ENSC 427: Communication Networks
Spring 2011

Final Project: Final Report
Examining Wireless Mobile Network Routing Protocols Through Simulation
www.sfu.ca/~dja6

Damir Jungic
301069480
dja6@sfu.ca
Group 9

Abstract

I will examine mobile ad hoc network routing, specifically the protocols used in wireless mobile networks. Due to the lossy characteristics of wireless links, mobile ad hoc networks need to dynamically determine optimal paths for packets. An experimental protocol, the Better Approach to Mobile Ad hoc Networking (B.A.T.M.A.N.), is implemented using NS2. This protocol is then simulated in order to validate its unique approach to mobile ad hoc networking. Further, the B.A.T.M.A.N. protocol is compared to protocols that use a more traditional methodology, such as Optimized Link State Routing (O.L.S.R.).

Table of Contents

Introduction

Wireless mobile networks are collections of routers and hosts that form networks over very lossy links. As a result of link failure, routes throughout the network are frequently created and removed. Wireless mobile routing protocols must therefore be able to determine paths to recipients in a highly dynamic manner.

Applications

Wireless mobile networks have many applications due to the lack of required infrastructure. Soldiers deployed deep in hostile territory can make ad hoc networks to stay connected. Disaster relief efforts are greatly assisted by the rapid deployment of wireless mobile networks for communication, particularly when the existing communications network has been damaged or overburdened as a result of the disaster.

Vehicular networks are a greatly researched application of mobile ad hoc routing protocols [1]. As a vehicle travels down a road, it becomes in contact with new road signs and other vehicles, while also losing contact with others. Each vehicle therefore perceives a greatly varying network topology – precisely the situation addressed by mobile ad hoc routing protocols.

Classifications of Mobile Ad hoc Routing Protocols

Wireless mobile routing protocols can broadly be placed in two categories: table-driven and on-demand [2]. Table-driven routing protocols maintain paths to all destinations in the network. The method used to spread the state of links throughout the network varies from one protocol to the next. On-demand routing protocols do not perpetually maintain state information about the network. Instead, a route discovery process is initiated whenever a route is needed. In this way, on-demand protocols do away with the control packet flooding needed in order to continuously maintain routes for table-driven protocols. However, the route initiation procedure will cause a delay before the data can start being transmitted.

This project will examine a particular table-driven wireless mobile routing protocol: the Better Approach to Mobile Ad Hoc Networking (B.A.T.M.A.N.)[3]. The B.A.T.M.A.N. protocol uses the presence (and absence) of control packets to indicate link quality. This strategy is unlike the traditional approach of using control packets that contain information about link state, as used in the Optimized Link State Routing Protocol (O.L.S.R) [4]. Creators of the B.A.T.M.A.N. protocol claim that frequent loss and garbling of control packets make them an ineffective way to disseminate specific link information through the network. Instead, the occasional erroneous or dropped control packet is used as an indication of signal quality in the B.A.T.M.A.N. protocol.

Motivation

The B.A.T.M.A.N. protocol was intended to be a better approach to mobile routing compared with protocols like O.L.S.R. Though both are table-driven, the amount of information on which routing decisions are bested varies greatly between the two protocols. O.S.L.R has each node contain full path information to every other node in the network. All nodes should therefore have near-exhaustive information about the network state, something that is not always possible in wireless mobile networks. Alternatively, B.A.T.M.A.N. nodes determine the quality of a next hop neighbor N to a particular destination D based on the number of D's control packets that have been received through N. In this way, information about paths is implicitly boiled down to neighbor quality – the only routing decision the node can make. Despite both being table-drive, the two protocols offer very different approaches towards wireless mobile networking: O.L.S.R. relies on each node having full information about the network, while B.A.T.M.A.N. relies on collective intelligence being spread amongst all network nodes.

It is therefore worthwhile to validate B.A.T.M.A.N.'s experimental approach, and compare it to the traditional methodology of O.L.S.R.

B.A.T.M.A.N. Protocol

The B.A.T.M.A.N. protocol relies on each network node periodically sending originator messages (OGM). These OGMs make nodes aware of one another. By receiving OGMs, each node attempts to determine which link-local neighbor serves as the best gateway to the node that sent the OGM. This is all the information on which routing decisions are made: for each known node in the network, a record is maintained of which link-local neighbor should serve as a gateway to this node.

Two familiar constructs are used to determine the best gateway: a sequence number and a sliding window. Sequence numbers are the main piece of information communicated in OGMs. A node transmits its OGM periodically, with each message having a sequence number one greater than the previous message. Using sequence numbers, nodes attempt to determine how many OGMs it has received from each node, and which link-local neighbor is sending the most OGMs. The neighbor that relays the (statistically determined) most OGMs for a given node will be the gateway for all traffic destined for the node where the OGM originated.

Each node is aware of its link-local neighbors - the nodes to which it is connected through a direct wireless link. For each of the known nodes in the network, each node maintains a set of its neighbors, each with a sliding window. The sliding window is used to count how many of the past WINDOW_SIZE sequence numbers from a particular originator the node has received through each of its neighbors. Every time a node receives an OGM with a new sequence number, the sliding window of the neighbor that relayed OGM moves forward so that the highest value is the new one. Figure 1 illustrates the sliding window with a WINDOW_SIZE of four.

