Binary Waypoint Geographical Routing in Wireless Mesh Networks

Eryk Schiller, Paul Starzetz, Franck Rousseau, Andrzej Duda
Grenoble Informatics Laboratory*
Grenoble, France
{schiller, starzetz, rousseau, duda}@imag.fr

ABSTRACT

We propose *Binary Waypoint Routing*, a novel geographical routing protocol for wireless mesh networks. Its idea is to learn and maintain source routes to a small number of nodes called *binary waypoints* that are placed in subspaces constructed as a result of binary space partitioning. A source node sends a packet to a waypoint for a given destination and intermediate nodes try to adapt the packet route by aiming at waypoints that are closer to the destination. Our simulation results show that the proposed scheme achieves high packet delivery rate with a traffic pattern similar to the Optimal Shortest Path Routing.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*

General Terms

Performance

Keywords

Wireless Mesh Networks, Geographical Routing

1. INTRODUCTION

We analyze *spontaneous wireless mesh networks* composed of a large number of wireless mesh routers that provide multi-hop connectivity to client stations. Such mesh networks begin to appear in highly populated areas and provide cheap network connectivity to a community of end users. They evolve in a spontaneous way—users or network operators place additional routers to increase coverage, density, and network capacity.

Geographical routing is particularly interesting for spontaneous wireless mesh networks: it does not require any information on the global topology, but rather it uses the position of the destination to forward packets. The most familiar scheme of geographical routing is *greedy forwarding* in which a node forwards a packet to the neighbor that has the least distance to the destination [1]. It relies on the information about the physical location of nodes, which can be provided by means of a global positioning techniques such as GPS. In the absence of GPS, for instance indoors, the location information can be obtained from positioning based on estimation of the signal strength [2] or through relative positioning of nodes with respect to few nodes with the exact geographical position.

Greedy forwarding guarantees loop-free operation, but packets may be dropped at *concave nodes* that have no further neighbors closer to the destination. Concave nodes usually appear at some places close to *voids*—uncovered areas or obstacles to radio waves in a given direction. A large amount of work considered methods for coping with voids, *face routing* being one of the first solutions [1, 9]. However, face routing requires the construction of a planar graph, i.e. a graph with no crossing edges. For a *Unit Disk Graph* (UDG), which is a common model for representing multi-hop wireless networks, the construction of a *Gabriel graph* [7] or a *Relative Neighborhood graph* [13] leads to a planar graph without any connectivity loss, but in real wireless environments, the conditions for obtaining planar graphs are not satisfied due to asymmetric links and not circular radio coverage, so that routing protocols based on face routing fail to provide sufficient packet delivery rate [10].

In this paper, we present a novel geographical routing protocol that offers high packet delivery rates in wireless mesh networks with voids. The idea of *Binary Waypoint Routing* is to learn and maintain source routes to a small number of nodes called *binary waypoints* that are placed in subspaces constructed as a result of binary space partitioning. A set of subspaces whose size decreases exponentially, covers the whole addressing space. To forward a packet, a node sends it to the waypoint corresponding to the subspace of the destination. Each intermediate node tries to adapt the route by aiming at its waypoint that may be closer to the destination.

The source routes to waypoints provide a means for reaching any destination in the addressing space. As the number of subspaces grows in a logarithmic way with the space size, the size of routing tables in any node remains very small compared to the size of the mesh network. The proposed scheme does not require explicit construction of routing ta-

*LIG is a joint research laboratory of CNRS (*Centre National de la Recherche Scientifique*), INRIA (*Institut National de Recherche en Informatique et Automatique*), Grenoble INP (*Institut Polytechnique de Grenoble*), UJF (*Université Joseph Fourier*), and UPMF (*Université Pierre-Mendès-France*).

MSWiM'08, October 27–31, 2008, Vancouver, BC, Canada.

bles and its operation does not depend on any graph construction algorithm such as UDG, so the protocol works for any type of wireless connectivity between nodes.

In this paper, we explain the principles of the protocol and show how it guarantees packet delivery. We also evaluate its performance through simulations and compare it with other approaches (Greedy Routing and Optimal Shortest Path). In our simulations, we randomly place wireless nodes in a circular arena and create connectivity based on the UDG model. Even if UDG is only a first-order approximation of real wireless connectivity, such a model enables performance comparisons between the routing protocols and results in an initial insight into main performance trends. We observe forwarding performance of different routing protocols for randomly chosen pairs of a source and a destination. Our comparisons show that our scheme achieves high packet delivery rate without the need for learning the global topology. It results in a traffic pattern similar to the Optimal Shortest Path Routing.

The rest of the paper is organized as follows. We start with a short review of the related work (Section 2). Then, we explain the principles of the proposed routing scheme (Section 3) and analyze its performance (Section 4). Finally, we present some conclusions (Section 5).

2. RELATED WORK

Several authors have addressed the problem of greedy geographical routing in wireless mesh or ad hoc networks [3,4,8]. Bose *et al.* have done a pioneering work on greedy geographical routing in which a node chooses to forward a packet to the neighbor closest to the destination [1]. To cope with *concave nodes*, the authors propose a method called *face routing* to surround voids. As stated previously, face routing suffers from several problems, the main being the need for construction of a planar graph, which is difficult in realistic wireless environments [10]. To the best of our knowledge, there is no efficient and localized planarization algorithm proposed for a general connectivity graph (not UDG).

Kuhn *et al.* study asymptotically different geographical routing protocols and compare their performance [11]. Frey considers scalable geographical routing protocol and discuss recent improvements to greedy forwarding [6]. De *et al.* give the bounds on the hop distance in greedy routing [5]. In our previous work, we focused on the analysis of packet losses in greedy geographical routing due to *concave nodes* on the route to the destination [12].

