International
Journal
Of Advanced Research in Engineering & Management (IJAREM)

ANALYSIS OF QOS ISSUES OF MODIFIED DSR

[1]Zaiba ishrat, [2]Dr. Pankaj Singh, [3]Mohd. Shadab
[1]Assistant Professor, IImt Gr.Noida
[2]Professor, Accurate,Greater Noida
[3]Sr. Faculty, NIELIT Chandigarh

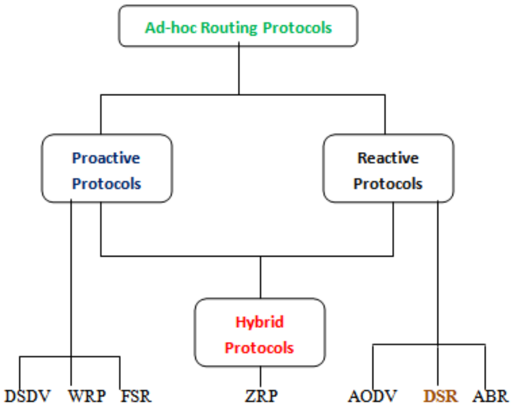
Abstract— Ad hoc networks (MANETs) are infrastructure less, autonomous networks comprised of wireless mobile computing devices. MANETs [1] are peer to peer networks in which all the nodes in the network have the same capability and communicate with each other without the intervention or need of a centralized access point or base-station.. Several routing protocols have been proposed for mobile ad hoc networks. These can be categorized as proactive (also known as tabledriven) protocols, reactive (known as source initiated or demand-driven) protocols or the hybrid of the reactive and proactive protocols[1,2]. .Dynamic Source Routing which is a reactive routing protocol adapts quickly to routing changes when host movement is frequent yet requires little or no overhead during periods in which hosts move less frequently. The DSR is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes.[3]This paper compare the performance of DSR with modified DSR.A modified DSR use the path ranking technique to improve its performance. The modified DSR is simulated on Ns2 and compared with the existingDSR protocol.
Keyword: DSR,MDSR,NS2

I. Introduction

Mobile ad hoc network is a collection of mobile nodes communicating through wireless channels without any existing network infrastructure or centralized administration.[4] Due of the limited transmission range of wireless network, multiple "hops" are needed to exchange data across the network. Routing protocols used in ad hoc networks must automatically adjust to environments that can vary between the extremes of high mobility with low bandwidth, and low mobility with high bandwidth[1,4].

The basic routing problem is that of finding an ordered series of intermediate nodes that can transport a packet across a network from its source to its destination by forwarding the packet along this series of intermediate nodes

The challenge in creating a routing protocol for ad hoc networks is to design a single protocol that can adapt to the wide variety of conditions that can be present in any ad hoc network over time. The routing protocol must perform efficiently in environments in which nodes are stationary and bandwidth is not a limiting factor. Yet, the same protocol must still function efficiently when the bandwidth available between nodes is low and the level of mobility and topology change is high.[2] Because the environment can change unpredictably, the routing protocol must be able to adapt automatically. Several routing protocols have been proposed for mobile ad hoc networks. [1,2]These can be categorized as proactive (also known as table driven) protocols, reactive (known as source initiated or demand-driven) protocols or the hybrid of the reactive and proactive protocols. A categorization of the prominent ad hoc routing protocols is shown in Fig1.



DSDV : Destination Sequence Distance Vector
WRP : Wireless Routing Protocol
FSR : Fisheye State Routing
ZRP : Zone Routing Protocol
ADOV : Ad hoc On Demand Distance Vector
DSR : Dynamic Source Routing
ABR : Associativity Based Routing

Fig 1. Categorization of ad hoc routing protocols

This paper is organized as follow: Section I gives the Introduction of the Routing protocols. Section II is helpful to understand the background of DSR Section III explains basic operations of DSR. Section IV modified DSR using path ranking technique in detail and the last section V gives the result that will compare the performance of DSR with MDSR using NS2 and the last section of the paper is conclusion and followed by the references.

