Performance Comparison Of Ad Hoc Wireless Network Routing Protocols

Humaira Ehsan[1] and Zartash Afzal Uzmi[2]
[1]*National University of Computer and Emerging Sciences, Islamabad, Pakistan.*
[2]*Lahore University of Management Sciences, Lahore, Pakistan.*
humaira_79@yahoo.com , zartash@lums.edu.pk

Abstract

In recent years many protocols for ad hoc wireless networks have been developed but very little information is available on the performance of these protocols. After describing the desirable characteristics of an ad hoc network routing protocol this paper provides an overview of four existing ad hoc wireless routing protocols, which are Distance sequence distance vector (DSDV), Ad hoc on demand distance vector (AODV), Dynamic source routing (DSR) and Temporally ordered routing algorithm (TORA). The performance comparison of these four protocols is based on simulations performed using network simulator-2. The performance parameters analyzed are the mobility rate, network load and network size. The paper describes all the parameters used for the simulations in detail and then compares each routing protocol's simulation results before arriving at a conclusion as to which is the best one for ad hoc networks.

1. Introduction

Wireless networks emerged in the 1970's, since then they have become increasingly popular. The reason of their popularity is that they provide access to information regardless of the geographical location of the user. Wireless networks can be classified into two types i.e. infrastructured and infrastructureless networks.

Infrastructured wireless networks, also known as cellular networks, have fixed base satiations which are connected to other base stations through links. Mobile nodes communicate with one another through these base stations.

Infrastructureless wireless networks, also known as ad hoc wireless networks, are a collection of wireless mobile nodes that does not have any predefined infrastructure or centralized control such as base stations. Ad hoc wireless networks are different from other networks because of following characteristics: absence of centralized control, each node has wireless interface, nodes can move around freely which results in frequent changes in network topology, nodes have limited amount of resources and lack of symmetrical links i.e. transmission does not usually perform equally well in both directions.

2. Routing in Ad Hoc Wireless Networks

In wired networks in order to obtain the shortest path usually Distance Vector or Link state routing protocols are used. These protocols do not perform well in ad hoc wireless networks because wireless networks have limited bandwidth and there is no central control. Therefore, modifications to these routing protocols or totally new routing protocols are required for the ad hoc wireless domain. To perform well, routing protocols for ad hoc wireless networks should address the following issues:

- *Finding an optimal route:* The protocol should find an optimal route based on the optimality metric chosen. The metric for deciding optimality can be hop count, delay, bandwidth, load or reliability, etc.
- *Energy efficient:* Many nodes in ad hoc wireless networks have limited battery power so they need to use energy optimally. The protocol should have minimum possible processing and transmission requirements.
- *Bandwidth efficient:* In these networks as each node has a limited amount of energy and bandwidth so the protocol should aim to generate minimum possible traffic. This can be done by reducing the number of periodic updates.
- *Convergence:* Whenever the topology of the network changes, the protocol should be able to converge to a stable state in a reasonably small amount of time.
- *Loop free:* Loops are formed in the network when some packets remain in the network for some time without reaching their destination. A good routing

protocol should ensure that it is loop free because loops waste a lot of bandwidth and packets in loops may never reach their destination.

- *Route Failure and recovery:* Some mechanism should be defined to discover route failures and propagate that information. The new configuration should converge fast to a stable state.
- *Stale routes handling:* As the nodes of the networks are mobile, routes become stale as a result of node movement. The protocol should describe a mechanism to handle stale routes.
- *Nodes movement speed:* The protocol should perform well for all speeds and types of node movements.
- *Partitioned Networks:* As a consequence of the free movement of nodes, it is possible that few nodes become isolated from rest of the nodes that is they get out of the transmission range of the remaining nodes. In this manner, different groups of nodes will be formed and such a scenario is characterized as a partitioned network. The routing protocol should address this issue and suggest some appropriate strategy to handle this scenario.
- *Nonsymmetrical Links :* As the links in wireless networks are normally not symmetric i.e. bidirectional, therefore the protocol should accommodate unidirectional links. There are many existing protocols which assume that the links are symmetric and their functionality and performance is severely affected assuming a unidirectional scenario.
- *Multiple Routes:* A routing protocol should be able to provide multiple routes for a single source-destination pair in order to reduce congestion on a particular route. Multiple routes are more useful in mobile networks because in these networks the frequent movement of nodes causes a lot of link failures. If alternate route are available, they can reduce delay and improve the packet delivery ratio.
- *Load Balancing:* The protocol should not overload one node and should be designed to keep the load even on all nodes. This will also help in avoiding the occurrence congestion near certain nodes.
- *Scalability:* The performance of the protocol should not be effected by increasing or decreasing the number of nodes in the network.
- *Sleep function:* As nodes in an ad hoc wireless network are energy constrained therefore some nodes may decide to go to sleep (inactive mode) for some random period of time. The protocol should be able to handle such nodes without causing any effect on the rest of the network.

