PERFORMANCE ANALYSIS OF DSDV WITH OTHER MANET ROUTING PROTOCOL USING RANDOM WAYPOINT MOBILITY MODEL IN NS-3

Saurabh kumar soni*

Prof.Prem Narayan Arya*

ABSTRACT

Routing protocols are a critical aspect to performance in mobile wireless networks. The development of new protocols requires testing against well-known protocols in various simulation environments. In this paper, present an overview of several well-known MANET routing protocols and the implementation details of the DSDV routing protocol using Random Way Point Mobility Model in the ns-3 network simulator.

Categories and Subject Descriptors

I.6 [Simulation and Modeling]: General, and Mobility Model; C.2.2 [Computer- Communication Networks]: Network Protocols -- routing protocols.

Keywords: DSDV, ns-3, DSDV model implementation, MANET routing protocol, ns-3 simulation methodology.

* Samrat ashok technological institute, Vidisha(m.p).

1. INTRODUCTION

Wireless mobile ad hoc networks (MANETs) [1] that do not require infrastructure to operate have been the subject connectivity in the face of wireless channels and nodes moving out of range of significant research. In MANETs, nodes self-organise and act as both end systems and as intermediate systems. The two major challenges to routing in MANETs are the dynamic topologies that result from mobility, and maintaining from one another. Many routing protocols for MANETs have been proposed, the most prominent in theresearch community: DSDV [8], due to their early emergence and varied characteristics. The operation and performance of these four protocols provide an important baseline to which new protocols should be compared. In particular, DSDV provides the proactive distance vector case.

Simulation has been the backbone of MANET research [4], since the simulation environment provides easily accessible resources to study new protocols and models. The ns-2 simulator [5] has been widely used due to its open-source model appropriate for the academic research community. In response to a number of deficiencies, the ns-3 discrete event network simulator [6] is under development, providing greater exibility, modularity using C++, evolvability, and support for heterogeneity including hybrid wired and wireless models.

Despite its advantages, ns-3 is relatively new with few protocol models yet incorporated into its release distribution [9]; existing built-in MANET protocols are limited to the optimized link state routing (OLSR) and ad hoc on-demand distance vector (AODV) protocols. Thus we have developed an ns-3 implementation of the destination sequenced distance vector (DSDV) protocol. DSDV is one of the earliest MANET routing protocols proposed [13], and provides a baseline proactive distance vector algorithm for performance comparisons. In this paper we describe our ns-3 implementation of the DSDV routing protocol. The rest of the paper is organised as follows: Section 2 presents background and related work on MANET protocols. Section 3 presents the details of the DSDV module implementation in ns-3. Baseline performance evaluation and is presented in Section 4. Finally, Section 5 presents our conclusions and future work,.

2. BACKGROUND & RELATED WORK

In this section we present background information about MANET routing protocols, and describe the protocol simulated in this paper: DSDV in ns-3.

2.1 Routing Protocol Types

MANET routing protocols can be classified into two categories based on their update mechanisms: proactive routing protocols and reactive routing protocols.

2.1.1 Proactive Table-Driven Routing Protocols

Proactive routing protocols maintain routing information of all the nodes in the network and add new routes or update existing routes by periodically distributing routing information among each other. One advantage of doing so is that routes to any destination are ready to use when needed. However, this is ofset by the overhead of route updates in response to mobility, for which nodes may have to wait anyway. For a large dynamic network, convergence may not be possible. Routing tables grow with the size and density of the network, rather than the number of routes actually needed. The overhead of flooding route advertisements to maintain convergence is a major drawback of proactive protocols.

2.1.2 Reactive On-Demand Routing Protocols

Unlike proactive routing protocols, reactive routing protocols construct routes only when they are required. Thus the nodes using reactive routing protocols do not need to update their routing tables as frequently and do not maintain routes for all nodes in the network. When a node using a reactive protocol requires a route to a new destination, it initiates a route request and must wait until the route is discovered. Reactive routing protocols have the disadvantage of delay in finding routes to new destinations traded against the savings of not needing to maintain tables for all possible routes.

2.2 MANET Routing Protocol

In this section we summarize the MANET routing protocol considered in this paper that have been implemented in ns-3 as of release 3.9.

