Simulation Based Performance Evaluation of Mobile, Ad hoc Network Routing Protocols [*]

Samir R. Das, Robert Castañeda and Jiangtao Yan
Division of Computer Science
The University of Texas at San Antonio
San Antonio, TX 78249-0667
U.S.A.
Email: {`samir,rcastane,jyan`}`@cs.utsa.edu`

Abstract

In this paper we evaluate several routing protocols for mobile, wireless, ad hoc networks via packet level simulations. The ad hoc networks are multi-hop wireless networks with dynamically changing network connectivity owing to mobility. In the protocol suite includes several routing protocols specifically designed for ad hoc routing, as well as more traditional protocols, such as link state and distance vector, used for dynamic networks. Performance is evaluated with respect to fraction of packets delivered, end-to-end delay and routing load for a given traffic and mobility model. Both small (30 nodes) and medium sized (60 nodes) networks are used. It is observed that the new generation of on-demand routing protocols use much lower routing load, especially with small number of peer-to-peer conversations. However, the traditional link state and distance vector protocols provide, in general, better packet delivery and end-to-end delay performance.

1 Introduction

A mobile, *ad hoc* network [5] is an autonomous system of mobile hosts connected by wireless links. There is no static infrastructure such as base stations. If two hosts are not within radio range, all message communication between them must pass through one or more intermediate hosts that double as routers. The hosts are free to move around randomly, thus changing the network topology dynamically. Thus routing protocols must be adaptive and able to maintain routes in spite of the changing network connectivity. Such networks are very useful in military and other tactical applications such as emergency rescue or exploration missions, where cellular

---

[*] A preliminary version of this work appeared in the proceedings of the Seventh International Conference on Computer Communication and Networks (IC3N), Lafayette, LA, Oct. 1998.

infrastructure is unavailable or unreliable. Commercial applications are also likely where there is a need for ubiquitous communication services without the presence or use of a fixed infrastructure. Examples include on-the-fly conferencing applications, networking intelligent devices or sensors etc.

Interest in such dynamic wireless networks is not new. It dates back to the seventies, when the U.S. Defense Research Agency, DARPA worked on PRNET (Packet radio Network) [19] and SURAN (Survivable Adaptive Networks) [32] projects. They supported automatic route set up and maintenance in a packet radio network with moderate mobility. Interest in such networks has recently grown due to the common availability of wireless communication devices that can connect laptops and palmtops and operate in license free radio frequency bands (such as the Industrial-Scientific-Military or ISM band in the U.S.). In an interest to run internetworking protocols on ad hoc networks, a new working group for Mobile, Ad hoc Networking (MANET) has been formed within the Internet Engineering Task Force (IETF) [23], whose charter includes developing a framework for running IP based protocols in ad-hoc networks. Interest has also been partly fueled by the recent IEEE standard 802.11 [8] that includes the MAC and physical layer specifications for wireless LANs without any fixed infrastructure.

Routing protocols in packet-switched networks traditionally use either *link-state* or *distance-vector* routing algorithm [20]. Both algorithms allow a host to find the next hop neighbor to reach the destination via the “shortest path.” The shortest path is usually in terms of the number of hops; however, other suitable cost measures such as link utilization or queueing delay can also be used. Such shortest path protocols have been successfully used in many dynamic packet switched networks. Prominent examples include use of link state protocol in OSPF (Open Shortest Path First) [24] and use of distance vector protocol in RIP (Routing Information Protocol) [17] for interior routing in the Internet. Even though, any such protocol would, in principle, work for ad hoc networks, a number of protocols has been specifically developed for use with ad hoc networks. The primary motivation is that the shortest path protocols, either link-state or distance vector, take too long to converge and have a high message complexity [5]. Because of the limited bandwidth of wireless links, message complexity must be kept low. Also, potentially rapidly changing topology makes it important to find routes quickly, even if the route may be suboptimal [5].

Several new ad hoc routing protocols have been developed with this basic philosophy. They, however, vary widely in characteristics. For example, some of these protocols are variations of distance vector routing. Some protocols explicitly maintain redundant routing paths so that alternatives are available when a route changes. Some recently proposed protocols use a *reactive* approach for route discovery and maintenance, instead of the more traditional, *proactive* approach [16]. In a reactive approach protocols are “source initiated;” routes are discovered and maintained on an as needed basis, thus circumventing large overheads of always maintaining routes between all possible source and destination pairs. The protocols are briefly reviewed in the following section.

Even though many protocols have been proposed, their comparative performance is not well understood. Current literature reports only a limited amount of performance study and when performance is reported, typically comparison has been made only to a selected few protocols (typically only to link state and distance vector protocols). Specifically, the protocols proposed for mobile, ad hoc routing have not been evaluated against one another. Our goal in this paper to address this inadequacy by a thorough performance study of several key protocols in the same framework to better understand their comparative merits and suitability for deployment under different scenarios. To this end we use an existing, packet level, routing simulator called *MaRS* [1] with added mobility modeling capability. A suite of ad hoc routing protocols is evaluated under varying mobility and traffic models. Traditional link state and distance vector protocols are included in the suite to provide a point of comparison.

The rest of the paper is organized as follows. In Section 2, the dynamic routing protocols in packet networks are reviewed with a special emphasis on the protocols evaluated in this paper. Section 3 describes the simulation platform used in evaluation and Section 4 presents the performance results. Related work and conclusions are presented in Section 5 and Section 6 respectively.

