The Utility of Hello Messages for Determining Link Connectivity

Ian D. Chakeres
Dept. of Electrical & Computer Engineering
University of California, Santa Barbara
idc@engineering.ucsb.edu

Elizabeth M. Belding-Royer
Dept. of Computer Science
University of California, Santa Barbara
ebelding@cs.ucsb.edu

Abstract

Future wireless applications will take advantage of rapidly deployable, self-configuring multihop ad hoc networks. Because of the difficulty of obtaining IEEE 802.11 feedback about link connectivity in real networks, many multihop ad hoc networks utilize hello messages to determine local connectivity. This paper uses an implementation of the Ad hoc On-demand Distance Vector (AODV) routing protocol to examine the effectiveness of hello messages for monitoring link status. In this study, it is determined that many factors influence the utility of hello messages, including allowed hello message loss settings, discrepancy between data and hello message size and 802.11b packet handling. This paper examines these factors and experimentally evaluates a variety of approaches for improving the accuracy of hello messages as an indicator of local connectivity.

Keywords

ad hoc networking, hello messages, mobile networks

INTRODUCTION

Infrastructured IEEE 802.11b networks are becoming ubiquitous. These networks offer high bandwidth wireless connectivity well-suited for a variety of traffic types, including multimedia distribution. One drawback of infrastructured networks is the complexity of deploying and configuring these networks. Ad hoc networking protocols do not suffer from this limitation. By using a multihop ad hoc network connectivity is maximized.

For quality multimedia sessions, routing paths between nodes in an ad hoc network must be continually monitored. Numerous ad hoc routing protocols [1, 3, 4, 7, 12] make use of periodic broadcast messages to determine local connectivity. Also, because of the difficulty of obtaining IEEE 802.11 feedback about link connectivity in real networks, many current protocol implementations utilize hello messages [2, 3, 6, 9, 10].

The basis of using hello messages to determine connectivity stems from the assumption that reception of a hello message indicates a viable communication channel with the source of the hello. This mechanism works well on wired networks, which experience few packet losses and connectivity changes. However, when used in wireless ad hoc networks the effectiveness decreases due to many factors. Some of the factors that have significant effect are: hello loss settings, hello packet size and 802.11b packet handling.

The Ad hoc On-demand Distance Vector (AODV) routing protocol [11, 12] is a reactive protocol designed for routing in ad hoc mobile networks. In this paper an implementation of AODV is utilized to determine the effectiveness of hello messages for determining local connectivity. A variety of approaches for improving the accuracy of hello messages as an indicator of local connectivity are examined.

AODV PROTOCOL OVERVIEW

The AODV protocol is a reactive routing protocol; routes are determined only as needed. When a route is required, AODV uses a route discovery process to learn a route. Once a route is established, it is maintained as long as it is needed through a maintenance procedure. These two operations are described in detail in subsequent sections.

AODV maintains routes using a soft state approach; if a route is not used it is expired after a specified time. AODV may use either of two methods to detect breaks in a route: link layer feedback or hello messages. Due to the difficulty in obtaining link layer feedback, only AODV's operation using hello messages is described in this paper.

Hello Messages

Network connectivity may be determined through the reception of broadcast control messages. Any broadcast control message also serves as a hello message, indicating the presence of a neighbor. When a node receives a hello message from its neighbor, it creates or refreshes the routing table entry to the neighbor (see Figure 1(a)). To maintain connectivity, if a node has not sent any broadcast control message within a specified interval, a hello message is locally broadcast. This results in at least one hello message transmission during every time period. Failure to receive any hello message from a neighbor for several time intervals indicates that neighbor is no longer within transmission range, and connectivity has been lost.

Two variables control the determination of connectivity using hello messages: HELLO_INTERVAL and ALLOWED_HELLO_LOSS. HELLO_INTERVAL specifies the maximum time interval between the transmission of hello messages. ALLOWED_HELLO_LOSS specifies the maximum number of periods of HELLO_INTERVAL to wait without receiving a hello message before detecting a loss of connectivity to a neighbor. The recommended value for HELLO_INTERVAL is one second and for ALLOWED_HELLO_LOSS is two [11]. In other words, if a