3. BINARY WAYPOINT ROUTING

Our previous work [12] shows that packet losses only occur in fairly few concave nodes that are topology defects: they only have neighbors in the backward direction and they appear at the border of concave voids in the mesh. The idea of Binary Waypoint Routing is to define a forwarding scheme that attempts to go around voids based on routes learnt from destinations.

3.1 Binary Address Space Partitioning

Assume that each node knows its geographical coordinates $a_i = (x_i, y_i) \in \mathcal{A}$, where $\mathcal{A} = [0,1] \times [0,1]$ is a finite square addressing space.

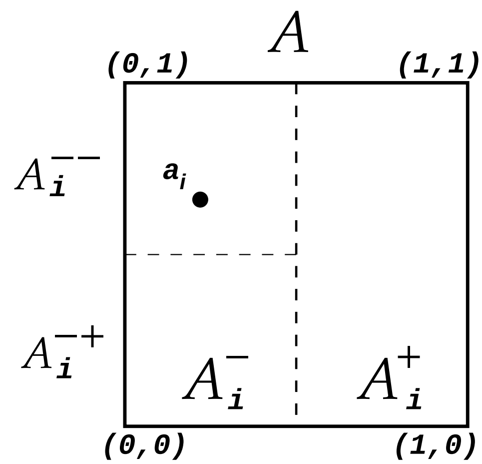

Figure 1: Binary address space partitioning

Node a_i divides the addressing space into an unbalanced binary tree of progressively smaller subspaces $\mathcal{S}_i$ that fulfill the following conditions:

$$\bigcup_j \mathcal{S}_i^j = \mathcal{A},\ i = 1, ..., N,\ j = 0, ..., m \quad (1)$$

$$\forall j \neq k : \mathcal{S}_i^j \cap \mathcal{S}_i^k = \emptyset \quad (2)$$

Partitioning terminates when the length of the shortest border edge of the smallest subspace is not shorter than parameter d_l, which can be the maximal or the average distance to an immediate neighbor node. In the end of partitioning, the smallest subspace contains node a_i.

Figure 1 illustrates partitioning of the addressing space. At the beginning, the tree of node a_i is empty: $\mathcal{S}_i = \emptyset$. Let us define two subspaces of node a_i*:

- $\mathcal{A}_i^+$ such that it does not contain node a_i, i.e. $a_i \notin \mathcal{A}_i^+$, in our case $\mathcal{A}_i^+ = [0.5, 1] \times [0, 1]$,
- $\mathcal{A}_i^-$ such that it contains node a_i, i.e. $a_i \in \mathcal{A}_i^-$, in our case $\mathcal{A}_i^- = [0, 0.5] \times [0, 1]$.

The node inserts $\mathcal{A}_i^+$ into $\mathcal{S}_i$, i.e. $\mathcal{S}_i^0 = \mathcal{A}_i^+$ and checks whether it can further partition $\mathcal{A}_i^-$ by comparing its border edge to d_l. If so, it partitions $\mathcal{A}_i^-$ into $\mathcal{A}_i^{-+}$ such that $a_i \notin \mathcal{A}_i^{-+}$ and $\mathcal{A}_i^{--}$ such that $a_i \in \mathcal{A}_i^{--}$, and inserts $\mathcal{A}_i^{-+}$ into $\mathcal{S}_i$, i.e. $\mathcal{S}_i^1 = \mathcal{A}_i^{-+}$. Partitioning continues until the space is too small for further splitting: the last subspace $\mathcal{A}_i^{-\cdots-}$ that contains node a_i is inserted into $\mathcal{S}_i$.

Partitioning returns a set of $m+1$ subspaces S_i that fulfill Eqs. 1, 2—they cover the addressing space and their size decreases exponentially:

$$\mathcal{S}_i = \left\{\mathcal{A}_i^+, \mathcal{A}_i^{-+}, \ldots, \mathcal{A}_i^{-\cdots-+}, \mathcal{A}_i^{-\cdots--}\right\}. \quad (3)$$

Figure 2 shows the resulting unbalanced binary tree $\mathcal{S}_i$ of length $m+1$. Note that the area of the address subspaces varies from very large to very small. As binary partitioning divides the size of an address subspace by two, the number of subspaces in $\mathcal{S}_i$ grows logarithmically with the number of nodes in the network:

$$m \sim \log N. \quad (4)$$

*Note that the node can split square spaces either vertically or horizontally, but rectangles need to be cut into squares.

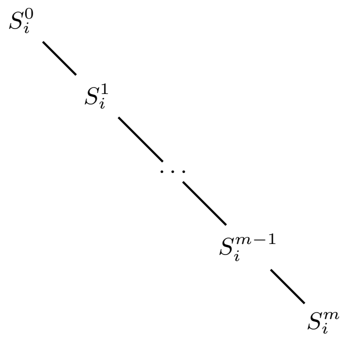

Figure 2: Unbalanced binary tree of addressing subspaces

3.2 Routing tables

Node a_i maintains a routing table $\mathcal{R}_i$ of size $m+1$. Entry j of the routing table contains a source route to node $a_j \in \mathcal{S}_i^j$ called a *binary waypoint* for subspace $\mathcal{S}_i^j$:

$$\mathcal{R}_i[j] = (a_1, a_2, \ldots, a_j) \text{ such that } a_j \in \mathcal{S}_i^j \quad (5)$$

i.e. the source route is composed of all intermediate nodes between node a_i and waypoint a_j in $\mathcal{S}_i^j$. So, the routing table $\mathcal{R}_i$ contains one route to a waypoint node per subspace.