2 Routing Protocols for Ad hoc Networks

2.1 Link State Protocols

Each node maintains its own view of the network topology, including link costs of all its outgoing links. To keep views up-to-date, each node broadcasts the link costs of all its neighbors[1] to all other nodes in the network using flooding. This is done whenever there is a change in link costs. As a node receives this information, it updates its view of the network topology and applies a shortest path algorithm (Dijkstra's shortest path algorithm [9] in our simulations) to choose the next hop to a destination. Asynchronous link cost updates may give rise to short-lived routing loops; however, they disappear by the time update messages have propagated throughout the network [20]. In our simulation we used the SPF implementation of link state protocol as described in [33].

2.2 Distance Vector Protocols

In the distance vector approach, for each destination i, every node j maintains a set of distances or costs, $d_{ik}(j)$, where k ranges over the neighbors of i. Node k is treated as the next hop node

---

[1] In a mobile, ad hoc network any node can potentially be a neighbor.

for a data packet destined for i, if $d_{ik}(j) = \min \forall k\{d_{ik}(j)\}$. To keep these distances up-to-date, whenever there is any change of this minimum distance because of link cost changes, the new minimum distance is reported to the neighboring nodes. If, as a result, a minimum distance to any neighbor changes, this process is repeated. This technique is the classical distributed Bellman-Ford algorithm [2].

Routing loops, both short-lived and long-lived, are possible in distributed Bellman-Ford. There is also a possibility of *counting-to-infinity* problem, where it takes a large number of update messages to detect that a node is unreachable [20]. Several protocols have been proposed to avoid long-lived loop and counting to infinity problems. They typically work by increasing the amount of information exchanged between nodes or providing some sort of inter-nodal coordination. For example, in the Border Gateway Protocol (BGP) the entire path between the source and the destination is sent instead of just the distance [31]. In DUAL (Distributed Update Algorithm) [12] inter-nodal coordination is achieved via a technique known as *diffusing computation.*

We focus our attention on two distance vector protocols. The first, extended Bellman-Ford achieved good performance for stationary networks in earlier simulation studies [33]. The second, DSDV [30], was specifically proposed for mobile, ad hoc networks.

Extended Bellman Ford Extended Bellman-Ford [4] augments the classical Bellman-Ford by maintaining on node j, (in addition to the set of distances $d_{ik}(j)$) a set of nodes $N_{ik}(j)$, which immediately precede the destination i in the path from j to i via neighbor k. Then it is possible for the source node i to construct the whole path to the destination, by repeatedly using the preceding node $N_{ik}(j)$ as a new destination. It can be shown that the protocol is free from both long-lived loop and counting-to-infinity problems, if each node avoids sending route change updates to a neighbor for any destination d, if that neighbor is in the path to d.

We used the EXBF implementation described in [33] for our evaluation. Here, periodically or whenever a failure or reconnect occurs, link costs are recalculated, and if there is a change in the minimum distance, the new minimum distance is reported to the neighboring nodes. Several protocols proposed in the literature are also based on a similar idea of maintaining the second-to-last hop (predecessor) for the shortest path to each destination to achieve loop freedom. See [25] and the references therein.

DSDV The *destination sequenced distance vector* or DSDV protocol [30] has been specifically targeted for mobile networks. DSDV augments the classical, distributed Bellman-Ford by tagging each distance entry $d_{ik}(j)$ by a sequence number that originated in the destination node i. Each node maintains this sequence number, incrementing it each time the node sends an update to the neighbors. The sequence number is disseminated in the network via update messages. The destination sequence number is used to determine the "freshness" of a route. Always the latest sequence number is used for updating routes. For equal sequence numbers, the one with the smallest distance metric is used. It has been shown that DSDV avoids long-lived loops and counting to-infinity problems [30]. However, in our knowledge its performance has not yet been reported in literature.

2.3 Multipath Protocols – TORA

The unique feature of the *temporally ordered routing algorithm* or TORA [28] is maintaining multiple routes to the destination so that many topological changes do not require any reaction at all, as having just a single route is sufficient. The protocol reacts only when *all* routes to the destination are lost. In that case routes are re-established via a temporally ordered sequence of diffusing computations, which are essentially *link reversals* (to be described momentarily). In the event of network partitions, the protocol is able to detect the partition and erase all invalid routes.

TORA is based, in part, on the classical work by Gafni and Bertsekas [11], who consider a similar problem of maintaining a *destination oriented* directed acyclic graph (DAG) in the face of topological changes. A DAG is considered destination oriented, if for every node, there is a path to a given destination. If link failures make such a graph "destination disoriented," a series of link reversals ensue so that the graph again becomes destination oriented in finite time. The graph is initially constructed (route discovery or construction phase) in a "source-initiated" fashion, using a *query* flood followed by *update* routing packets. From that point it is maintained (route maintenance phase) using link reversals alone, whenever topological change causes a node to lose its last downstream link. If the destination becomes unreachable because of a network partition, the protocol erases (route erasure phase) all invalid routes.

TORA uses a notion of node "height" to maintain the destination oriented DAG. Each node maintains a height and exchanges this value with each neighbor. The significance of the height is that a link is always directed from a "higher" node to a "lower" node. Note that this notion of height and link directions are destination specific. Independent copies of the protocol runs for each possible destination node in the network.

