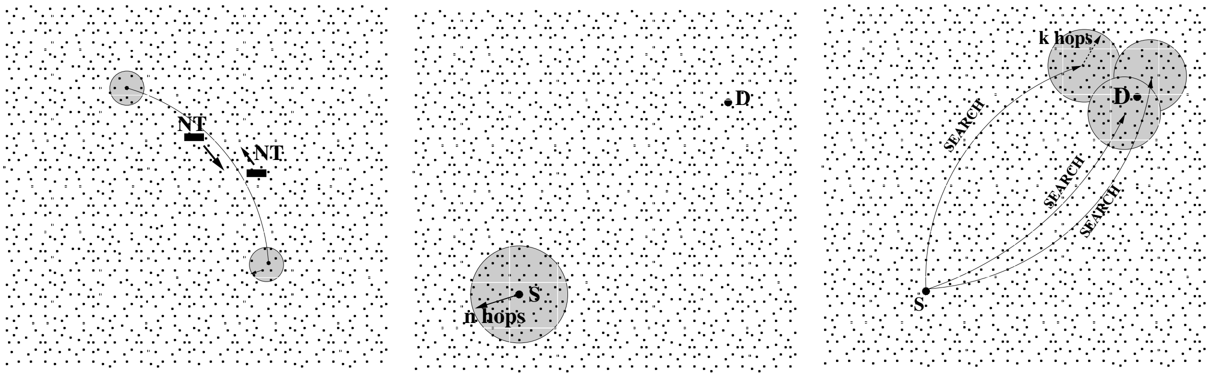
In case no routing information is collected in this phase (to either the destination node or to neighbors), a larger area search becomes necessary. A second search, however, may generate even more signaling overhead and longer delays compared with blind-flooding techniques. Therefore, it is important to choose the right value of n in order to find routing information related to the destination node in the first attempt and with the minimum amount of signaling overhead. In Section 4 we give some guidelines about choosing the initial value of n.

$Target\ search$: Upon collecting routing information about past neighbors of the destination node, the source node sends few unicast packets to each neighbor found, we call these control packets "Search" (see Figure 2c). We recommend sending 2-3 Search packets to each neighbor found because Search packets may not always reach the target neighbor because heavy traffic congestion or broken routes may be present.

Upon receiving a Search packet, a neighbor node checks whether it has routing information about the destination node, otherwise it constructs a new RREQ packet with the IP address of the destination node as the searched node. This new RREQ is flooded to an area limited by k hops. In case the destination node is found in this search, it answers with a RREP to the neighbor that made the request which then forwards the RREP back to the source node so communication can be started.

The scope of the second flooding (controlled by the k parameter) can be determined according to the last encounter time with the destination node as well as node mobility. An old recorded encounter with the destination node or a high mobility scenario makes the routing information to become obsolete faster as time passes, making it necessary to search in a larger area. A discussion about how to choose the parameter k is presented in Section 4. Under this scheme, only a small region of the network near the destination node will be flooded by control packets as it is illustrated in Figure 2c.

In NARD, the final route between source and destination nodes is the concatenation of two routes, a first route from the source node to a neighbor node, and a second route from that neighbor to the destination node. This final



(a) Neighbor Discovery (b) Neighbor search (c) Target search

Figure 2. NARD Operation: (a) Neighbor discovery phase. Nodes transmit data or Hello packets and keep track of overheard nodes in a neighbor table (NT). End points of a connection exchange their corresponding neighbor tables. (b) Neighbor-search phase: A source node floods an area limited to n hops and receives routing information related to either the destination node or about past neighbors of the destination node. (c) Target search phase: A source node has recovered routes to past neighbors of the destination node and sends Search packets (unicast) to each neighbor found. These neighbors then perform a new search limited to k hops looking for the destination node.

route, however, may have more hops than necessary compared with a route obtained by a traditional MANET protocol using blind-flooding. In such cases, there are already route shortening algorithms [1] available that can be used to remove some unnecessary links. Even without considering route shortening mechanisms, the great reduction of control packets achieved by NARD justifies the occasional increased hop count.

