A Performance Comparison of MDSDV with AODV and DSDV Routing Protocols

A. Etorban etorban@macs.hw.ac.uk
Peter J.B King pjbk@macs.hw.ac.uk
Phil Trinder P.W.Trinder@hw.ac.uk

School of Mathematical and Computer Sciences (MACS)
Heriot Watt University at Edinburgh UK

Abstract

We present a systematic comparative evaluation of a new multipath routing protocol for MANETS. The new protocol, called Multipath Destination Sequenced Distance Vector (MDSDV) is compared with two known protocols DSDV and AODV. MDSDV finds disjoint paths which do not have any common nodes between a source and destination, and we outline some adaptation of MDSDV over previous work. We evaluate the protocols on a range of MANETS with between 10 and 80 nodes, either static or highly dynamic nodes, and slow, medium or fast node speeds. The protocol comparison metrics are Packet Delivery Fraction (PDF), end-to-end delay, and data dropped.

Keywords: Ad Hoc Networks, Routing protocols, Proactive Routing, Performance Evaluation

1 Introduction

Nodes in ad hoc networks are distinguished by their limited resources such as power and memory as well as mobility. Due to the limited transmission range of the nodes, multiple hops may be needed for a node to send data to any other node in the network. Thus each node acts as a host and router. If a node needs to communicate with another that is outside its transmission range, an intermediate node acts as a router to relay or forward packets from the source to the destination. For this purpose, a routing protocol is needed. Nodes in ad hoc networks are free to move over a certain area. Because of this movement, the network topology frequently changes. This means that we need a routing protocol that quickly adapts to topology changes. Routing protocol design is an important and essential issue for Ad Hoc networks due to dynamism of the network. Routing protocols can be classified as three types; reactive, proactive, and hybrid routing protocols. In addition they can be categorized by the number of paths they retain to each destination: single path or multipath. The single path category includes protocols such as AODV [15], DSR [7], TORA [14], DSDV [16], ZRP [4], where each node has just one path for each destination. AODV is an on-demand (reactive) routing protocol where the routes are created as needed, and maintained during the communication session. Nodes get routes by broadcasting a route request (RREQ) message and receive a route reply (RREP) message from the destination or from an intermediate node. During the route discovery, a node broadcasts a RREQ message for a desired destination. Nodes that have a route to the desired destination respond to the RREQ message by sending a RREP message to the source. Nodes that have no route to the desired destination rebroadcast the RREQ message. As a node discovers a link failure, it sends a route error (RERR) message to a list of nodes (precursors). DSR is also a reactive routing protocol with different mechanism, where each route request records its traversed path; also the route reply sent back to the source includes the complete path between the source and the desired destination [3]. DSDV is a table-driven (proactive) routing protocol where nodes periodically broadcast their routing tables to neighbouring nodes. Routing updates from any node are tagged with increasing sequence number by 2 to distinguish between stale and new routes. Before accepting the update message, the received node checks if the sequence number specified in the update message is higher than the sequence number recorded in its own routing table. As a node discovers a broken link it increases the sequence number and broadcasts an update packet. DSDV uses both full and incremental updates of routing tables to reduce the routing overhead. The main advantage of proactive routing protocol is that a route to any destination is available even if it is not needed. In contrast to single path routing, multipath routing protocols is a useful technique of finding multiple routes between a source and destination which can be used to compensate for the topology change caused by the movement of nodes, and provide load balancing in ad hoc networks. Multipath has been one of the most important issues in the area of routing, and several multipath algorithms have been proposed by researchers such as AOMDV [10][11][13], AODVM [12], AODVM-PSP [5][6], MOR [1], and NDMR [9], Cheng [8] argues that most of the multiple path discovery algorithms were developed on the DSR protocol [7] such as [17], by adding a node list in each header of the RREQ and RREP control packets which leads to an extra overhead into data packets in large networks. Also, he stated that few multipath protocols were developed on AODV such as AOMDV [9] that does not need to record all the nodes in the headers of the control packets, but it aims to create disjoint paths. The rest of this paper is organized as follows. Section 2 outlines the new MDSDV routing protocol. The simulation methodology is described in section 3. We present the first systematic evaluation of MDSDV in section 4. Finally we conclude by summarizing the results and discussing implications in section 5.