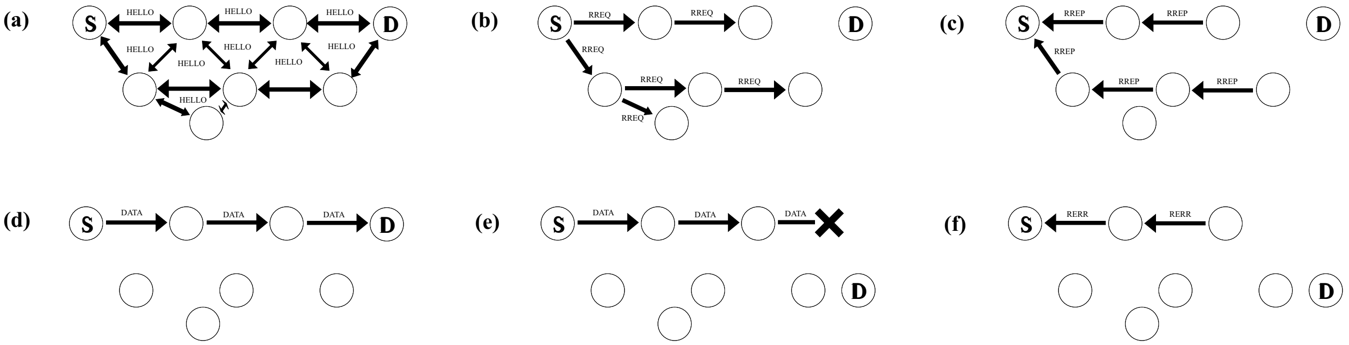

Figure 1. AODV Operation.

hello message is not received from a neighbor within two seconds of the last message, a loss of connectivity to that neighbor is determined.

Route Discovery

When a source needs to send packets to a destination, it first must determine a path for communication. The source node begins route discovery by broadcasting a route request (RREQ) message containing the IP address of the destination. When an intermediate node receives the RREQ, it records the *reverse route* toward the source and checks whether it has a route to the destination. If a route to the destination is not known, the intermediate node rebroadcasts the RREQ. RREQ propagation is illustrated in Figure 1(b).

When the destination, or an intermediate node with recent information about a route to the destination, receives the RREQ, a route reply (RREP) is generated. The RREP is unicast back to the source using the *reverse route* created by the RREQ. For example, in Figure 1(c) two nodes have recent information about the destination because hello messages are being used. These two nodes unicast a RREP to the source. As the RREP propagates toward the source, a *forward route* to the destination is created at each intermediate hop. When a RREP reaches the source, the source records the route to the destination and begins sending data packets to the destination along the discovered path, as illustrated in Figure 1(d). If more than one RREP is received by the source, the route with the lowest hop count to the destination is selected.

Route Maintenance

When a link breaks along an active path, the node upstream of the break detects the break (see Figure 1(e)) and creates a route error (RERR) message. The RERR message lists all destinations that are now unreachable, due to the link break. The node then sends the RERR message toward the source. Each intermediate hop deletes any broken routes and forwards the RERR packet toward the source, as shown in Figure 1(f). When the source receives the RERR packet it determines whether it still needs the route to the destination. If so, the source creates a RREQ and begins the route discovery process again.

IEEE 802.11B OVERVIEW

The MAC layer protocol used for transmitting unicast packets in the IEEE 802.11 standard is the Distributed Coordination Function (DCF) [5]. This standard uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for unicast data transmissions between neighboring nodes. A node wishing to unicast a data packet to its neighbor broadcasts a short RTS control packet. When its neighbor receives the packet, it responds with a CTS packet. Once the source node receives the CTS, it transmits the data packet. After receiving this data packet, the destination then sends an acknowledgment (ACK) to the source, signifying reception of the data packet. The use of the RTS-CTS control packets reduces the potential for the hidden-terminal problem.

Broadcast data packets, RTS and CTS control packets are sent using the unslotted Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA) [5]. When a node wishes to broadcast a packet, it first senses the channel. If it does not detect an on-going transmission, it broadcasts the packet. On the other hand, if it does detect a transmission, it calculates a random backoff time and then waits this amount of time before attempting the transmission again.

The IEEE 802.11 standard [5] specifies two data rates, 1 and 2 Mbps. The IEEE 802.11b standard [5] introduces higher data rates, 5.5 and 11 Mbps. These higher data rates are achieved by different coding schemes at the physical layer. The MAC layer operation is identical to IEEE 802.11, as described above. The IEEE 802.11b standard also allows for automatic rate changing, as long as both the source and the destination support the desired rate.

EXPERIMENTS

The AODV implementation [2] is a user space daemon. The implementation includes buffering during route discovery and jitter between sending of hello messages. Hello messages are sent at a periodic rate minus some random jitter. This is necessary to combat synchronization of hello messages, which results in hello message losses.

