Performance Comparison and Analysis of Ad Hoc Routing Algorithms *

Azzedine Boukerche
Parallel Simulation and Distributed Systems (PARADISE) Research Laboratory
Dept. of Computer Science, University of North of Texas
Email : boukerche@cs.unt.edu

Abstract

A Mobile ad-hoc network (Manet) is a system of wireless mobile nodes dynamically self-organizing in arbitrary and temporary network topologies. People and vehicles can thus be internetworked in areas without a pre-existing communication infrastructure, or when the use of such infrastructure requires wireless extension. Therefore, such networks are designed to operate in widely varying environments, from military networks to low-power sensor networks and other embedded systems. Frequent topology changes caused by node mobility make routing in ad hoc wireless networks a challenging problem. In this paper, we focus upon routing protocols for ad hoc networks. We study and compare the performance of several routing protocols. A variety of workload and scenarios using a variety of mobility, load and size of the ad hoc network were simulated.

1 Introduction

Ad hoc wireless networks are composed of mobile stations communicating solely through wireless channels [14]. Such networks are expected to play an increasingly important role in future civilian and military setting, being useful for providing communication support where no fixed infrastructure exists or the deployment of a fixed infrastructure is not economically profitable, and movement of communicating parties is possible. Applications of ad hoc wireless networks include military operations (communication in a hostile environment), rescue operations (rapid deployment of a communication network where infrastructures do not exist or have been damaged), and sporadic happenings coverage (intense utilization of a communication network for a very limited time).

In ad hoc wireless networks, a message sent by a mobile may be received simultaneously by all the nodes in its vicinity, i.e., by all of its *neighbors*. Messages directed to mobiles not within the sender's transmission range must be forwarded by neighbors, which thus act as *routers*. Due to mobility, it is not possible to establish fixed paths for message delivery through the network. Therefore, a number of routing protocols have been proposed for ad hoc wireless networks [1], [5] [7], [6], [14] derived from *distance-vector* [12] or *link-state* [13] routing algorithms. Such protocols are classified as *proactive* or *reactive*, depending on whether they keep routes continuously updated, or whether they react on demand. In this paper, we propose to compare and study the performance of the following ad hoc routing protocols: AODV, CBRP and DSR using extensive simulation experiments.

The remainder of the paper is organized as follows. Section 2 briefly reviews the three on-demand routing protocols: AODV, CBRP and DSR. An improvement for further reducing their routing overhead are also discussed. In Section 3, we analyze the differences between these protocols that may affect their performance. Section 4 presents the simulation experiments we carried out to study and compare the performance of the three routing protocols, followed by the conclusions in Section 5.

2 Ad Hoc Routing Protocols

While the first two of the protocols we consider in this paper uses source routing, the third one avoids the routing workload at intermediate nodes, and is based on an improved distance-vector routing algorithm which is able to avoid routing loops.

2.1 Dynamic Source Routing (DSR)

The first protocol we studied is the *Dynamic Source Routing* (DSR) protocol [1]. With source routing, the sender of a packet determines the complete route from itself to the destination, and includes the route in the packet. All the intermediate hosts forward the packet based on this predetermined route (called source route). No routing decision is made at the intermediate hosts.

DSR offers a number of potential advantages for routing in ad hoc networks. First, a host dynamically

*This work was supported by UNT Faculty Research Grant.

0-7803-7001-5/01/$10.00 © 2001 IEEE

discovers a route only when it needs to send a packet through that route. There are no periodic routing messages. In addition, DSR only monitors the operations of the routes in use. Once there is a link failure in a route, the source (sender) of the route is notified immediately. As a result, DSR can quickly adapt to topological changes caused by node movement, which may often occur in a mobile wireless network. Furthermore, DSR is able to compute correct routes in the presence of asymmetric (uni-directional) links, another possible situation in wireless networks.