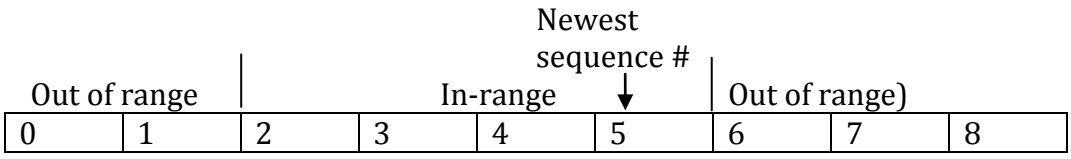

Figure 1: Sliding Window with a WINDOW_SIZE of 4

Nodes select which neighbor is the best gateway to a particular destination by seeing which neighbor has relayed the most in-range sequence numbers for that destination.

When a node A receives and OGM originated by node B through some neighbor C, node A needs to determine whether or not it should be rebroadcasted. Though there are several conditions to be checked, the most significant is whether or not neighbor node C is the best link to destination B. If the neighbor B, through which the OGM was received, is indeed the best next hop to C, then the OGM is rebroadcasted. In this way, the neighbors of A that receive B's OGM through A have implicit information path quality through C – the next hop towards B after A. Without this important check, the neighbors of A that receive B's OGM would have an erroneous view of the network. What if A has many neighbors that provide mediocre paths to destination C? If node A were to rebroadcast C's OGMs regardless of which neighbor they came from, then A's neighbors might think A has a very strong path to C! This is however not the case: A will provide a path to C through only one of its mediocre neighbors, and therefore provide a mediocre connection.

It can be difficult to describe and understand a protocol without a concrete example. The next subsection serves to clarify all the devices used by the B.A.T.M.A.N. protocol by putting them in context.

Example – Basic Behavior

A simple example can illustrate the basic behavior of route selection in the B.A.T.M.A.N. protocol. Only a small subset of the protocol's activity will be considered, in order to illustrate basic functionality. Assume all sliding windows are initially empty. Figure 2 presents a simple network topology, where N1..4 are nodes and L1..4 are links.

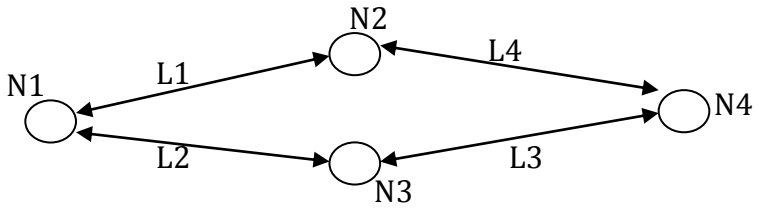

Figure 2: Simple Network Topology

Suppose N4 is broadcasting its originator message (OGM) with sequence number 1. Both nodes N2 and N3 will receive this OGM, and update their sliding windows for

N4. Both nodes N1 and N2 will then forward this OGM to N1. Note that N1 has a different sliding window for each of its link local neighbors N2 and N3, for destination node N4. The node N1 therefore updates both of these sliding windows, updating the sliding windows for N4. Since N1 has two neighbors, N2 and N3, that both have 1 sequence number in the sliding window for destination N4, packets destined for N1 will go through either node (this is undefined behavior).

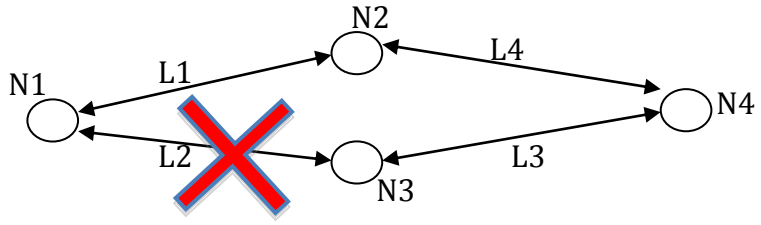

Figure 3: Link Failure in the Network

Let us suppose the link L2 goes down, as commonly occurs in wireless mobile networks. When N4 sends its next OGM packet with sequence number 2, at first hop it is received by nodes N2 and N3. The packet with sequence number 2 will then be forwarded to N1 only by N2, since the link between N3 and N2 is down. Since this OGM from N4 was relayed through N2, the node N1 will update its sliding window for packets originally from N4 that came through N2 to include the sequence number 2. The node N1 now has a link-local neighbor (N2) that has more in-range sequence numbers for OGMs originating from N4 than any other link-local neighbor (more than N3, which in this case only has 1 in-range entry for the sequence number 1). The node N1 therefore correctly concludes that all packets destined for node N4 should be sent to N2.

Example – A More realistic situation

The previous example demonstrated the role played by originator messages (OGMs) and sliding windows in their most basic form. Now a situation that shows the more realistic behavior of the protocol will be presented.

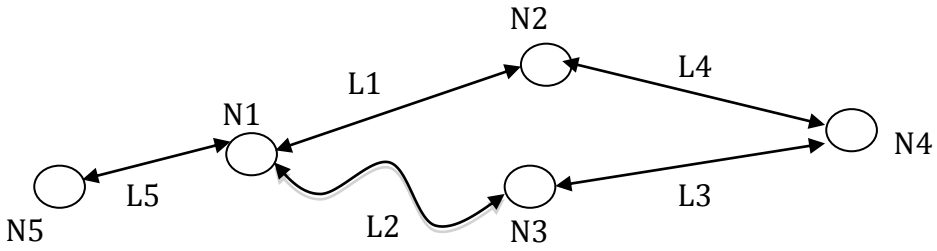

Figure 4: A More Realistic Network Topology

The network depicted in Figure 4 resembles the network in the previous example, with the exceptions that link L2 is now of poor quality and a new node N5 has been added through link L5.

