A Performance Comparison of Routing Protocols for Ad Hoc Networks

Azzedine Boukerche *
Parallel Simulations and Distributed Systems (PARADISE)
Research Laboratory
University of North of Texas
Email : boukerche@cs.unt.edu

Abstract

Ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration. Because of the limited range of each host's wireless transmission, to communicate with hosts outside its transmission range, a host needs to enlist the aid of its nearby hosts in forwarding packets to the destination. However, since there is no stationary infrastructure such as base stations, each host has to act as a router for itself. In this paper, we focus upon on-demand schemes. We study and compare the performance of the following three routing Protocols AODV, CBRP and DSR.

1 Introduction

An ad hoc network is a collection of wireless mobile hosts forming a temporary network without the aid of any established infrastructure or centralized administration [11]. Because of the limited range of each host's wireless transmission, to communicate with hosts outside its transmission range, a host needs to enlist the aid of its nearby hosts in forwarding packets to the destination. However, since there is no stationary infrastructure such as base stations, each host has to act as a router for itself. Hence a routing protocol for ad hoc networks is a protocol that will be executed on every host and is therefore subject to the limit of the resources at each mobile host. A number of routing protocols have been proposed for ad hoc wireless networks [1], [9], [15], [6], derived from *distance-vector* [13] or *link-state* [14] 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 our recent work [4], we have proposed an ad-hoc routing protocol using a randomized version of DSDV and based upon Markov modeling

In this paper, we focus upon the on-demand ad hoc routing protocols. We study and compare the three algorithms known as AODV, CBRP and DSR.

*This work was supported by UNT Faculty Research Grant.

2 Ad-Hoc Routing Protocols

The basic idea of on-demand routing algorithms is to find and maintain a route only when it is used for communication. This idea proves to be especially efficient in ad hoc networks where routes are usually temporary.

Dynamic Source Routing (DSR): 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 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.

Ad-hoc On-Demand Distance Vector Routing (AODV): AODV [5] 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.

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 [8], 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 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*, version 2.1*b*1 [12].

In our experiments, traffic sources are CBR, i.e., 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 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. 1500m$\times$300m field with 50 nodes was used in our experiments. 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 frequncy 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.

In the course of our experiments, we consider the following two parameters: (1) *Throughput* — ratio of the data packets delivered to the destination to those generated by the CBR sources; and (2) *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. The two metrics are the most important metrics for best-effort traffic. Note that these metrics are not completely independent. For example, a larger overhead may cause lower throughput and longer delay.