2.2.1 DSDV

The DSDV (destination-sequenced distance vector) protocol [8] uses the Bellman-Ford algorithm to calculate paths. The cost metric used is the hop count, which is the number of hops it takes for the packet to reach its destination. DSDV is a table-driven proactive protocol, thus it maintains a routing table with entries for all the nodes in the network and not just the neighbors of a node. The changes are propagated through periodic and trigger update mechanisms used by DSDV. Due to these updates, there is a chance of having routing loops within the network. To eliminate routing loops, each update from the node is tagged with a sequence number. The sequence number from each node is independently chosen but it must be incremented each time a periodic update is made by a node. The sequence number of normal update must be an even number, since each time a periodic update is made the node increments its sequence number by 2 and adds its update to the routing message it transmits. The node cannot change the sequence number of other nodes. If a node wants to send an update for an expired route to its neighbors, only then it increments the sequence number of the disconnected node by 1. The nodes receiving this update will then look at the sequence number and if it is odd, will remove the corresponding entry from the routing table. Mobility of the nodes in MANETs causes route uctuations, for which DSDV uses settling time to dampen.

2.3 Previous DSDV Simulations

DSDV has been implemented and analysed in a number of simulation tools. A discrete event, packet-level routing simulator called MaRS (Maryland Routing Simulator) was used to evaluate DSDV performance under different network scenarios [2], assumed a channel bandwidth of 1.5 Mb/s with all data packets 512 B long, and interarrival times exponentially distributed with a mean of 300 ms. This analysis showed that for fraction of packets delivered, the proactive DSDV routing protocol outperforms on-demand protocols for both low and high mobility cases. Furthermore, the average end-to-end delays for DSDV show the minimum delay characteristics.

Most relevant to this paper is the ns-2 predecessor to ns-3.. DSDV exhibits low delay because only packets belonging to valid routes at the sending instant. However, DSDV has the highest overhead because of the table updates flooded throughout the entire network. This analysis also shows DSDV's inability to converge when the mobility is high, especially at high loads.

The ns-2 implementation of DSDV cannot be ported to the ns-3 environment due to the significantly different simulation architecture and code structure, however the ns-2 implementation was used to provide insight and guide design decisions for our ns-3 implementation.

3. DSDV MODULE FOR ns-3

This section describes our implementation of DSDV, which has been included in ns-3.10 stable release [7]. The main components of the DSDV implementation are routing update mechanisms, route table creation, and route maintenance. DSDV maintains valid routes and flushes out invalid routes based on the periodic update interval.

3.1 Routing Table

The structure of the DSDV *RoutingTable* is implemented as follows. Each entry implemented by the *RoutingTableEntry* class corresponds to a node in the network and the entry is mapped to that node's IP address. Every entry stores the following attributes of a node: its IP address, interface address, a pointer to its ns-3 net device, last known sequence number of the node, hop-count to reach the node, timestamp of the last update received for the node, and the settling time for that node. Also, we maintain a boolean value that specifies whether the entry for this node has changed since the last periodic update. This helps filter DSDV updates that are broadcasted through the trigger update mechanism. The *RoutingTable* class has methods to add, delete, update, look up, and print entries. It also defines the event functions explained in section 3.3.2.

3.2 Routing Advertisements

A node combines all the DSDV messages that it has to transmit into a single packet over *RouteAggregationTime*, if *Route-Aggregation* is enabled. However, to keep the packet size under the maximum transmission unit (MTU) in the implementation, we split the packets and send them separately if the packet size is longer than the MTU. As mentioned earlier, DSDV sends both periodic update messages and trigger messages. As soon as the routing protocol in the node is initialised, the node broadcasts its DSDV update message to the network to announce its presence. Each node will periodically broadcast its own routing table and all the nodes that are in range of this advertisement will use this information to update their routing tables. They may further trigger these updates to other nodes in their broadcast range. This mechanism is also used to keep the neighborhood relationship alive. One of the attributes that can be set for the routing protocol is the duration between these periodic updates, known as periodic update interval, using *PeriodicUpdateInterval*. It specifies the time duration for which a node has to wait before broadcasting its routing table. A node uses the trigger update mechanism when there are only a few updates to be transmitted. However, if the node identifies that the number of updates sent per trigger is comparable to that of a periodic update, then it sends a periodic update instead.