Node N4, along with all other nodes, is again periodically transmitting its originator messages (OGMs). Some of N4's OGMs will arrive at N1 through N2, and others will arrive through N3. Which of these next hop neighbors should serve as N1's gateway to N4? Table 1 demonstrates an example OGM information map that the node N1 could have developed.

Table 1: OGM information stored at node N1

Originator (destination)	Neighbor	In Window Range Packet Count
N2	N2	15
	N3	6
	N5	0
N3	N2	12
	N3	4
	N5	0
N4	N2	11
	N3	3
	N5	0
N5	N2	0
	N3	0
	N5	15

Looking at the entry for originator N4, we see that neighbor N2 passed along 11 recent OGMs from N4, while neighbor N3 has only passed along 3 of N4's OGMs (due to the lossy connection between N1 and N3). The node N1 therefore correctly uses neighbor N2 as the best next hop for all data destined for node N4, and avoids the lossy link.

It is also illustrative to consider which of N4's OGMs N1 will pass along to N5. Node N1 will only pass along OGMs that arrived through the best link for the originator of the OGM: in this case, N4's OGMs will only be passed along when they arrive through N2. If this weren't the case – if N1 rebroadcasted regardless of which neighbor passed along the OGM – node N5 would base the quality of N1 as a next hop option on the quality of *all of N1's neighbors*. However, N1 will only ever use a single neighbor as a best hop, and therefore should only rebroadcast OGMs when they arrive through the best next hop to the originator of the OGM.

Implementation

The B.A.T.M.A.N. protocol was implemented based on a RFC Draft submitted in April of 2008 [3]. This document provides many of the structures and much of the logic necessary for implementing the protocol. A brief overview of implementation details will be provided here.

Structure of a B.A.T.M.A.N. Node

Figure 5 shows the basic structure of the batman node.

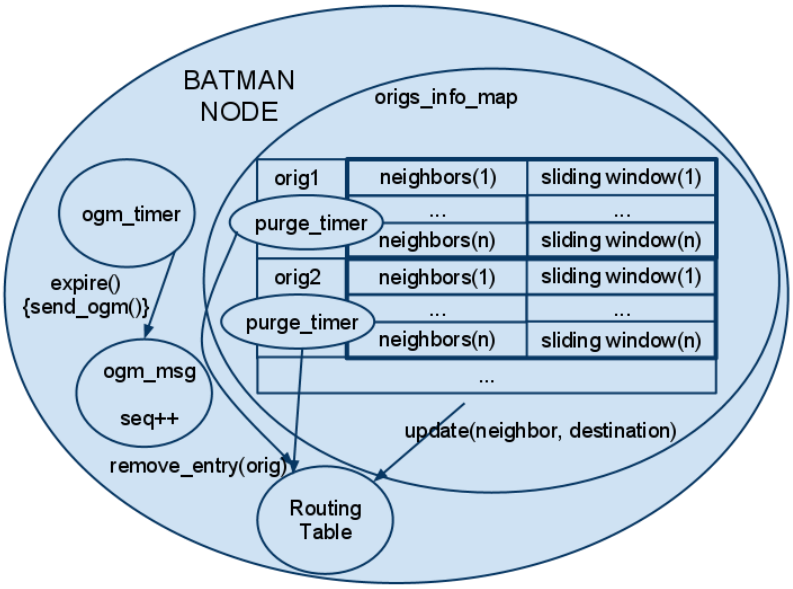

Figure 5: Basic Structure of a B.A.T.M.A.N. Node

Specifically, Figure 5 depicts:

- A node info table: for each known node (originator) in the network, the node info table includes:
  - A set of all neighbors.
    - Each of these neighbors then has a sliding window with a in-window range packet count.
  - A purge timer
    - If the node doesn't receive an OGM from a particular originator for a certain amount of time, it will assume there is no longer a path to that originator and will delete the corresponding entry in the routing table.
- An OGM timer
  - Broadcasts this node's OGM at a regular interval, plus some jitter.
- A Routing table
  - Maps all known nodes to their determined best next-hops, based on the information stored in the node info table, and is potentially updated whenever a new OGM arrives

How Packets are Received

The flowchart presented in Figure 6 shows the decisions that are made every time an OGM or data packet is received.