At the beginning, routing table $\mathcal{R}_i$ is empty. We assume that each packet records the route taken towards a destination: it is a list of all nodes that have forwarded a packet between source node a_s and forwarding node a_f:

$$r = (a_s, \ldots, a_f) \quad (6)$$

When node a_f receives a packet, it analyzes its source a_s and identifies to which subspace $\mathcal{S}_f^s$ it belongs. Then, it fills $\mathcal{R}_f[s]$ with the inverted route $r^{-1} = (a_f, \ldots, a_s)$ which leads to a_s, so a_s becomes the waypoint node a_w for the subspace $\mathcal{S}_f^s$.

Nodes keep the best routes to waypoints in all subspaces. To compare routes they need to use a metric—we propose the following one: the *average distance per hop* of route r defined as:

$$H(r) := \begin{cases} |a_f, a_s|/\text{Length}[r] & \text{Length}[r] \neq 0, \\ 0 & \text{Length}[r] = 0, \end{cases} \quad (7)$$

where $|a_i, a_j|$ is the Euclidean distance between nodes a_i and a_j, Length$[r]$ is the length of route r in hops.

If the routing table $\mathcal{R}_f[s]$ is already loaded, node a_f replaces its routing table entry with the route recorded in a received packet, if its average distance per hop is greater than that of the route stored in the routing table entry, i.e. $H(r) > H(\mathcal{R}_f[s])$

3.3 Binary Waypoint Forwarding

When a source node a_s wants to send a packet to destination a_d, it first finds route $r_d = \mathcal{R}_i[d]$ to waypoint node a_w in subspace $\mathcal{S}_i^d$ such that $a_d, a_w \in \mathcal{S}_i^d$. If the route exists it then inserts the route in the packet and sends it to the first node in the source route. If the routing entry is empty node uses greedy forwarding and sends the packet to the neighbor that has the least distance to the destination.

Each node that forwards the packet checks whether it has a better waypoint node for the destination, e.g. an intermediate node a_j compares its route $r'_d = \mathcal{R}_j[d]$ such that $\mathcal{S}_j^d : a_d \in \mathcal{S}_j^d$ with the remaining source route of the packet. Assume that route r'_d has a waypoint $a_{w'}$, if $|a_w, a_d| > |a_{w'}, a_d|$ or the remaining source route in the packet is empty the intermediate node replaces it with its route r'_d, because its waypoint is closer to the destination. At the end node sends the packet to the first node in the source route or uses greedy forwarding if remaining source is still empty.

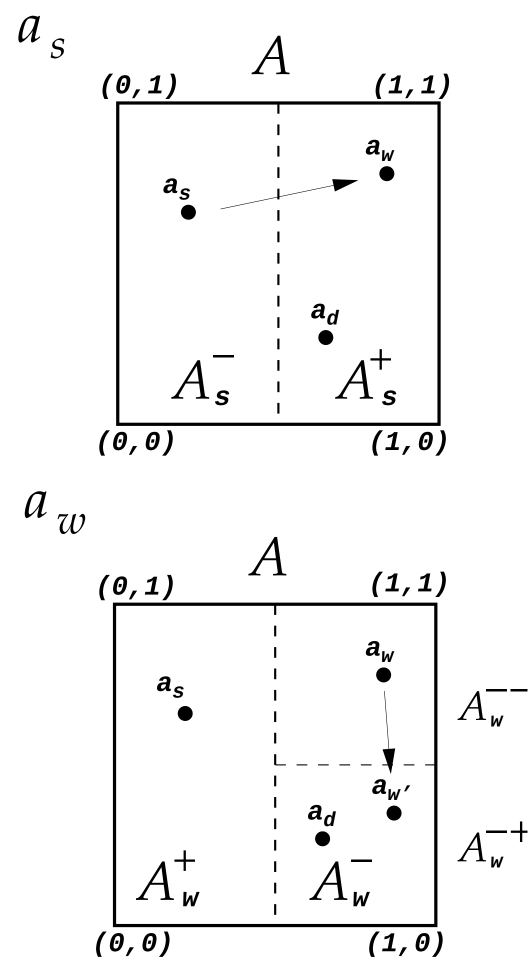

Figure 3: The worst case in Binary Waypoint Forwarding. The view on the addressing space of nodes a_s and a_w.

Let us show that when routing tables $\mathcal{R}_i$ contain sufficient information (all nodes have at least one route to each of their subspaces), waypoint forwarding guarantees packet delivery. Consider Figure 3: in this example destination a_d is in subspace $\mathcal{A}_s^+$ of source node a_s. It sets the route of a packet to waypoint a_w in $\mathcal{A}_s^+$: $r_d = \mathcal{R}_s[\mathcal{S}_s^0 = \mathcal{A}_s^+]$. Assume the worst case in which there is no intermediate node along route r_d that knows a better route to the destination. When waypoint node a_w receives the packet, it finds new route $r'_d = \mathcal{R}_w[\mathcal{A}_w^{-+} = \mathcal{S}_w^1]$ with another waypoint node $a_{w'}$. As the size of each subspace exponentially decreases, this forwarding process advances the packet into an increasingly smaller area near the destination. In the last step, the packet arrives in a node within distance d_l from the destination. This last node can directly deliver the packet to the destination.