2 An Overview of the MDSDV Routing Protocol

This section briefly outlines the MDSDV protocol and describes how it has evolved beyond the initial design presented in [8].

2.1 MDSDV Design

A novel multipath routing protocol for MANETS based on DSDV was proposed in [8]. Due to mobility, the network connectivity changes frequently and by

Field name	Description
Destination node	Address of the destination node
Next hop	The first hop to the destination
Second hop	the second hop to the destination
Number of hops	Number of hops to the destination
Link-id	An identifier generated by the new node for the newest routes
Sequence number	A sequence number to distinguish stale routes
Time	The time that the path has been discovered

Table 1: Routing table structure (RT) entry

maintaining multiple disjoint paths, one can increase the probability that source can reach the destination via one of the known paths. Multiple paths can also be employed to achieve better load balance and improve quality of service in bandwidth constrained MANETs [2]. The proposed routing protocol uses a method to find number of disjoint paths, which do not have any common nodes between a source and destination. Two new fields called *second hop* and *link-id* which is generated by the destination have been used to get these disjoint paths. Each node maintains its routing table that lists a number of paths for each destination. The routing table is used to transmit packets through the ad-hoc network. The nodes update their routing tables periodically. Table (1) shows the structure of a routing table entry.

Each node broadcasts a hello message periodically. Neighbours receiving such a message add a new entry as a route for the sender and unicast a full dump of their routing tables to the hello message sender. As the node receives a full dump, it starts to create its routing table, creates a link number and broadcasts an update packet. Upon receiving an update packet, a node updates its routing table, updates the update packet and broadcasts the update packet. As soon as the node discovers a link failure, it broadcasts an Error packet containing the link number belonging to the unreachable node. Next, it updates its routing table by deleting entries that use the unreachable node as a first hop. Any node that receives this error packet; checks its routing table and deletes entries that have the same link-id. The received node rebroadcasts the error packet only if its routing table is updated, otherwise the packet is discarded. The intermediate node is not forced to use a certain path to forward the data.

2.2 MDSDV Design Evolution

Broadcasting large number of control packets (hello, update, full dump, error packets, and failure message) causes very high overhead. So, some modifications have been made to the original MDSDV protocol described in [8] to reduce the overhead to get better performance. Instead of broadcasting a hello message periodically and broadcasting an update packet immediately as soon as receiving a full dump, the algorithm is changed as follows. Periodically, the node checks if there is any change in its routing table. If so, it broadcasts an update packet contains the changes; otherwise it broadcasts a hello message. Upon receiving an update packet, a node updates its routing table regarding the received update packet. When a node receives any routing packet (hello message, full dump, update packet, or error packet), it seeks for a direct route (1 hop) to the packet sender. If the route is found, the receiver just updates the timeout of the sender, otherwise it unicasts a full dump of its routing table to the packet sender considering it as a new neighbour. As a node receives an error packet, it does not need to rebroadcast the error packet, but updates its routing table by deleting the entries that have the link id which is included in the error packet. Because error packet is limited, failure message is invoked only when an intermediate node tries to forward a data packet to a specified node and this node is unreachable; It unicasts a failure message to the data packet sender, so that the previous hops and the source stop sending data along this route. Also, when a nodes plans to send a data packet, it includes the second hop in the header of the packet to enforce the intermediate node to use a specific path. If the intermediate node has the specified path, it updates the second hop field of the header, and forwards the packet. Otherwise, it unicasts a failure message so that the previous hops and the source stop sending data along this route, and seeks for an alternative path to forward the received packet. Instead of using two tables, Neighbours table is cancelled, and a timeout field is used to distinguish the one hop neighbours. Upon receiving a control message from a neighbour, the timeout is updated.

MDSDV uses 5 different routing packets that are described as follows:

Hello message: The node broadcasts this type of message when it has no change in its routing table.