The two main operations of DSR are route discovery and route maintenance. When a host wants to send a packet and there is no route to the destination currently available in its route cache, the host initiates a route discovery. The discovering process is straightforward. The initiator broadcasts a route request to its neighbors. A route request contains the address of the destination host as well as a route record which records the hosts that the request has passed. Upon receiving a route request, a host checks if it knows a route to the destination or itself is the destination. In both cases, the complete route from the initiator to the destination is found. This route is then replied to the initiator. Otherwise, the host appends its address to the route record and re-broadcast the route request to its neighbors. Because of the broadcasting, a host may receive multiple copies of the same route request. To avoid repeatedly processing the same request, each host maintains a list of the IDs of the recently seen requests. A host can also detect that a request has gone through a cycle if it finds its address already listed in the route record of the request. In both cases, the host discards the route request and does nothing further.

Routes may become invalid due to the host movement. To quickly adapt to this change, each host constantly monitors the links it uses to forward packets. If a host in a route finds out that it cannot forward packets to the next host in the route, (many wireless networks support a hop-by-hop acknowledgment at the data link level), it immediately sends a route error packet to the source of the route. Therefore, the source host is able to quickly detect an invalid route and stop using it any longer.

2.2 On-Demand Distance Vector Routing (AODV)

AODV [6] shares the same on-demand characteristics as DSR, but adopts a very different mechanism to maintain routing information. In AODV, each host maintains a traditional routing table, one entry per destination. Each entry records the next hop to that destination and a sequence number generated by the destination which indicates the freshness of this information. In addition, each entry also records the addresses of active neighbors through which packets for the given destination are received. Therefore, once the corresponding link of this entry is down, the upstream hosts using this link can be notified immediately.

Like DSR, AODV discovers a route through network-wide broadcasting. The source host starts a route discovery by broadcasting a route request to its neighbors. In the route request, there is a requested destination sequence number which is 1 greater than the destination sequence number currently known to the source. This number prevents old routing information being used as reply to the request, which is the essential reason for the routing loop problem in the traditional distance vector algorithm. Unlike DSR, the route request doesn't record the nodes it has passed but only counts the number of such nodes. Instead, each node the request has passed sets up a temporary reverse link pointing to the previous node from which the request has come, so that the reply can be returned to the source host. An intermediate node can reply to a request only if it has a route entry for the destination which has the same or higher destination sequence number than the requested number. A route reply contains the total hop count of the route and its destination sequence number. As a reply travels back to the source, each intermediate node sets up the forward link as a route entry and records the destination sequence number. If the node receives further route replies later, it updates its routing entry and propagates the reply back to the source only if the reply has either a greater destination sequence number, or the same sequence number with a smaller hop count.

Route maintenance in AODV is similar to DSR. An invalid link can be detected through link layer acknowledgement, or by letting each host broadcasting periodic hello messages to neighbors. Hello messages can also be used to discover neighbors. Whenever a link in use is no longer valid, the upstream host of that link immediately notifies the active neighbors of the link, which in turn notify their active neighbors for the route and so on until the source hosts using that link are reached. The notification is done by sending an unsolicited route reply with a fresh sequence number and hop count of ∞. The fresh destination sequence number makes the active neighbors unconditionally updates their corresponding route entries, and the ∞ hop count simply means the route is no longer valid.

2.3 Cluster Based Routing Protocol (CBRP)

Another way to reduce flooding traffic is to establish some kind of hierarchy among mobile hosts, and query only those high-level hosts in the hierarchy which has the information about the low-level hosts under them.

In the CBRP protocol [9], mobile hosts form clusters. The head of a cluster knows the addresses of its members. Hence, broadcasting route requests only to the cluster heads is equivalent to broadcasting to every host in the network.