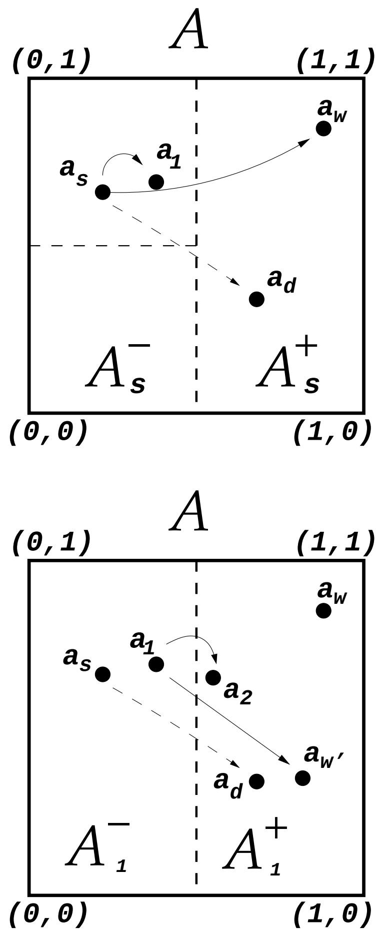

Figure 4: Route adaptation

Figure 4 illustrates route adaptation when an intermediate node knows a better route to the destination. Assume that source node a_s sends a packet to destination node a_d. As previously, it finds that a_d lies in $\mathcal{A}_s^+$ and sets the packet source route $r_d = \mathcal{R}_s[\mathcal{S}_s^0 = \mathcal{A}_s^+]$ to waypoint a_w in $\mathcal{A}_s^+$. Assume that the next hop on the route is node a_1. It finds that the destination lies in its $\mathcal{A}_1^+$ and it has a route $\mathcal{R}_1[\mathcal{A}_1^+]$ with waypoint node $a_{w'}$ that is closer to destination a_d. It thus updates the packet source route with the new route and sends the packet towards the new waypoint $a_{w'}$. In this way, a packet can advance towards the destination in few hops even if at the beginning the source node only knows a route to a waypoint node, which may be far away from the destination.

3.4 Voids and obstacles

Figure 5 shows the behavior of the protocol when there is a void in the network. Source node a_s wants to send a packet to destination node a_d in A_s^+. Assume that the void shadows some part of the subspace $\mathcal{A}_s^+$ with respect to node a_s: the source can receive packets along straight routes from the white area of subspace $\mathcal{A}_s^+$, but it does not receive packets from the shadow region. Recall that at the beginning nodes use greedy forwarding that results in almost straight routes.

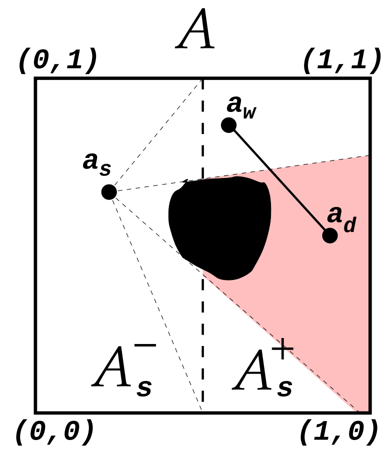

Figure 5: Going around voids

The source node thus learns routes to nodes from the visible area and stores them in its routing table. Assume that the source node has a route to waypoint a_w for subspace $\mathcal{A}_s^+$. At the beginning, packets will go to waypoint a_w, which is a good thing, because they will not go in the direction of the void. A packet on the route to waypoint a_w will probably pass through other nodes that have already received packets from nodes closer to the destination. It is likely, because we can connect a_w and a_d with a straight line: nodes close to a_w receive packets coming from regions near a_d and they record this information in their routing tables, which is then used to adapt routes of packets going to the destination.

4. PERFORMANCE OF BINARY WAYPOINT ROUTING

In this section, we present simulation results of the proposed routing scheme in a large scale wireless mesh network.

4.1 Unit disk graph

We generate our simulations based on a model of wireless mesh networks considered previously [12]: a network is composed of N nodes uniformly distributed in a disk $\mathbb{D}$ of radius L with area $Z = \pi L^2$. We assume that disk $\mathbb{D}$ is embedded in the addressing space $\mathcal{A}$. The probability density function of node position (x, y) is thus:

$$p_l(x,y) = \begin{cases} 1/Z & (x,y) \in \mathbb{D} \\ 0 & (x,y) \notin \mathbb{D} \end{cases} \tag{8}$$

We assume that nodes use omnidirectional antennas and their transmission range is R with a perfect propagation model: there is a link in the connectivity graph between nodes a_1, a_2, if $|a_1, a_2| < R$. Under such assumptions, probability p_r (resp. q_r) that node a_2 is inside (resp. outside) the disk of center a_1 and radius R is a Bernoulli distribution with parameters

$$p_r = \frac{R^2}{L^2} \quad \text{and} \quad q_r = 1 - p_r \tag{9}$$

so that the mean node degree $\bar{k}$ is Np_r and its variance $\mathrm{Var}[k] = Np_rq_r$. For large values of N, the distribution of the node degree converges to the normal (Gaussian) one.