It is important to note that although the n and k parameters appear to have similar roles (i.e., they both limit the scope of the flooding) finding a good value for them obeys totally different basis. The parameter n on one hand, is related to how much traffic is present in the network while the k parameter, on the other hand, is related to the dispersion of nodes as time passes. A detailed discussion of choosing the values of n and k is presented below.

4. Selecting the scope of the neighbor and target searches

The performance of NARD, as many other routing protocols used in ad hoc networks, is influenced by node and network dynamics [6] [7]. In NARD, in particular, choosing the scope of the first and second searches (the n and k parameters) plays a key role in the performance and accuracy of the route-discovery phase using NARD.

4.1. The scope of the neighbor search

The scope of the neighbor-search in which a source node searches for routing information related to either the destination node or about recent neighbors of the destination node is controlled by the n parameter. Choosing n large increases the area of the search and thus the probability of finding more routing information that may lead toward finding the destination node, however more signaling overhead is generated. Clearly an opposite tradeoff applies when n is small. However, intuition suggests that node density and traffic conditions impact the selection of n. With a high node density, for example, more nodes overhear each transmission on the average, however, neighbor information is useless unless there is data traffic that propagates this local information to other parts of the network (remember that source and destination nodes exchange their neighbor tables at the beginning of a connection). In practice, with no global view of density and traffic conditions in the network, a node needs to guess the value of n based on its local view of the network only. For example, a node can approximate the total number of connections in the network by extrapolating the number of connections (either source or destination) it overhears within its range. Now we present a simple method that can be used to choose the initial value of n.

Figure 3 illustrates a network where node S is about to send a RREQ limited to n hops looking for routing information related to node D. Following the notation in Figure 3, let EP_{A_I} be the number of End Points of connections (either source or destination) that are located within an area

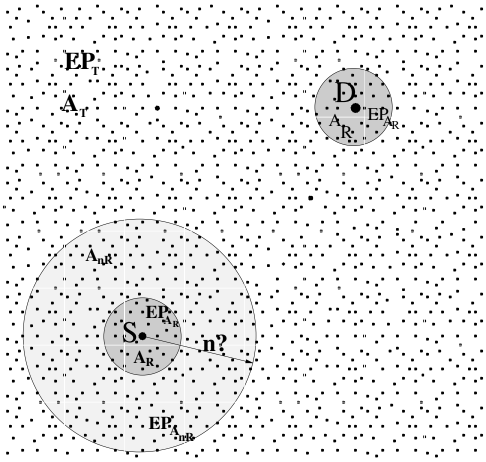


Figure 3. Parameters involved in the selection of n

of A_I square meters. As we just mentioned, the source node can determine EP_{A_R} (the number of end-points located within a circular area of radius R) by overhearing traffic from its neighbors. Assuming end points in the network are homogeneously distributed [1], the source node can approximate the number of end-points located within A_{nR} (the equivalent area of a limited flooding to n hops as

$$EP_{A_{nR}} \approx \frac{A_{nR}}{A_R} EP_{A_R} = n^2 EP_{A_R} \qquad (1)$$

thus a linear increase in the value of n increases the area of search geometrically. Finally, lets define the parameter β as the number of the end points located within $EP_{A_{nR}}$ that have the other end of the connection in a node located within the coverage area of the destination node. The value of β is quite important for NARD because it represents how many nodes within A_{nR} know about the location of recent neighbors of the destination node. We can then approximate β as

$$\beta \approx EP_{A_{nR}} \frac{A_R}{A_T - A_{nR}} \qquad (2)$$

where A_T is the total area of the network. The value of A_T is unknown to the source node, however, we believe its value can be estimated indirectly. One way to do this, for example, is to monitor the length of the longest detected route in the network, and then assume that this length is the diameter of the network. In DSR, for example, route length can be estimated by counting the number of intermediate hops of the routing field located in the header of each data packet.

[1] Various mobility models have this property however the random way point model, for example, creates higher density of nodes in the center of the network and this type of behavior needs further study.