Since ad hoc network has no established infrastructure and its topology is constantly changing, the cluster formation must be self-contained and able to adapt to host movement. In addition, the formation should not incur too much overhead both on the computation workload of the mobile hosts and on the network traffic. CBRP uses a simple cluster formation strategy. The diameter of a cluster is only two hops and clusters can overlap. The cluster head is just the node whose IP address is the smallest among its neighbors. At any time, a node is in one of the three states: a cluster member, a cluster head, or undecided, meaning still searching for its host cluster. Every node broadcast a hello message to its neighbors periodically. At the beginning, all nodes are in the undecided state, and after a while the nodes with the smallest IP address among their neighbors elect themselves as cluster heads. After that, when a cluster head receives a hello message from an undecided neighbor, it sends out a triggered hello message which notifies that neighbor about the existence of the cluster. Upon receiving the triggered hello message from a cluster head, the undecided node changes its state to a member and records the cluster head's address. It is possible that a node gets responses from multiple heads. In that case, the node becomes member of each of the clusters. If a cluster member hasn't received a hello message from any of its head for a while, the node goes back to the undecided state and searches for clusters again.

In order to broadcast route requests among the cluster heads, each cluster head must know the addresses of its neighboring cluster heads. This adjacent cluster discovery is done by having each node maintain a *cluster adjacency table*, which stores the addresses of the neighboring cluster heads and the *gateway* node through which that head can be reached. Since clusters are only two-node wide, a member node is able to find out its neighboring cluster heads through the hello messages from its neighbors which are members of those clusters. A cluster head can then inspect the hello messages of its members which contain their cluster adjacency tables to get the information about the neighboring heads.

With all these information at hand, a route discovery starts with the source host broadcasting a route request to its neighbors, one of which is the cluster head. Subsequently, the request is flooded to the neighboring cluster heads though the gateway nodes, and so on until the request reaches the cluster head of the destination host which unicasts the request to the destination. The route request only records the cluster heads it has passed. Therefore, upon arriving at the destination, the request has the whole path from the source to the destination in terms of cluster heads. The actual route is calculated during the returning of the route reply. Each cluster head along the returning path tries to find out the optimal hop-by-hop route (maybe bypassing itself) from the previous node to the next cluster head in the path.

The rest of CBRP is almost the same as DSR. CBRP uses source routing. Currently used routes are monitored and route errors are notified to the source host immediately. Since a host can detect its current neighbors through their hello messages, it always tries to find a shorter route to forward a data packet by forwarding the packet to the furtherest node in the source route which has become its neighbor. As a result, shorter routes are reflected very quickly. A host can also use the neighbor information to do local route repair. Once a link is down, the upstream host checks to see if the next hop or some hop after that can be reached through one of its neighbors (a node's hello messages also include its neighborhood information, so its neighbors know their two-hop away nodes). In the case where hosts are not moving very fast, this local repair turns out to be efficient and avoids unnecessary route re-discovery.

3 Comparison of Ad hoc Routing Protocols

All three protocols are on-demand protocols. Hence they share certain advantages. For one thing, on-demand protocols can almost always guarantee to use valid routes. (Each discovered route is stored in the route cache for a short life time, so there is a slight chance that they may get stale). For another, route maintenance in all three protocols are done by real-time monitoring rather than periodic updates. As a result, they can quickly respond to topological changes which might be frequent in an ad hoc network.

Despite those common features, there are several important differences between these protocols, which may give rise to significant performance differentials. Table 1 summarizes the differences.

(1) Number of route discoveries: DSR makes the most uses of a route discovery due to the use of source routing. Once a route is found, nodes along the path can communicate with each other using the same route. And since the whole route is carried in each data packet, intermediate nodes have the opportunity to inspect the whole route and extract information

	AODV	CBRP	DSR
Number of route discoveries	Most	Less	Least
Hello messages	Small size, used as a supplement for neighbor detection.	An i integral part of cluster formation and maintenance. Size may be large.	No hello messages.
Source routing	Not used.	May cause bottleneck as too many nodes use the same route.	Same as CBRP
Broadcasting of route requests	Network-wide broadcasting, causing large overhead.	Only broadcasts to cluster heads, but needs to establishing and maintaining clusters. Also a more complicated route discovery is used.	Network-wide broadcasting, causing large overhead.