The following parameters can be used to compare the performance of various routing protocols [6, 9]:

1. Packet delivery fraction: Throughput
2. Average end to end delay of data packets
3. Rate of out of order delivery
4. Path Optimality
5. Routing traffic generated or Routing Overhead
6. MAC layer traffic generated

The following parameters can be varied while benchmarking the performance of routing protocols [10]:

1. Network size-measured in the number of nodes
2. Network connectivity: Average number of nodes connected to a single node.
3. Link capacity: Link speed measured in bits/second
4. Fraction of unidirectional links:
5. Offered load: Traffic pattern
6. Mobility
7. Fraction of sleeping nodes

There are three main categories of routing protocols for ad hoc wireless networks which are:

1. Table driven routing protocols
2. On demand routing protocols
3. Hybrid routing protocols

We will discuss only first two categories in this paper.

3. Table Driven Protocols

These protocols are also called proactive protocols. These protocols find routes between all source-destination pairs in the network and maintain the latest routes information by sending periodic route update messages. The updates are sent even if no change in topology has occurred. In this category, protocols have been developed by modifying the distance vector and link state algorithms. Protocols store routing information into various routing tables. Because of periodic updates these protocols converge very slowly and generate a lot of routing overhead that is why they are not very suitable for ad hoc wireless networks. Some of the existing table driven ad hoc routing protocols are:

1. Destination Sequenced Distance Vector
2. Clustered Gateway Switch Routing
3. Wireless Routing Protocol

3.1. Destination Sequences Distance Vector

This protocol is an adaptation of the Routing Information Protocol (RIP). It adds a sequence number to the RIP routing table. This sequence number field is used to differentiate between stale and fresh routes [12].

Each node maintains a routing table which contains next hop information for all reachable destinations. Each entry of the routing table consists of destination address, the number of hops required to reach the destination and the sequence number received from that destination. The sequence number associated with each route is used to determine the freshness of a route. Whenever a node receives new information about a particular route it compares sequence numbers and the one with the greatest sequence number is kept while the other one is discarded. If it receives two updates with the same sequence number then the one with lower number of hops is used.

The routing table is updated by periodic advertisements or whenever new information is available. Nodes send two types of updates i.e., full dump or incremental updates. Full dumps are sent periodically while incremental updates are event driven i.e., whenever some route changes its update is sent to the neighbors. In full dumps the whole routing table is sent while in incremental updates just the latest updated information is sent.

The performance of the protocol critically depends on the periodic update interval value. If this value is very small then there will be a very large routing overhead because of the full dumps and incremental updates and if this value is very small then there will be delays in getting the latest route information. This protocol is highly unfavorable for networks which have high mobility and a large number of nodes.

4. On Demand Protocols

Protocols in this category do not maintain the valid routes all the time. Routes are discovered only when they are required that is why these protocols are called on demand routing protocols. A few existing on-demand routing protocols are:

1. Dynamic Source Routing
2. On-Demand Distance Vector Routing
3. Temporally Ordered Routing Algorithm

4.1. Dynamic Source Routing

In source routing the sender determines the entire path through which the packet should travel and then it explicitly appends that path in the packet header. Source routing can be dynamic or static. This protocol uses dynamic source routing.

Each node maintains a route cache, entries in the route cache contain complete paths to different nodes. The route is determined either by making a hit in the cache or by a route discovery process. When a source node needs to send a packet to another node it first checks its cache. If an entry for that particular destination is present in the cache then it is used directly otherwise a route discovery process is initiated and this process continues recursively until the complete path to the destination is computed. Once a route is known it is then appended to the packet header and the packet is forwarded along that particular route [5].