Finally the value of n is found from (2) as

$$n \approx \frac{\sqrt{\frac{\beta A_T}{\pi(EP_{A_R}+\beta)}}}{R} \qquad (3)$$

It is possible that this initial choice for n may result in retrieving no routing information or retrieving too much of it. In both cases it is necessary to use an adaptive algorithm to adjust its value in further searches according to how successful the initial value of n was (using Equation 3). A simple algorithm that can perform this task is the following:

i) compute n using Equation 3 and launch the neighbor search flooding

ii) increase the value of n by one if no routing information was retrieved in the previous attempt

iii) repeat (ii) until routing information is retrieved

iv) decrease the value of n in further searches if too many nodes replied back to the source node (i.e., $\beta >> 1$). Here we recommend β to be set to two or three because Search packets may either get lost before reaching its intended neighbor, or because neighbors may not find the destination node during the second search.

4.2. The scope of the target search

Let us consider the problem of determining the scope of the second search so that nodes that were originally neighbors can still be reachable within k hops after some time. How fast nodes move around and how old was the last encounter time with the destination node determine the optimum scope of the second search. An old entry in a neighbor table or high mobility conditions (i.e., fast spreading of the nodes) makes necessary to find the destination node in a larger area (higher overhead).

Let us consider an ad-hoc network and let us place the origin of the coordinate system at the position occupied by the destination node at time zero. Let us assume that all nodes in the network move according to a two-dimensional random walk mobility model as follows. Every T seconds each node randomly chooses whether to move left or right and independently whether to move north or south. Once the moving direction has been individually defined, they move s meters in that direction during T seconds. It can be easily shown that, after t seconds, the distance r measured from the origin to the position of the destination node follows a Rayleigh distribution [12]. Therefore, the probability that the destination node remains within a circle of radius C, after t seconds, is given by the Rayleigh CDF as follows,

$$P\{r < C\} = 1 - e^{\frac{-C^2}{2\alpha t}} \qquad 0 < r < \infty, t > 0 \quad (4)$$

where $\alpha = s^2/T$. Equation 4 can be used in order to determine the required search area that would allow us to find the destination node at time t with an arbitrarily high probability ϵ. Thus, the target search should expand over a circular area of radius C given by

$$C = \sqrt{-2\alpha t \;\; ln(1-\epsilon)} \tag{5}$$

Now let us consider the set of one-hop neighbors of the destination node at time zero (i.e., nodes that were originally located within R meters from the destination node). We are interested in locating this set of nodes at time t with probability ϵ. Since the diffusion process affects all nodes in the same manner, we can locate the destination node and its former neighbors in a circular area of radius $2C + R$. Thus we can approximate k as

$$k \approx \frac{2C}{R} + 1 \tag{6}$$

5. Performance Evaluation

NARD was implemented using the NS2.28 network simulator. In this implementation of NARD we used the DSR routing protocol as a starting point. We replaced the route-discovery mechanism of DSR with NARD instead. As we mentioned before, NARD is not tied to any routing protocol in particular, and it can be used with any reactive protocol replacing its route-discovery mechanism. In order to control the number of hops that a route request (RREQ) is propagated in the network, we modified the time-to-live value of the RREQ (16 hops is the default in DSR), so that we can control its value to a fixed small number.

In what follows, we present an experimental performance evaluation of NARD under different network conditions to stress its advantages and disadvantages. In all cases, mobile nodes use the standard IEEE 802.11 MAC protocol running at 2 Mbps and nodes move according to the random way-point mobility model [10]. Table 1 shows the parameters used in the experiments.