II. DSR Protocol Working

The *Dynamic Source Routing* protocol (DSR) [Johnson 1994, Johnson 1996a, Broch 1999a] is a simple and efficient routing protocol designed specifically for use in multi-hop wireless ad hoc networks of mobile nodes. DSR is reactive protocol. Reactive protocol discovers route only when you need it means it works on traffic demand. Therefore it saves energy and bandwidth during inactivity. The Dynamic Source Routing protocol (DSR) is based on source routing, which means that the originator of each packet determines an ordered list of nodes through which the packet must pass while traveling to the destination.[3] The key advantage of a source routing design is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets that they forward, since the packet's source has already made all of the routing decisions. The DSR protocol consists of two basic mechanisms: Route Discovery and Route Maintenance. Route Discovery is the mechanism by which a sender node wishing to send a packet to a destination D obtains a source route to D.[5]

Route Discovery, in which a node S is attempting to discover a route to node D. To initiate the Route Discovery, S transmits a ROUTE REQUEST message as a single local broadcast packet, which is received by (approximately) all nodes currently within wireless transmission range of S. Each ROUTE REQUEST message identifies the initiator and target of the Route Discovery, and also contains a unique *request id*, determined by the initiator of the REQUEST. Each ROUTE REQUEST also contains a record listing the address of each intermediate node through which this particular copy of the ROUTE REQUEST message has been forwarded.[5,6] This route record is initialized to an empty list by the initiator of the Route Discovery.

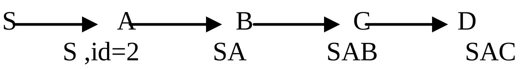


Fig 2: Route Discovery example

When another node receives a ROUTE REQUEST, if it is the target of the Route Discovery, it returns

ROUTE REPLY message to the initiator of the Route Discovery, giving a copy of the accumulated route record from the ROUTE REQUEST; when the initiator receives this ROUTE REPLY, it caches this route in its Route Cache for use in sending subsequent packets to this destination. Otherwise, if this node receiving the ROUTE REQUEST has recently seen another ROUTE REQUEST message from this initiator bearing this same request id, or if it finds that its own address is already listed in the route record in the ROUTE REQUEST message, it discards the REQUEST. Otherwise, this node appends its own address to the route record in the ROUTE REQUEST message and propagates it by transmitting it as a local broadcast packet (with the same request id).[3]

Route Maintenance: When sending or forwarding a packet to some destination D, Route Maintenance is used to detect if the network topology has changed such that the route used by this packet has broken. Each node along the route, when transmitting the packet to the next hop, is responsible for detecting if its link to the next hop has broken.[3,6] For example, in the situation illustrated in Figure 3, node S has originated a packet for D using a source route through intermediate nodes A, B, and C. In this case, node S is responsible for receipt of the packet at A ,node A is responsible for receipt at B, node B is responsible for receipt at C, and node C is responsible for receipt finally at the destination D. If the packet is retransmitted by some hop the maximum number of times and no receipt confirmation is received, this node returns a ROUTE ERROR message to the original sender of the packet, identifying the link over which the packet could not be forwarded. For example, in Figure 3, if B is unable to deliver the packet to the next hop C, then B returns a ROUTE ERROR to S, stating that the link from B to C is currently "broken.".Therefore Node S removes this broken link from its cache and retransmitted the packet using another route.if no route is available in cache than use route discovery method.[7]

S ⟶ A ⟶ B   C ⟶ D

Fig. 3 : Route maintenance

Route Cache: The optimization of DSR is Route cache. Each node caches a new route it learns by any means.e.g. When node S finds route [S, E, F, J, D] to node D, node S also learns route [S, E, F] to node F. When node K receives Route Request [S, C, G] destined for node, node K learns route [K, G, C, S] to node S. A node receiving a ROUTE REQUEST for which it is not the target, searches its own Route Cache for a route to the target of the REQUEST. If found, the node generally returns a ROUTE REPLY to the initiator itself rather than forwarding the ROUTE REQUEST. In the ROUTE REPLY, it sets the route record to list the sequence of hops over which this copy of the ROUTE REQUEST was forwarded to it, concatenated with its own idea of the route from itself to the target from its Route Cache.[5,7]
However, before transmitting a ROUTE REPLY packet that was generated using information from its Route Cache in this way, a node must verify that the resulting route being returned in the ROUTE REPLY, after this concatenation, contains no duplicate nodes listed in the route record.

