Demonstrating the Resilience of Geographical Routing to Localization Errors

Stefano Basagni
ECE Department
Northeastern University
Boston, MA, U.S.A.
E-mail: basagni@ece.neu.edu

Michele Nati and Chiara Petrioli
Dipartimento di Informatica
Università di Roma "la Sapienza"
Roma, Italy
E-mail: {nati,petrioli}@di.uniroma1.it

1. Introduction

This demonstration concerns Geographic Forwarding (GF) as an effective solution for data dissemination (from sensors to a *sink*) in wireless sensor networks (WSNs). In particular, we focus on demonstrating the different degrees of resilience of a recent solution, ALBA–R [1], to localization errors, which are highly likely to occur in WSNs.

GF routing protocols are based on the nodes knowing their own location information as well as that of the sink, which is the intended destination of a packet. When a node has a packet to send it attempts to forward it in the direction of the sink. Several forwarding methods have been proposed [2]. For instance, according to a typical GF routing protocol, like GeRaF [3], forwarding happens by the sender node s requesting which among its neighbors can relay the packet and provide a positive advancement to the sink. Nodes that receive this request, based on the distance from s to the sink carried by the request message, candidate themselves as relays. At this time, s chooses one of them that provides high advancement as relay, and sends the packet to it. Improvements can be obtained over this basic scheme by favoring relays which are less congested and enabling transmissions of bursts of packets back to back, as happens in ALBA [4].

With respect to these simple and effective forwarding methods, problems may arise when no nodes exist in the direction of the sink that can relay the packet for a given node. In this case the packet is stuck at a so called "dead end" node. For this problem, many solutions have been proposed, that range from planar graph traversal to flooding-based techniques, to cost-based techniques, where the cost is usually the distance from the sink [5]. One recently proposed solution, termed ALBA–R [1], for instance, enhances ALBA with a mechanism for guaranteeing the delivery of data packets even in presence of dead ends in the network.

The aim of this demonstration is that of showing the capabilities of ALBA–R of efficiently performing GF, and in particular its resilience to localization errors. We demonstrate that, while in presence of both dead ends and localization errors ALBA packet delivery ratio suffers quite remarkably, ALBA–R is able to deliver all packets to the sink, even those from dead ends and even when the estimated location of a node is considerably distant from it actual position.

2. ALBA–R

In this section we briefly review ALBA–R and in particular the "rainbow" mechanism for dealing with dead ends, which extends ALBA into ALBA–R.

For energy conservation purposes, nodes follow asynchronous awake/asleep schedules. When a node has a packet to transmit it initiates a contention, asking its currently awake neighbors toward the sink (eligible relays) to volunteer as relays. All the eligible relays of a node compute two values, namely, the Geographic Priority Index (GPI), which gives an indication of the packet advancement that the node can provide, and the Queue Priority Index (QPI), which is a measure of forwarding effectiveness as perceived by the relay. Based on their GPI and QPI eligible relays are partitioned into regions which are scanned sequentially. The contention mechanism leads to the selection of a relay in the first non-empty region, i.e., the one with eligible relays with the highest GPI and QPI. In this way both positive advancement and balancing of the traffic among the possible relays are enabled.

Now onto the rainbow mechanism. Let x be a node engaged in packet forwarding. Let us denote with F the portion of node x transmission area where relays offering positive advancement are located. Similarly, we call F^C the remaining part of x's transmission area (Figure 1).

With $C_1, \ldots, C_h$ we indicate the *colors* that nodes assume according to their perceived ability to forward packets directly to the sink or not. Initially, all nodes are colored C_1 (for instance, yellow, Y) and function according to the advancement rules of ALBA–R. If

1-4244-1455-5/07/$25.00 2007 IEEE

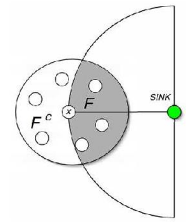

Figure 1. Forwarding regions

there are no connectivity holes all nodes remain yellow. However, if a node x is unable to relay a packet for more than a fixed number of attempts, it deduces that it may be a dead end (see [1] for a the calculation of a reasonable and effective number of attempts and further details). In this case x acts as follows. It increasingly lowers its participation in contentions initiated by other nodes, since it is likely not to be able to advance packets toward the sink. When this probability reaches 0 it changes color (say, it becomes a red, R, node). The first action is important for the node to progressively realize that it is on a dead network branch and must stop uselessly volunteering as a relay. The second action modifies the behavior of the node so as to route both incoming and locally originated packets around the hole. Red nodes behave differently in the relay search phase, as they try to send the packet away from the sink by searching for yellow or red relays in region F_C. When a yellow node (which has a greedy route available) is reached, regular ALBA–R operations are resumed. From that point on the path going toward the sink is made up only of yellow nodes. Each hop provides a positive advancement toward the sink.