Table 1: **Differences between AODV, CBRP, and DSR**

from it. Because of this, one route discovery actually results in lots of route information and can be used by many nodes. In addition, if promiscuous receiving mode is used, a node can get even more routing information from data packets targeted at other nodes. Therefore, DSR has the least route discovery. And so does CBRP, which uses source routing and similar strategy to extract routing information from source routes. Not using source routing, each route discovery in AODV can find only one route. Hence the number of route discoveries in AODV is the most among the three protocols.

(2) Hello messages: DSR does not have periodic hello messages at all. AODV uses hello messages as a supplement for detecting link failure. It also serves as a one-hop route discovery. So AODV is able to avoid route discoveries whose source and destination are neighbors. Hello messages are the integral part of CBRP and the size of a hello message may be large as it contains the neighbor table and cluster adjacency table of the sender. As a result, while CBRP uses hello messages to establish clusters and in turn reduce the flood in route discovery, the hello message itself is another kind of overhead.

(3) Source routing: Source routing makes a route discovery more profitable. But it also has disadvantages. First, the extensive use of information in a source route might not always be a good idea. With so many nodes making use of the same route, the route becomes a bottleneck and traffic congestion may happen. Also, once the route has a link failure, many nodes will be affected. Furthermore, it is possible that a node extracts route information from a stale source route. Using the invalid route information, the node will waste time and cause unnecessary overhead. Second, carrying a source route in every data packet incurs overhead too, although in most cases the source route is negligible small comparing with the size of the data packet it is attached to.

(4) Broadcasting of route requests: Both DSR and AODV use network-wide flooding for route discovery. CBRP only broadcasts route requests to cluster heads, largely reducing the flooding traffic. But to set up and maintain clusters consumes both network bandwidth and host computation time. In addition, CBRP's route discovery consists of two phases. First, the route request reaches the destination and finds out the sequence of cluster heads from the source to the destination. Then the whole route is calculated as the route reply is returned through those cluster heads. This complicated procedure helps find a shorter route. (In DSR and AODV, the route taken by the first request reaching the destination is returned, which might not be a short route). But it also makes the route discovery longer.

4 Simulation Experiments

As each protocol has its own advantages and disadvantages, none of them can be claimed as absolutely better than the others. To see how the features of each protocol affect their performance, we did a performance comparison using the implementations of these protocols in *ns-2*[11].

The implementations of all three protocols are based on the CMU Monarch extension. Recently, the Monarch research group in CMU extended *ns-2* with support for simulating the physical, data link and MAC layer of multihop wireless networks. The distributed coordination function (DCF) of IEEE 802.11 for wireless LANs is used as the MAC layer. The 802.11 DCF uses Request-to-send (RTS) and Clear-to-send(CTS) control packets for unicast data transmission to a neighboring node. The RTS/CTS exchange precedes the data packet transmission and implements a form of *virtual carrier sensing* and channel reservation. Data packet transmission is followed by an ACK. Broadcasting data packets and the RTS control packets are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance (CSMA/CA) is used to transmit these packets. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN. WaveLAN is a shared-media radio with a nominal bit-rate of 2Mb/sec and a nominal radio range of 250 meters. A detailed description of simulation environment and the models is available in [2], [11].

The traffic and mobility models are the same as [8], which is a common test scenario used by other performance comparison papers [2] [10]. Traffic sources are CBR (continuous bit-rate). The source-destination pairs are spreaded randomly over the network. By changing the total number of traffic sources, we get scenarios with different traffic loads. For small traffic loads (10, 20, 30 sources), the packet rate at the source node is 4 packets/sec. For 40 sources a smaller rate of 3 packets/sec for 50 nodes and 2 packets/sec for 100 nodes is used, since higher rate will cause very high network congestion. Only 512 byte data packets are used. The mobility model uses the *random waypoint* model [2] in a rectangular field. Two field configurations are used: (1) 1500m × 300m field with 50 nodes and (2) 2200m × 600m field with 100 nodes. Each node starts its journey from a random location to a random destination with a randomly chosen speed uniformly distributed between 0-20m/sec. Once the destination is reached, another random destination is targeted after a pause. Varying the pause time changes the frequency of node movement. For the set of tests with 50 nodes, the total simulation time is 900 seconds, and each data point in the following figures is the average of 5 runs with the same scenario configuration but different random seeds. For the 100-node tests, because they are too slow, the simulation time is 500 seconds, and each data point is got with 3 runs. Identical mobility and traffic scenarios are used across protocols.