In route discovery process the sender generates a route request packet. The route request packet contains a route record in which the sequence of next hop information is stored along with a unique request id. The pair (source address, request id) uniquely identifies each route request packet. The source node broadcasts the route request packet. All the neighbors receive this request. If any of them has corresponding entry in its cache for the destination node it will send a route reply to the initiator with the complete route in it. If an entry is not present in its cache it will further broadcast the route request.

A route reply packet may be routed back to the source using the path listed in the route request packet or the node may also use some other route from its own cache. One of the improvements to the protocol suggests that the nodes should use exponential backoff while sending a route reply so that the node which has shortest path is able to send first. Nodes also work in promiscuous mode so that they can learn new routes from various route requests as well as reply and error packets which are not destined for them. But this promiscuous listening increases CPU overhead as a greater number of packets need to be processed.

This protocol does not send any periodic updates but still has routing overhead because of the fact that it embeds the whole route in every packet. This overhead increases with an increase in mobility of users and with bursty traffic.

4.2. Ad hoc on Demand Distance Vector

In AODV Each node maintains a routing table but unlike the DSDV protocol it does not necessarily contain route to all other nodes. It uses a broadcast route discovery method similar to dynamic source routing. Instead of source routing it dynamically creates entries in the routing tables of intermediate nodes.

Whenever a packet is generated for a particular node for which there is no entry in the routing table a route request message is broadcasted. Each neighboring node receives that packet and checks its own routing table. If there is no entry in the routing table this node also broadcast the packet and also records in its table the address of the node from which it received the route request packet. This entry is used in future for establishing the reverse path. These entries are kept in the routing table for a period of time in which the route request packet can propagate through the whole network and produce a route reply packet. The request message is forwarded until it reaches some node which has a fresh entry for the destination in its routing table or it reaches the destination. That final node then sends a route reply packet. The entries in the routing tables of the intermediate nodes form the reverse path.

The route reply packet travels along the reverse path. Each node which receives the route reply packet sets a forward pointer to the node from which that packet was received. In this way a forward path is created from the source to the destination on which data packets travel later on.

This protocol assumes that all links are symmetric i.e. there are no unidirectional links. Based on this assumption it uses the same path to send a reply message. It uses sequence numbers to determine which routes are fresh and which are stale [2].

This protocol uses a periodic hello messages to determine local connectivity. This mechanism is also used to determine link failures. The routing overhead for this protocol is not as much as that for DSDV but it increases with an increase in the number of nodes. The protocol finds multiple routes between a source and destination pair. This avoids overhead of performing a new route discovery if one of the links on one of the routes fail and also allows the user to select and control routes for load balancing or any such tasks. The route cache is very useful in low mobility scenarios but for high mobility when links are changing very quickly this cache can become an overhead.

4.3. Temporally Ordered Routing Algorithm

TORA is designed to minimize reaction to topological changes. A key concept in its design is that it decouples the generation of potentially far-reaching control message propagation from the rate of topological changes [7].

The basic functionality of the protocol consists of: creating routes, maintaining routes and erasing routes. The protocol models the network as a graph; initially all the edges in the graph i.e., links in the network are undirected. Each link can be undirected or directed from node i to node j or directed from node j to node i. Each node maintains a metric "height". This metric is used in assigning directions to links with each neighbor.

Routes can be created in reactive or proactive mode. Reactive mode route creation requires establishing a series of directed links from the source to the destination node. This is done by constructing a directed acyclic graph rooted at the destination using a query/reply process. When a route is required the source broadcasts a QRY (query) packet to its neighbors. The QRY packet is propagated until it is received by one or more routers that have a route to the destination. The router that has a route to the destination sends an UPD (update) packet to all its neighbors. The node which receives a UPD packet sets its height one greater then the height of the node from which it received the UPD packet. In a proactive mode the destination initiates route creation by sending a OPT (optimization) packet, which is then processed by the neighbors and forwarded further.

Route maintenance is performed only for routers that have a non null height. Routers with a null height are not used for computations. Reaction to link failure is initiated only when a node loses its last downstream link. The protocol is designed such that the number of nodes that participate in the failure reaction is minimum. No reaction is initiated to link activation.