In the initial route construction phase, the height of a node carries the notion of distance (in hops) of the node from the destination. However, this distance information is eventually lost during route maintenance phase. Since multiple routes are maintained in TORA, an obvious question is the choice of route. Two alternatives are suggested – choosing a neighbor randomly so that the loads are more or less evenly distributed or choosing the lowest neighbor [28]. We have chosen the latter in our simulations.

2.4 On Demand Protocols

Link state and distance vector protocols are primarily *proactive* protocols in the sense that routes are maintained to all potential destinations (possibly all nodes in the network) all the time, whether are not all such routes are actually used [15]. Route maintenance can be a large overhead because of a significant amount of route update traffic, especially for large networks. *Reactive* or on-demand protocols, on the other hand, create and maintain routes only on "as needed" basis. Thus, when a route is needed, some sort of global search procedure is employed. The family of classical flooding algorithms belong to the reactive group. Two examples of recently proposed reactive protocols are DSR and AODV. Note that TORA, described earlier is also partially reactive in the sense that route creation is initiated on demand. However, route maintenance is done on a proactive basis such that multiple routing options are available in case of link failures.

DSR Dynamic source routing or DSR [18] uses a technique where the source of a data packet determines the complete sequence of nodes through which to forward the packet; the source explicitly lists this route in the packet's header. DSR builds routes on demand using flooded query packets that carry the sequence of hops they passed through. Once a query reaches the destination, destination replies with a reply packet that simply copies the route from the query packet and traverses it backwards.[2] Each node has a *route cache*, where complete routes to desired destinations are stored as gleaned from the reply packets. These routes are used for data packets. Route failure is detected by the failure of an attempted message transmission. Such a failure initiates an error packet sent backward to the source. The error packet erases all routes in the route caches of all intermediate nodes on its path, if the route contains the failed link.

DSR has an unique advantage by virtue of source routing. As the route is part of the packet itself, routing loops, either short- or long-lived, cannot be formed as they can be immediately detected and eliminated. This property opens up the protocol to a variety of useful optimizations. For example, a flooded query can be quenched early by having any non-destination host reply to the query if that host has a route to the intended destination. A node can learn a route to a destination while passing on route reply packets. Also, routes can be improved by having nodes

---

[2] We assume that the wireless links are symmetric, which may not be the case in practice. DSR can tolerate asymmetric links by, for example, using an independent route discovery from destination back to the source. We did not consider asymmetric links in our evaluations in this work.

promiscuously listen to conversations between other nodes in proximity. We haven't implemented this last optimization, however.

AODV Ad hoc, on-demand distance vector protocol or AODV [29] is an on-demand variation of distance vector protocols. AODV uses destination sequence numbers like DSDV to determine freshness of routing information. In AODV, flooded requests are used to create route, with the destination responding to the first such request, much as in DSR. However, AODV maintains routes in a distributed fashion, as routing table entries, on all intermediate nodes on the route. Routing table entries are tuples in the form of <*destination, next hop, distance*>. Nodes forwarding queries remember the earlier hop taken by the query packet. This hop is used to forward the reply packet back to the source. The reply packet sets up the routing table entries on its path. AODV advocates use of "early quenching" of request packets, i.e., any node having a route to the destination can reply to a request. AODV also uses a technique called *route expiry*, where a routing table entry expires after a pre-determined period, after which fresh route discovery must be initiated.

AODV maintains the addresses of the neighbors through which packets destined for a given destination were received. A neighbor is considered *active* (for a destination), if it originates or relays at least one packet for that destination, within the past *active timeout* period. A routing table entry is *active* if it is used by an active neighbor. The path from a source to a destination via the active routing table entries is called an *active path*. On a link failure, all routing table entries are erased for which the failed link is on the active path. This is accomplished by an error packet going backwards to the active neighbors, which forward them to their active neighbors and so on. This technique effectively erases the route backwards from the failed link.

Neither DSR nor AODV guarantees shortest path. If the destination alone can respond to route requests (i.e., early quenching of route requests is not used) and the source node (and not an intermediate node) is always the initiator of the route request, the initial route may the shortest. But depending on the changes in topology this route may not always remain the shortest.

2.5 Other Protocols

Several other protocols have appeared in literature for mobile, ad hoc networks. Zone routing protocol (ZRP) [14, 15, 16] is a zone or cluster based routing protocol that is a hybrid between proactive and reactive routing. It is targeted for very large networks and divides the network into zones or clusters of nodes. The nodes within a zone is close to one another. Use of proactive routing is advocated within a zone and reactive routing across zones [15]. Cluster based routing, however, is not new. Quite a few cluster-based approaches have appeared in the past. See, for example, [32, 13, 22].

We have not yet included ZRP in our suite of protocols considered for evaluation because of a couple of reasons. ZRP can be viewed more a "routing framework" rather than an independent protocol, as potentially any proactive protocol can be employed for intrazone routing and any reactive protocol can be employed for interzone routing. Also, ZRP is suitable for a very large network such as a good number of clusters can be formed, while our study so far is concentrated on moderately sized network. Note that at the time of this writing, TORA, DSR, AODV and ZRP are the four protocols currently under study by the IETF MANET working group as candidate protocols for evaluation and standardization [23].