Red nodes may be unable to find routes that lead to the sink through red or yellow nodes only. In this case, a red node progressively stops offering itself as a relay for red nodes and changes its color again, turning to C_3 (e.g., blue, B). According to this new color, it resumes relay searches in F (instead of in F_C) but only looks for blue or red neighbors. Blue nodes do not volunteer to be next hops relays for red or yellow nodes, but they resort to them to find a route. Packets generated by a blue node will advance toward the sink through blue nodes till they reach a red node. Then they will travel away from the sink (via red nodes) till they reach a yellow node. From that point on, they will be routed on the "yellow brick route" (i.e., on a route to the sink traversing only yellow nodes). If blue nodes are unable to find blue or red eligible forwarders (blue or red neighbors in F) they switch to C_4 (violet). Similar to red nodes, violet nodes search into F_C for eligible relays. However only blue or violet nodes can answer violet nodes searching for relays. An example of coloring for a specific topology is illustrated in figures 2 through 5. The sink is the "green" square node. The figures show the progressive coloring of the nodes. In Figure 2 the colored nodes are all yellow. In Figure 3 the red nodes are added. The nodes that resulted to be blue are shown in Figure 4. The final coloring, obtained with the violet nodes, is shown in Figure 5.

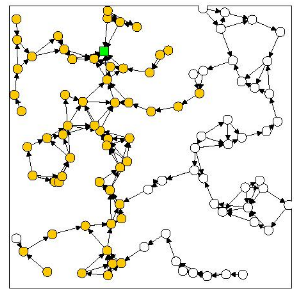

Figure 2. Yellow nodes

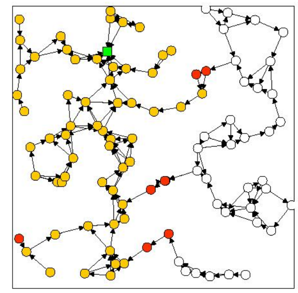

Figure 3. Y + R

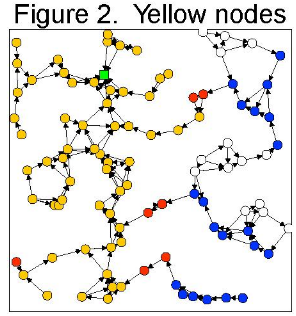

Figure 4. Y + R + B

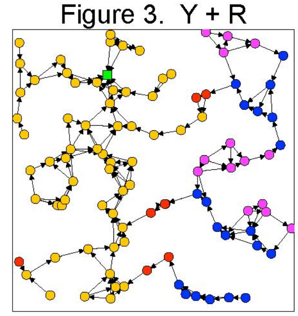

Figure 5. Y+R+B+V

The rainbow process can be generalized for any number of colors, $C_1, \ldots, C_h$. In particular, C_1-nodes are part of a direct route to the sink passing only through other C_1-nodes. When a node is forced to change to color C_k, it searches region F_C (if k is even) or region F (if k is odd). In both cases, nodes colored C_k only participate in the relay selection process initiated by C_k or C_{k+1}-nodes, and can forward their packets only to nodes colored C_k- or C_{k-1}.

It is possible to show [1] that the rainbow mechanism is always able to find a route around connectivity holes.

3. Demonstration Settings

A number of EYES sensor nodes [6] are deployed in order to form a multi-hop wireless networks. Each node is identified by a unique ID and has two kinds of coordinates: 1) Its *physical coordinates* indicate the real position of the node in the network and determine the actual network topology graph. 2)The node's *estimated coordinates* represent its position as estimated by the node by means of some localization protocol [7].