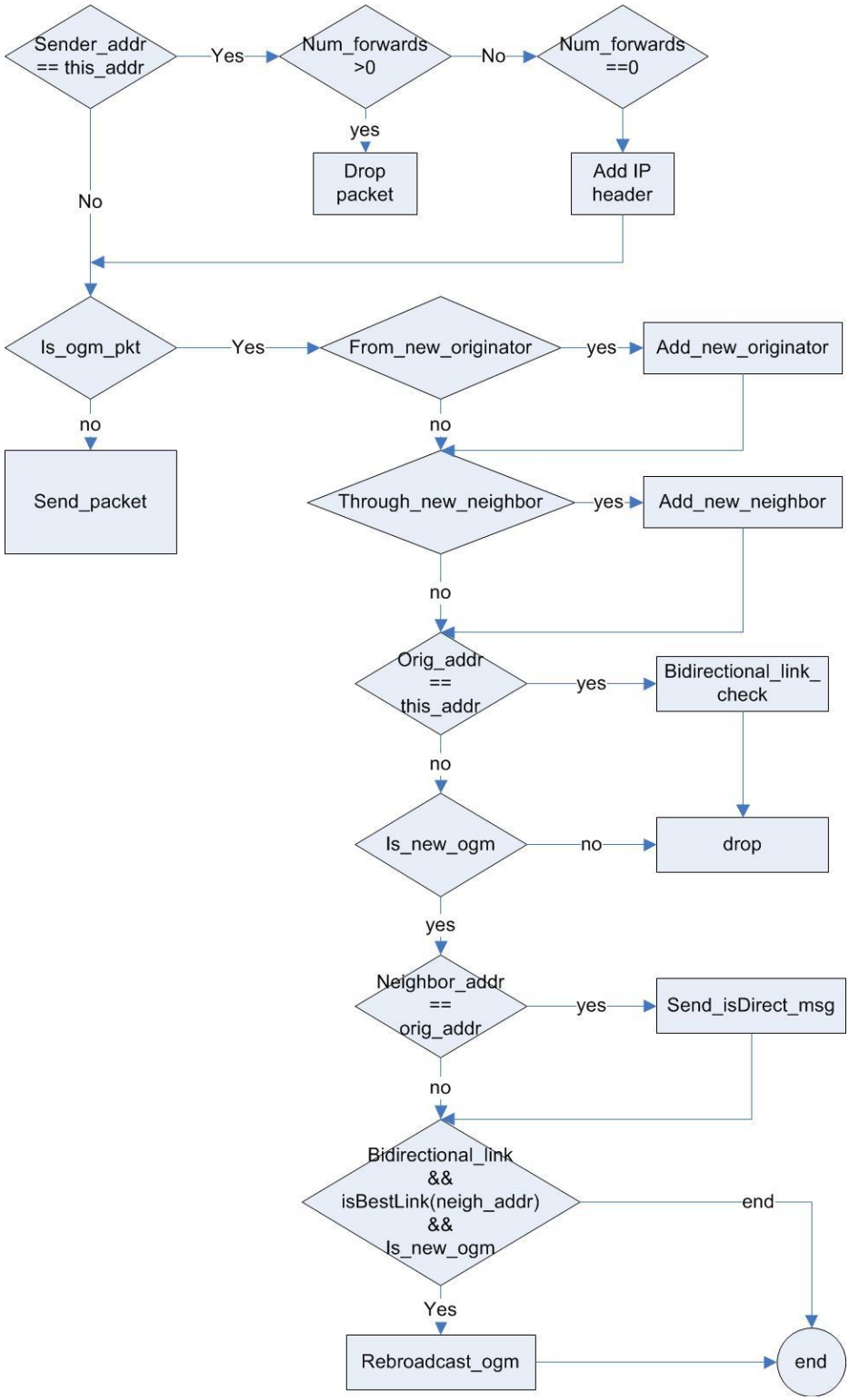

Figure 6: Flowchart Depicting Reception of a Packet

Particularly important considerations in receiving packets include:

- Verifying that if there is a loop, the packet must be dropped by checking the num_forwards field of the packet

- If the packet is an originator message (OGM):
    - If the OGM originated at this node, perform a bidirectional link check. This verifies that neighbors are connected via bidirectional links.
    - If it is a duplicate OGM (non-new), the packet should be dropped.
    - If it isn't a duplicate, we update the originators information for the neighbor through which the OGM was received.
    - If the neighbor was the originator, we send an is-direct link message.
    - If the OGM was received through a bidirectional link, which is also the best link to the originator of the OGM and this was a new OGM, then we will rebroadcast the OGM.
- If the packet isn't an OGM, it's a data packet that we need to route.

Omissions

For the purposes of simplifying the NS2 implementation of the B.A.T.M.A.N. protocol, certain facets of the protocol will not be included. Omitted portions of the protocol will include:

- Version checking – it will be assumed that all nodes are running the same version of the protocol.
- HNA extensions – HNA messages allow nodes to communicate their connectivity to non-B.A.T.M.A.N. networks. Interfacing with various types of networks will not be addressed in the simulations presented here.
- Bidirectional links only – the protocol only functions with bidirectional links. A complete implementation would include verification of each link's bidirectional capability. However, such checking will be omitted from the NS2 model used for simulation.

The above omissions will not interfere with the examination of the B.A.T.M.A.N. protocol as a wireless mobile networking protocol.

Simulation

Several simulations scenarios were executed in order to validate the use of the B.A.T.M.A.N. protocol as a mobile ad hoc routing protocol.

Scenario 1

Scenario 1 will serve as a reference for Scenario 2. Figure 7 depicts the network topology for this scenario.

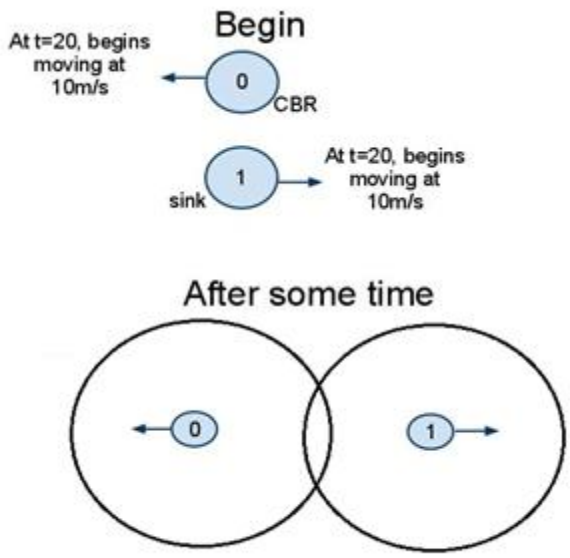

Figure 7: Network Topology for Scenario 1

Only two nodes are present, and 20 seconds into the simulation they start moving away from each other at 10m/s. Constant bit rate traffic begins to flow from node 0 to node 1 at 10s. The simulation uses the standard antenna power supplied by NS2, which has a range of 250m. Figure 8 shows the receiving bit rate at node 1.