4.1 Performance Metrics

Three key performance metrics are evaluated in our experiments:

Throughput — ratio of the data packets delivered to the destination to those generated by the CBR sources.

Average end-to-end delay of data packets — 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 time.

Normalized routing overhead — this metric has two variants: *packet overhead* is the number of routing packets "transmitted" per data packet "delivered" at the destination; *byte overhead* is the number of bytes of routing packets "transmitted" per data byte "delivered" at the destination. Each hop-wise transmission of a routing packet is counted as one transmission. Finally, we also compared the number of routing requests and replies between CBRP and DSR to see how much of them can be reduced with CBRP's cluster structure.

The first two metrics are the most important metrics for best-effort traffic. The routing load metric evaluates the efficiency of the routing protocol. Note that these metrics are not completely independent. For example, a larger overhead may cause lower throughput and longer delay. On the other hand, a shorter delay may not necessarily imply a higher throughput, since delay is only measured on those successfully delivered packets. Also notice the scenario tested here is simply a random situation. Real-world ah hoc networks usually have special traffic and mobility models. The difficulty here is that different applications have different scenarios, and it is not very clear what the typical scenario of a specific application is.

In Figures 1, 2, we report the results obtained with respect to the throughput for all routing protocols using a variety of networks size model and scenarios. In all the testing scenarios, the two source routing protocols demonstrate high quality in delivering packets — more than 95% in the case of 50 nodes and mostly above 90% for 100 nodes. AODV has difficulty when the nodes are moving fast (corresponding to smaller pause time), with a throughput less than 80%. As discussed previously, source routing reveals more information in one route discovery than AODV. Therefore, within the same time more routes are discovered and so more packets can be delivered. AODV catches up when the mobility of the nodes gets lower. This is because routes become more stable and so eventually everybody can find all the routes it ever needs. Between DSR and CBRP, CBRP has a better throughput in the 100-node scenario. This better scalability comes from its largely reduced flooding for route discovery.

In Figures 3, 4, We report the results we obtained with respect to the delays. As we can see, among the three protocols, AODV has the shortest end-to-end delay of no more than 0.05 second. Besides the actual delivery of data packets, the delay time is also affected by route discovery, which is the first step to begin a communication session. The source routing protocols have longer delay because their route discovery takes more time as every intermediate node tries to extract information before forwarding the reply. And the same thing happens when a data packet is forwarded hop by hop. Hence, while source routing makes route discovery more profitable, it slows down the transmission of packets. CBRP is even more time-consuming because of its two-phase route discovery. The task of maintaining cluster structure also takes a piece of each host's CPU time.

In Figures 5 - 8, we report the results obtained with respect to the overhead of all routing protocols. As we can see, without any periodic hello messages, DSR outperforms the other two protocols in terms of overhead. In most cases, both the packet overhead and the byte overhead of DSR are less than half of the overhead of CBRP and less than a quarter of AODV's overhead. AODV has the largest routing load, (in the 50-node cases, as many as 6.5 routing packets per data packet and 2 routing bytes per data byte), because the number of its route discoveries is the most and the discovery is network-wide flooding. CBRP has much smaller flooding range — as shown in Figure 7 and 8, the number of its route requests and replies is constantly half of that of DSR. But its hello messages outweigh this gain. And since the size of CBRP hello messages can be large, its byte overhead is still more than DSR's (in the 50-node cases, more than twice as much as DSR's). When there are more connections, more routing is needed and so the proportion of hello messages in the total overhead becomes smaller. As the result, CBRP and AODV gets closer to DSR.