Full_dump packet: as soon as the node receives a hello message, update packet, or error packet, it checks if it has a direct route to the packet sender. If so, it reschedule its timeout, otherwise, it unicasts a full dump of its routing table to the packet sender.

Update packet: The node broadcasts this kind of packet if its routing table is changed.

Error packet: If the node discovers a broken link, it broadcasts this kind of packet.

Failure message: when the node tries to forward a data packet through the specified route, and has no such route, it unicasts a failure route to the source asking not to use the route anymore. When a route breaks the source still keeps on sending data packets unaware of the link breakage. This leads to a large number of data packets being dropped.

3 Simulation Methodology

Each routing protocol has its own strategy and advantages. None can be considered as better than the others in all situations. We use the network Simulator ns-2 [17] commonly used in the ad hoc networking community to compare MDSDV with two known protocols DSDV and AODV. Each simulation runs 10 times and the figures reported are the mean results. Tables 2 and 3 summarize the parameters used in the communication and movement models.

3.1 Communication Model

In our simulation, we use constant bit rate (CBR) traffic sources. When defining the parameters of the communication model, we experimented with 10, 20, 30,

Parameter	**Value**
Traffic type	CBR (UDP)
Number of nodes	10, 20, 30, 40, 50, 60, 70, and 80 nodes
Number of data sources	10 CBR/UDP connections
Transmission rate	4 packets/second

Table 2: Parameters of communication model

Parameter	**Value**
Simulator	Ns-2
Simulation time	300 seconds
Area of the network	600m x 600m
Number of nodes	10, 20, 30, 40, 50, 60, 70, and 80 nodes
Pause time	1 and 300 seconds
Maximum speed of nodes	1, 15, and 30 meters per second.
Transmission range	250 m
Mobility model	Random waypoint

Table 3: Parameters used in the movement model

40, 50, 60, 70, and 80 nodes; the sending rate used was 4 packets per second, and the network contains 10 CBR sources.

3.2 Movement Model

In our simulations, we use the random waypoint model [1][9] where each node begins the simulation by remaining stationary for pause time seconds. It then selects a random destination in the 600m x 600m space and moves to that destination at a speed distributed uniformly between 1 and some maximum speed. Upon reaching the destination, the node pauses again for pause time seconds, selects another destination, and proceeds there as previously described, repeating this behaviour for the duration of the simulation. We fix the area to be 600x600 meters, and the simulation time to be 300 seconds. Meanwhile, we vary the number of nodes to compare the protocols performance for low and high density, pause times 1 and 300 seconds (1 as continuous motion and 300 as a static network), and speed of nodes 1, 15, and 30 m/sec (low, medium, and high speed). Table 3 lists the movement parameters of the simulations.

3.3 Performance Metrics

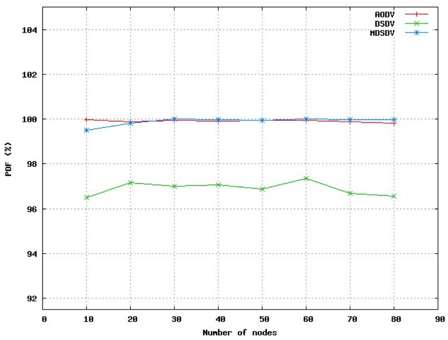
We report three performance metrics for the protocols: packet Delivery Fraction (PDF), Average End-to-End delay (ms), and Data dropped. The Packet Delivery Fraction (PDF) is a measure of throughput, i.e. how much data is successfully delivered by a protocol, and is the percentage of successfully delivered packets of those generated by the source. Average End-to-End delay measures the communication latency of a protocol and it is the mean time that elapses to deliver each packet. Data dropped is a measure of data lost by the protocol and includes the data that the source or intermediate nodes drop during the simulation time.