A special node, called the *inspector*, directly connected to a laptop, is used for three important operations. 1) *Node programming*: Protocol code is transferred (wirelessly) from the laptop to each node through the inspector. 2) *Network configuration*: A specific network topology (including the selection of the sink) can be manually or randomly generated using the laptop, and communicated to the nodes. The inspector also communicates the estimated position to each node, sending to the nodes information about which node is going to be the source of the packets, and which metrics should be monitored by the sensor (e.g., either light or temperature). 3) *Network monitoring* allows us to visualize on the laptop the state of each node in the network as well as the path followed by data packets in real-time, thus being able to control route formation and the protocol adaptiveness to changes in the topology. In this case the inspector acts like a "sniffer." Thanks to the inspector no other node involved in the demonstration is directly connected to the laptop.

3.1. Packet structure

Each *data packet* transmitted or relayed comprises the following fields. (a) A payload carrying the sensed information such as temperature or light (2 Bytes). (b) The coordinates of the source node (8 Bytes). (c) The "color" of the transmitting node (which is an attribute of ALBA–R nodes) (1 Byte), and (d) a sequence number (1 Byte) of the packet. This information is needed for computing the packet delivery ratio.

3.2. Software control tool

We developed a software tool, running on the laptop, to aid in the programming of the sensor nodes and to collect real-time information about the network status (which node has a packet, packet progress toward the sink, etc.). Our tool is made up of three main components: 1) The **Configuration Tool** is used to send commands and to code the nodes in the network through the inspector. For instance, the packet source node is selected through this component of our software, as it is the selection of the metric to be monitored. This tool is also used to select the protocol to be deployed. Currently nodes are programmed to run both ALBA and ALBA–R. 2) The **Topology Tool** shows the network topology as a graph and the colors currently assumed by each of the network nodes according to ALBA–R. By using different colors we highlight the path followed by the packets. By means of this tool it is also possible to arbitrarily modify the estimated coordinate of each node by just dragging and dropping the selected node on its new position. This can be interactively done during the demo, and hence by the demo attendants. 3) The **Application Tool** shows a chart of the reported metrics. The values of light or temperature carried by the packets received by the sink are plotted on the laptop screen.

4. Demonstration

We demonstrate how ALBA–R is an effective solution for dealing with both dead ends and localization errors.

4.1. ALBA–R and dead ends

We start the demonstration without localization errors. The user, via the inspector, selects the source of the packets (node 5 in Figure 6), and the source starts generating packets at a given rate (1 packet every 3s). In the following figures each node is depicted by a circle. The number within the circle is the unique ID of the node. The number in brackets aside each node represent its Euclidean distance from the sink. When packet transmission begins, the Topology Tool shows the different path followed by each packet. Nodes 1, 2, 3, 4, 6, 7 and 10 are yellow nodes (identifiable by the "Y" next to the node). No other node is able to transmit packets to the sink unless it chooses a different color. For instance, this is the case of our selected source, node 5. Through the rainbow mechanism *all* nodes are able to find a route to the sink. Consider, for instance, nodes 5 and 8. They do not have yellow neighbors in F, but they do in F_C (node 5 has node 4 as a yellow neighbor, and node 8 has the yellow neighbor 7). Therefore, they become red. Node 9, in turn, becomes blue, since it has red nodes (nodes 5 and 8) in its F region.

4.2. Resilience to node failure/removal

In order to demonstrate ALBA–R adaptivity we show how the protocol continues to work properly when some node are manually disconnected from the network. If a sequence of nodes in the direction of the sink exists ALBA–R is always able to find it. In Figure 8 we show the case when node 4 is removed. Node 5, no longer having yellow nodes in F_C, turns blue and looks for (blue or red) relays in F. Since there are no such nodes (node 9 is in node 5 F_C region), it switches to violet, looks in F_C and finds node 9 as a suitable relay.

4.3. ALBA–R and localization errors

We demonstrate the complete resilience of ALBA–R to localization errors. Through the Topology Tool we are able to modify the topology of the network, so that a node will be assigned coordinates different from its real ones.