4.2 Comparisons with Greedy Routing and Optimal Shortest Path

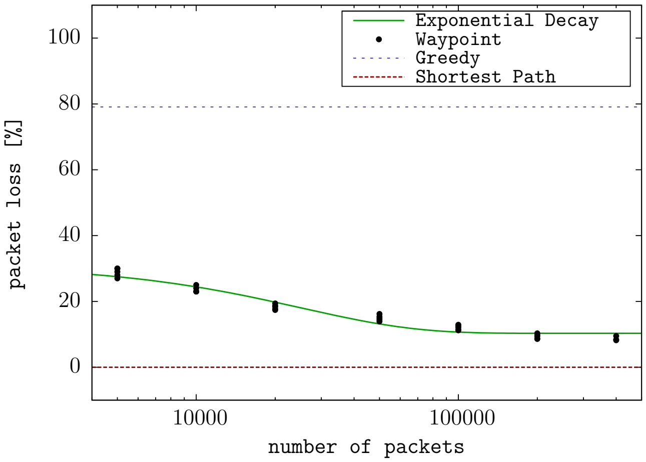

Figure 6: Binary Waypoint Routing, $N = 10,000, R = 0.01425$

We generate a random uniformly distributed network with $N = 10,000$ nodes in a disk embedded in the address space $\mathcal{A} = [0,1] \times [0,1]$. We assume $L = 1.0$ and the radio range of $R = 0.01425$. Under these conditions after 10 simulations the average node degree is equal to 9 ± 0.03. In each simulation, we randomly choose a pair of source and destination for 200,000 packets. They are routed according to the Greedy Geographical Routing. We measure the packet loss rate p_e and obtain the value of $p_e = 0.791 \pm 0.023$, which is fairly high. Next, we simulate the behavior of Binary Waypoint Routing: we have measured p_e as a function of the total number of packets that have been sent in the network (cf. Figure 6). We can see that after 100,000 packets, which is 10 packets per node on the average, nodes have learned routes to waypoints and sending more packets almost does not improve the delivery rate in the network. The loss rate for Binary Waypoint Routing attains $p_e \sim 9\%$. The figure also presents the performance of the Optimal Shortest Path Routing in which all nodes have the complete knowledge of the network topology and use the shortest route to the destination. We obtain very small value of $p_e = 570 \pm 365$ ppm, which is not zero, because the simulated mesh network is not fully connected for $R = 0.01425$.

Figure 7 shows the total number of packets delivered after a specific number of hops for each routing scheme. We can see that Greedy Routing cannot forward packets to destinations that are far away. Moreover, we can see that the shape of the curve for Binary Waypoint Routing is not much different from the shape for Optimal Shortest Path Routing, but more packets are lost.

We ran another simulations for a larger network ($N = 40,000$), but for the radio range $R = 0.00731$ such that the average node degree is almost the same as in the first experiment: 9.01 ± 0.03. We obtain $p_e = 0.91 \pm 0.005$ for Greedy Routing. Binary Waypoint Routing attains almost the same performance as in the first experiment: $p_e \sim 9\%$ (cf. Figure 8). This means that the delivery rate remains constant for a given average node degree even though we have increased the network area four times.

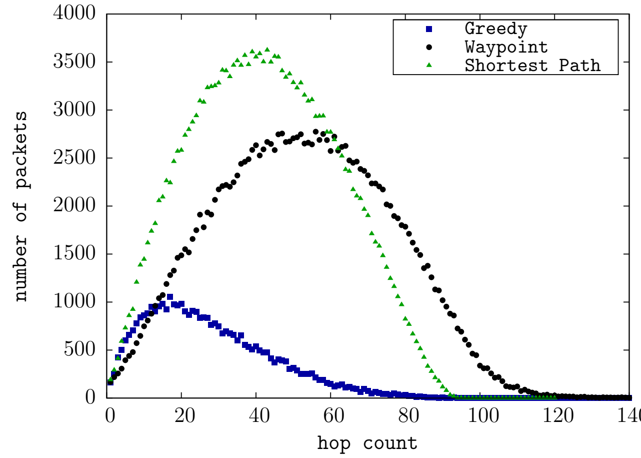

Figure 7: Packet statistics for parameters $N = 10,000, R = 0.01425$ **and** $200,000$ **packets**

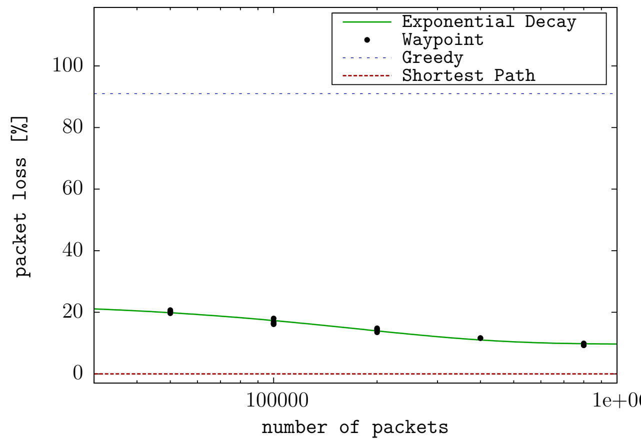

Figure 8: Binary Waypoint Routing, $N = 40,000, R = 0.00731$

Figure 9 presents hop statistics for this case: the average hop count for Binary Waypoint Routing is 102.5 and 82.4 hops for the Optimal Shortest Path. This means that routes in Binary Waypoint Routing are on the average almost 25% longer.

To decrease the packet loss rate in Binary Waypoint Routing we need to increase average node degree—in the next simulation, we have increased the coverage area up to $R = 0.015$ for $N = 10000$ nodes and after 10 simulations we obtained the average node degree equal to 10.03 ± 0.07. Figure 10 shows that the network under Binary Waypoint Routing tends to the state with a smaller loss rate than in the previous case: it achieves the loss rate of 5%.

4.3 Routing in a network with voids

We use a simple model of concave voids (cf. Figure 11) for which we can easily modify size and shape by varying its parameters. A concave void placed inside our circular arena results in deleting all mesh routers lying inside the black area.