3 Simulation Model

A discrete event, packet-level, routing simulator called *MaRS* (Maryland Routing Simulator) [1] was used for comparative performance evaluation. MaRS is a flexible platform developed specifically for evaluation and comparison of network routing algorithms. MaRS was used previously for comparative evaluation of link-state and distance-vector routing protocols for the NSFNET T1 backbone network with the possibility of link failures [33]. We augmented MaRS to provide node mobility. The nodes can move around in a rectangular region according to a given mobility model (to be described momentarily). Each node has a fixed radio range and has a link to every other node in the system. If the other node is not within range the link cost is infinity. Otherwise, the link cost is modeled by the *hop-normalized delay* function, same as the revised ARPAnet cost metric [21, 33].

Each node is modeled by a store-and-forward, queueing station, and is characterized by parameters such as buffer space and processing speed. Each link is characterized by a bandwidth and propagation delay. A link is modeled as an FCFS queue with service time as the transmission time. Currently, our study is limited to network layer details. Thus, no link layer details, such as MAC protocol, multiple-access interference or link errors, are modeled, nor are any physical, radio channel level details.

The routing protocol is modeled as an independent routing module, one at each node, which maintains routing information (such as next-hops, distances, routing table etc. depending on the protocol used) and responds to routing packets and link status changes. Routing packets are distinct from data packets in the simulator and are used for route maintenance. The nodes forward data packets via the next hop link as per the routing information provided by the routing module. If the next hop link is broken or there is no next hop information available, data packets are dropped until some usable next hop information is available. In source routed protocols, however, the data packets themselves contain the route gleaned from the route cache maintained by the routing module.

Workload is defined in terms of *connections*. A connection is a unicast conversation between a source and a sink. The source and sink are modules associated with nodes. Several workload models are provided in MaRS. In this paper, however, we use the simplest model, similar to a datagram, where the source generates data packets destined for the sink at a steady rate. This traffic is characterized by a packet length and a random (exponentially distributed in our simulation) inter-packet generation interval. There is no flow or congestion control.

3.1 Detecting link status changes

An important feature of mobile networks is detection of link failures or appearances. This can be done in a few ways, such as periodic link status sensing/probing by so-called *hello* messages. Link layer protocols that use acknowledgments can also be used to detect link failures. Sine no link layer details are modeled, a link layer event is generated automatically whenever a link fails or reappears, i.e., a node goes out or in range. The routing protocol responds to this event. No hello messages have been modeled.

3.2 Simulation Parameters

Physical network We assume a channel bandwidth of 1.5 Mbits/sec. This data rate is similar to what is obtainable from the current generation wireless LAN products using IEEE 802.11 [8] or similar standard. Since no multiple-access contention or interference is modeled, each link essentially enjoys the entire channel bandwidth while transmitting packets. In the simulation model, a packet can be unicast (received only by a specific neighbor) or broadcast (received by all neighbors). Broadcast transmissions are modeled as a sequence of unicast transmissions on all active links of a node, but the packet is counted only once in simulation statistics. Data packets are always unicast.[3] Routing packets can be broadcast or unicast depending on the protocol requirement.

All nodes are assumed to have adequate buffer capacity for buffering packets awaiting forwarding. Data packets are processed (includes parsing the header, consulting the routing table or cache and adding the packet to the appropriate outgoing packet queue) in parallel. Data packet processing times are fixed. Routing packets have higher priority over data packets in the node's outgoing packet queue. Routing packets are processed sequentially. Routing packet processing time and routing packet sizes depend on the routing protocol being used. Data packet sizes are defined

---

[3] Passive eavesdropping may improve performance of some routing protocols such as DSR. This could be modeled using broadcast transmission. However, we did not model eavesdropping yet.

by the workload model plus a fixed, small header. However, for source routing the header length is variable and can be long depending on the length of the route. Packet processing times are estimated via independent simulation runs by timing the processing codes in the simulator itself.

Mobility Nodes move around in a rectangular region of size 1000 m × 1000 m according to a mobility model. The nodes have a constant radio range of 350m. Nodes are constantly moving, thus putting stress on the routing protocols. The node movements, however, are discretized for ease of modeling in a discrete event framework. Each node chooses a direction, speed and distance of move based on a pre-defined distribution and then computes its next position P and the time instant T of reaching that position. Similarly, a new "move" is again computed at simulation time T. A node computes its neighborhood after each such move, thus generating link failure and link repair events that in turn drive the routing protocol.

For the experiments described in this paper, the speed of each move is uniformly distributed between a given range (0.4 – 0.6 m/sec for low mobility experiments and 3.5 – 4.5 m/sec for high mobility experiments), distance is exponentially distributed with a mean of 5 m, and the direction is uniformly distributed within $[-\pi/8, +\pi/8]$ with respect to the direction of the previous move. Note that in the chosen mobility model, the nodes are always moving (*albeit* in discrete time) without stationary intervals. This presents a stress case for the routing protocols. All simulations are run for 10,000 simulated seconds.

Workload A simple workload model is used. All data packets are 512 bytes long, and interarrival times are exponentially distributed with a mean of 300 ms. There is no acknowledgment, or flow or congestion control in the workload model. Flow or congestion control mechanisms will be influenced by the routing dynamics and thus will change the load on the network. It is not clear how this will influence our performance metrics and how comparison should be made across routing protocols with very different dynamics. Thus the simplest datagram workload model has been chosen. Workload traffic is always between a pair of source and sink nodes, called a *connection*. The number of such pairs or connections is varied over a wide range in the simulation experiments. In the performance plots, it is presented in terms of no. of connections per node in the network.