4 Results and Analysis

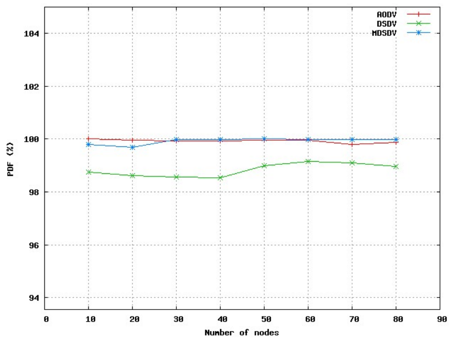
The simulation results bring out several differences in the three protocols. We discuss them in the following subsections.

4.1 Packet Delivery Fraction

Figures 1,2, and 3 compare the data packet delivery fraction of the three protocols as network size grows.

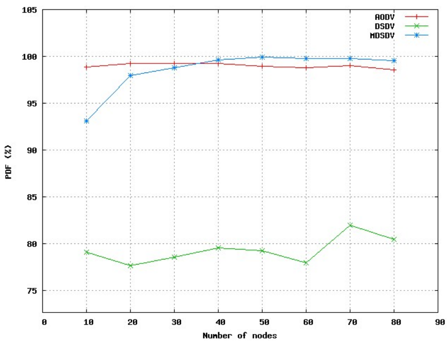


(a) PDF vs. network size, Pause Time 1 sec

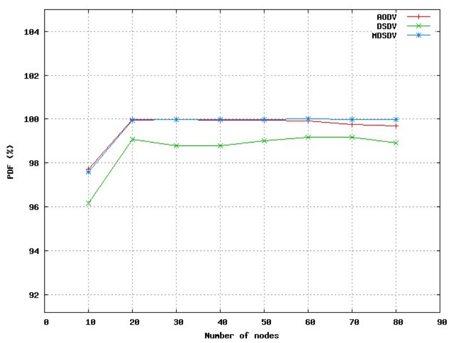


(b) PDF vs. network size, Pause Time 300 sec

Figure 1: PDF vs. network size at low speed



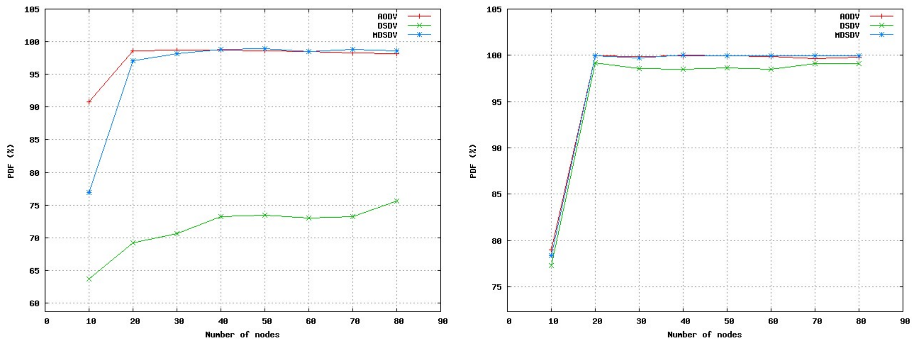
(a) PDF vs. network size, Pause Time 1 sec



(b) PDF vs. network size, Pause Time 300 sec

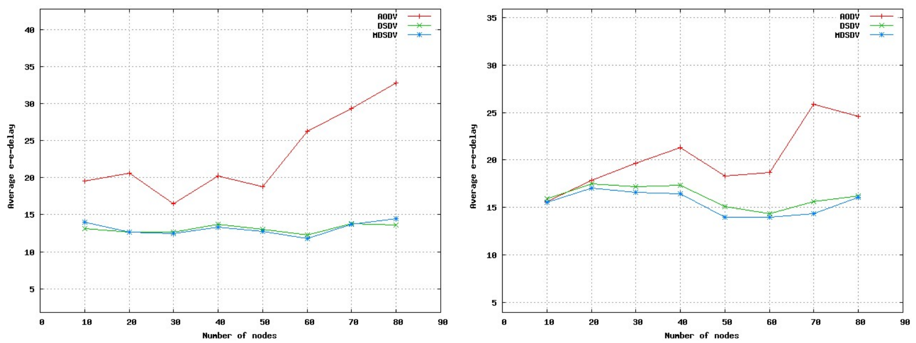
Figure 2: PDF vs. network size at medium speed