5. Performance Comparison

For the purpose of a performance comparison detailed performance simulations are performed for four main ad hoc routing protocols i.e. DSR, AODV, DSDV and TORA. The simulations are done using ns-2. We have used three metrics i.e. normalized routing overhead, packet delivery fraction and average end to end delay in our simulations to measure performance.

- *Normalized routing overhead:* This is the number of routing packets transmitted per delivery of a data packet. Each hop transmission of a routing packet is counted as one transmission. This factor also tells us something about the scalability of the routing protocol. If routing overhead increases with the increase in mobility then that protocol is not scalable.
- *Packet delivery fraction:* It is the ratio of data packets received to packets sent. This also tells us about the number of packets dropped and throughput of the network.
- *Average end to end delay:* This is the difference between sending time of a packet and receiving time of a packet. This includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times[9].

5.1. Traffic Pattern

The traffic sources used in the simulations are continuous bit rate (CBR). "TCP sources are not used because it offers a conforming load to the network, meaning that it changes the time at which it sends packets based on its perception of the network's ability to carry packets. As a result, both the time at which each data packet is originated by its sender and the position of the node when sending the packet would differ between the protocols, preventing a direct comparison between them." [6]

For the simulations the sending rate is fixed to 4 packets per second and the number of CBR sources is varied. Varying CBR sources is equivalent to varying the sending rate.

Traffic is generated using the following parameters:

Traffic Type: CBR

No of nodes: 50

No of sources: 10, 20, 30 sources

Rate : 4 packets per second

5.2. Movement Model

The node movement generator of ns-2 is used to generate node movement scenarios. The parameters this movement generator takes as input are number of nodes, pause time, maximum speed, field configuration and simulation time. The parameter which is of primary importance is *pause time*. Pause time basically determines the mobility rate of the model, as pause time increases the mobility rate decreases.

At the start of the simulations nodes are assigned some random position within the specified field configuration, for pause time seconds nodes stay at that position and after that they make a random movement to some other position. The movement speed is uniformly distributed between 0 and *maximum speed*.

The following parameter values are used for generating various mobility models:

Number of nodes: 50 nodes

Pause times : 0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100 seconds,

Maximum speed: 20m/s

Field configuration: 1500x300

Simulation time: 100 seconds.

6. Performance Results

The simulation results are presented in this section in the form of line graphs. Graphs show comparison between the four protocols and between a different numbers of sources on the basis of the aforementioned metrics as a function of pause time.

6.1. Normalize Routing Overhead:

Graphs below show a comparison between all four protocols on the basis of normalized routing overhead using a different number of sources.