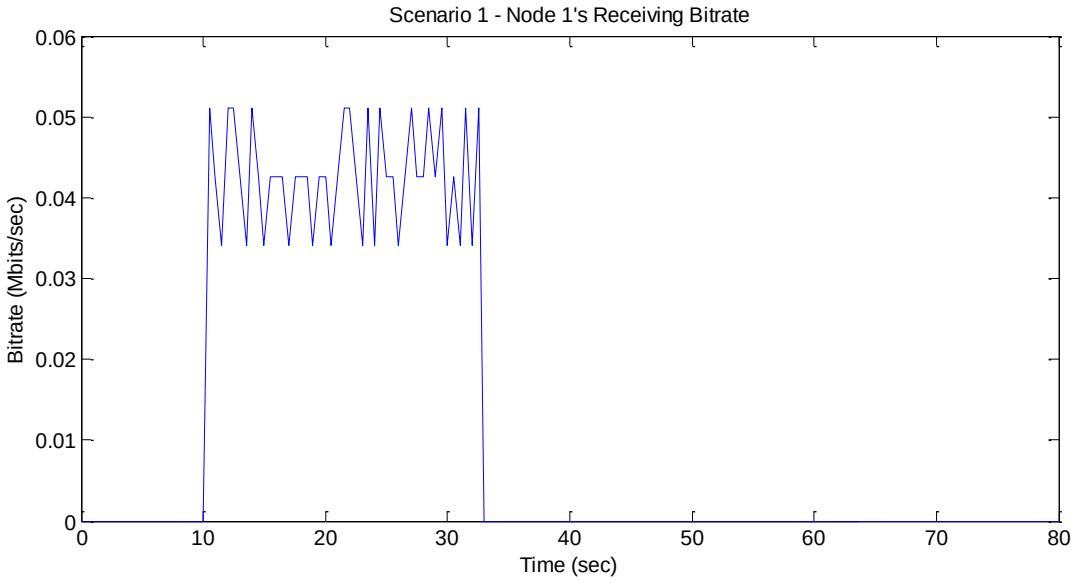

Figure 8: Node 1's Receiving Bit rate for Scenario 1

From the plot presented in Figure 8, it is clear that data is arriving at Node 1 starting at 10s, when the traffic starts, to around 32.5 seconds, when the nodes become out of range. Moving at 10m/s for around 12.5 seconds, the nodes are a distance

2*10*12.5=250m away from one another when they become out of range – exactly what we would expect.

Important Note: Figure 8, as well as other plots presented in this report, displays a saw-toothed graph where a constant bit rate is expected. This is because the bit rate points do not all consider the same amount of packets. Therefore, it is the average of this plot that indicates the constant bit rate.

Scenario 2

Scenario 2 involves 3 nodes, two of which move in the same way as in Scenario 1. The remaining node doesn't move, and will serve to route packets between the nodes that are moving away from one another. The topology is presented in Figure 9.

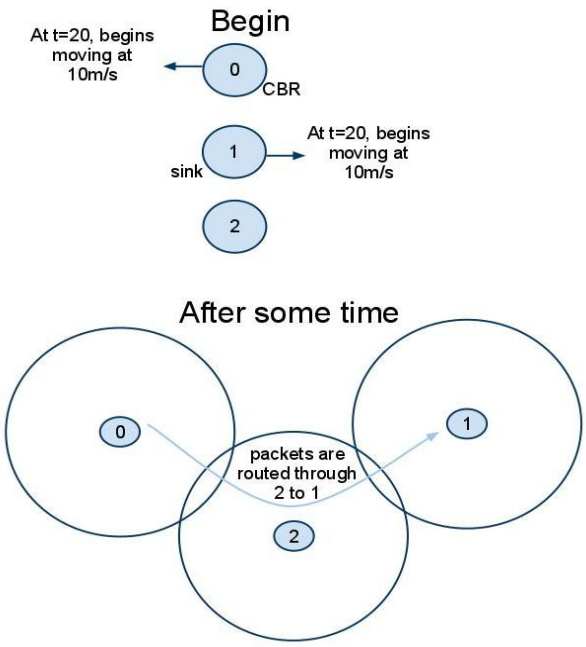

Figure 9: Network Topology for Scenario 2

Traffic will initially flow directly from node 0 to node 1. However, as the nodes move away from one other, a point will be reached where direct communication is no longer possible. We expect this point to be at the same time where communication was lost in Scenario 1. The B.A.T.M.A.N. protocol should then determine that node 0 should send all data destined for node 1 to neighbor node 2. Of course, there will also be a point where node 0 moves too far away from node 2, and communication between node 0 and node 1 will again no longer be possible. Node 1's receiving bit rate is presented in Figure 10.