Parameter	Value
Data rate	1pkt/s
Packet size	100 bytes
Node speed	0, 2 and 10 m/s
Scenario size	2200x1200 (m^2)
number of connections	0, 10, 30, 60, 80, 100
Transmission range	250 m
n	1
k	3
prop. model	two-ray ground
scenario generator	setdest
pause time	0 seconds

Table 1. Simulation Parameters

On each simulation, a number of Z random connections are created before a route-discovery search is performed with NARD for the $Z+1$ connection (scenarios for Z equal to 0, 10, 30, 60, 80 and 100 previous connections were considered). The results shown in Figure 4 and Figure 5 are related to the $Z + 1$ connection only, and are compared to those obtained by standard DSR (blind-flooding) under similar conditions. All connections are of type UDP/CBR transmitting a 100-byte long packet every 1 second. We fixed $n = 1$ and $k = 3$ in order to compare the performance of the protocol with respect to the number of connections only. In section 5.1 we compare NARD with FRESH and we will use the optimum values of n and k derived previously. The points shown in Figures 4 and 5 are the result of averaging over 10 different scenarios with different seed numbers which modified the location and mobility patterns of each node in the simulation.

Figures 4a, 4b and 4c show the performance of NARD for the $Z + 1$ connection for 0, 2 and 10 m/s, respectively. This includes

i) *Neighbors found*: the number of found routes to different neighbors of the destination node. This information is collected by the source node during the neighbor-search phase.

ii) *Neighbors reached*: the number of neighbors reached by the source node with unicast Search packets, and

iii) *Destination reached*: the number of neighbors reached by Search packets that could find the destination node during the second search.

Figure 4a shows the performance of NARD with no mobility. As expected, we observe that the number of neighbors found in the neighbor-search phase increases as the number of connections in the network rises. This is a direct result of the neighbor discovery phase where source-destination pairs exchange their corresponding neighbor tables. Probably the main drawback of NARD is that it requires some background traffic to work properly. The good news is that even for few connections (e.g., 10 connections in Figure 4a) some information about neighbors of the destination node was retrieved during the neighbor-search phase. In this figure we also observe that most Search packets sent by the source node reached their intended neighbor except for high traffic conditions. Packets that did not reach the intended neighbor were dropped by the network because of congestion. Once a neighbor node received the Search packet, it always found the destination node during the second search except again when heavy traffic congestion was present.

Figures 4b and 4c show the performance of NARD when nodes move at 2 m/s and 10 m/s, respectively. Mobility brings both benefits and disadvantages to the performance of NARD. When nodes move, they move around with their corresponding knowledge about routing informa-

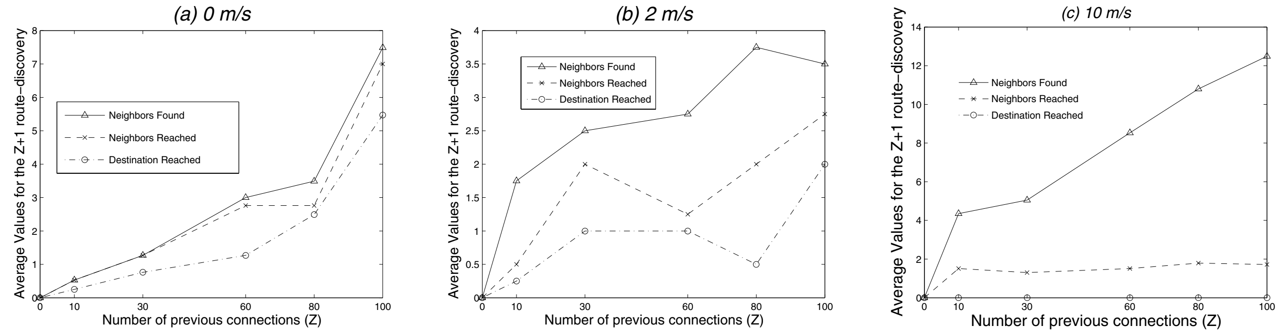


Figure 4. NARD Performance for the $Z + 1$ route-discovery at 0 m/s (a), 2 m/s (b) and 10m/s (c)

