OPNET DSR Verification and Validation

I. Dynamic Source Routing Protocol Verification and Validation Implementation

1.1 Overview

This appendix presents the design and implementation of the Dynamic Source Routing (DSR) protocol. The first area discussed is the modifications and additions made to the OPNET DSR model implementation from the National Institute for Standards and Technology (NIST) [PRP00]. Second, a list of system parameter settings will be described and explained. A list of the workload parameter settings will also be described and explained. Finally, some of the problem areas encountered while implementing this verification and validation model will be discussed.

1.2 Validation and Verification of the OPNET DSR Model

The first step in the process of this research was to develop a model that accurately represented the DSR protocol. The OPNET network simulation tool was chosen for this research. Since NIST had developed a DSR model in OPNET [PRP00], it was chosen as a starting point for this research. The NIST model [JMHJ99], the third version of the specification for DSR. A number of critical areas in the model were either implemented incorrectly or left out altogether. The NIST model was updated to DSR specification version 5 [JMHJ01] so that the model would reflect a current DSR protocol for MANETs. The following list of areas were either modified or added to the NIST model. The only capabilities that were not implemented were the piggybacking of multiple packets into one packet and the Implicit Flow State for DSR (which now has its own specification separate from the DSR specification).

1. Packet Sizes and Formats. The packet formats and field sizes were not in compliance with the specification for DSR. For instance, all of the address fields were 8 bits instead of 32. While this was all that was needed to hold the address for simulations, it is not the true size of the fields and will impact the load to the network.

2. Route Cache. The NIST model only implemented a single route to every destination. The route cache, as defined in the specification, should allow for more than one route to a destination. While modifying the NIST model for this validation and verification it was discovered that the route cache and the route caching strategy have a large effect on the performance of the network. If only one route is maintained and that route "breaks" then a new route discovery sequence will have to be initiated. If routes are cached, the node can simply look into its route cache for the next available route. Not caching routes proved to increase the number of routing packets by as much as two and three times, which also obviously increases the load of the network, the end-to-end delay, throughput and so on. Therefore, the route cache was modified to handle up to 100 routes per destination with a caching strategy that prioritizes the route based on when the route was added to the cache, the size of the route, as well as how the route was discovered.

3. Packet Salvaging. Packet Salvaging, as defined in [JMHJ01], allows for a intermediate node to look for an alternate route in its cache to a particular destination if an error was received for the source route defined in the data packet. This was not implemented in the NIST model. Packet salvaging is used extensively to not only increase the packet delivery ratio, but to also clean out invalid routes from the route cache of all neighboring nodes.

4. Error Packet Handling. Upon receiving an error for a particular data packet an intermediate node would transmit an error packet back to the source along the reverse path of the source route. However, only nodes in the reverse path would clean out their cache from the invalid link, leaving neighboring nodes with this erroneous route information to possibly use in the next route discovery. This was modified to meet the specification such that all nodes overhearing the error packet would clean out their route cache as well.

5. Promiscuous Listening. The route cache was only updated from route reply packet, thus when a link went bad a new route discovery would have to take place. Along with the addition of multiple routes in a cache, the model was updated such that all nodes could promiscuously listen and gather route information from data packets,

request packets and reply packets. This greatly improves the possibility of having a valid route available in the cache.

6. Retransmissions. In the NIST model, if an error occurred when sending a data packet the packet was automatically dropped. The specification calls for two retransmissions before packet salvaging, so this was implemented. It has been argued that this is not needed when DSR is implemented over 802.11, but experimentation showed that when the network was congested, these retransmissions were extremely beneficial.

7. Send Buffer. The send buffer is used to hold data packets waiting to be sent to their destination. The send buffer was not being checked when a route was added to the cache to see if any packets were waiting on that route information, which could unnecessarily increase the end-to-end delay of the data packets. The send buffer was also not regularly checking the packets to verify they had not expended their maximum lifetime limit. These problems were corrected for this research.

8. Random Waypoint Mobility. NIST implemented the billiard mobility model for this DSR implementation, which is described in NIST's documentation for the DSR model [PRP00]. While this is not incorrect, the billiard mobility model was not found anywhere else in the literature reviewed. The random waypoint mobility model was the model of choice for all published DSR research data. NIST had developed the random waypoint model for OPNET in its implementation of the AODV MANET routing protocol [Gue01, PRD01], so that mobility model was modified and incorporated into this DSR model.

9. Data Packet Transmission Delay Window. The DSR specification states that a source node should not send an "unbounded" number of packets along a route without the source node allowing for a route error. However, nowhere in the literature review was a transmission delay window between sending packets down the same source route specified. Thus, through experimentation as well as trial and error, an effective delay time was determined. If multiple packets are waiting in the send buffer for a particular destination, the node should wait 30 milliseconds per hop after receiving a successful acknowledgement from the next hop in the source route before sending the next data packet down that same route. This allowed enough time for the data

packet to be transmitted to the destination and for an error packet to make it back to the source node should an error occur.