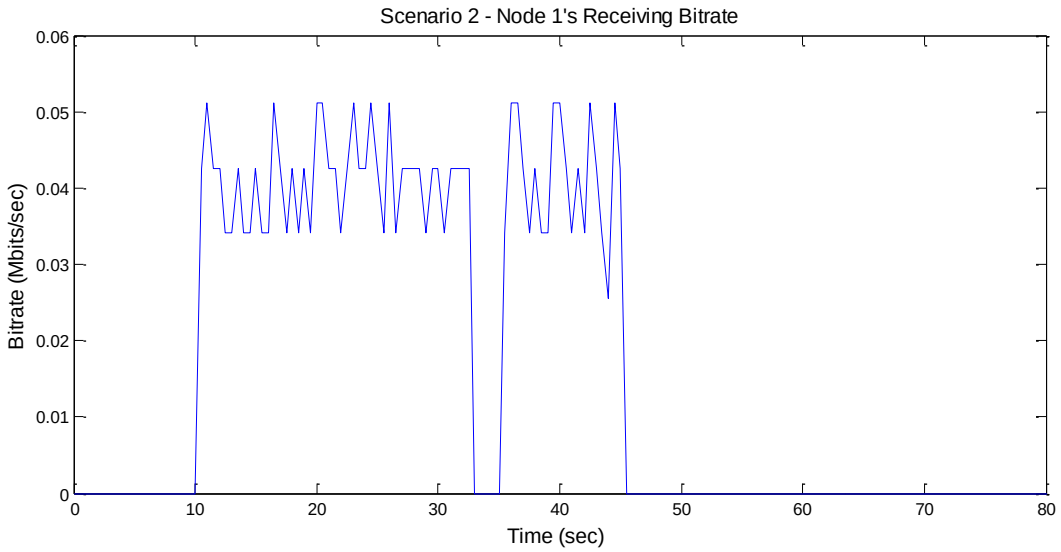

Figure 10: Node 1's Receiving Bit rate for Scenario 2

As expected, the transmission continues for an additional 12.5s – the amount of time nodes 0 and 1 remain within the range of node 2. However, the bit rate drops to zero exactly where nodes 1 and 2 lose direct communication. The loss of data during this period is a result of the time it takes the routing protocol to change node 0's best hop neighbor for destination 1 from 1 to 2. This can be demonstrated by observing the routing table and node info stored at node 0, presented here in Listing 1 (at time 33s into the simulation).

Listing 1: Node 1's Routing table and Node Info Data at 33s

```
P 33.000000 _0_ Routing Table
P    dest next
P    1    1
P    2    2
P 33.000000 _0_ node Data
P ogm_1_   | neigh_1_      | packetCount: 33
P          | neigh_2_      | packetCount: 31
P ogm_2_   | neigh_1_      | packetCount: 30
P          | neigh_2_      | packetCount: 33
```

The next hop entry for packets destined for node 1 is still node 1, even though node 1 is no longer in range. This is because more of node 1's OGMs have still been received through node 1 (33 packets compared to 31). However, now as node 1 continues to broadcast OGMs, they will *only* arrive to node 0 through node 2. In this way, it is only a matter of time until node 0's routing table is updated to reflect the correct best next hop towards destination node 1. Listing 2 shows node 0's data three seconds late at 36s.

Listing 2: Node 1's Routing table and Node Info Data at 36s

```
P 36.000000 _0_ Routing Table
P    dest next
P    1    2
P    2    2
P 36.000000 _0_ node Data
P ogm_1_   | neigh_1_       | packetCount: 33
P          | neigh_2_       | packetCount: 34
P ogm_2_   | neigh_1_       | packetCount: 30
P          | neigh_2_       | packetCount: 36
```

Listing 2 shows that node 0 has node 2 as the next best hop for packets destined for node 1. This is indeed true; nodes 1 and 0 are not within range of one another, and can therefore not communicate directly. We see that OGMs through node 2 have remained at 33, as expected. The updating of node 0's routing table is what allows the constant bit rate traffic to resume arriving at node 1, as depicted by the bite rate plot in Figure 10.

Scenario 3

Scenario 3 involves one network topology but two simulations: one simulation using the Dynamic Source Routing (D.S.R.) protocol to perform the routing, and one using the B.A.T.M.A.N. protocol. The network topology is presented in Figure 9, and is taken from the Running Wireless Simulations in NS section of Marc Greis' tutorial. The simulation file presented in the tutorial uses D.S.R. as a routing protocol, and will in this way be used as the basis of comparison for when B.A.T.M.A.N. is used for routing instead. The specific types of data, constant bit rate and FTP, are not significant for this simulation. The topology and transfer only serve to accommodate a comparison between B.A.T.M.A.N. and D.S.R.

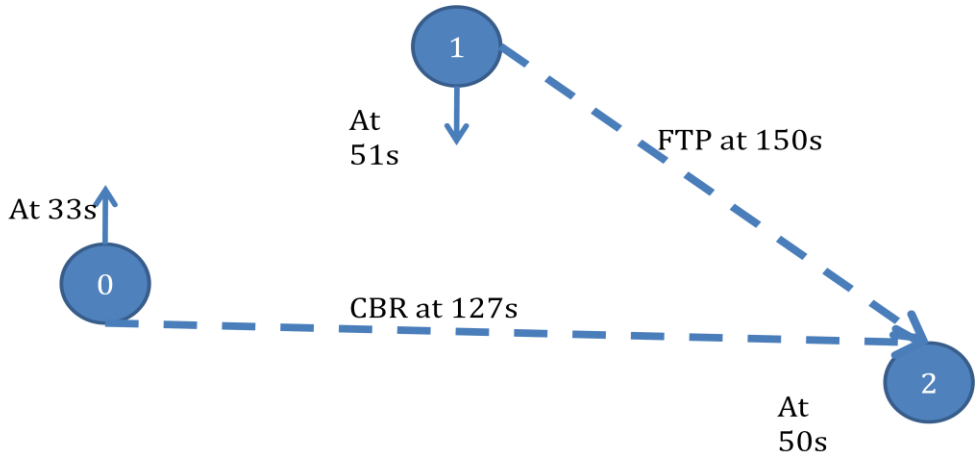

Figure 11: Network Topology for Scenario 2

The node movement is based on a random movement generation program shipped with NS2. Both nodes 0 and 1 attempt to send data to node 2, though only node 1 will be within range of node 2. Node 1 will therefore be responsible for routing node 0's data to node 2. Figure 12 shows the data being received at node 2, from both source nodes 0 and 1, while using DSR for routing.