4 Performance Results

We have simulated 30 and 60 node mobile, ad hoc network with respect to the above mobility and workload models. All protocols are studied with respect to three key performance metrics:

- *Fraction of packets delivered:* measured as a ratio of the number of data packets delivered to the destination and the number of data packets sent by the sender. Data packets may be dropped *en route* for two reasons: the next hop link is broken when the data packet is ready to be transmitted, or no routing table (cache) entry exists for the intended destination.

- *End-to-end delay:* measured in ms. This delay includes processing and queueing delays in each intermediate node.

- *Routing load:* measured in a normalized fashion in terms of number of bytes of routing packets transmitted per byte of data packets transmitted. The latter includes only the data packets finally delivered at the destination and not the ones that are dropped. The transmission on each hop is counted once for both routing and data packets. This gives an idea of network bandwidth consumed by routing packets with respect to "useful" data packets.

The first set of figures present the fraction of packets delivered for all protocols for low (Figures 1 and 3) and high (Figures 2 and 4) mobilities. Note the excellent behavior on the part of all link state and distance vector protocols, but considerably lower packet delivery fraction for on-demand protocols as well as for TORA. On-demand protocols (DSR and AODV) drop a considerable number of packets during the route discovery phase, as route acquisition takes time proportional to the distance between the source and destination. The situation is similar with TORA. Packet drops are fewer with proactive protocols as alternate routing table entries can always be assigned in response to link failures. In SPF, an alternate route is assigned from the current node's view of the state of all links in the network. In EXBF and DSDV, an alternate minimum cost route is found via a different neighbor. However, no such alternative is available for DSR and AODV and thus packets are dropped until route can be repaired. TORA, surprisingly, offers the lowest packet delivery fraction in spite of its multipath capability. In our observation, the key reason for this is that the initial route discoveries take longer. This affects the performance most when there is a reconnect after a network partition. In addition, in TORA wireless links have a sense of direction, which is maintained by the protocol. Because of the asynchrony in the distributed implementation, there can be short-lived inconsistencies about the sense of the direction of a link as perceived by the nodes at the end-points of this link (e.g., during link reversal). If the network is congested and the queueing delays are high, this inconsistency can persist for a while causing both delay and loss of packets. This more than offsets the advantages gained by the multipath nature of the protocol.

We also note here in passing that TORA can be quite sensitive to the loss of routing packets compared to the other protocols. Loss of certain types of routing packets (e.g., UPD, using the terminology in [28]) can put the routing tables towards an inconsistent state, which may take a while to recover. In the most recent specification of TORA [27] it is recommended that TORA be run with an encapsulation protocol called IMEP [7] that guarantees reliable, in-order delivery of routing packets. We did not, however, feel it necessary to implement IMEP in our simulation model as loss of routing packets is unlikely in the absence of any link layer model when the link is up.

Buffering data packets while route discovery in progress has a great potential to improve DSR, AODV and TORA performances. However, this alternative has not been evaluated. We also have not used early quenching of route request packets by a non-destination node in AODV. We have noticed that AODV performs very poorly by picking up stale routes, if early quenching is used. It affects both its packet delivery and delay performance significantly. DSR, on the other hand, does not demonstrate any significant performance differential with or without the use of early quenching.

AODV has a slightly worse packet delivery performance than DSR because of higher drop rates. AODV uses route expiry, dropping some packets when a route expires and a new route must be found. This, however, gives better quality routes to AODV in general.

The average end-to-end delays are shown in Figures 5, 6, 7 and 8 for both network sizes and for low and high mobilities. The shortest path protocols (SPF, EXBF and DSDV) show the minimum delay characteristics. AODV and DSR show worse characteristics as their routes are typically not the shortest. Even if the initial route discovery phase finds the shortest route (it typically will), the route may not remain the shortest over a period of time due to node mobility. Also, note that in AODV routes are maintained as a soft state, i.e., routes expire after a timeout interval and fresh route discovery is initiated. Accordingly, AODV performs a little better delay-wise and can possibly do even better with some fine-tuning of this timeout period by making it, for example, a function of node mobility. TORA has the worst delay characteristics because of the loss of distance information with progress.

The routing load characteristics shown in Figures 9, 10, 11 and 12 are interesting. Note that the routing load varies over a very wide range and hence the plots use a logarithmic scale for the vertical axis. SPF expends significantly more routing load than the other protocols. The distance vector protocols, EXBF and DSDV, have very similar routing loads, and much lower than SPF. DSR and AODV perform very well, particularly for smaller number of connections, with DSR often outperforming AODV. The excellent routing load performance of DSR is due to the optimizations possible by virtue of source routing. TORA's performance is not very competitive with the distance vector and on-demand protocols. We conjecture that it is due to the fact network partitions cause TORA to do substantial work to erase routes even when those routes are not in use.

It appears that the theoretical "worst" case communication complexity (number of messages required to adapt to a link failure/recovery) does not provide much insight into the average case behavior obtained via simulation. For example, SPF has a worst case communication complexity of $O(2e)$ [20], where e is the number of links in the network. On the other hand communication complexity in protocols based on distributed Bellman-Ford, such as EXBF and DSDV, is exponential in the number of nodes N in the network [4], which should be much higher. However, SPF has a much higher routing load in the simulations.