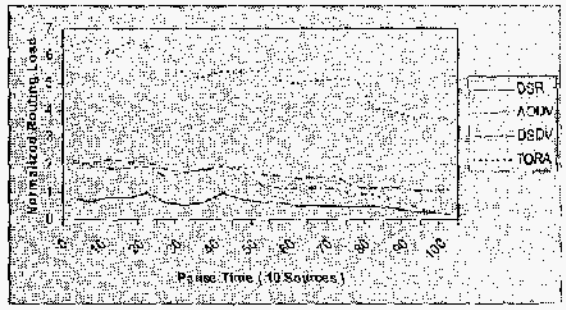


Figure 1: Normalized routing load of protocols for 10 sources

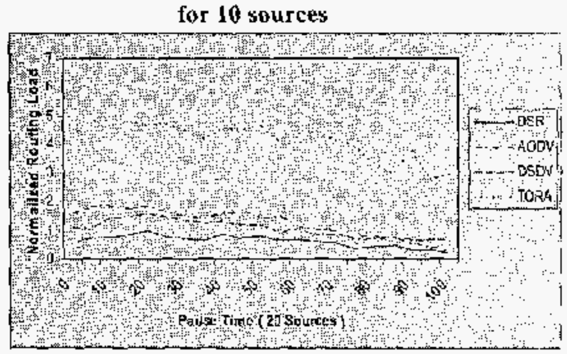


Figure 2: Normalized routing load of protocols for 20 sources

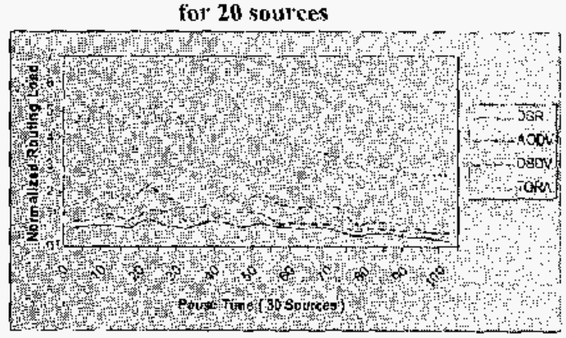


Figure 3: Normalized routing load of protocols for 30 sources

The routing overhead for TORA is most for all the cases because of the fact that it has overhead of both table driven and on-demand routing techniques.

As DSDV is a table driven routing protocol its overhead is almost the same with respect to node mobility.

When number of sources are 10 AODV performs better then DSDV but as number of sources increases AODV overhead becomes more then DSDV. This is because of the reason that AODV is on demand routing protocol so as the number of sources increases the number of routing packets also increases.

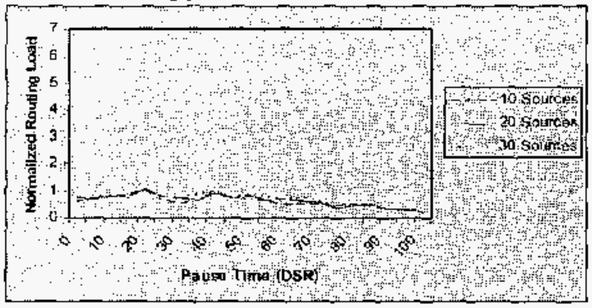


Figure 4: Normalized routing load of DSR with various number of sources

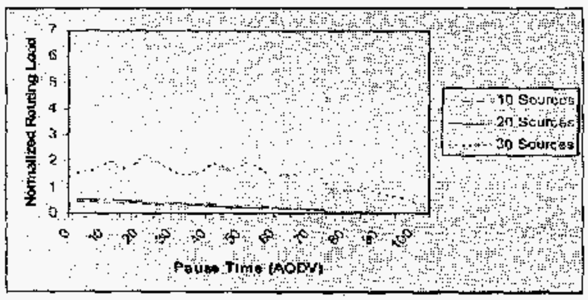


Figure 5: Normalized routing load of AODV with various number of sources

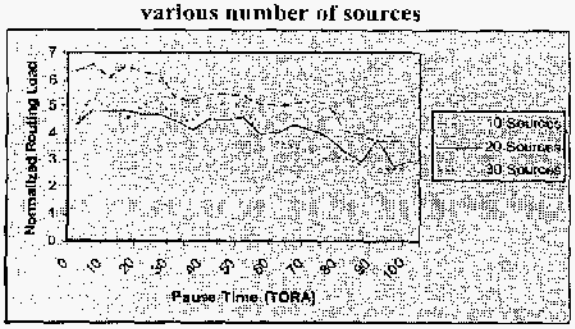


Figure 6: Normalized routing load of TORA with various number of sources

DSR's performance is best as it has the least overhead for all the cases. This is because DSR uses caching and it is more likely to find routes in its cache which result in lesser number of route discovery requests then other protocols.

For DSDV, DSR and AODV overhead decreases as mobility rate decreases and it converges to zero as pause time reaches 100 seconds i.e. mobility reaches zero.

6.2. Packet Delivery Fraction

Graphs 7-13 show a comparison between all four protocols on the basis of packet delivery fraction using a different number of sources.