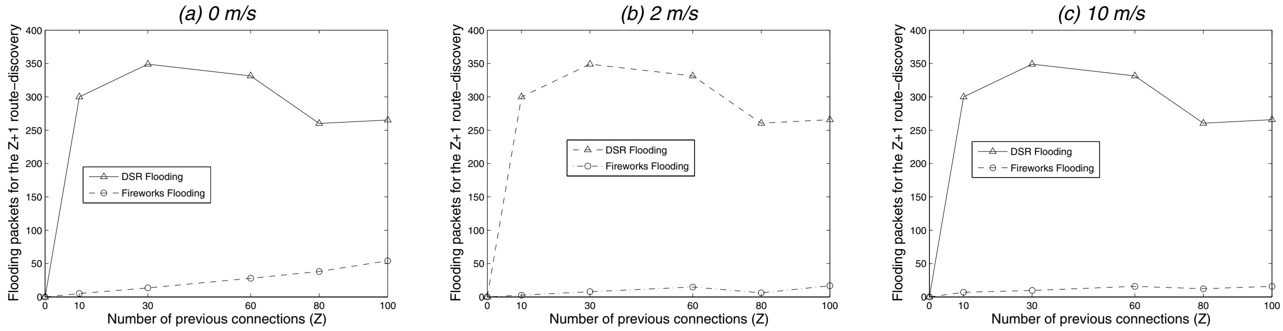


Figure 5. Flooding Performance (a) 0 m/s, (b) 2 m/s and (c) 10 m/s

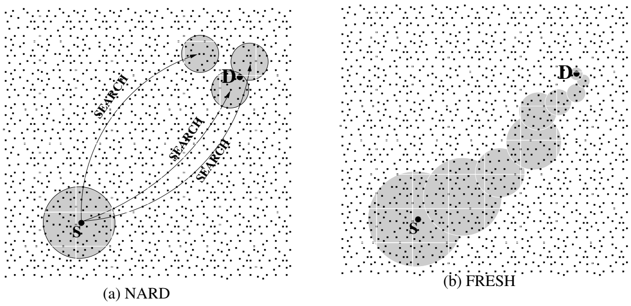
tion to other nodes in the network. Since mobility makes easier for a node to meet with other nodes, a node increases its overall knowledge of routing information about other nodes as time passes. This is in contrast with the static case where a source node can not retrieve routing information from nodes outside the area covered by the first search. This phenomenon can be observed in Figure 4b where the number of neighbors found in the first search rises faster with respect to the number of connections compared with Figure 4a. On the other hand, mobility makes routing information less reliable because nodes move around as times passes. This can be observed in Figure 4b where some Search packets did not reach their intended neighbor due to broken routes. An extreme example of the impact of mobility can be seen in Figure 4c where only few Search packets reached their intended neighbors, and even if they did, no neighbor found the destination node during the second search. This poor performance can be corrected once the optimum value of k is chosen according to Equation 6. Remember that we use a fixed value of k ($k = 3$) in Figure 4c to focus on the impact of background traffic on NARD's performance only. In this figure it is likely that the destination node was already located outside the scope of a flooding limited to 3-hops from the neighbor node sending the RREQ.

Figures 5a, 5b and 5c show the number of signaling packets generated by NARD for the $Z + 1$ route-discovery only (this includes the sum of the signaling packets generated during neighbor search and target search phases). For comparison purposes we also plot the signaling overhead of DSR (blind flooding) in the same experiment. As we can observe in these figures, NARD outperforms DSR by a great factor. This again is due to the fact that NARD floods only two small regions of the network as opposed to DSR where the entire network is flooded with control packets.

5.1. NARD versus FRESH

Because FRESH represents the state of the art in efficient route-discovery and possibly the closest competitor we compare its performance with NARD. In FRESH, nodes keep a record of their most recent encounter times with all nodes. Instead of searching for the destination node directly, the source node searches in a limited portion of the network for any node that has encountered the destination node more recently than the source node did itself. This intermediate node (i.e., anchor) then performs a second limited-area search for another node that have encountered the destination even more recently and so on. This procedure continues until the destination is finally reached (see Figure 6b).