Most protocols benefit to some degree as the number of connections grow large. This is because a single route repair can potentially benefit many connections. Thus routing load do not increase as much as the data load with increasing number of connections. This effect is the most pronounced for proactive, shortest-path protocols where routing load is independent of the data load. Also note that the routing load differentials between the protocols become smaller as the number of connections grow large. Thus the proactive, distance vector protocols may be favored at large number of connections as they provide better packet delivery fractions and end-to-end delay characteristics. We also note in the passing, that DSR uses somewhat more bandwidth (10–20% in our experiments) because of source routing that increases the size of the header in data packets. Even counting this in as a part of routing load DSR is very competitive with AODV. However, this bandwidth usage is expected to increase for larger networks and may make DSR less attractive.

5 Related Work

Some simulation studies of the routing protocols evaluated here have been presented earlier in the literature. However, in our knowledge they have not yet been thoroughly evaluated against each other in the same framework.[4] In [26] Park and Corson compared TORA with an ideal link-state routing protocol and demonstrated superior performance of TORA. In their chosen network model, however, there is no true node mobility. The network is connected in a "honeycomb" pattern and links go up and down at some rate with the average network connectivity held constant artificially. In an earlier work [6], Corson and Ephremides presented simulation results of three protocols on which TORA is based in part. Here again stationary networks are chosen with links going up and down at random intervals.

Johnson and Maltz presented the simulation study of DSR in [18]. They simulated a true, mobile network. Their results indicate that DSR is able to find close to optimal routes on an

---

[4] We are, however, aware of a recently completed work on comparative simulation study of the ad hoc routing protocols in CMU [3], which is similar in spirit as ours. But details are not available to us at the time of this writing.

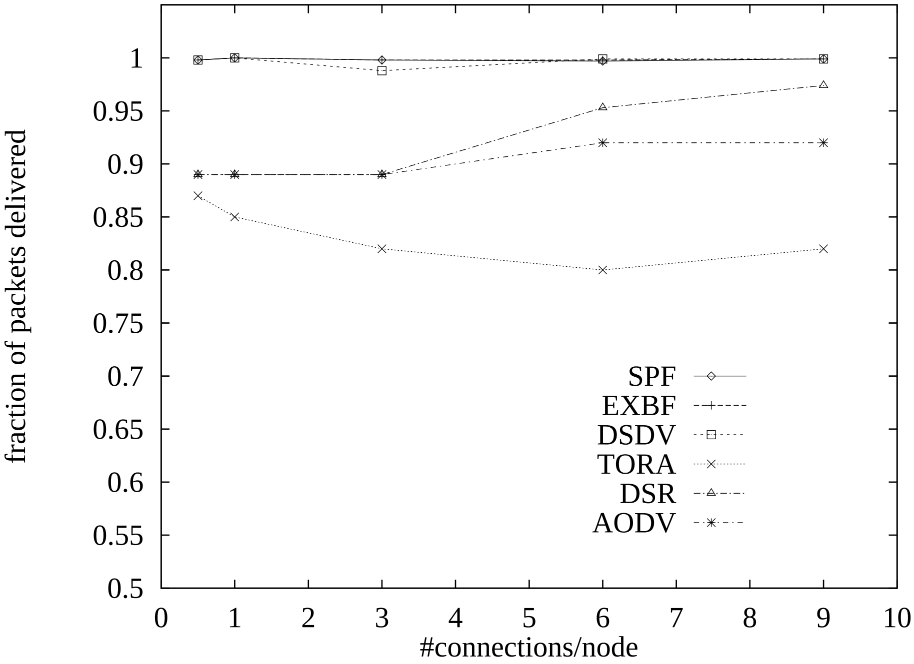

Figure 1: Fraction of packets delivered in the 30 node network for all routing protocols for the low mobility case.

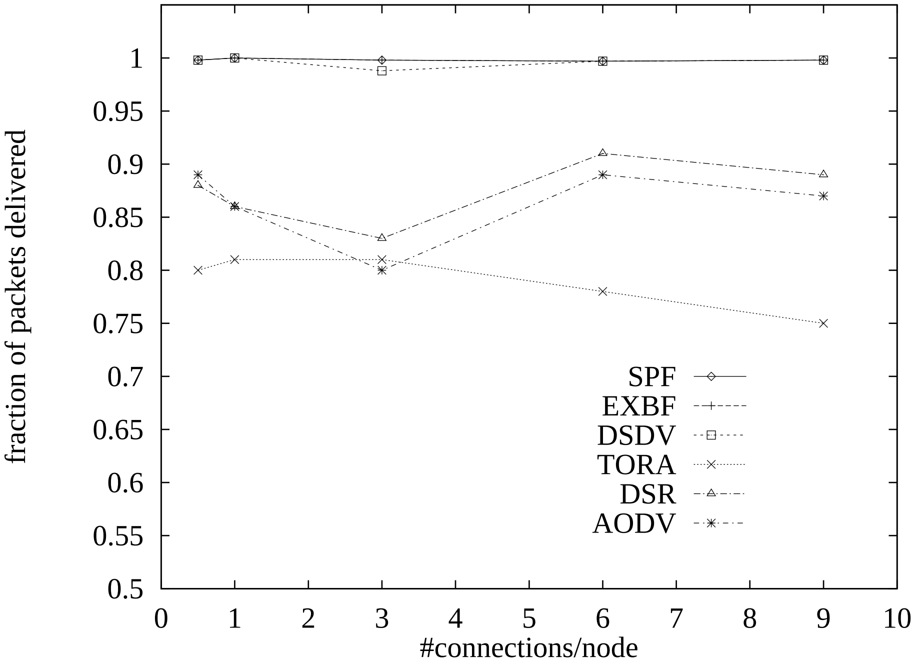

Figure 2: Fraction of packets delivered in the 30 node network for all routing protocols for the high mobility case.