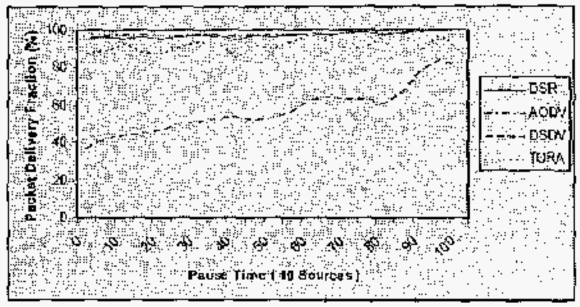


Figure 7: Packet delivery fraction of protocols for 10 sources

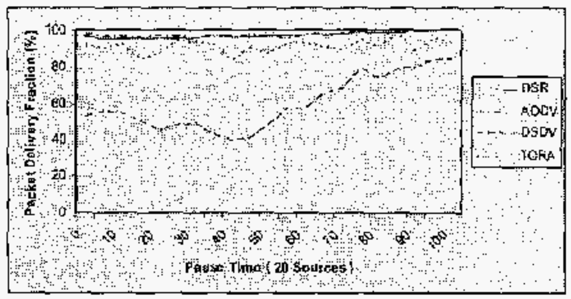


Figure 8: Packet delivery fraction of protocols for 20 sources

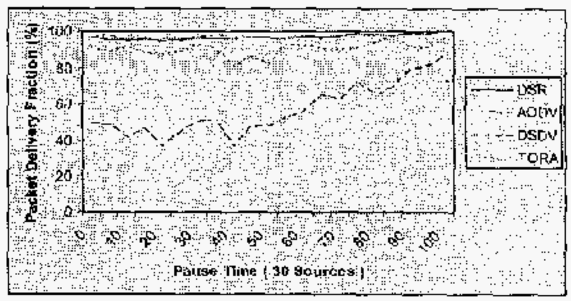


Figure 9: Packet delivery fraction of protocols for 30 sources

All on demand protocols converge to almost a 100% packet delivery fraction when the pause time reaches 100 i.e., there is no mobility. DSR and AODV performs the best; their packet delivery is almost independent of the number of sources as it is obvious from graphs 11-12 that varying number of sources does not effect DSR and AODV that much.

DSDV performance is worst when mobility is high. This poor performance is because of the reason that DSDV is not on demand and it keeps only one route per destination therefore lack of alternate routes and presence of stale routes in the routing table when nodes are moving at higher rate leads to packet drops.

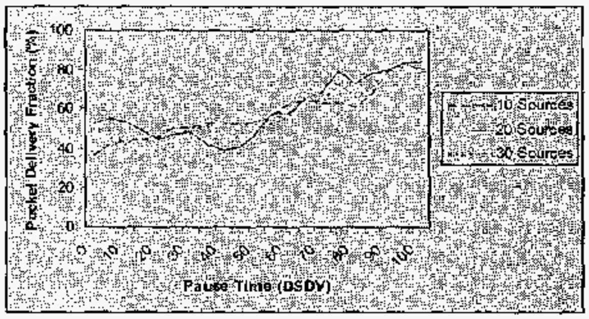


Figure 10: Packet delivery fraction of DSDV with various number of sources

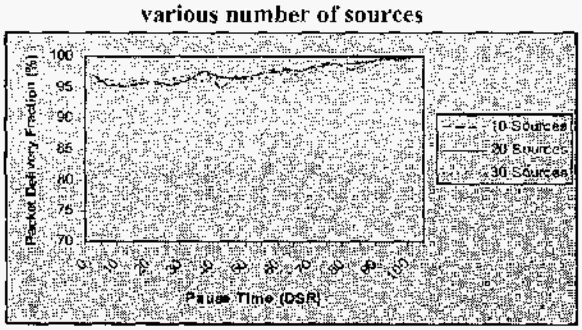


Figure 11: Packet delivery fraction of DSR with various number of sources

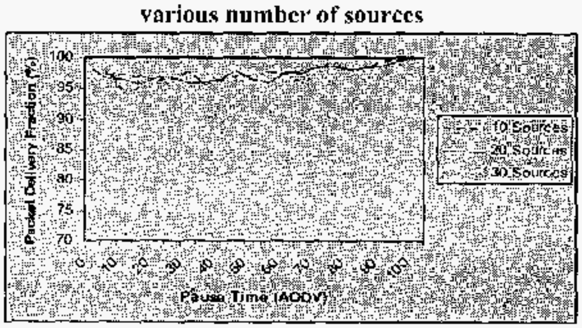


Figure 12: Packet delivery fraction of AODV with various number of sources

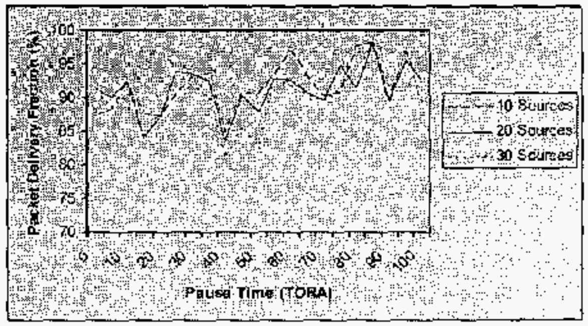


Figure 13: Packet delivery fraction of TORA with various number of sources

6.3. Average End to End Delay

Figure 14-17 shows comparison between all the four protocols on the basis of average end to end delay using a different number of sources.

