Performance Evaluation of Ad-hoc Routing Protocols through NCTUns 6.0

Wajid Mumtaz
Member, IEEE
Centre for Advanced Studies in Telecommunication (CAST)
COMSATS Institute of Information Technology
Islamabad, Pakistan

Beenish Niaz
Centre for Advanced Studies in Telecommunication (CAST)
COMSATS Institute of Information Technology
Islamabad, Pakistan

ABSTRACT

The idea of ad hoc networking is gaining importance as more and more dynamic routing protocols have been designed and a lot of research with state of the art network traffic simulator has been carried out. Routing protocols solve many problems of ad hoc networks. In this paper we are aiming to provide a performance comparison of three routing protocols i.e. ADV, AODV and DSDV. NCTUns 6.0 is used as a simulator. The performance metrics for evaluating the protocols are number of packet dropped, number of packet collision and throughput.

General Terms

Vehicular Adhoc Networks, Algorithms

Keywords

Vehicular Ad-hoc networks, Adhoc On-Demand Distance Vector (AODV), Adaptive Distance Vector (ADV), and Destination-Sequenced Distance Vector (DSDV).

1. INTRODUCTION

Wireless Ad hoc networks are ubiquitous in everyday life. The popularity increase is due to their demand in the personal as well as industrial and military applications. Everyone wants to be stay connected on the go. Some examples of Ad hoc networks are communication of vehicles with roadside installed communication equipments, business people sharing information during a meeting etc. Mobile Ad hoc Networks (MANETs)[7], [11] and Vehicular Adhoc Networks(VANETs)[6], [11] are subclasses of wireless Adhoc networks. A lot of research is going on for routing protocols for Adhoc networks. Various simulators are used for this purpose. NCTUns is one of the simulators. NCTUns is an innovative simulator with an integrated network and traffic simulation capabilities. It uses the real world TCP/IP protocols for realistic simulations. NCTUns uses kernel reentering simulation methodology. Protocols regarding MANETs and VANETs can be simulated. NCTUns is an open source simulator and its code can be modified for simulating new research ideas. This flexibility makes it ideal for researchers to evaluate new network protocols without worrying about the other layers. Present work focuses performance evaluation of network protocols especially the ADV, AODV and DSDV .For this purpose the chosen performance metrics are as follows

- Packet collision: Packet collision can result in a loss of packet integrity and compared the network performance. In a network when two or more nodes try to transmit packets at the same time packet collision may result.

- Number of Packet dropped: The number of data packets that cannot be transmitted successfully to the destination.

- Throughput: Throughput measures effectiveness of the network in delivering data packets, that is, how well does the network deliver packets from source to destination [1], [2],[3].

Remainder of the paper is organized as follows. Section II provides information related work in this area. Section III provides an overview of the three routing protocol AODV, ADV and DSDV. Section IV describes the experimental setup and discusses the simulation results. Section 5 concludes the paper.

2. BACKGROUND

For our performance comparison study, the authors pick Adhoc On-Demand Distance Vector (AODV), Adaptive Distance Vector (ADV) and Destination-Sequenced Distance Vector (DSDV) routing protocols. A brief description of these protocols is as follows

2.1 AODV

The AODV [14] protocol is designed to handle mobile traffic with adhoc behavior. It is a reactive routing protocol; therefore, routes are determined when needed. It offers adaptation to dynamic link conditions while ensuring low processing and memory overhead. It also offers loop freedom. It can perform both unicast and multicast routing. The multicast group members are tracked through the use of trees, which represent the group members and the nodes that connect them. Each node contains a table for tracking the network topology. This requirement of route is dependent on the source node. As long as the route is valid it is timely updated and is tracked by a sequence number. Each route has its own sequence number which ensures the freshness of the route through proper and timely updates. A new route entry is entirely dependent on there quest of source node.

AODV works with a route request/reply query cycle which requires three commands: RREQ, RREP and RERR. These command messages are sent through 654 port running UDP and applies normal IP addressing. AODV works only when a new destination is requested, otherwise it is not required as long as a route to the requested destination is valid. A broadcast request is generated using RREQ. In response a reply is generated by the node only if it’s the destination node or either a node with a valid route is found. Such nodes are called intermediate nodes and the check on the validity of their route is dependent on how fresh it is in terms of the sequence number which needs to be at least as great as the RREQ.