One more feature of DSDV routing protocol is the settling time, which is used to prevent the advertisement of an unstable route that arrives at the node before a stable route. Since DSDV uses broadcasting to propagate these changes, it would create unnecessary overhead in the network. Thus, a node waits for the period of *SettlingTime* before propagating any update. However to make sure updates for stable routes are not delayed, we use the attribute *WeightedFactor*. This is used to calculate the weighted average of the settling times for the updates received from a node. If the update is for an old and stable route, the settling time decreases. A node can not process multiple update messages simultaneously. If the nodes are highly mobile, the node might have to send many updates as there would be a lot of route changes. This will lead to more overhead in the network that may increase the number of collisions. To reduce overhead, we use *RouteAggregation*. This optional feature enables multiple update messages to be sent out as a single update message. The period over which routes are aggregated can be modified by *RouteAggregationTime attribute*.

3.3 Processing of Updates

As mentioned in Section 3.2, a node might receive many updates stacked within a single packet. Since the DSDV header size is fixed to 12 bytes, we iterate over the packet until it is empty to extract all the 12B DSDV control messages. Each message is processed as it is extracted. We first verify the destination address in the extracted message. If it is same as the node's IP address, the message is discarded. If not, the protocol verifies whether the received update is for a new route with a valid sequence number. In this case the route is added to the permanent routing table and broadcast immediately. Otherwise, if the node already has an update for that IP address the protocol verifies the sequence number. If the sequence number is odd and if the node from which this update was received is the next hop neighbor in the table, then the route is deleted from the routing table and triggers an update of this broken route to other nodes immediately. However if the sequence number is valid we have three cases in which the sequence number can relate to the sequence number from the table:

- Received > Local : The protocol veri_es the received hop-count with the local value of hop count. If they are not equal, the node updates its local entries in the advertising routing table and waits for settling time period if *SettlingTime* is enabled. This is implemented using events in ns-3. This mechanism is explained briey in section 3.3.2. If the received hop-count is same as the local value, then the node does not wait for the settling time interval as this is an update for the stable route.

- Received = Local : If the received hop-count is less than the local value, then the local value is updated and the protocol waits for settling time to make sure that this update is not an unstable one. If it does not receive any further update for that destination address, the protocol updates the permanent table with this update and triggers this update back to all its neighbors. However if the received hop-count is greater than or equal to the local value, the message is discarded.

- Received < Local : The protocol discards this update message as it already has a most recent update from that destination.

After processing messages from the packet, the *SendTriggeredUpdate* method will be called. *SendTriggeredUpdate* iterates over the advertising routing table, computes all the needed updates, and creates a new packet with these updates and broadcasts.

3.4 Parameter Tuning

An advantage of DSDV is that it is relatively simple compared to other MANET routing protocols. It is also similar to the conventional wired distance-vector routing protocols, with only minimum adaptations made. However, the drawback of DSDV is that its periodic overhead for broadcasting is unavoidable even if the network is static. If the node density increases in the network, the routing table will also become larger. This leads to more updates with larger packet sizes. With a highly dynamic network, the routing updates may take up the available bandwidth of channel. Furthermore, before the time of update, intermediate nodes may use stale information to forward packets. Thus proper choice of *PeriodicUpdateInterval* and SettlingTime is important in a highly mobile environment.

4. Random Waypoint

The Random Way Point Mobility Model[10] includes pauses between changes in direction and/or speed. A Mobile node begins by staying in one location for a certain period of time (i.e. pause). Once this time expires, the mobile node chooses a random destination in the simulation area and a speed that is uniformly distributed between [min-speed, max-speed]. The mobile node then travels toward the newly chosen destination at the selected speed. Upon arrival, the mobile node pauses for a specified period of time starting the process again. The random waypoint model is a commonly used mobility model in the simulation of ad hoc networks. It is known that the spatial distribution of network nodes moving according to this model is non uniform. However, a closed-form expression of this distribution and an in-depth investigation is still missing. This fact impairs the accuracy of the current simulation methodology of ad hoc networks and makes it impossible to relate simulation-based performance results to corresponding analytical results. To overcome these problems, it is presented a detailed

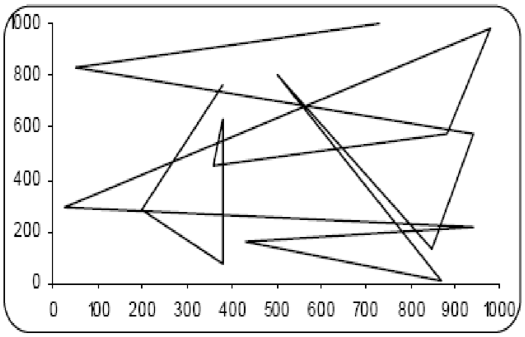

Figure1 Node Movement in Random Way Point.

analytical study of the spatial node distribution generated by random waypoint mobility. The movement trace of a mobile node using the Random Waypoint model is shown in figure 1. It is considered that a generalization of the model in which the pause time of the mobile nodes is chosen arbitrarily in each waypoint and a fraction of nodes may remain static for the entire simulation time.