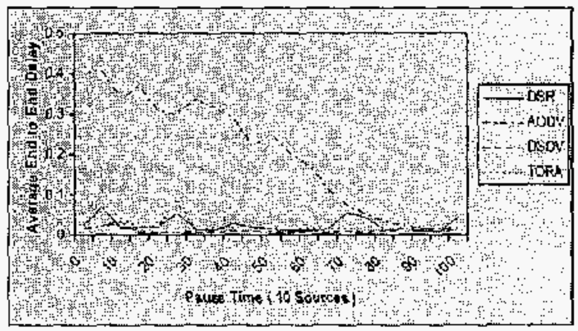


Figure 14: Average end to end delay of protocols for 10 sources

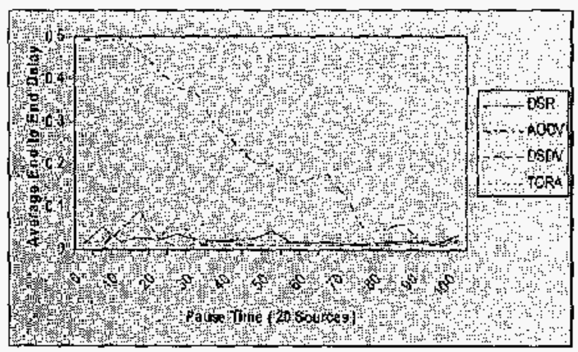


Figure 15: Average end to end delay of protocols for 20 sources

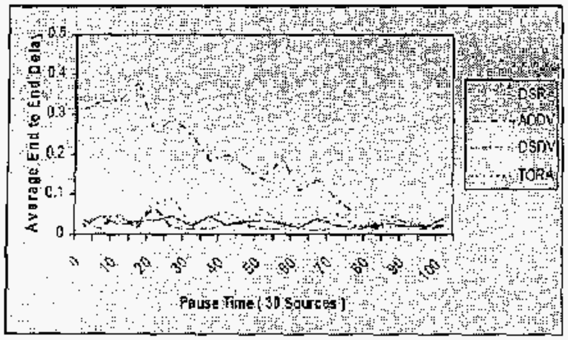
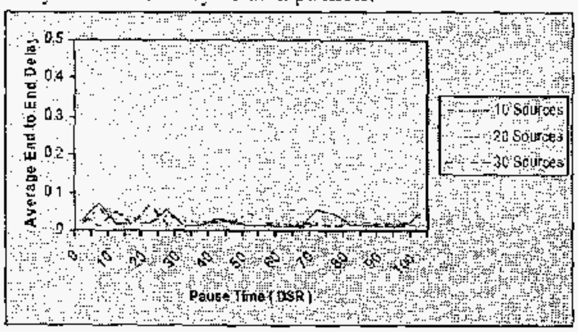
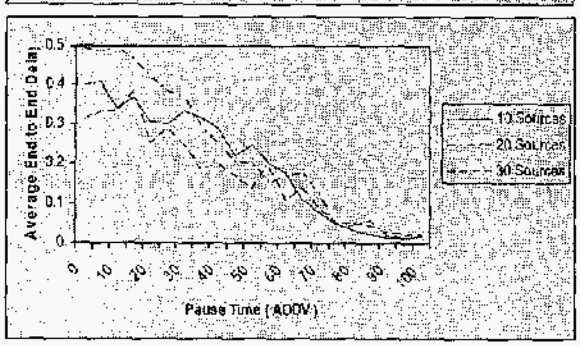


Figure 16: Average end to end delay of protocols for 30 sources

The average end to end delay for DSDV, DSR and TORA is fairly below 0.1 second for all cases.