We place a void with parameters $b = 0.4$, $c = 0.4$ in the center of the circular network with parameters $N = 10,000, R = 0.01425$. We have observed in 5 simulations

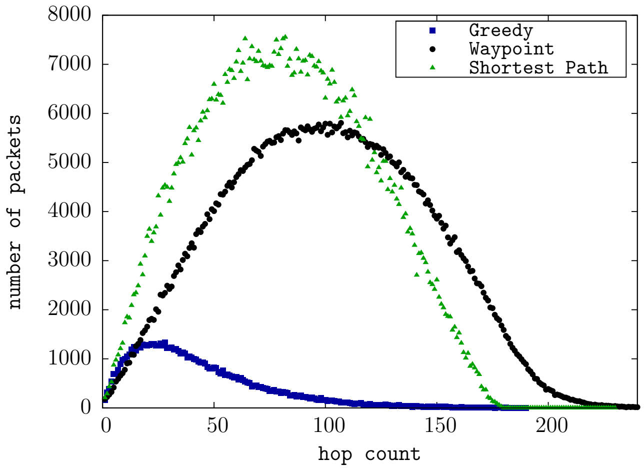

Figure 9: Packet statistics for parameters $N = 40,000, R = 0.00731$, and $800,000$ packets

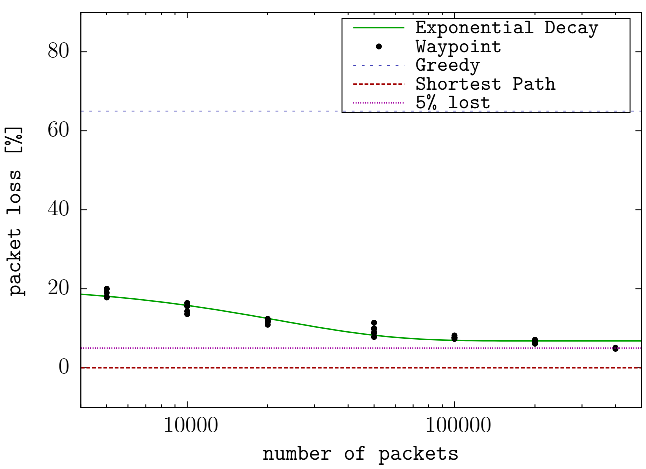

Figure 10: Binary Waypoint Routing, $N = 10,000, R = 0.015$

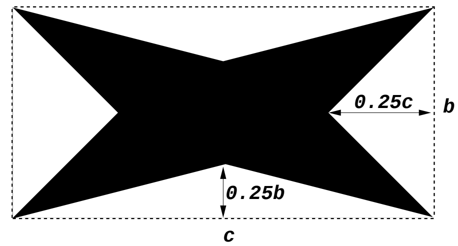

Figure 11: Model of a void.

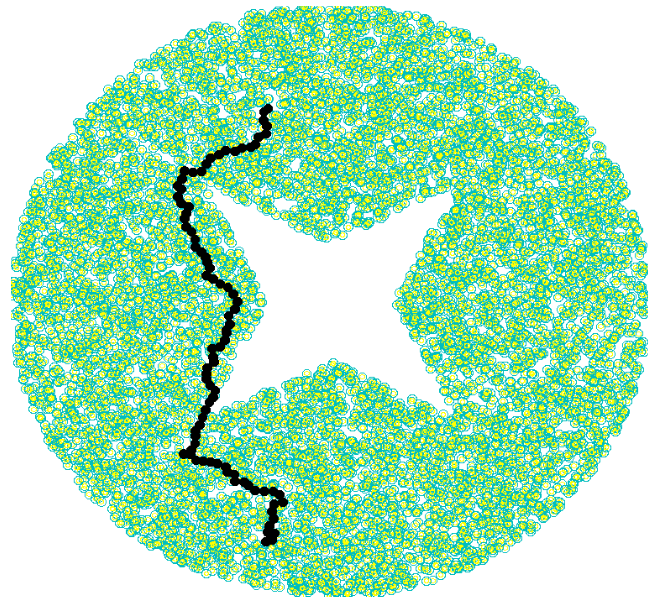

Figure 12: Network with a void, an example route used in Binary Waypoint Routing

that the void with the chosen parameters removes $N' = 1093 \pm 29$ nodes on the average. We have generated 400,000 packets per simulation between random pairs of sources and destinations. The packet drop rate p_e for Greedy Routing significantly grows to $p_e = 0.85 \pm 0.01$. Figure 13 presents its traffic map that shows packet loss and delivery statistics at nodes for a single simulation: a black bar represents the number of packets dropped by a node whereas a red one represents the number of packets forwarded by the same node. We can see an empty area in the center of the network and large loss bars at the border of the concave region.

We have simulated a mesh network with the same parameters, but operating under Binary Waypoint Routing. After 5 simulations we obtained the packet drop rate of $p_e = 0.11 \pm 0.015$. Figure 12 shows an example route used in Binary Waypoint Routing in this network while Figure 14 presents the corresponding traffic map. We can see that the border of the void does not drop as many packets as in the greedy case. Note also that the neighborhood of the void does not forward as many packets as under Greedy Routing. We can observe that Binary Waypoint Routing prefers certain paths related to the topology of the void and results in a cobweb-like traffic pattern observed also for the Optimal Shortest Path Routing (cf. Figure 15). Moreover, we can see that traffic does not go along the concave face of the void as it is the case for face routing (e.g. GPSR). Rather, the nodes have organized themselves into a more efficient ring-like structure to route around the void. We have also observed networks with voids placed in other regions, not only in the center. The packet loss probability p_e remains constant and does not depend on the placement of the void.