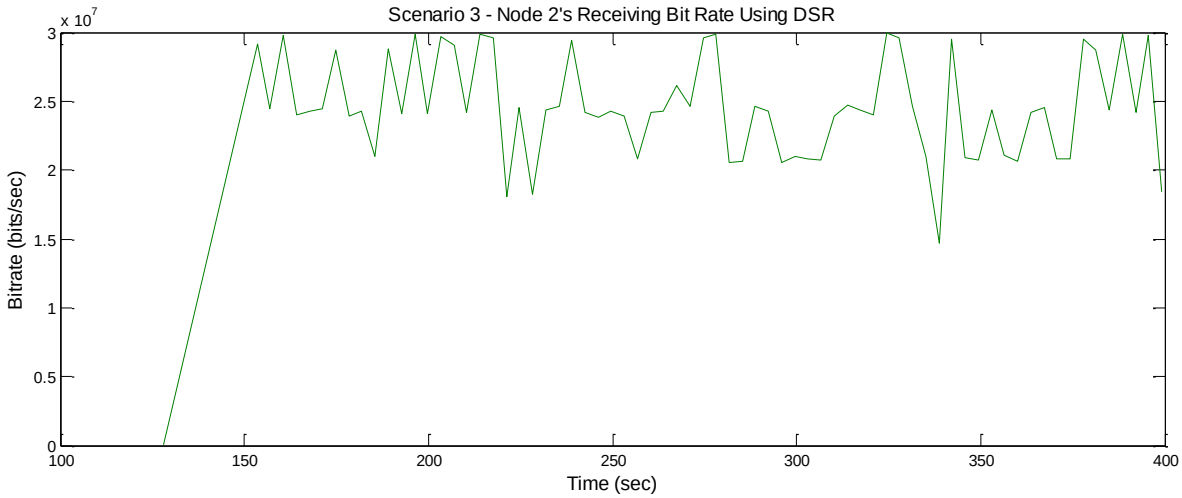

Figure 12: Data Received at Node 2 using DSR

The bit rate plot in Figure 12 shows data beginning to be received at 127s and then consistently received thereafter. Figure 13 shows the same network topology and node movement being simulated, except using B.A.T.M.A.N. for routing instead of DSR.

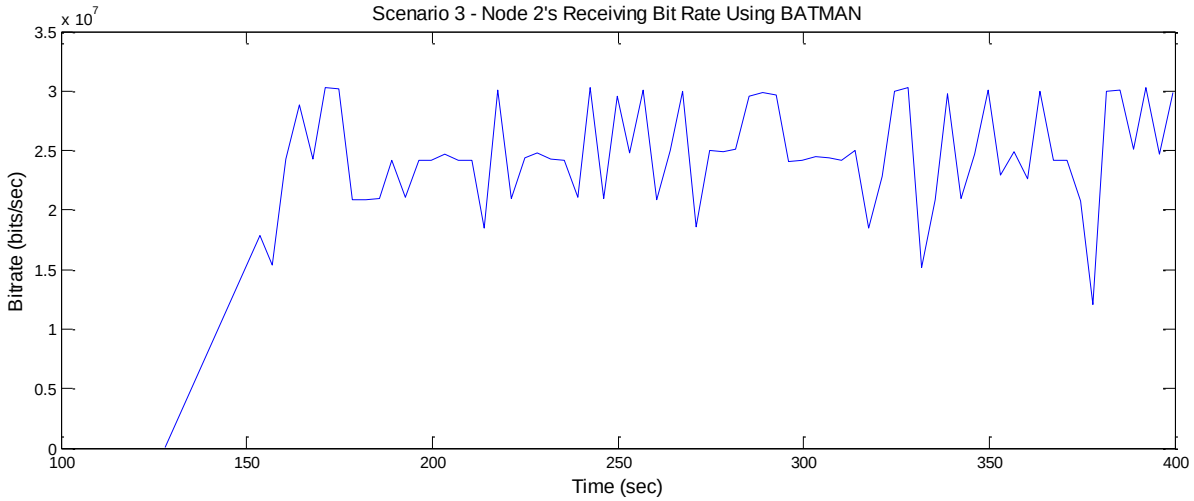

Figure 13: Data Received at Node 2 using BATMAN

The bit rate received at node 2 using B.A.T.M.A.N. looks very similar to previous simulation, which used D.S.R. This is what we expect – if B.A.T.M.A.N. is a viable mobile ad hoc routing protocol, it should behave similarly to establish routing

protocols such as DSR. There are many behavior details that can be compared between B.A.T.M.A.N. and other mobile ad hoc routing protocols can be made through simulation, though these are beyond the scope of this project. Further work is discussed in the last section of this document.