The experiments were performed in two separate environments: in a lab and in a field. In the lab all the laptops were located on the same desk and connectivity was controlled us-

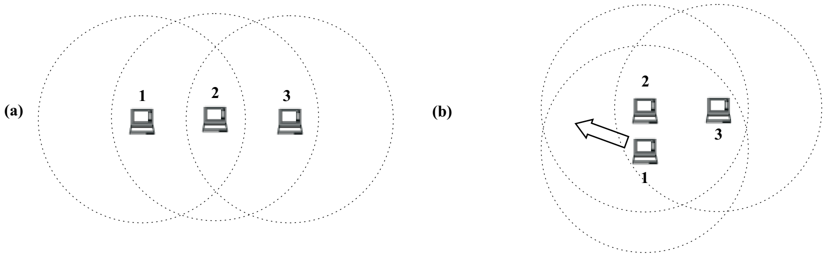

Figure 2. Experimental Topologies.

ing *iptables* for MAC layer filtering. *iptables* is also used to simulate mobility by instantaneously changing connectivity between nodes. The lab provided a controllable environment for testing the implementation. Lab tests also provide a benchmark with which to compare the in-field results.

The field tests occurred in a large open field. There were no obstacles or objects within 75 meters of any node. This significantly reduces the negative effects, such as multi-path, caused by obstacles. In these experiments, connectivity is controlled by distance.

Two topologies are used in the experiments. The first topology, as shown in Figure 2(a), is a simple multihop network that consists of three nodes organized linearly. The transmission range of the nodes is depicted by the dotted circles. This topology was chosen as a verification of route discovery, as well as to provide a baseline with which to compare other results.

In the second topology, nodes 2 and 3 are stationary and in the same position as in the static topology. Node 1 begins one meter from node 2 and moves away from node 3, as indicated by the arrow in Figure 2(b). The final orientation of the three nodes is identical to the static topology. Node 1 is mobile for one minute while moving from its initial to final position; approximately 2.7 kilometers per hour. In the lab mobility is simulated by controlling connectivity using *iptables*.

For each experiment, node 1 was the traffic source and node 3 was the destination. The data traffic consisted of 512-byte UDP packets, unless otherwise noted. The data packets were transmitted at a rate of ten packets per second. There were 1000 total data packets originated by the source in each test.

Both topologies were tested in the lab and in the field. The default rate setting for 802.11b was 11 Mbps. Two values for ALLOWED_HELLO_LOSS were examined: the recommended value of two, as well as an experimental value of three. These tests are referred to as 1/2 and 1/3, respectively, indicating that one hello must be received in every two (three) hello intervals to indicate connectivity. An ALLOWED_HELLO_LOSS of three is more tolerant to packet loss. Three runs of each of the described tests were performed. The results were then averaged to determine the performance.

For the experiments three Dell Latitude C610 laptops were used to run the AODV routing daemon. The laptops have Mobile Pentium III-1000/766 MHz processors and 256 MB of RAM. The operating system utilized was Linux kernel version 2.4.7-10. For wireless connectivity, Lucent Orinoco IEEE 802.11b wireless cards were used with the Orinoco driver (wvlan). The wireless cards were set in ad-hoc mode on channel 1. The sensitivity of the antenna was set to its highest setting, the default setting. The RTS/CTS setting was set to 1 byte; DCF is used for all unicast packets larger than 1 byte. No WEP encryption was utilized.

During initial testing for the field tests many factors affected the quality of the wireless channel, including the distance between communicating nodes, the height of the laptops from the ground, the relative orientation of the laptops to each other and the ground, the settings for the IEEE 802.11b hardware (rate, sensitivity, transmit power, etc.), ambient weather (temperature, moisture, etc.) and location of the experiments.

For this reason the following design experiment choices were made for the field tests. All the test were run on the same day over a five hour period to minimize any effect due to ambient weather. For the field tests, the laptops were placed on pedestals at one-half meter above the ground to adjust their range to a usable distance. The distance between the nodes varied throughout the day, and was configured as needed to acquire a multihop network. With the laptops one-half meter above the ground, the range of the wireless cards varied between 45 and 55 meters. To combat issues related to the relative orientation of the laptops to each other and the ground, all the laptops faced the same direction, with their screens open facing node 1 and keyboards parallel to the ground for all tests.

RESULTS AND DISCUSSION

The results for static topology are presented in Table 1. When compared with the 1/3 results, the in-lab 1/2 results show a slight degradation in packet delivery. The cause of this behavior is due to false detection of a link break resulting from lost hello messages. As expected the 1/3 performance is better as it is more tolerant to hello messages loss. The in-field performance is further decreased due to more packet losses, caused by multi-path, fading, and other real-world effects.