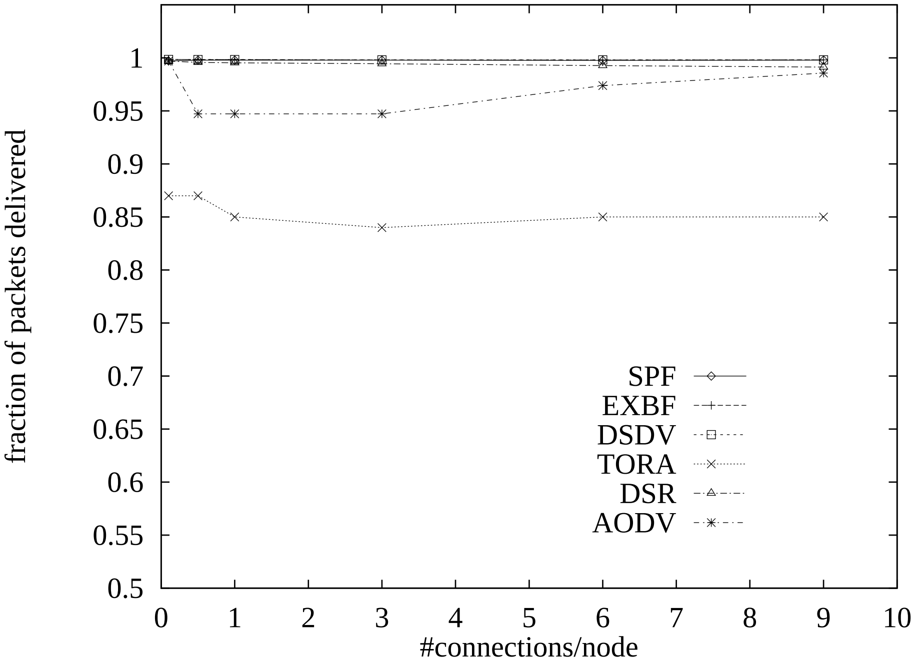

Figure 3: Fraction of packets delivered in the 60 node network for all routing protocols for the low mobility case.

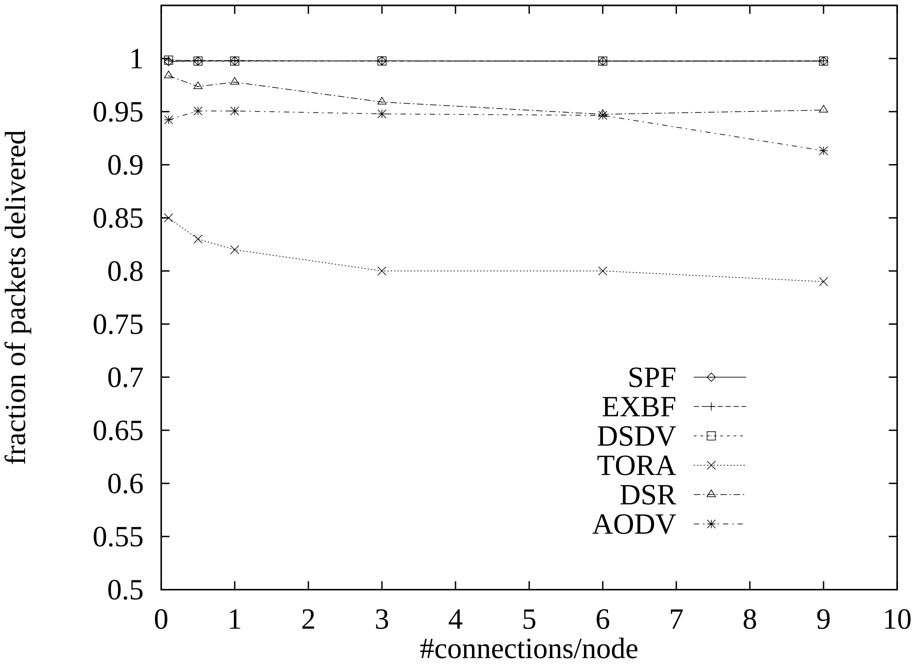

Figure 4: Fraction of packets delivered in the 60 node network for all routing protocols for the high mobility case.