III. MDSR

In modified DSR we don't use simple hop count as in DSR, a new path metric use which decide a route in Modified DSR (MDSR). To maintain the new routing metric a ranking is assigned to a node so that whole path can be ranked. So in case of multiple paths from source to destination, a path which has highest path ranking is chosen.

Algorithm to decide the path metric;
Each node assigns the rank for other node according to its packet dropping and successful transmission property. Path ranking is determined by taking average of the rankings of each node in the path as this allows choosing shortest path algorithm if no metric is given to nodes. In case of more than one path to the destination path will be chose which have high path metric

.**Algorithm for assigning rank to a node**
1. For a neutral node, that is a new node, is given a ranking of 0.5.
2. Ranking of each node is done with highest ranking of 1.0 to make sure that if all are neutral nodes then shortest path first is chosen.
3. For every 200ms the ranking of nodes on active path is incremented by 0.01.

4. Packet is dropped on a link and if a node becomes un-reachable to other nodes than its ranking is reduced by 0.05
5. Lower limit of a neutral node is assigned 0.0.
6. Changes on the rankings of other nodes than one mentioned above are not performed.
7. Any misbehaving node given a ranking of -100.
8. If the simulation is run for long period of time then the negative rankings can be reset after a long timeout period.
9. In case when no node is found that can be given packet to forward, Send Route Request is given.

Therefore, we will choose the path which is having highest path metric .

IV. Simulation Tool NS2

ns2 is the de facto standard for network simulation. Its behavior is highly trusted within the networking community. It is developed at ISI, California, and is supported by the DARPA and NSF. ns2 is an object oriented simulator, written in C++, with an OTcl interpreter as a frontend. This means that most of the simulation scripts are created in Tcl. If the components have to be developed for ns2, then both tcl and C++ have to be used. The simulator supports a class hierarchy in C++, and a similar class hierarchy within the OTcl interpreter[9]. The two hierarchies are closely related to each other; from the user's perspective, there is a one-to-one correspondence between a class in the interpreted hierarchy and one in the compiled hierarchy. The root of this hierarchy is the class TclObject. Users create new simulator objects through the interpreter; these objects are instantiated within the interpreter, and are closely mirrored by a corresponding object in the compiled hierarchy[10]. The interpreted class hierarchy is
automatically established through methods defined in the class TclClass. User instantiated objects are mirrored through methods defined in the class TclObject. There are other hierarchies in the C++ code and OTcl scripts; these other hierarchies are not mirrored in the manner of TclObject [8].

Performance metric: Two performance parameter is considered for comparing the performance of MDSR with DSR which are throughput and end to end delay.

1. Throughput: It is defined as the ratio of data packets received to the destination to those generated by source. Throughput is average rate of packets successfully transferred to their final destination per unit time.

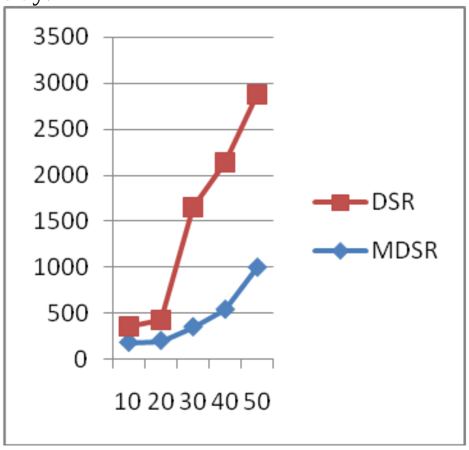
2. End-to-End Delay: It is the average delay time for a data packet travelling from its source to destination. It signifies the amount of time taken by packet from source to destination. The delay time of all successfully received packets is summed, and then the average delay time is calculated.