Data delivery is measured as packet delivery fraction (PDF) and network size is between 10 and 80 nodes. We investigate static networks with pause time 300 sec (simulation time) in figures 1(b), 2(b), and 3(b), and highly dynamic networks with pause time 1 second in figures 1(a), 2(a), and 3(a). We also investigate networks at slow node speeds (1 m/sec) in figures 1(a) and 1(b) and networks with high node speeds (30 m/sec) in figures 3(a) and 3(b). The data packet delivery fraction for MDSDV and AODV protocols are very similar in



(a) PDF vs. network size, Pause Time 1 sec (b) PDF vs. network size, Pause Time 300 sec

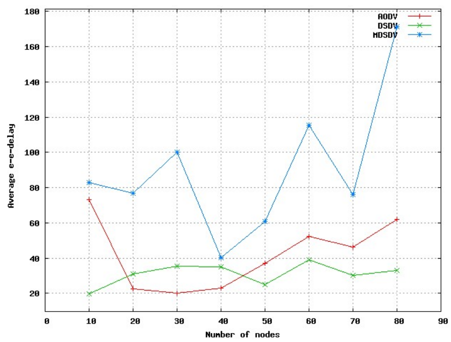
Figure 3: PDF vs. network size at high speed



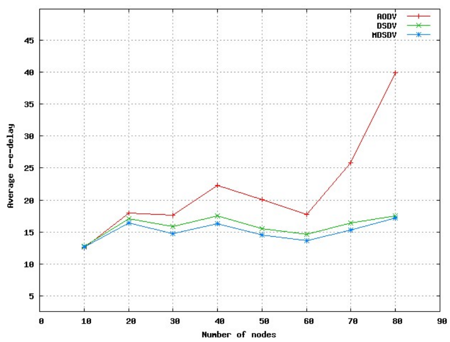
(a) Average End-to-End delay vs. network size, Pause Time 1 sec (b) Average End-to-End delay vs. network size, Pause Time 300 sec

Figure 4: Average End to End Delay vs. network size at low speed

most cases figures 1(a), 1(b), 2(b), and 3(b). Figure 3(a) shows that AODV delivers 14% more data than MDSDV in 10 node network at pause time 1 and node speed 30 m/sec. Also, figure 2(a) shows that AODV delivers 5% more data than MDSDV in 10 node network at pause time 1 and node speed 15 m/sec. This is because with a small number of nodes, any node has only a few neighbours and consequently the possibility of finding node-disjoint route is decreased. All figures show that MDSDV successfully delivered nearly 100% of data when the network size is greater than 20 nodes because of using an alternative path in case of a broken link. Also, all figures show that DSDV delivered less data than AODV and MDSDV. The reason for this is that MDSDV uses an alternative path and AODV broadcasts a route request immediately; whereas DSDV waits for an update packet. Also, as the node's speed increases, the difference in PDF becomes bigger; this is because as speed increases, the probability of link failure increases (figures 1(a), 2(a), and 3(a)).