10. Data Rate. The NIST DSR model implemented a 1 Mbps data rate. This normally would not have been a problem since it should be a matter of simply changing the data rate parameter to 2 Mbps to match the data rate of all other published data. However, there were implementation errors in the model when using any data rate other than 1 Mbps. These implementation errors caused the model to transmit at 5 Mbps for some packets and 1 Mbps for others even though the set data rate was 2 Mbps.

11. Jitter Delay. Jitter Delay causes a random delay between zero and 10 milliseconds for request and reply packets. This did not turn out to be of any great importance since 802.11 already implements its own random transmission delay, but it was added anyway to meet the specification call for a maximum jitter delay of 10 milliseconds.

12. RTS/CTS handshaking at the MAC layer. The RTS/CTS handshaking used by the IEEE 802.11 OPNET implementation was problematic (e.g., pointers to non-existent packets). This problem was also seen in other research using OPNET [Gue01] and had to be fixed to accurately simulate the DSR network. Without RTS/CTS an increased amount of collisions will occur causing possible transmission failures.

1.2.1 Verification and Validation Implementation. Once the DSR model had been updated to the specification standards, the verification and validation of the model was made by comparing the results to other published data [BMJ$^+$98, MBJJ99, DPR01]. In particular, the results from [BMJ$^+$98] were published in [Per01], so those were used by this research for verification and validation. The next two sections describe the parameter settings used for the model to accomplish the verification and validation. These settings were either stated explicitly in [BMJ$^+$98] or were inferred based on the research of the published data and expert opinion from pilot test simulations.

1.2.1.1 System Parameters.

The system parameters were set as shown below.

1. Data Rate - A data rate of 2 Mbps was used.
2. Simulation Area - An area of 1500 x 300 meters was used for the validation and verification of this model. This area represents a highway environment with the narrow width and long length.
3. Route Cache - The size of the cache refers to the number of routes a node's cache will maintain to any particular destination node. Pilot studies showed that a cache of 50 routes to every destination produced the best results under the implemented caching strategy.
4. Node Mobility - The random waypoint model as described in [BMJ+98] was implemented and used for this verification and validation of DSR.
5. Transmission Range - The nominal transmission range of the model was set to 250 meters. This is the range that was used for most of the published MANET research.

1.2.1.2 Workload Parameters.

1. Nodes - A total of 50 nodes were placed in the simulation area.
2. Source Nodes - 20 of the 50 nodes were used as data packet source generators for peer-to-peer connections.
3. Size of data packets - 64 byte packets were generated by the 20 source nodes.
4. Packet Interarrival - The data packets were generated at a constant rate of 4 packets/second.
5. Node Speed - The node speed is uniformly distributed between 0 and 20 meters/second.
6. Node Pause Time - The node pause time for the random waypoint mobility model is varied between 0, 30, 60, 120, 300, 600, and 900 seconds.
7. Hop Delay - The specification states that the Hop Delay should be twice the propagation delay and [BMJ+98] mentioned that the propagation delay is 600 microseconds. Thus, the hop delay was set at 1.2 milliseconds.

8. Transmission Delay Window - If there were multiple packets waiting in the send buffer for a particular destination, the source node would wait 30 milliseconds per hop after receiving a successful acknowledgement from the next hop in the source route before sending another data packet down the same route.

II. DSR Verification and Validation Implementation and Analysis

2.1 DSR Verification and Validation

In order to verify and validate that the DSR model being used was performing appropriately, simulations were configured and conducted according to previous research in this area [BMJ$^+$98, DPR01, MBJJ99]. The results were compared to the data provided in that research. In particular, the data provided by [BMJ$^+$98] is used as a comparison.

2.1.1 Verification and Validation Implementation. The basic implementation of the DSR model used for verification and validation included the parameter settings defined in Table 2.1. The performance metrics included the data packet delivery ratio and number of routing packets.

Table 2.1. Validation and Verification Workload Parameter Settings

Workload Parameter	**Setting**
Nodes in Simulation	50
Source Nodes	20, 30
Data Packet Size	64 Bytes
Mean Interarrival Time	0.25 seconds
Hop Delay	1.2 milliseconds
Packet Send Delay	30 milliseconds
Max Node Speed	20 meters per second
Node Pause Time	0, 30, 60, 120, 300, and 900 seconds
Simulation Area	1500 x 300 meters
Transmission Range	250 meters
Mobility Model	Random Waypoint

2.1.2 Verification and Validation Results. As can be seen in Figure 2.1, all of the data points from previous research implementing DSR in the NS-2 network simulator [Per01, BMJ$^+$98] were well above the 97 percent level and the delivery ratios encountered by the OPNET DSR model used in this research all fall above the 97 percent delivery as well. It should be pointed out that the data points from previous research as shown in these graphs are approximate. However, using these data points and assuming a 95 percent confidence interval we find that the two sets of data points are statistically equivalent.