5. DSDV MODULE EVALUATION

To evaluate the performance of our DSDV routing protocol implementation, we performed simulations using the ns-3.9 version of the network simulator1.

5.1 Performance Metrics

The performance metrics for evaluation of the DSDV routing protocol are packet delivery ratio (PDR), routing overhead, and delay.

- **Packet Delivery Ratio PDR**: The number of packets received divided by the number of packets sent by the application.
- **Routing Overhead**: The fraction of bytes used by the protocol for DSDV control messages.
- **Delay**: The time taken by the packet to reach the destination node's MAC protocol from the source node's MAC protocol.

5.2 Simulation Setup

Parameter	Value
Mobility Model	Random Waypoint model
Nodes (WifiNode's)	95 nodes
Simulation Time	100s
Packet Size	1000 bytes
Sink Nodes	17
Update Interval	15s
Node Speed	10 m/s
Traffic Model	Constant Bit Rate
Settling Time	6

We performed the simulations over an area of 1500m2. All the simulations were averaged with each simulation running for 1500s. The communication model is peer-to-peer communication with as many ows as the number of nodes in the network. We initially performed some simulations with 1000 byte packets but observed that the PDR was low, therefore we used a packet size of 64 bytes based on previous study. All the nodes are configured to send 4 packets/s. Using this lower packet size, we can correctly evaluate the performance of the protocol. We use the ns-3 On-Off application to generate CBR (constant-bit rate) trafic. The 802.11b MAC is used over Friis propagation loss model to limit the transmission ranges of nodes. The transmit power was set to 8.9048 dBm to achieve a 250 m transmission range. The mobility model used is random waypoint with random velocities from 0 { 20 m/s and pause times of 100 { 800 s. We used the default DSDV parameters values described above except for PeriodicUpdateInterval which was varied among f4, 5, 8, 12, 15, 30g s and SettlingTime which was varied among f0, 1, 2, 3, 4, 5, 6g s.

5.3 Simulation Analysis

6. CONCLUSIONS

This paper present the implementation of the DSDV MANET routing protocol in ns-3. The results indicate that DSDV overhead increases as the node density increases. PDR performance of the DSDV is inversely affected as the overhead increases.As part of the future work, we will explore the effects of buffering on sparser networks and also to analyse the routing performance of the MANET protocols including DSR.

7. REFERENCES

[1] M. Conti and S. Giordano. Multihop Ad Hoc Networking: The Theory. IEEE Communications Magazine, 45(4):78 {86, April 2007.

[2] S. R. Das, R. Casta~neda, and J. Yan.

Simulation-based Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks. Mob. Netw. Appl., 5(3):179{189, September 2000.

[3] D. Johnson, Y. Hu, and D. Maltz. The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4. RFC 4728 (Experimental), Feb. 2007.

[4] S. Kurkowski, T. Camp, and M. Colagrosso. MANET Simulation Studies: The Incredibles. SIGMOBILE Mob. Comput. Commun. Rev., 9(4):50{61, October 2005.

[5]The network simulator: ns-2. http://www.isi.edu/nsnam/ns, December 2007.

[6]The ns-3 network simulator. http://www.nsnam.org, July 2009.

[7] Dsdv code review. http://codereview.appspot.com/1668042.

[8]C. E. Perkins and P. Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In Proceedings of the ACM Conference on Communications Architectures, Protocols and Applications (SIGCOMM), pages 234{244, 1994.

[9] E. Weingartner, H. vom Lehn, and K. Wehrle. A Performance Comparison of Recent Network Simulators. In IEEE International Conference on Communications (ICC), pages 1{5, June 2009.

[10]M. K. Jeya Kumar and R.S. Rajesh. Performance Analysis of MANET Routing Protocols in Different Mobility Models. IJCSNS International Journal of Computer Science and Network Security, VOL.9 No.2, February 2009.