Discussion

The simulation scenarios presented in the previous section serve to validate the B.A.T.M.A.N. protocol as a mobile ad hoc networking protocol. It was observed that the protocol correctly alters routing tables in order to provide paths to destinations, despite changing network topology. An apparent shortcoming was observed in Scenario 2, when there was a brief loss of data until the routing table was updated to reflect the current topology. Some loss in inevitable in mobile ad hoc networks – any protocol can only respond so quickly to change topology. What determines how quickly a protocol responds is the frequency with which control packets (OGMs in the case of B.A.T.M.A.N.) are transmitted. Observing that the current frequency is one OGM per second, it is clear that a data stream with a packet inter-arrival time of less than a second will experience loss before the protocol has an opportunity to update. If this loss of data was a major concern, it is possible to increase the frequency with which OGMs are transmitted. However, increasing the amount of OGMs sent has the drawback that a high percentage of packets that need to be processed would then be OGMs instead of data. The relatively small size of OGMs, 12 bytes plus IP header, is an attractive feature of B.A.T.M.A.N. when used in an environment where it is necessary to transmit control packets with a high frequency.

The B.A.T.M.A.N. protocol has the further advantage of quick processing over many other protocols. With each OGM received, the only route that might change is the next hop neighbor to the destination where the OGM originated. That is, at most one routing table entry changes with each received OGM. This varies greatly to protocols that exchange link state information, such as O.L.S.R., that might have to recalculate entire routing tables if enough link state information changes. The reduced amount of processing required by B.A.T.M.A.N. is a significant advantage, though this is feature not easily observed through simulation with a tool such as NS2.

The comparison between B.A.T.M.A.N. and D.S.R. served to indicate that B.A.T.M.A.N. behaves similarly to existing routing protocols. Comparing the two protocols in detail, however, would likely require significantly more simulation. Dynamic Source Routing, as the name suggests, isn't a table-driven protocol like B.A.T.M.A.N. Instead, Dynamic Source Routing finds paths only when they are required. In this way, D.S.R. doesn't need to disseminate control information. However, if an active path becomes invalidated do to link failure, the protocol then needs to go through the involved process of find a new path. A general comparison between the different approaches between table-driven and on-demand protocols is beyond the scope of this project. However, armed with the implementation of the

B.A.T.M.A.N. protocol provided by this project, many comparisons can now be conducted through simulation. Future work is discussed in the next section.
The main focus of this project was to develop a working implementation of the B.A.T.M.A.N. protocol in NS2. As a result, thorough examination of the protocol's behavior through simulation was not possible. However, there are valuable heuristic comparisons to be made between the innovative approach taken by the B.A.T.M.A.N. protocol and the more traditional approaches taken by algorithms such as D.S.R. and O.L.S.R. The next section discusses the vast amount of future work possible now that the B.A.T.M.A.N. protocol has been implemented in NS2.

Conclusion and Future Work

Implementing the Better Approach to Mobile Ad hoc Networking routing protocol and running B.A.T.M.A.N. basic simulations in NS2 served to validate the innovating approach taken by B.A.T.M.A.N. The comparison between B.A.T.M.A.N. and established routing protocols concluded that their behavior was similar – further justification of using B.A.T.M.A.N as a mobile ad hoc networking protocol.

However, with a working implementation of the B.A.T.M.A.N. protocol in NS2, there is very much to be explored. Much more in depth comparisons can be made between the B.A.T.M.A.N. protocol and other routing protocols. The responsiveness to failure, amount of control packet overhead, percentage of time spent processing control packets, and many more features of routing protocols all ought to be thoroughly examined through simulation. In this way, the specific strengths and weaknesses of B.A.T.M.A.N. as a mobile ad hoc routing protocol can be determined.

The implementation of B.A.T.M.A.N. produced by this project can still be greatly improved upon. As mentioned previously, certain facets of the protocol were not implemented due to time constraints. Further work would involve implementing bidirectional link checking, direct-link neighbor checking, and interfacing with other networks (HNA extensions).

This project provides an implementation of the B.A.T.M.A.N. protocol, and validates its use. Further, this project can serve as the basis for significant future work.

References

[1] Joo-Han Song; V.W.S. Wong; V.C.M. Leung; , "Wireless Location Privacy Protection in Vehicular Ad-Hoc Networks," *Communications*, 2009. ICC '09. IEEE International Conference on , vol., no., pp.1-6, 14-18 June 2009 doi: 10.1109/ICC.2009.5199575

[2] E.M. Royer; Chai-Keong Toh; , "A review of current routing protocols for ad hoc mobile wireless networks," *Personal Communications*, IEEE , vol.6, no.2, pp.46-55, Apr 1999 doi: 10.1109/98.760423

[3] C. Aichele; M. Lindner; A. Nuemann; S. Wunderlich; , "Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.)," IETF, Internet-Draft, 2008. URL: http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00

[4] T. Clausen; P. Jacquet; , "Optimized Link State Routing Protocol (OLSR)," IETF, Experimental, 2003. URL: http://tools.ietf.org/html/rfc3626

[5] R.E. Kahn; S.A. Gronemeyer; J. Burchfiel; R.C. Kunzelman; , "Advances in packet ratio technology," *Proceedings of the IEEE* , vol.66, no.11, pp. 1468-1496, Nov. 1978 doi: 10.1109/PROC.1978.11151

[6] C.V. Lopes; P. Baldi; , "A survey, classification and comparative analysis of medium access control protocols for ad hoc networks," *Communications Surveys & Tutorials*, IEEE , vol.6, no.1, pp.2-16, First Quarter 2004 doi: 10.1109/COMST.2004.5342231