5 Conclusion

Ad hoc networks are useful for providing communication support where no fixed infrastructure exists or the deployment of a fixed infrastructure is not economically profitable, and movement of communicating parties is allowed. Applications of ad hoc networks include military operations (communication in a hostile environment), rescue operations (rapid deployment of a communication network where infrastructures do not exist or have been damaged), and sporadic happenings coverage (intense utilization of a communication network for a very limited time). Due to the mobility, routes connecting two nodes may change. Therefore, it is not possible to establish a priori fixed paths for message delivery through the network. As a consequence, the routing in ad hoc wireless networks is hard and a challenging problem.

In this paper, we focus upon the routing problem in ad hoc networks. We have discussed the differences among the three ad hoc routing protocols, AODV, CBRP and DSR. We have presented an extensive performance of ad hoc routing protocols using a variety of workload such mobility, load and size of the ad hoc networks. Our results indicate that the two source routing based protocols, DSR and CBRP, have very high throughput while the the distance-vector based protocol, AODV, exhibits a very short end-to-end delay of data packets. Furthermore, despite its improvement in reducing route request packets, CBRP has a higher routing overhead than DSR because of its periodic hello messages. DSR has much smaller routing overhead than AODV and CBRP. and AODV has the largest overhead among the three protocols.

We plan to investigate and study on how to improve and reduce further the routing overhead using history and setting up structures like clusters and using GPS for instance [3].

References

[1] D. B. Johnson and D. A. Maltz. "*Dynamic Source Routing in Ad Hoc Wireless Networks*", Mobile Computing, Chapter 5, pages 153-181, Kluwer Academic Publishers, 1996.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu, and J Jetcheva. "*A performance comparison of multi-hop wireless ad hoc network routing protocols*", Mobicom'98, October 1998.

[3] A. Boukerche, "*Ad Hoc Routing Protocols Using GPS*", Technical Report UNT, In Preparation.

[4] A. Boukerche, "*A Performance Comparison of Routing Protocols for Ad Hock Networks*", IEEE Parallel and Distributed Computing for Wireless and Mobile Systems, LNCS, Springer-Verlag 2001.

[5] A. Boukerche, S. K. Das, and A. Fabbri "*Analysis of Randomized Congestion Control with DSDV Routing in Ad Hoc Wireless Networks*" IEEE MASCOTS'2000, pp.65-72.

[6] C. E. Perkins, E. RM. Royer, and Samir R. Das. "*Ad Hoc On-Demand Distance Vector (AODV) Routing*", IETF MANET Working Group, Internet-Draft, March 2000.

[7] C.E. Perkins, P. Bhagwat, Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, *ACM SIGCOMM'94*, pp. 234-244.

[8] S. R. Das, C.E. Perkins, and E. M. Royer. "*Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks*", INFOCOM'2000.

[9] M. Jiang, J. Li, and Y.C. Tay. "*Cluster Based Routing Protocol (CBRP)*", IETF MANET Working Group, Internet-Draft, 1999.

[10] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. "*Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks*", Mobicom'99.

[11] UCB/LBNL/VINT Network Simulator. "*http://www-mash.cs.berkeley.edu/ns/*".

[12] G.S. Malkin, M.E. Steenstrup, Distance-Vector Routing, in: *Routing in Communications Networks*, Ed. M.E. Steenstrup (Prentice-Hall, 1995), pp. 83-98.

[13] J. Moy, Link-State Routing, in: *Routing in Communications Networks*, Ed. : M.E. Steenstrup (Prentice-Hall, 1995), pp. 135-157.

[14] S. Ramanathan, M.E. Steenstrup, A Survey of Routing Techniques for Mobile Communications Networks, *ACM MONET*, Vol.1 (1996), pp. 98-104.