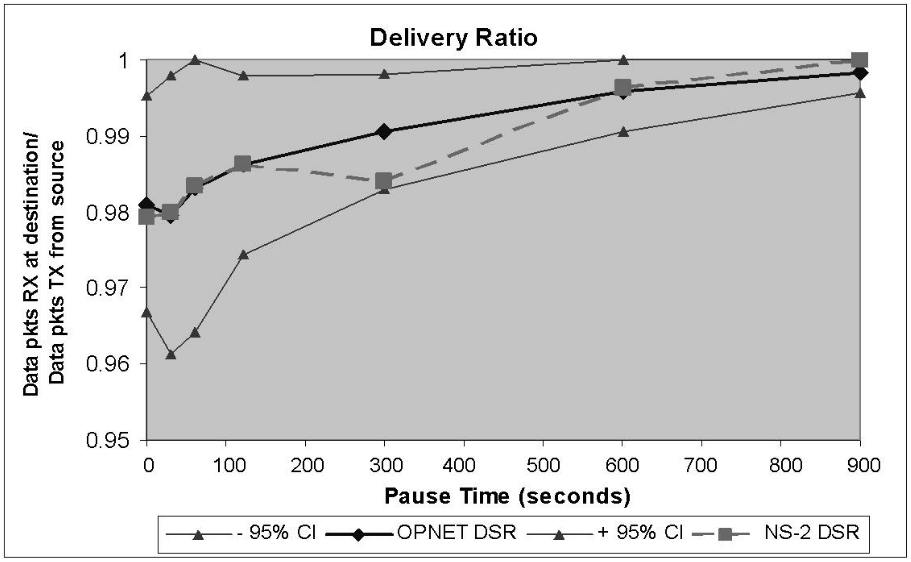

Figure 2.1. DSR Delivery Ratio Comparison for Validation and Verification

As shown in Figure 2.2, the number of routing packets seen by the OPNET DSR model is statistically equivalent with the data provided by previous research in NS-2. Based on these metrics the OPNET DSR implementation produces similar results to that of previous implementations. Therefore, the OPNET DSR implementation is a valid and verified DSR model.

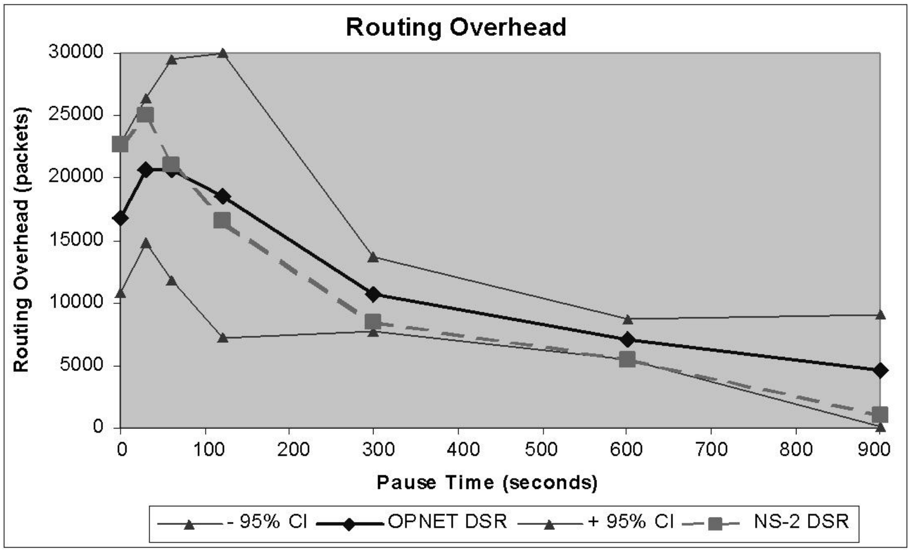

Figure 2.2. DSR Routing Packet Comparison for Validation and Verification

Bibliography

[BMJ+98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu, and Jorjeta Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Mobile Computing and Networking*, pages 85–97, 1998.

[DPR01] Samir Ranjan Das, Charles E. Perkins, and Elizabeth E. Royer. Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks. *IEEE Personal Communications*, 8:16–28, 2001.

[Gue01] Lyes Guemari. An OPNET model implementation for Ad-hoc On demand Distance Vector Routing Protocol. Master's thesis at the Information Technology Laboratory of the National Institute of Standards and Technology, August 2001.

[JMHJ99] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet draft, Internet Engineering Task Force MANET Working Group, October 1999. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt.

[JMHJ01] David B. Johnson, David A. Maltz, Yih-Chun Hu, and Jorjeta G. Jetcheva. The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks. Internet draft, Internet Engineering Task Force MANET Working Group, March 2001. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-05.txt.

[MBJJ99] D. Maltz, J. Broch, J. Jetcheva, and D. Johnson. The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks. *IEEE Journal on Selected Areas in Communication*, 17:17–25, August 1999.

[Per01] Charles E. Perkins. *Ad Hoc Networking*. Addison-Wesley, 1 edition, 2001.

[PRD01] Charles E. Perkins, Elizabeth M. Royer, and Samir R. Das. Ad hoc On Demand Distance Vector (AODV) Routing. Internet draft, Internet Engineering Task Force MANET Working Group, March 2001. http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-08.txt.

[PRP00] Xavier Pallot, Nicolas Roux, and Jean-Sebastien Pegon. README File for NIST DSR Model. File and Model found at "http://w3.antd.nist.gov/wctg/DSRreadme.pdf", December 2000.