Once the destination is found; a RREP is unicasted all the way back to the originator. Therefore each node keeps a track on the incoming requests which are used for unicasting back to the originator of the RREQ from the nodes which fulfills their quest. Apart from this the AODV protocol is activated whenever a broken link is detected which leads to the generation of RERR message. The RERR requires a precursor list in order to work. The precursor list contains the IP addresses of all nodes that are likely to be used by the next hop for reaching the destination. This implies that the working of the protocol can be divided into two processes one being the send request and other being the receive event. Since each node is working as AODV router therefore it can either initiate a send request process or receive an event receive process. All this process of requesting and replying requires a table to be created and maintained. AODV has to deal with route table management in order to function with best results. These tables are to keep records of all possible routes either they are long or short lived. There are some specific fields required in the table for each route entry by AODV. These fields are destination ID address, destination sequence number, valid destination sequence number flag, network interface, hop count, next hop, other state and routing flag, list of precursors and lifetime. The other thing AODV had to manage very carefully is the sequence number. The broken links needs to be marked invalid if it's broken so that the problem of routing loops can be avoided.

2.2 DSDV

DSDV [9] serves for networks not centralized to any server or base stations. The mobile hosts participating in the formation of a network can enter and leave any time without harming the integrity of the protocol. This particular nature makes it fit for ad-hoc networks. DSDV effectively removes the looping problem by assigning sequence numbers to each root. DSDV uses distance vector algorithm for finding the next hop in the network as compared to link-state method. DSDV treats each mobile host participating in network as a specialized router which provides a mechanism for mobile computer to exchange data along the dynamically changing topology of network. The routing information is disseminated either by broad casting or multicasting the updates periodically or incrementally. This allows a new mobile host to enter and leave the network anytime without hampering the performance of network. The data packets are used for information sharing which follows the routing tables stored at each mobile host. DSDV requires each mobile host to maintain the routing tables and advertise them periodically and incrementally. The routing table contains a list of all available destinations and number of hops to reach each of them, this hop count is termed as metric. Each route table entry is tagged with a sequence number generated by the destination. Each root has an associated metric and a sequence number. The selection of a new route depends on both of these parameters. The consistency of the routing tables is an important issue and DSDV removes it by periodically broadcasting or multicasting the routing information. Each mobile host broadcast the route table changes as it finds the information to be significant enough to be disseminated. The mobile hosts involved in transfer of data with each other broadcast the necessary information periodically say once every few seconds. Upon the reception of new information, by a mobile host, it is compared with the routing table entries. Comparing the freshness of the sequences numbers associated with the routing table entries and the new information received recently, if the sequence number is newer than already saved ones that route table entry is replaced. Otherwise the new information is discarded. Mobile hosts which cause broken links as they move around in the network and can be detected if no information is received from them after a certain time interval. A broken route is described by infinity metric. When a link to a next hop is broken, any route through that next hop is assigned infinity metric and an updated sequence number. In normal situation the sequence number is generated by the respective mobile host but to describe the broken link it is generated by the mobile host which detects the broken link. In this case the sequence number generated is odd as compared to even sequence numbers in normal cases. Each mobile host can advertise its view of the network anytime asynchronously. For a large population of mobile hosts there should be some adjustments be made in time between broadcasts of routing information packets. DSDV proposes two types: One carries all the available information called" full dump" and the second type carries only the information changed since the last full dump called as "incremental". By design the incremental should fit in one network protocol data unit (NPDU) and a full dump require multiple NPDUs. The full dump is transmitted relatively infrequently as the mobile hosts remain stationary. When the movement is frequent and the size of an incremental becomes equal to the size of NPDU then a full dump can be scheduled.

2.3 ADV

ADV [10] is based on the distance vector algorithm which uses sequence numbers. The Long Lived Loop problem can be solved by the use of sequence numbers. Apart from this it helps the originator nodes to keep track of the freshest route and help in reducing the overhead required in order to organize the table. This overhead reduction is one of the major features of this algorithm. ADV adaptively reduces the overhead in the route table maintenance by either varying the updating frequency or the route table size in accordance with the traffic and the mobility speed. Maintenance of routing table: ADV uses control packets to maintain the route table at each node. These packets are broadcast and can be categorized into two processes that are connection oriented processes and other being the connection termination processes. These processes are helpful in the dynamic maintenance of routes that correspond to the active connections available. These processes are initiated either at the source node or the destination nodes. In ADV an update is not triggered whenever there is a change in the metric for a routing entry. It may happen either when a need is advertized by the neighbor or it's a request for forwarding nodes to keep the routs fresh. This implies that in ADV the updation is neighbor dependent and obtaining a valid route is a time taking process. The other case may be when there is a request for a fresh route to an active receiver. In ADV these routes are obtained from neighbors as compared to the other on-demand protocol where intermediate nodes rebroadcast for fresh routes. In ADV the updates at a node are triggered under three conditions:

1) If the node has some buffered data due to lack of route

2) If the fresh routes are requested by one or more neighbors

3) Forwarding nodes for updating valid and invalid routes

for keeping them fresh and updated. These updates can be processed as they are indicated or as they occur. But to reduce the overhead these updates are weighted. No update is triggered until sufficient need is met. So when ever any of these conditions is met a trigger update will be generated leading to high overhead. So in order to reduce the overhead the nodes are kept in wait. In the mean while the triggers are quantified for all the events that require an update. For this purpose a variable is used that is incremented every time any of the above conditions is met. This variable is called "Trigger Meter". Four constants are associated with the 'Trigger Meter': TRGMETER FULL,TRGMETER HIGH, TRGMETER MED and TRGMETERLOW. Three of these define the priority level of the node with which it should trigger an update while the fourth one which is TRGMETER FULL; is used to generate a full update. This happens through a check that is preformed every time an increment is made. If level of the trigger meter is exceeding the TRGMETER FULL value it implies a full update. A partial update is generated when the trigger meter crosses a threshold value. After an update has been generated the trigger meter is set to zero. The trigger threshold is a dynamic variable whose value is set based on the history of the trigger meter values for partial updates thus the adaptive nature. Other than this the trigger meter is also used to define time bar for any two consecutive updates by a node. This value is optimized in order to reduce the overhead. Usually it's at least 500ms.This implies whenever a threshold value is crossed and the minimum time has not lapsed no update will be triggered. This time factor is not applicable with full updates.

3. EXPERIMENTAL SETUP

As NCTUns is open source and Linux based, the present work is carried out in NCTUns 6.0 and Fedora 12. It can work with lower versions as well but for that some patches have to be installed first. Each protocol is evaluated against the above mentioned three performance metrics. The results are plotted against varying number of network nodes with a multiple of five nodes starting from five to forty five. The nodes participating in the simulation are wireless Adhoc nodes with IEEE 802.11(b) standard. The traversal path for each node is generated randomly. The simulations are carried out for 400 steps of simulation time. To have realistic results each protocol is simulated 8 times with number of nodes varying from 5 to 45, having a total of 24 simulations. The packet size is fixed. Initially the nodes are placed at some positions and then they move according to the randomly generated traversal path by the simulator. The moving speed of each node is constant throughout the simulations. Identical mobility and traffic scenarios are used for fairness purposes across different simulations. The simulation parameters are summarized in Table 1 as follows:

Table 1. Simulation Parameters

Parameter	Value
Simulation time	5 mins
PHY-MODEL	802.11b
Number of Nodes	5 to 45
Node Movement	Random
Channel Frequency	2.4 GHz

4. RESULTS AND DISCUSSION

In this section the results for variations in performance metrics with number of nodes is presented.

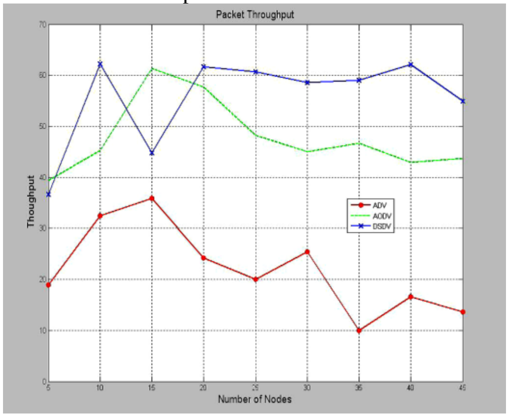

Fig 1: Diagram for packet throughput

Figure 1 is the plot showing results of simulations for throughput variation with increase in the number of nodes. The DSDV outperforms in terms of throughput. The throughput decreases with number of network traffic in case of ADV.

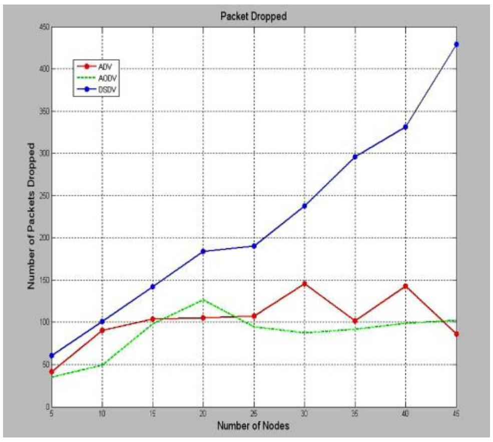

Fig 2: Diagram of packet dropped

Figure 2 is the plot showing results of simulations for packet dropped. As can be seen from the figure that the number of packet dropped increases with network traffic. The AODV and ADV behave similarly as compared to DSDV**.**