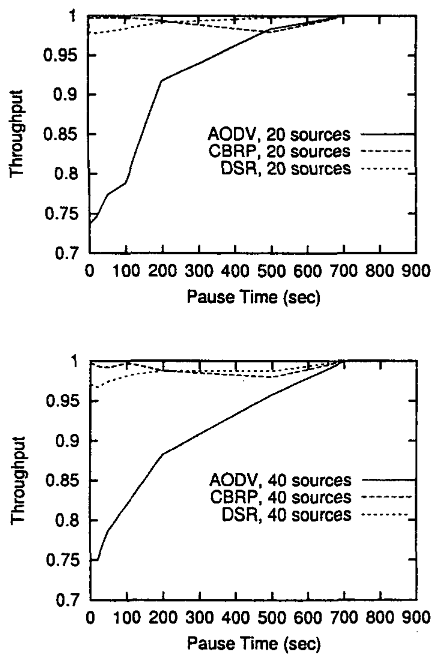

Figure 1: **Data packet throughput for the 50-node model with various numbers of traffic sources**

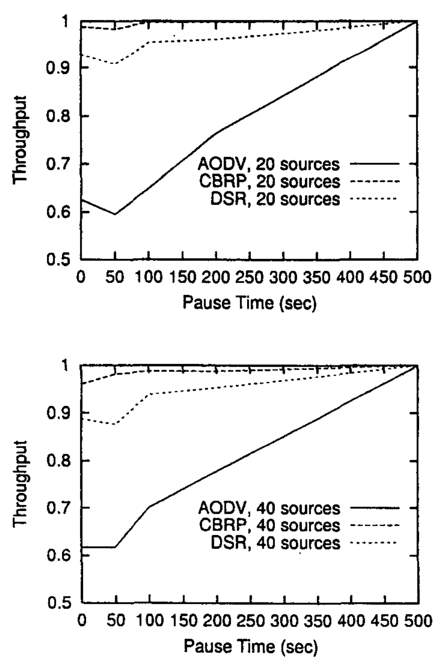

Figure 2: **Data packet throughput for the 100-node model with various numbers of traffic sources**

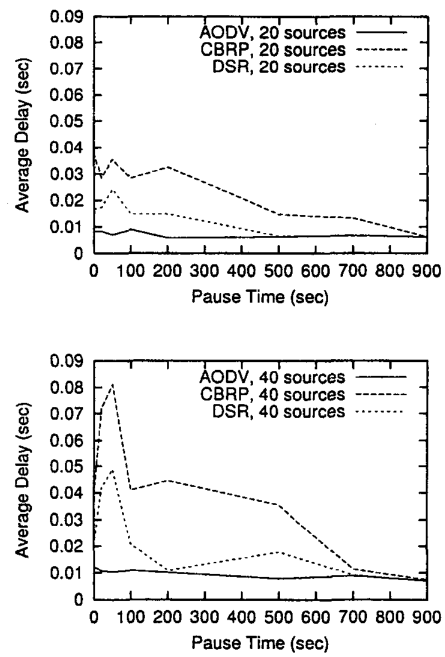

Figure 3: **Average data packet delay for the 50-node model with various numbers of traffic sources**

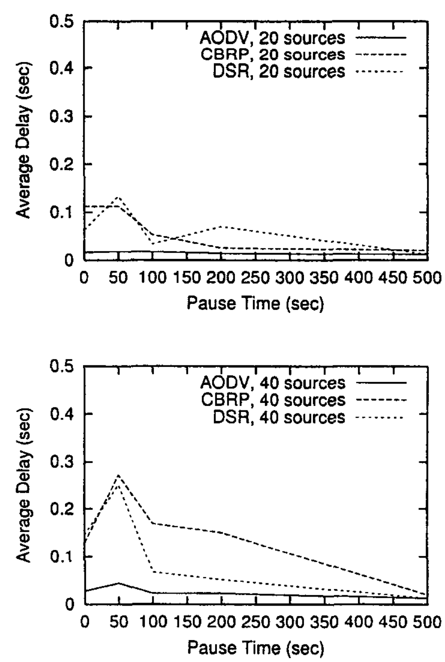

Figure 4: **Average data packet delay for the 100-node model with various numbers of traffic sources**