Table 1. Static Topology Results.

Test Scenario	Connection Strategy	Data Rate	Percent Delivered
lab	1/2	11 Mbps	99.9
lab	1/3	11 Mbps	100
field	1/2	11 Mbps	99.1
field	1/3	11 Mbps	99.8

Table 2. Mobile Topology Results.

Test Scenario	Connectivity Strategy	Data Rate	Percent Delivered
lab	1/2	11 Mbps	97.4
lab	1/3	11 Mbps	96.2
field	1/2	11 Mbps	60.7
field	1/3	11 Mbps	57.0

Table 2 presents the performance for the mobile topology. In the lab there is a slight decrease in throughput because node 1 believes (due to received hello messages) that it can communicate with node 3 while it cannot; for a few seconds packet are lost. In the field tests the throughput is extremely low. Figure 3 illustrates the reception of data packets during a single test run; many data packets were lost as the distance between the two nodes increased. In Figure 3, node 1 continues to receive hello messages directly from node 3 until 60 seconds have elapsed; it is therefore sending data directly to node 3 during this time. After 60 seconds, hello messages are no longer received from node 3 and a link break is detected. A multihop route through node 2 is then discovered. This multihop route is used for the remainder of the test. These data packet losses are not experienced in the in-lab tests because connectivity is binary (on/off), as it is controlled by *iptables*. Table 2 also shows that the 1/2 connectivity strategy outperforms 1/3. This is because 1/2 causes routes to timeout more quickly, resulting in prompt route discovery. 1/3, on the other hand, continues to send packets along a route where data packets are not being received. Because of its higher performance in this mobile experiment, the 1/2 connectivity strategy is used for the remainder of the experiments.

Figure 3. Packet Reception for 11 Mbps Experiment.

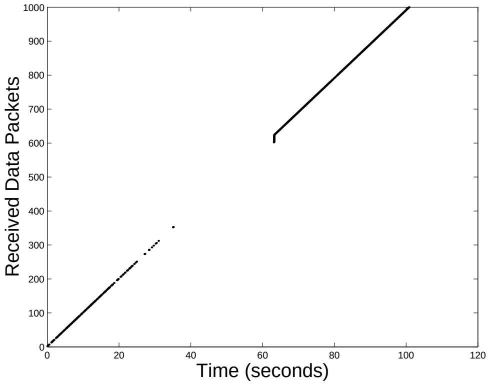

Table 3. Size Variation Results.

Test Scenario	Connectivity Strategy	Hello Size	Data Rate	Percent Delivered
field	1/2	20 bytes	11 Mbps	60.7
field	1/2	512 bytes	11 Mbps	80.8

Table 4. Rate Variation Results.

Test Scenario	Connectivity Strategy	Data Rate	Percent Delivered
field	1/2	11 Mbps	60.7
field	1/2	auto	74.3
field	1/2	1 Mbps	84.5

To examine why hello messages were being received during a portion of the test but data packets were not, further experiments were run. There is a large size discrepancy between data packets (512 bytes) and hello messages (20 bytes). To examine whether packet size has an effect on the reception rate, another set of tests were run using 512 byte hello messages. Table 3 shows the significant improvement in delivery of the data packets in the field when the size of hello messages is increased. The increase in hello message size decreases the probability of reception and the effective range of hello messages. Therefore the difference in reception range between data and hello messages is decreased.

This improvement still does not account for all packet losses, so further examination is required. It was determined 802.11b transmits broadcast packets at a lower data rate, as opposed to the configured rate (i.e., 11 Mbps). Broadcast packets are sent at a lower rate to guarantee backward compatibility with 802.11. Depending on hardware and software, broadcasts occur at 1 or 2 Mbps. This results in hello messages having a much higher reception rate and larger range than data packets (see Figure 4). Consequently, connectivity is assumed because hello messages are being received;

Figure 4. Communication Range.

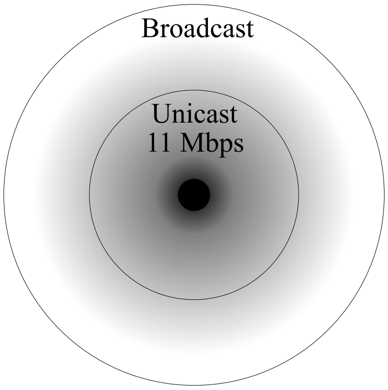

Table 5. Best Performance.