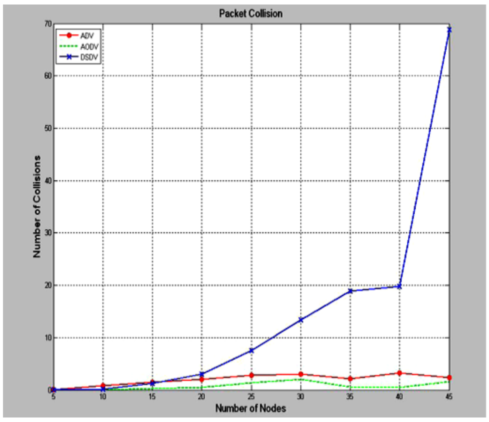

Fig 3: Diagram for packet collision

Figure 3 is the plot showing results of simulations for packet collision, due to increase in packet collision the packet dropped factor also increase and is evident from the plot.

5. CONCLUSION

In this paper we investigated the performance evaluation of AODV, DSDV and ADV routing protocols. Our results show that the protocols are quite capable to handle the dynamic requirements of VANETS. NCTUns is used for providing the scenario and experimental statistic. As it is evident from the results that packet throughput decreases as the network traffic increases and the conversely the packet dropped count and packet collision count increases with the network traffic.

6. ACKNOWLEDGMENT

The authors would like to thank the guidance of their teachers and their colleagues who inspire and facilitates them for the accomplishment of this task.

7. REFERENCES

[1] Abdul Hadi Abd Rahman, Zuriati Ahmad Zukarnain, “Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks”. European Journal of Scientific Research, ISSN 1450-216X Vol.31 No.4 (2009), pp.566-576 European Journals Publishing, Inc.2009http://www.eurojournals.com/ejsr.htm.

[2] Naga .V. Yedida, Rajesh Reddy Challa, “Performance Comparison of AODV, DSR and OLSR Routing Protocols in Static Scenarios”, Center for Advanced Computer Sciences. E-mail: nxy4835,rxc2763@louisiana.edu.

[3] Md. Anisur Rahman, Md. Shohidul Islam and Alex Talevski, “Performance Measurement of Various Routing Protocols in Ad-hoc Network”, Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol. I IMECS 2009, March 18 - 20, 2009, Hong Kong.

[4] Ioanna Stamouli, “Real-time Intrusion Detection for Ad hoc Networks”, A dissertation submitted to the University of Dublin, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science.

[5] M. Saravan akarthikeyan, K. Angayarkanni and Dr. S. Sujatha, Member IAENG, “Throughput Enhancement in Scalable MANETs using Proactive and Reactive Routing Protocols”.

[6] Aamir Hassan, “VANET Simulation”, Technical report, IDE0948, May2009, Master’s Thesis in Electrical Engineering.

[7] Thomas D. Dyer, Rajendra V. Boppana , “A Comparison of TCP Performance over Three Routing Protocols for Mobile Ad Hoc Networks”, Computer Science Division, The Univ. of Texas MobiHOC 2001, Long Beach, CA, USA ACM 2001 1-58113-390-1/01/10.

[8] Shie-Yuan Wang, Chih-Che Lin, “NCTUns 5.0: A Network Simulator for IEEE 802.11(p) and 1609 Wireless Vehicular Network Researches”, Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan.

[9] Charles E. Perkins, Pravin Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers”, SIGCOMM 94-8/94 London England UK 1994 ACM 0-89791 -682-4/94/0008.

[10] Rajendra V. Boppana, Satyadeva P Konduru, “An Adaptive Distance Vector Routing Algorithm for Mobile, Ad Hoc Networks”, IEEE INFOCOM 2001.

[11] INTER-VEHICLE COMMUNICATION SYSTEMS: A SURVEY.

[12] Valarmathi and R. M. Chandrasekaran, “Effect of Mobility and Traffic flow on DSR protocol in a predefined Mobile Ad Hoc Network”, IJCSNS International Journal of Computer Science and Network S 10 security,VOL.9 No.4, April 2009.

[13] Guoyou He, “Destination-Sequenced Distance Vector (DSDV) Protocol”, Networking Laboratory, Helsinki University of Technology, ghe@cc.hut.fi.

[14] C. Perkins, “Ad hoc On-Demand Distance Vector (AODV) Routing”, University of California, Santa Barbara, S. Das, University of CincinnatiJuly 2003.