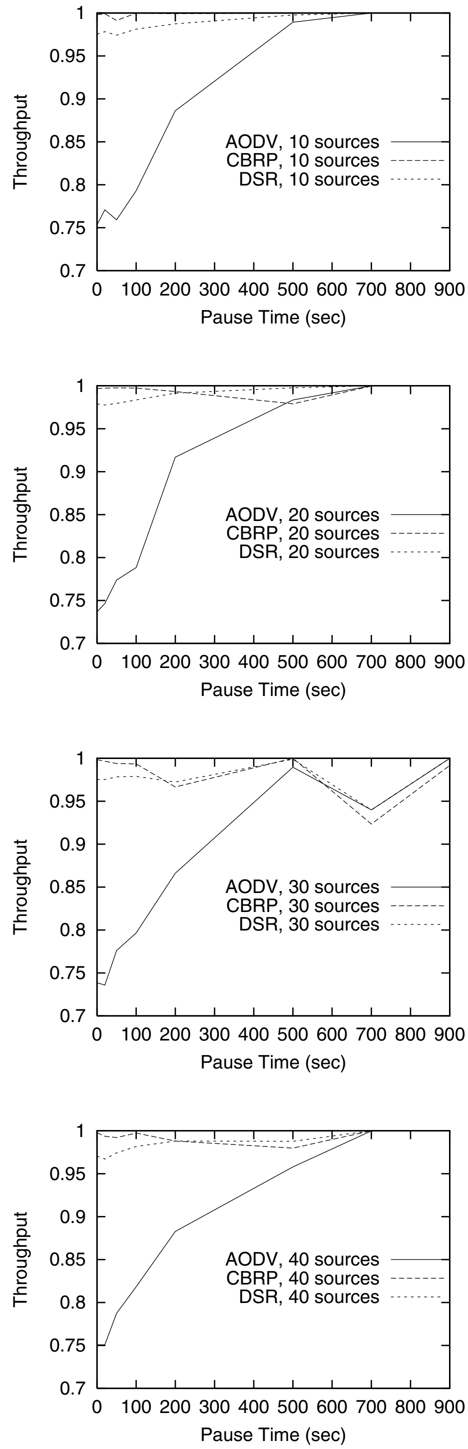

Figure 1: Data packet throughput vs. numbers of traffic sources

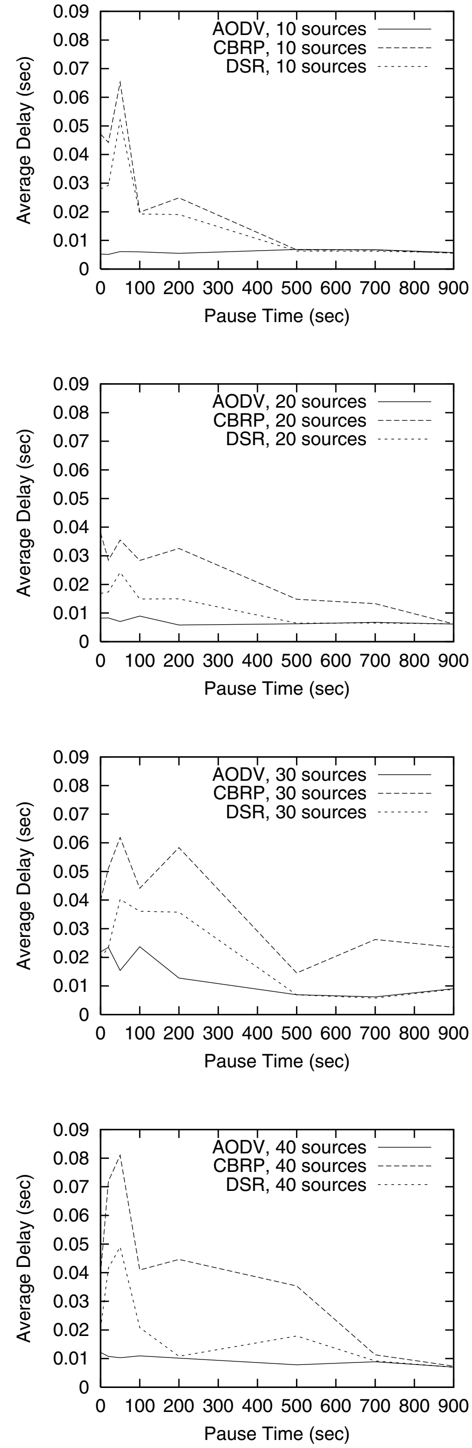

Figure 2: data packet delay vs. number of traffic sources

On the other hand, a shorter delay may not necessarily imply a higher throughput, since delay is only measured on those successfully delivered packets.

As shown in Fig. 1. 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. 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.

As shown in Fig. 2 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.

4 Conclusion

Ad hoc wireless networks are composed of mobile stations communicating solely through wireless channels. Such networks are expected to play an increasingly important role in future civilian and military setting. In this paper, we focus upon routing problem in ad hoc networks. in particular the AODV, CBRP and DSR routing protocols. We have presented an extensive simulation studies and compare these 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 throughputs while the the distance-vector based protocol, AODV, exhibits a very short end-to-end delay of data packets.

As GPS system becomes more and more common in wireless devices, it will give a whole new source of information. Knowing its current location and the locations of other nodes, a node can quite efficiently reduce the area for searching its destination node. By combining the GPS system with the current protocols [3], we expect large enhancement on routing overhead, which makes those protocols eventually practical for use in the real world.

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**, S. K. Das, and A. Fabbri "Analysis of Randomized Congestion Control with DSDV Routing in Ad Hoc Wireless Networks" Special Issue on Mobile Computing. To appear in Journal of Parallel and Distributed Computing (JPDC) (Short version appear in MASCOTS'2000, pp.65-72)

[4] A. Boukerche,"*Efficient Ad Hoc Routing Protocols Using GPS*", Technical Report UNT 2001.

[5] A. Boukerche,"*Performance Comparisons of Ad Hoc Routing Protocols*", Technical Report UNT 2000.

[6] C. E. Perkins and E. M. Royer. "*Ad-hoc On-Demand Distance Vector Routing*", 2nd IEEE Workshops on Mobile Computing Systems and Applications, 1999.

[7] C.E. Perkins, P. Bhagwat, Highly Dynamic Destination Sequenced Distance Vector Routing (DSDV) for Mobile Computers, *Proceedings of ACM SIGCOMM'94*, London, UK (Aug-Sept 1994), 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, March 2000.

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

[10] Z.J. Hass and M.R. Perlman. "*The Zone Routing Protocol (ZRP) for Ad Hoc Networks*", IETF MANET Working Group, Internet-Draft, June 1999.

[11] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. "*Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks*", Mobicom, August 1999.

[12] G.S. Lauer, Packet-Radio Routing, in: *Routing in Communications Networks*, Editor: M.E. Steenstrup (Prentice-Hall, 1995), Ch.11, 351-396.

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

[14] G.S. Malkin, M.E. Steenstrup, Distance-Vector Routing, in: *Routing in Communications Networks*, Editor: M.E. Steenstrup (Prentice-Hall, 1995), Ch.3, 83-98.

[15] J. Moy, Link-State Routing, in: *Routing in Communications Networks*, Editor: M.E. Steenstrup (Prentice-Hall, 1995), Ch.5, 135-157.

[16] S. Murthy, J.J. Garcia-Luna-Aceves, An Efficient Routing Protocol for Wireless Networks, *Mobile Networks and Applications*, Vol.1 (1996), 183-197.