In FRESH, the search cost of a single search (i.e., the



(a) NARD

(b) FRESH

Figure 6. Flooding footprint of NARD (a) and FRESH (b)

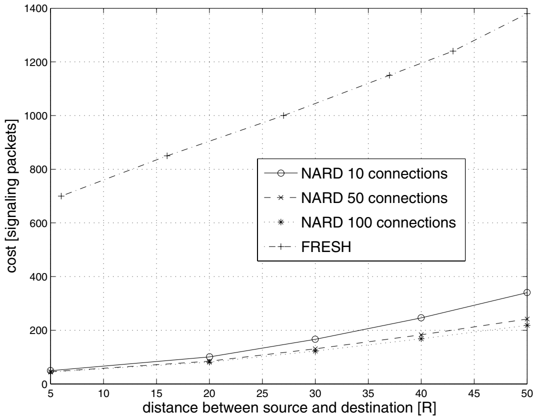


Figure 7. FRESH versus NARD (k constant)

area of the network covered during the search) which originates at node S and terminates at node D is found as $C_{FRESH}(S,D) = (\alpha(|X_S - X_D|))^2$ for some $1 < \alpha < 2$. The cost is quadratic with the distance because the number of packet transmissions generated by the search is proportional to the number of nodes located in a circular area of radius $|X_s - X_f|$ where the flooding takes place. Here α models the fact that the radius of the search will on average be larger than the distance between the two nodes (in Figures 7 and 8 the authors used α = 1.3). In FRESH a route-discovery involves N consecutive searches as depicted in Figure 6b having the following total search cost in terms of packet transmissions [4]:

$$C_{FRESH} = \rho \sum_{i=1}^{N} (\alpha(|X_i - X_{i+1}|))^2 \quad (7)$$

where ρ is the node density and X_i is the position of the ith anchor. We can find an equivalent cost function for NARD as:

$$C_{NARD} = \rho(nR)^2 + L\beta\frac{|X_S - X_D|}{R} + \rho\beta(kR)^2 \quad (8)$$

The terms in Equation 8 account for the number of signaling packets generated during neighbor-search (first term), transmission of L Search unicast packets to each of the β neighbors (second term), and packets generated during target search (third term), respectively. Here we used $\beta = 3$ and $\epsilon = 99\%$.

Figure 7 and Figure 8 compare the signaling overhead generated by FRESH and NARD during a route-discovery under similar network settings. We took the values of FRESH shown in both figures directly from [4] and no proactive signaling overhead from Hello packets was included for either protocol. In both figures we use $D = 1$ (unit density) and $R = 1$ (unit radius) as in [4]. In Figure 7 we kept constant the scope of the second flooding (k=3) and focused on the behavior of the first flooding with respect to the background traffic only. As we can see in this figure the amount of background traffic has only a limited impact on search cost for NARD. Opposite to Figure 7, in Figure 8 we kept constant the scope of the first flooding (n=3) and focused on the behavior of the second flooding with respect to the age of encounter times and the speed of movement. It can be observed in this figure that only when encounter times are old in high mobility conditions NARD incurs in high costs.

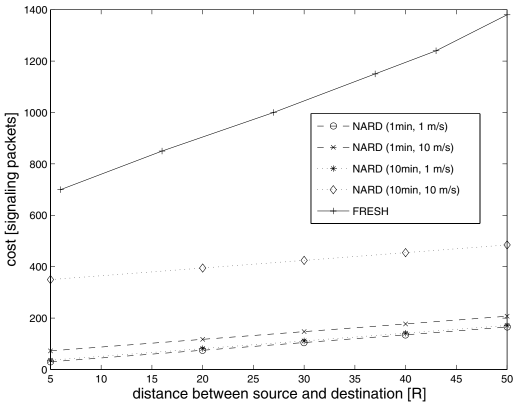


Figure 8. FRESH versus NARD (n constant)

As we can see in Figure 7 and Figure 8, NARD generates less signaling packets than FRESH and this advantage increases as the source-destination distance increases. We believe that the fact that NARD uses unicast packets to cover a significant portion of the search (see Figure 6a) is the key reason of its improved performance. FRESH uses node mobility to actually *move* location information across the network, NARD on the other hand, uses background traffic to perform a similar task. Since usually traffic forwarding moves information faster than moving nodes, NARD will have fresher information about the location of the destination node, which translates into less flooding areas and less signaling overhead.