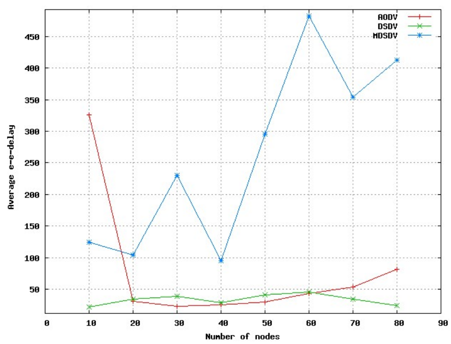


(a) Average End-to-End delay vs. network size, Pause Time 1 sec

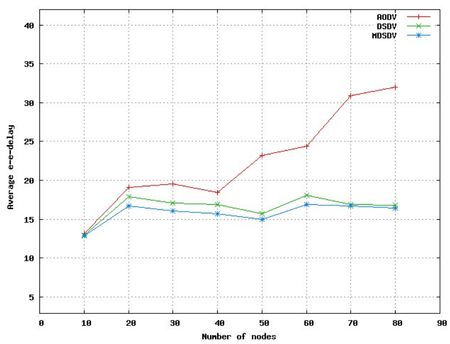


(b) Average End-to-End delay vs. network size, Pause Time 300 sec

Figure 5: Average End to End Delay vs. network size at medium speed



(a) Average End-to-End delay vs. network size, Pause Time 1 sec



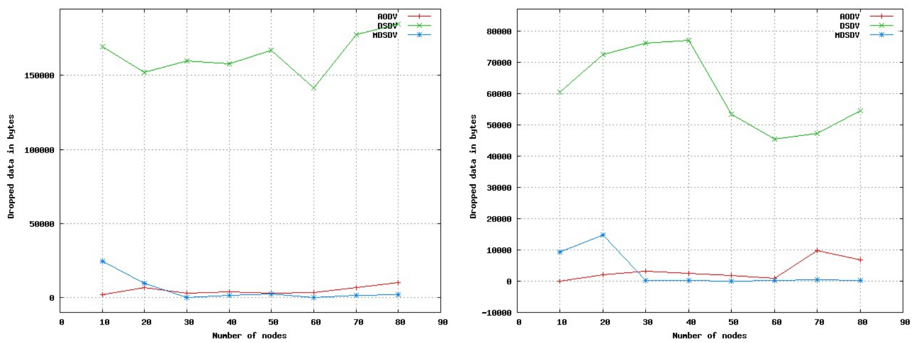
(b) Average End-to-End delay vs. network size, Pause Time 300 sec

Figure 6: Average End to End Delay vs. network size at high speed

4.2 Average End to End Delay

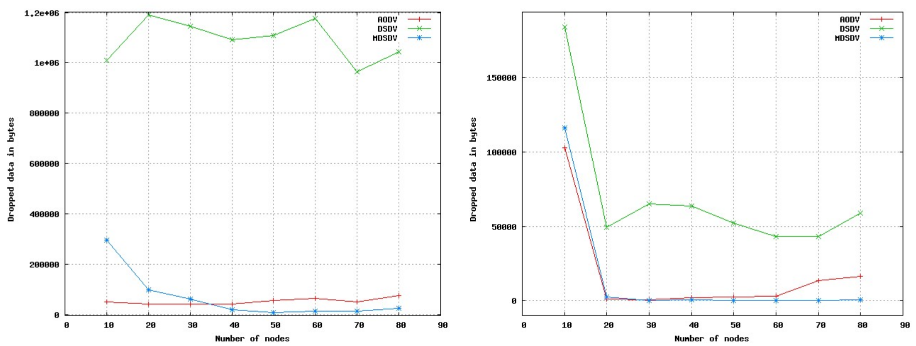
Figures 4, 5, and 6 show the comparison between the three protocols on the basis of average delay versus network size (number of nodes) with two pause times (1 second and 300 seconds). Figures 4(a), 5(a), and 6(a) represent dynamic networks where the pause time is 1 second; whereas figures 4(b), 5(b), and 6(b) represent static networks where the pause time is 300 seconds (simulation time). Compared to AODV and DSDV, figures 4(a), 4(b), 5(b), and 6(b) show that MDSDV has less delay in static networks or when the speed is very low (1 m/s). The same figures show that AODV has the greatest delay (typically more than 15 ms) and this increases as the number of nodes increases. Figure 5(a) and figure 6(a) show that MDSDV has the worst delay at medium and high speed when the pause time is very small (continuous motion). As a result of node's movement, a node becomes far from another node which causes a broken link that leads to broadcast an error packets, also a node becomes a new neighbour to another node which leads to the unicast of full dump. In consequence, the number of routing packets increases and the network becomes busy by broadcasting and unicasting these control packets.