Test Scenario	Connectivity Strategy	Data Rate	Hello Size	Data Size	Percent Delivered
lab	1/2	1 Mbps	512 bytes	512 bytes	96.4
field	1/2	1 Mbps	512 bytes	512 bytes	87.4

however, data packets are not received because they are sent at a higher data rate and therefore have a shorter range of reception. To verify the effect of transmission rate on packet reception, tests were run with a data rate set to 1 Mbps and the auto rate setting. The auto rate setting performs automatic rate adjustment; during the tests the data rate should decrease (from 11 Mbps to 1 Mbps) as nodes 1 and 3 separate. The results from this test are shown in Table 4. The percentage of packets received increases for both auto and a fixed 1 Mbps rate. This further confirms that as the differences between hello messages and data packets decrease, their relative range and reception rate converge.

This data also correlates with the large hello message test above. 512 byte hello messages transmitted at 1 Mbps have a larger range than 512 byte data packets sent at 11 Mbps (even ignoring the overhead of DCF on unicast data packets), because of the difference in rate.

Based on the results above, an experiment using the most effective connectivity strategy, hello message size and 802.11b rate was executed. These results, shown in Table 5, show an improvement of 44% over the initial results (see Table 2). It is believed that the additional 9% differential between the in lab and in field tests is due to random packet loss, reception of spurious hello messages and the difference in the handling of broadcast and unicast packets by 802.11b.

CONCLUSION

To increase the effectiveness of hello messages, their reception characteristics should be equal to that of data packets. To make the reception of hello messages equal to data packets the two must have similar characteristics of size, rate and handling by the hardware/software. The reception of hello messages will then correctly indicate that reception of data packets will occur, and better throughput will result.

Other methods of increasing the utility of hello messages may be used in conjunction with those discussed in this paper to further improve performance. For example, turning off RTS/CTS transmissions or dropping control packets based on their received signal to noise ratio [8].

ACKNOWLEDGMENTS

The laptops used in this study were donated by Intel Corporation as part of a UC Core grant.

REFERENCES

[1] Bhargav Bellur and Richard G. Ogier. A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks. *Proceedings of IEEE INFOCOM*, pages 178–186, New York, NY, March 1999.

[2] Ian D. Chakeres. AODV-UCSB Implementation from University of California Santa Barbara. <http://moment.cs.ucsb.edu/AODV/aodv.html>.

[3] Thomas Clausen, Philippe Jacquet, Anis Laouiti, Pascale Muhlethaler, Amir Qayyum, and Laurent Viennot. Optimized Link State Routing Protocol. *Proceedings of the IEEE INMIC*, Pakistan, 2001.

[4] Mario Gerla, Xiaoyan Hong, and Guangyu Pei. Landmark Routing for Large Ad Hoc Wireless Networks. *Proceedings of IEEE GLOBECOM 2000*, pages 1702–6, San Francisco, CA, November 2000.

[5] IEEE Computer Society. IEEE 802.11 and 802.11b Standards, 1999.

[6] Luke Klein-Berndt. Kernel AODV from NIST. <http://w3.antd.nist.gov/wctg/aodv_kernel/>.

[7] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, and Robert Morris. A Scalable Location Service for Geographic Ad Hoc Routing. *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom)*, pages 120–130, Boston, MA, August 2000.

[8] Henrik Lundgren, Erik Nordström, and Christian Tschudin. Coping with Communication Gray Zones in IEEE 802.11b based Ad hoc Networks. Technical Report 2002-022, Uppsala University Department of Information Technology, June 2002.

[9] Erik Nordstrom and Henrik Lundgren. AODV-UU Implementation from Uppsala University. <http://www.docs.uu.se/∼henrikl/aodv/>.

[10] Richard G. Ogier, Fred L. Templin, Bhargav Bellur, and Mark G. Lewis. Topology Broadcast Based on Reverse-Path Forwarding. *IETF Internet Draft, draft-ietf-manet-tbrpf-05.txt*, March 2002. (Work in Progress).

[11] Charles E. Perkins, Elizabeth M. Belding-Royer, and Samir Das. Ad hoc On-Demand Distance Vector (AODV) Routing Protocol. *IETF Internet Draft, draft-ietf-manet-aodv-10.txt*, January 2002. (Work in Progress).

[12] Charles E. Perkins and Elizabeth M. Royer. The Ad hoc On-Demand Distance Vector Protocol. Charles E. Perkins, editor, *Ad hoc Networking*, pages 173–219. Addison-Wesley, 2000.