Finally, we have generated a more complex network topology with two voids shown in Figure 16. The figure shows an example route of a packet under Binary Waypoint Routing. It is important to note that the route is admittedly suboptimal, but it adapts to the mesh topology. Even though we started the learning process of Binary Waypoint Routing with greedy geographical forwarding that prefers straight paths, we have obtained a curved route. The packet loss rate of Binary Waypoint Routing is $p_e = 0.127$ after 400,000 packets, which is much better than in the greedy case ($p_e = 0.871$). Greedy Geographical Routing simply does not work well in such complex topologies. We obtain the value of $p_e = 6450$ ppm > 0 for the Optimal Shortest Path routing, because some parts of the graph are not connected. The average hop count for Binary Waypoint Routing

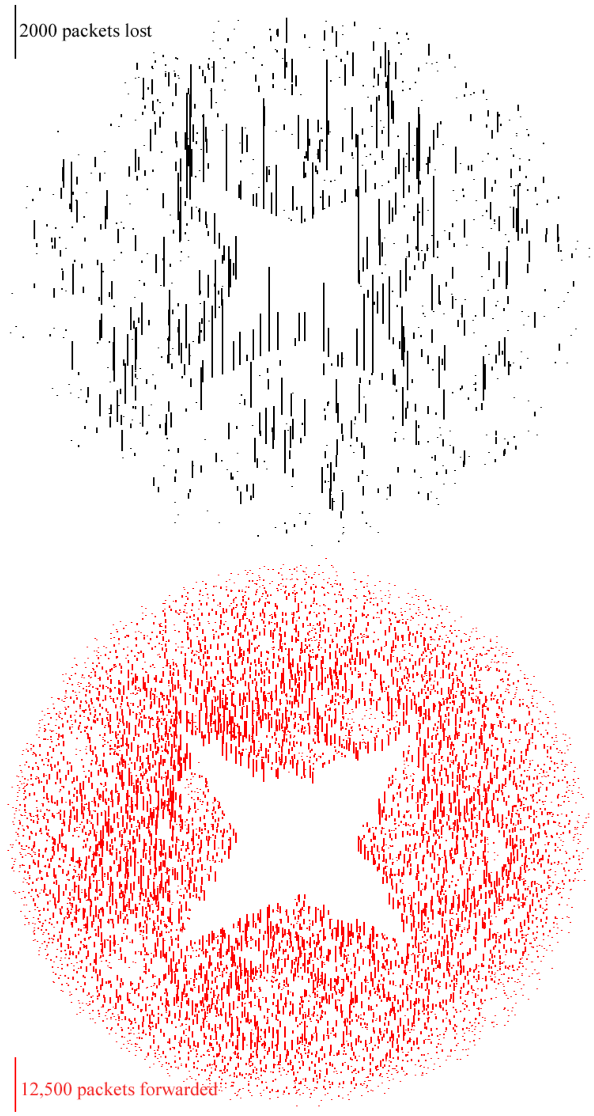

Figure 13: Traffic map for Greedy Routing

is 74.5, which is still larger than in the case of the Optimal Shortest Path routing: 58 hops.

5. CONCLUSION

We have presented *Binary Waypoint Routing*, a novel geographical routing protocol for wireless mesh networks based on the idea of learning and maintaining source routes to a small number of nodes in disjoint addressing subspaces.

Our simulation results show that the proposed scheme achieves high packet delivery rate with a traffic pattern similar to the Optimal Shortest Path Routing. A higher node degree (the number of neighbors) results in an improved packet delivery rate, so we can achieve desired protocol performance by constructing a network with a required node density. The proposed scheme does not require explicit construction of routing tables and its operation does not depend on any graph construction algorithm such as UDG, so the protocol works for any type of wireless connectivity between nodes.

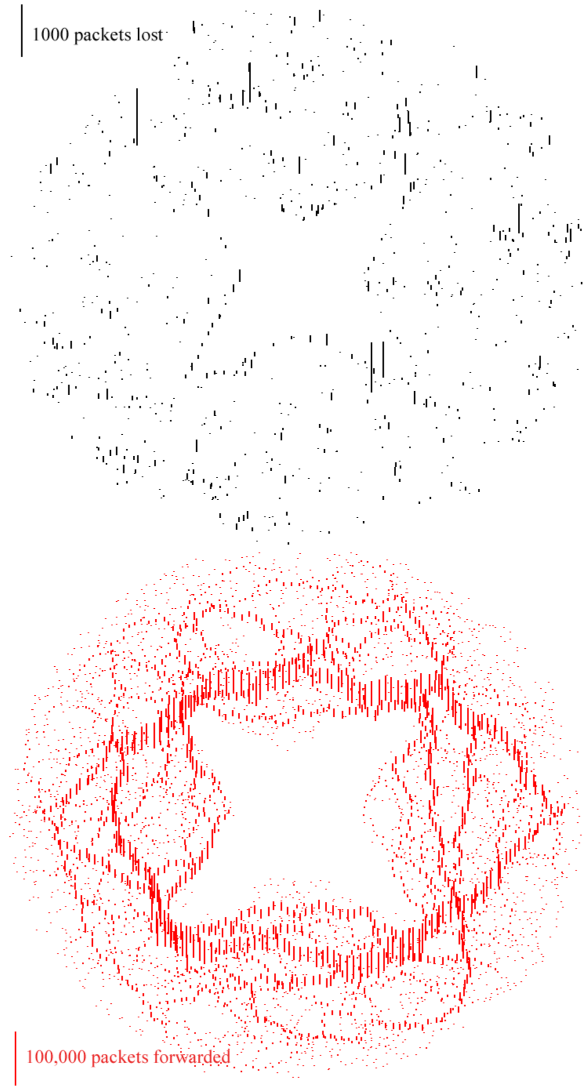

Figure 14: Traffic map for Binary Waypoint Routing

Acknowledgements

This work was partially supported by the European Commission project WIP under contract 27402 and the French Ministry of Research project AIRNET under contract ANR-05-RNRT-012-01.