All the above mentioned performance metrics are quantitatively measured. For a good routing protocol, throughput should be high whereas end to end delay parameters value should be less. We used the above performance metrics and quantitatively measured against number of nodes.

V. RESULT

5.1 End To End Delay

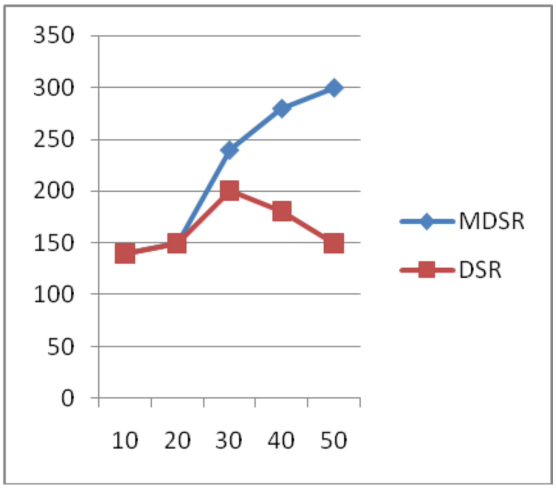
It is shown from the graph that average end-to-end delay is lower when number of nodes is lower and it will increase when number of node increases. It is clear from the graph that after doing the modification in DSR it is showing less average end-to-end delay.



NO. of Nodes vs delay

5.2 Throughput

It is shown by the graph that throughput is less when number of nodes is lower and it increase when number of node increases. It is clear from the graph that after doing the modification in DSR it is showing increased throughput as compared to existing DSR.



No. of nodes Vs. Throughput

VI. CONCLUSION

In this paper we use the concept of path ranking in DSR and shown that it has very good effect on the performance of existing DSR. Simulation results demonstrated in terms of throughput, end-to-end delay and against number of nodes shows that the modified DSR performs lot better as compared to existing DSR. In future work we will compare the MDSR with other existing protocol in MANET.

References

[1]. Chai-Keong Toh. Ad Hoc Mobile Wireless Networks. Prentice Hall PTR, Upper Saddle River, New Jersey, 2002.

[2]. D. Kim, J. Garcia and K. Obraczka, "Routing Mechanisms for Mobile Ad Hoc Networks based on the Energy Drain Rate", IEEE Transactions on Mobile Computing. Vol 2, no 2, 2003, pp.51-173.

[3]. David B. Johnson and David A. Maltz. "Dynamic Source Routing in Ad Hoc Wireless Networks". In Ad Hoc Wireless Networks, Mobile Computing, T. Imielinski and H. Korth (Eds.), Chapter 5, pp 153-181, Kluwer Academic Publishers, 1996.

[4]. Andreas Tønnesen. "Mobile Ad-Hoc Networks"

[5]. Yogesh Chaba, Yudhvir Singh, and Manish Joon, "Performance Evaluation and Analysis of Cluster Based Routing Protocols in MANETs", IEEE 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, pp 64-66.

[6]. Yudhvir Singh, Yogesh Chaba, Monika Jain, and Prabha Rani, "Performance Evaluation of On-Demand Multicasting Routing Protocols in Mobile Adhoc Networks", IEEE 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, pp 298-301.

[7]. *David B. Johnson David A. Maltz Josh Broch, "DSR"* http://www.monarch.cs.cmu.edu/

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

[9]. Giovanni Perbellini. "An introduction to NS-2", Università degli Studi di Verona, Facoltà di Scienze MM. FF. NN., Verona, 12/09/2005

[10]. Jaroslaw Malek. "Trace graph - Network Simulator NS-2 trace files analyzer" http://www.tracegraph.com

[11]. Elizabeth M. Royer and C-K Toh. "A Review of current Routing Protocols for Ad-hoc Mobile Wireless Networks", IEEE Personal Communications, Vol. 6, No.2, pp. 46-55, April 1999.

[12]. Tony Larsson amd Nisklas Hedman. "Routing Protocols in Wireless Ad-hoc Networks: A Simulation Study"