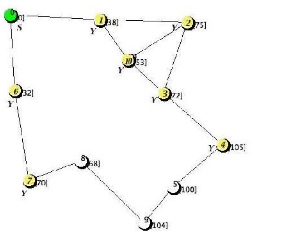

Figure 6. Yellow nodes

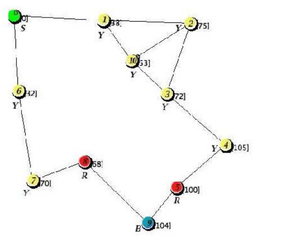

Figure 7. "R effect"

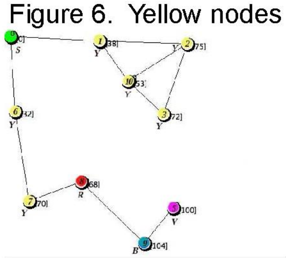

Figure 8. Removing a node

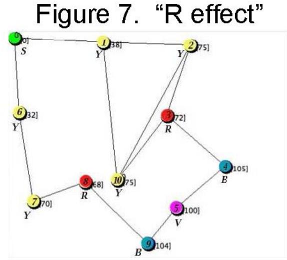

Figure 9. Localization error

An example is shown in Figure 9, where a node previously distant 53, node 10, from the sink (Figure 6) has been dragged at distance 75, i.e., in the place indicated by its estimated coordinates (instead of its real ones, as before). The link to other nodes will stay the same, since the capability of communication between two nodes is unaffected by the localization error.

The faulty coordinates are communicated to the node by the inspector. (In order to show that each node is using its estimated coordinates, the value of the coordinates is piggy-backed in each data packet transmitted from a given node to a given relay.)

After this configuration phase is completed, a node is designated as the source of the data packets. We therefore show, that, independently of the source and of the localization error that affect the perceived coordinates of some nodes, every packet is delivered to the sink. In other words, we show how the same mechanism that allows ALBA–R to effectively handle dead ends makes ALBA–R also transparent to localization errors. The path traversed by each packet is shown on the Topology Tool as the packet progresses toward the sink.

As an example, consider Figure 9 that shows the coloring of the nodes after node 10 coordinates are perturbed as described above. It is easy to observe that node 10 remains yellow, since it still has a yellow node in its forwarding region. However, node 3, previously yellow, does not have any more yellow nodes in its F region. Therefore, it searches in its F_C region, where it finds two yellow nodes. As a consequence, it turns red. For similar reasons, node 4 turns blue, and node 5 switches to violet. Node 5 has now the chance to alternatively select routes to the sink going through either node 4 or node 9 (as blue as node 4). The choice of the route is based on the current values of the GPI and QPI of nodes 4 and 9.

5. Acknowledgments

This paper was supported in part by the Italian project "Tecnologia e scienza per le reti di prossima generazione," (FIRB: RBIN047MH9), and by the National Science Foundation (Award NSF ECCS 0742054).

References

[1] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, "Efficient non-planar routing around dead ends in sparse topologies using random forwarding," in *Proceedings of the IEEE International Conference on Communications, ICC 2007*, Glasgow, Scotland, June 24–28 2007.

[2] I. Stojmenovic, "Position based routing in ad hoc networks," *IEEE Communication Magazine*, vol. 40, no. 7, pp. 128–134, July 2002.

[3] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaF) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, October-December 2003.

[4] P. Casari, M. Nati, C. Petrioli, and M. Zorzi, "ALBA: An adaptive load–balanced algorithm for geographic forwarding in wireless sensor networks," in *Proceedings of the 25th IEEE Military Communication Conference, MILCOM 2006*, Washington, DC, October 23–25 2006, pp. 1–9.

[5] D. Chen and P. K. Varshney, "A survey of void handling techniques for geographic routing in wireless networks," *IEEE Communications Surveys*, vol. 9, no. 1, pp. 50–, First Quarter 2007.

[6] P. J. M. Havinga, S. Etalle, H. Karl, C. Petrioli, M. Zorzi, H. Kip, and T. Lentsch, "EYES—energy efficient sensor networks," in *Proceedings of the 4th IFIP TC6/WG6.8 International Conference on Personal Wireless Communications (PWC) LNCS 2775, M. Conti, S. Giordano, E. Gregori and S. Olariu (eds.)*, Venezia, Italy, September 23–25 2003, pp. 198–201.

[7] A. Savvides and M. B. Srivastava, "Location discovery," in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. Piscataway, NJ and New York, NY: IEEE Press and John Wiley and Sons, Inc., April 2004, ch. 8, pp. 231–254.