6. REFERENCES

[1] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with Guaranteed Delivery in Ad Hoc Wireless Networks. *Wireless Networks*, 7(6):609–616, November 2001.

[2] S. Capkun, M. Hamdi, and J. P. Hubaux. GPS-Free Positioning in Mobile Ad-hoc Networks. *Cluster Computing*, 5(2):157–167, April 2002.

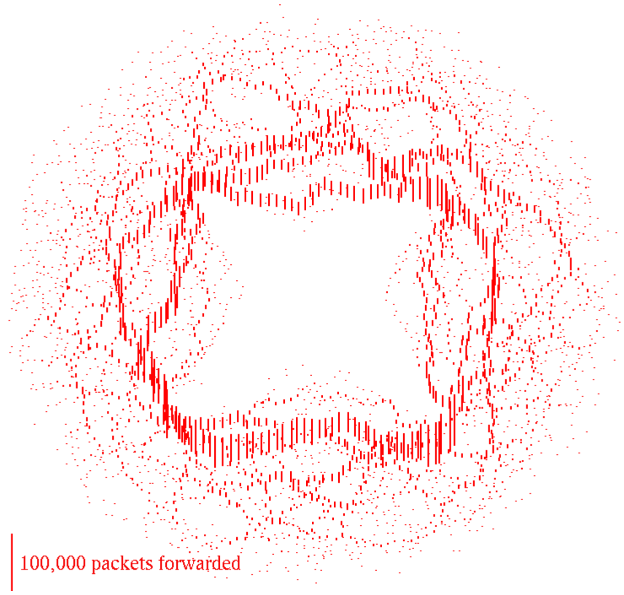

Figure 15: Traffic map for Optimal Shortest Path

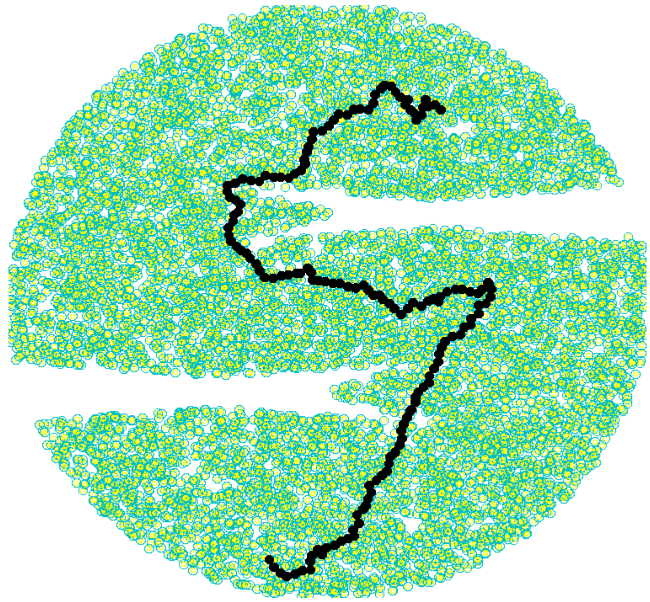

Figure 16: Example route under Binary Waypoint Routing in a complex topology with two voids

[3] N. Carlsson and D. L. Eager. Non-Euclidian Geographic Routing in Wireless Networks. *Ad Hoc Netw.*, 5(7):1173–1193, 2007.

[4] P. Casari, M. Nati, C. Petrioli, and M. Zorzi. Efficient Non Planar Routing Around Dead-Ends in Sparse Topologies Using Random Forwarding. In *Proc. of ICC*, Glasgow, UK, June 2007.

[5] S. De, A. Caruso, T. Chaira, and S. Chessa. Bounds on Hop Distance in Greedy Routing Approach in Wireless Ad Hoc Networks. *International Journal on Wireless and Mobile Computing*, 1(2):131–140, 2006.

[6] H. Frey. Scalable Geographic Routing Algorithms for Wireless Ad-Hoc Networks. *IEEE Network*, July/August 2004.

[7] K. Gabriel and R. Sokal. A New Statistical Approach to Geographic Variation Analysis. *Systematic Zoology*, 18:259–278, 1969.

[8] P. He, J. Li, and L. Zhou. A Novel Geographic Routing Algorithm for Ad Hoc Networks Based on Localized Delaunay Triangulation. In *Proc. of AINA*, Vienna, Austria, April 2006.

[9] B. Karp and H. T. Kung. Greedy Perimeter Stateless Routing for Wireless Networks. In *Proc. of MOBICOM*, Boston, USA, August 2000.

[10] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker. Lazy Cross-Link Removal for Geographic Routing. In *Proc. of SENSYS*, pages 112–124, 2006.

[11] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger. Geometric Ad-Hoc Routing: of Theory and Practice. In *Proc. ACM PODC*, 2003.

[12] E. Schiller, P. Starzetz, F. Theoleyre, and A. Duda. Properties of Greedy Geographical Routing in Spontaneous Wireless Mesh Networks. In *Proc. IEEE GLOBECOM*, 2007.

[13] G. Toussaint. The Relative Neighborhood Graph of a Finite Planar Set. *Pattern Recognition*, 12(4):261–268, 1980.