4.3 Data Dropped



(a) Data dropped in bytes vs. network size, Pause Time 1 sec (b) Data dropped in bytes vs. network size, Pause Time 300 sec

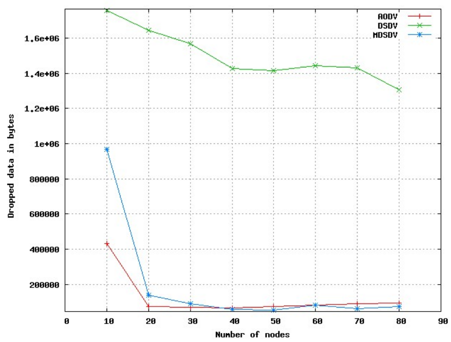
Figure 7: : Data dropped in bytes vs. network size at low speed



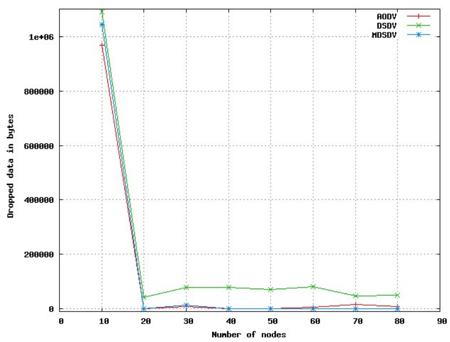
(a) Data dropped in bytes vs. network size, Pause Time 1 sec (b) Data dropped in bytes vs. network size, Pause Time 300 sec

Figure 8: : Data dropped in bytes vs. network size at medium speed

Figures 7, 8, and 9 show the comparison between the three protocols in terms of data dropped by each protocol. In all cases, DSDV is the worst in terms of data dropped because it waits for period of time to get new information; if no route available and a node using DSDV plans to transmit data, DSDV has to queue the packets; and the packets will be dropped in case of the queue is full. MDSDV drops slightly less data than AODV in the sparse network (networks with 20 nodes) because MDSDV uses an alternative path in case of link failure; whereas AODV broadcasts a route request in case of link failure and waits for some time to get new information by receiving a route reply; during this time



(a) Data dropped in bytes vs. network size, Pause Time 1 sec



(b) Data dropped in bytes vs. network size, Pause Time 300 sec

Figure 9: : Data dropped in bytes vs. network size at high speed

AODV queue the packets. Some times the packets are dropped because of expiration in the AODV queue is categorized as AODV queue timeout.

5 Conclusion

5.1 Summary

This paper evaluates a new proactive routing protocol called MDSDV for ad hoc networks. We have compared the performance of MDSDV with two major routing protocols AODV and DSDV using ns-2 simulations. The main results of our evaluation are as follows:

- The results show that the performance of AODV compared to MDSDV is very similar in terms of packet delivery fraction; whereas there is a big difference in the performance of DSDV compared to MDSDV, and this difference increases as the pause time decreases. As mentioned this is because in case of link failure, MDSDV immediately uses an alternative path (if found) to send data, AODV gets a new path by broadcasting a route request immediately, and DSDV waits for a periodic update or triggered update to get new information (section 4.1).

- In terms of Average end-to-end delay time, MDSDV has the shortest delay in static networks, whereas it has the worst delay for continuous motion networks. Also, we notice that there is a small difference between the delay of DSDV and MDSDV in static networks. AODV has the worst delay in static networks; whereas MDSDV has the worst delay in highly dynamic networks. The reason behind the big delay of MDSDV is the increasing of routing packets. In continuous motion; nodes go far from each other and become neighbours of each other frequently. This leads to broadcast of an error packet when a broken link occurs and unicast of a full dump when a node becomes a new neighbour to another node (section 4.2).

- In terms of data dropped, MDSDV consistently drops a little less data than AODV where the network has more than the 20 nodes. DSDV consistently drops much more data than AODV and MDSDV. This is because DSDV waits for a period of time to get new information. If a node using DSDV is transmitting data and link failure occurs, it queues the packets until getting new path for the desired destination. Node starts to drop the packets if the queue is full, and happens frequently especially at high speeds. MDSDV uses an alternative path in case of the used link break (section 4.3).