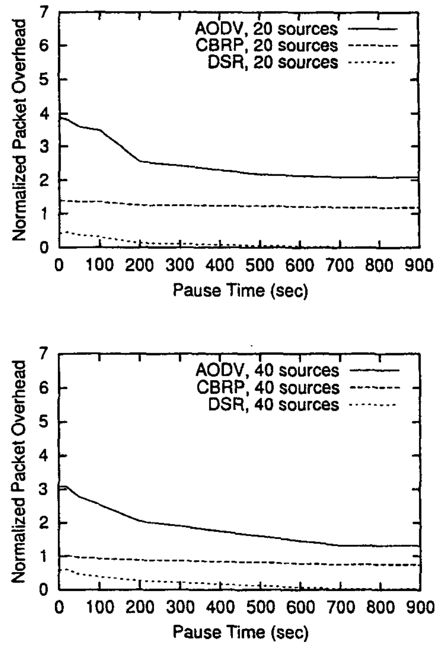

Figure 5: **Normalized packet overhead for the 50-node model with various numbers of traffic sources**

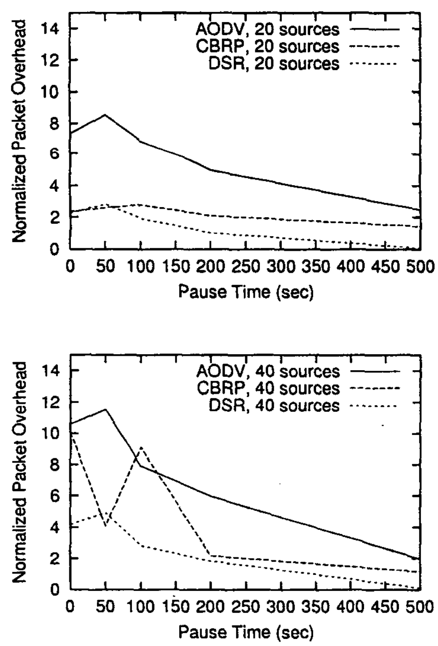

Figure 6: **Normalized packet overhead for the 100-node model with various numbers of traffic sources**

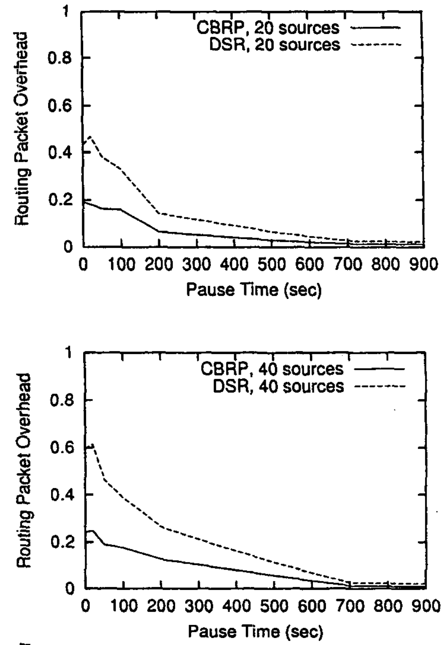

Figure 7: **Normalized packet overhead (excluding hello messages) for the 50-node model with various numbers of traffic sources**

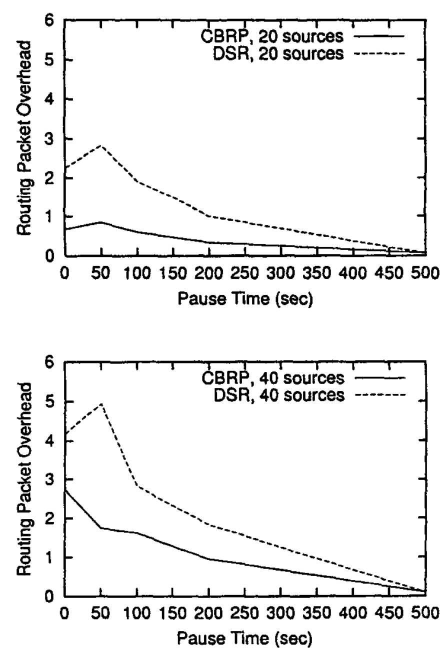

Figure 8: **Normalized packet overhead (excluding hello messages) for the 100-node model with various numbers of traffic sources**