For AODV the delay is much more then other protocols and it increases as the number of sources and mobility increases. As with an increased number of sources and high mobility there are more link failures therefore there are more route discoveries. AODV takes more time during the route discovery process as first it finds the route hop by hop and then it gets back to the source by back tracking that route. All this leads to delays in the delivery of data packets.





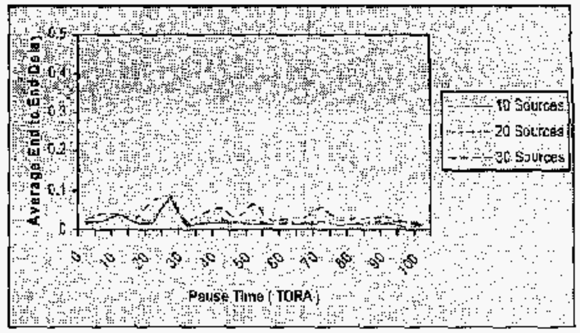


Figure 17: Average end to end delay of DSR AODV and TORA

7. Conclusion

This paper compared the four main ad hoc routing protocols. DSR, AODV and TORA are all on demand routing protocols which use different routing mechanism while DSDV is a table driven protocol.

Simulation results show that DSR outperforms all other protocols in all scenarios and for all performance metrics. DSR generates less routing load then AODV. AODV suffers from end to end delays while TORA has very high routing overhead. DSDV packet delivery fraction is very low for high mobility scenarios. The better performance of DSR is because it exploits caching aggressively and maintains multiple routes to destinations. This cache can become a problem if we increase the mobility and simulation time as then routes will be changing more frequently and cache will have stale routes mostly therefore in that case it will not help DSR in better performance.

8. References

[1] Andrea J. Goldsmith and Stephen B. Wicker, "Design Challenges For Energy Constrained Ad Hoc Wireless Networks," *IEEE Wireless Communication*, Aug. 2002, pp. 8-28.

[2] Charles E. Perkins, Elizabeth M. Royer, "Ad-Hoc on Demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. and Apps.*, Feb. 1999, pp. 90-100.

[3] David B. Johnson and Davis A. Maltz, "DSR in Ad-Hoc Multi Hop Network,".

[4] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks," *IEEE Personal Communication*, Apr. 1999, pp. 46-55.

[5] David B. Johnson and Davis A. Maltz, "Dynamic Source Routing in Ad Hoc Network," *T.Imielinski and H.Korth, Eds. Mobile Computing*, Ch. 5, Kluwer, 1996.

[6] J. Broch et al., "A Performance Comparison of Ad-Hoc MultiHop Wireless Networks Routing Protocols," *Proc. IEEE/ACM MOBICOM '98*, Oct. 1998, pp. 85-97.

[7] Vincent D. Park and M. Scott Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *Proc. INFOCOM '97*, Apr. 1997.

[8] David B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," *Proc. IEEE Wksp. Mobile Comp. and Apps.*, Dec. 1994, pp. 158-163.

[9] Charles E. Perkins and Elizabeth M. Royer,"Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks," *IEEE Personal Communication*, Feb. 2001.

[10] D. Chakeres and Elizabeth M. Royer, "The Utility of Hello Messages for Determining Link Connectivity".

[11] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *http://www.ietf.org/rfc-/rfc2501.txt , IETF Internet Draft*, Jan. 1999.

[12] Charles E. Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Comp. Communication Rev.*, Oct. 1994, pp.234-44.

[13] S. R. Das et al., "Comparative Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," *7th Int'l Conf. Comp. Communication and Networks*, Oct. 1998, pp. 153-161.

[14] P. Johansson et al., "Routing Protocols for Mobile Ad-hoc Networks – A Comparative Performance Analysis," *Proc. IEEE/ACM MOBNICOM '99*, Aug. 1999, pp. 195-206.

[15] Z. J. Hass, "A New Routing Protocol for the Reconfigurable Wireless Networks".

[16] Shree Murthy and J. J. Garcia-Luna-Aceves, "A Routing Protocol for Packet Radio Networks".

[17] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE Journal on Selected Areas in Commun., Special Issue on Wireless Ad Hoc Networks*, Aug. 1999, pp. 1395-1414.

[18] " The Network Simulator -2 ," http://www.isi.edu/nsnam/-ns/

[19] http://www.isi.edu/nsnam/ns/tutorial/