5.2 Discussion

From sections 4.1, 4.2, and 4.3 we observe that MDSDV behaves very well in terms of packet delivery fraction and data dropped in both static and dynamic networks. In other words, it delivers nearly 100% of data in dense networks (networks with more than 20 nodes). Also, speed of nodes does not effect on the behaviour of the protocol. In contrast, MDSDV is bad in terms of average end-to-end delay in highly dynamic networks (low pause time and high speed) as a result of the number of routing packets increasing.

Future Work MDSDV suffers from a large number of control packets, so our future work concentrates on minimizing the overhead. Also, we plan to test MDSDV with more data sources.

References

[1] E. Biagioni and S.H. Chen. A reliability layer for ad-hoc wireless sensor network routing. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 1–8, 2004.

[2] Y. Ganjali and A. Keshavarzian. Load balancing in ad hoc networks: Single-path routing vs. multi-path routing. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, volume 2, 2004.

[3] JJ Garcia-Luna-Aceves, M. Mosko, and C.E. Perkins. A new approach to on-demand loop-free routing in ad hoc networks. In *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, pages 53–62. ACM New York, NY, USA, 2003.

[4] Z.J. Haas. A New Routing Protocol for the Reconfigurable Wireless Networks. In *Proceedings of the IEEE International Conference on Universal Personal Communications (ICUPC*, pages 562–566, 1997.

[5] F. Jing, RS Bhuvaneswaran, Y. Katayama, and N. Takahashi. A multipath on-demand routing with path selection probabilities for mobile ad hoc networks. In *Wireless Communications, Networking and Mobile Computing, 2005. Proceedings. 2005 International Conference on*, volume 2, pages 1145–1148, 2005.

[6] R. S. Katayama Y. Takahashi N. Jing, F. Bhuvaneswaran. Multipath Routing Selection Strategies in Wireless Mobile Ad Hoc Networks. In *IEEE ICSCN*, pages 117–121, Anna University, Chennai, India, 2007.

[7] D.B. Johnson, D.A. Maltz, J. Broch, et al. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. *Ad hoc networking*, 5:139–172, 2001.

[8] P.J.B. King, A. Etorban, and I.S. Ibrahim. A DSDV-based multipath routing protocol for mobile ad-hoc networks. In *in The 8th Annual PostGraduate Symposium on The Convergence of Telecommunications, Networking and Broadcasting*, pages 93–98, The School of Computing and Mathematical Sciences, Liverpool John Moores University, 2007.

[9] X. Li and L. Cuthbert. Stable node-disjoint multipath routing with low overhead in mobile ad hoc networks. In *The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems, 2004.(MASCOTS 2004). Proceedings*, pages 184–191, 2004.

[10] M.K. Marina and S.R. Das. On-demand multipath distance vector routing in ad hoc networks. In *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, pages 14–23, 2001.

[11] M.K. Marina and S.R. Das. Ad hoc on-demand multipath distance vector routing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(3):92–93, 2002.

[12] S. Mueller and D. Ghosal. Analysis of a distributed algorithm to determine multiple routes with path diversity in ad hoc networks. In *Proceedings of the Third International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pages 277–285. IEEE Computer Society Washington, DC, USA, 2005.

[13] A. Nasipuri and S. Das. On-demand multipath routing for mobile ad hoc networks. In *Proceedings of IEEE ICCCN99*, pages 64–70, Boston, 1999.

[14] V.D. Park and M.S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Ieee Infocom*, volume 3, pages 1405–1413, 1997.

[15] C. Perkins, E. Belding-Royer, and S. Das. RFC3561: ad hoc on-demand distance vector (AODV) routing. *Internet RFCs*, 2003.

[16] C.E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Communication Review*, 24(4):234–244, 1994.

[17] L. Wang, M.D. Shu, L. Zhang, and O.W.W. Yang. Adaptive multipath source routing in ad hoc networks. In *Proceedings of IEEE International Conference on Communictions (ICC)*, pages 867–871, Finland, 2001.