We believe NARD and FRESH do not need to compete but in fact they can complement each other. Because NARD requires some background traffic to work properly, we imagine a framework where a routing protocol can use FRESH in cases where no background traffic is present, and then switch to NARD once some degree of background traffic is detected.

5.2. Future work

There are issues that remain open about NARD and we plan to study in the future. One of them is to study its performance under other mobility models. In NARD we assumed a homogeneous density of nodes in the network, however, a well-known feature of RWP is that it creates a greater density of nodes in the center of the network and this issue needs further study.

6. Conclusions

In this work a novel route-discovery protocol for ad hoc networks called NARD is presented. In NARD, a source node performs a limited-area search looking not only for a destination node, but also for past neighbors of the destination node. Because NARD floods only two small regions of the network, one around the source and another in the vicinity of the destination, it achieves a lower overhead compared with blind-flooding. We implemented NARD in NS2-28 where several scenarios were analyzed varying values such as background traffic and node mobility. From the results we observed that NARD generates less signaling overhead compared with blind-flooding in most scenarios. A downside feature of NARD is that it requires some background traffic to work properly. Fortunately, the simulation results show that even with little traffic NARD shows good performance. A comparison with FRESH showed NARD generates less signaling packets and this advantage increases as the source-destination distance increases. We believe NARD and FRESH can complement each other depending on the amount of background traffic in the network.

Acknowledgment: This work is supported in part by research funds from CONACyT grant 47197-Y and PAPIIT grant IN105307.

References

[1] David B. Jhonson, David A. Maltz, *The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks*, IETF MANET Working Group, November 2000.

[2] Jinyang Li, John Jannotti, *A Scalable Location Service for Geographic Ad Hoc Routing*, M.I.T. Laboratory for Computer Science, MOBICOM 2000, Boston MA, USA.

[3] Brad Karp, H.T. Kung, *GPSR: Greedy perimeter stateless routing for wireless networks*, ACM/IEEE MobiComm, August 2000.

[4] Henri Dubois_Ferrie, Matthias Grossglauser, Martin Vetterli, *Efficient Route Discovery in Mobile Ad Hoc Networks Using Encounter Ages*, School of Computer and Communications Sciences, EPFL, Laussane Switzerland.

[5] Gregory G. Finn, *Routing and Addressing problems in large metropolitan-scale internetworks*, ISI/RR-87, ISI, March 1987.

[6] Xiaolong Li, Dharma P. Agrawal, Qing-An Zeng, *Impact of Mobility on the Performance of Mobile Ad Hoc Networks*, Department of ECECS, University of Cincinnati, 2004.

[7] Guolong Lin, Guevara Noubir, Rajmohan Rajaraman, *Mobility Models for Ad hoc Network Simulation*, College of Computer & Information Science, Northeastern University, Boston, MA, InfoCom IEEE 2004. Published by Addison Wesley Professional, Series: Addison-Wesley Professional Computing Series, 1997

[8] Orlik Philip, Rappaport Stephen S., *Traffic performance and mobility modeling of cellular communications with mixed platforms and highly variable mobilities*, State University of New Cork, Stony Brook, New York, IEEE 1997.

[9] Jabbari Bijan, Zhou Yong, Hillier Frederick, *Random Walk Modeling of Mobility in Wireless Networks*, George Mason University, Fairfax, VA, USA, IEEE 1998.

[10] William Navidi and Tracy Camp, *Stationary Distributions for the Random Waypoint Mobility Model*, IEEE Transactions on Mobile Computing, Vol. 3. No. 1, March 2004.

[11] Srinivasan Keshav, *Fireworks: An Adaptive Group Communications Protocol for Mobile Ad Hoc Networks*, Lap Kong Law, Srikanth V. Krishnamurthy, and Michalis Falautsos, Networking 2005.

[12] Doganata Y. N. and T. X. Brown and T.X Posner, *Call Setup Strategy Tradeoffs for Universal Digital Portable Communications*, Computer Networks and ISDN Systems 20, North-Holland, 1990, pp. 455-464.