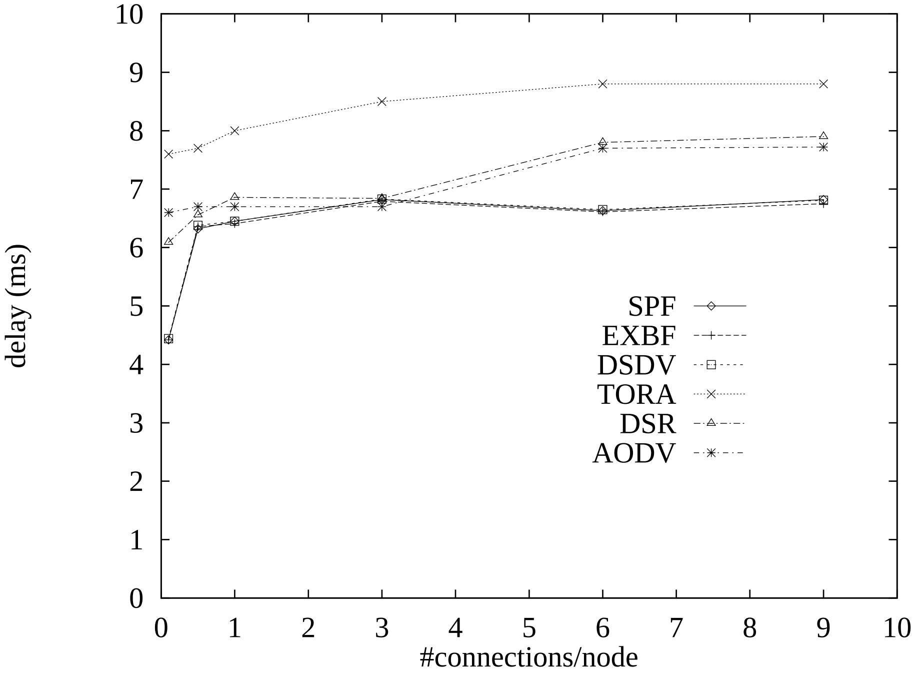

Figure 5: Average end-to-end delay in the 30 node network for all routing protocols for the low mobility case.

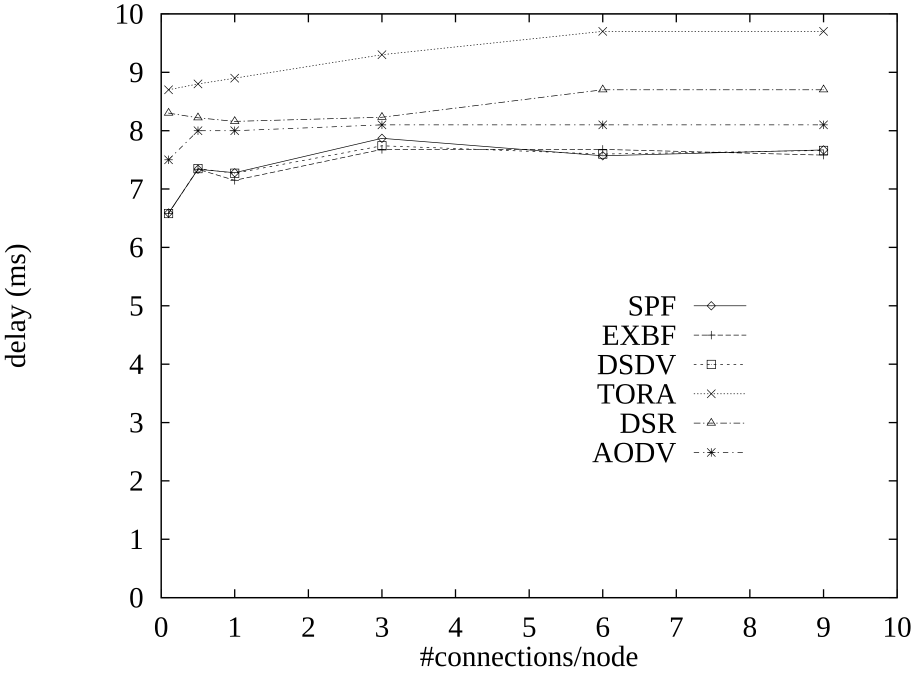

Figure 6: Average end-to-end delay in the 30 node network for all routing protocols for the high mobility case.

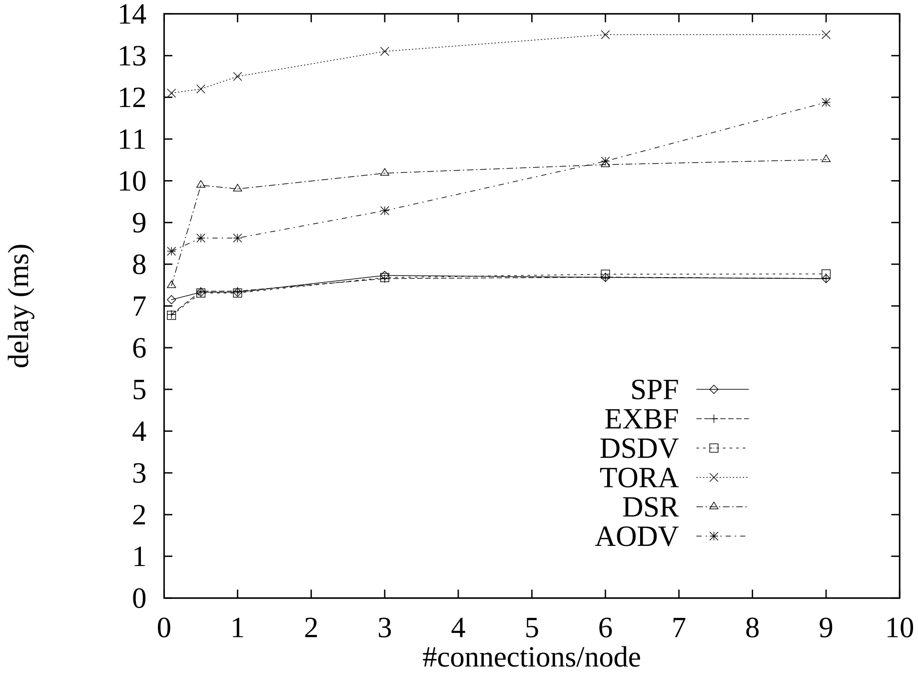

Figure 7: Average end-to-end delay in the 60 node network for all routing protocols for the low mobility case.

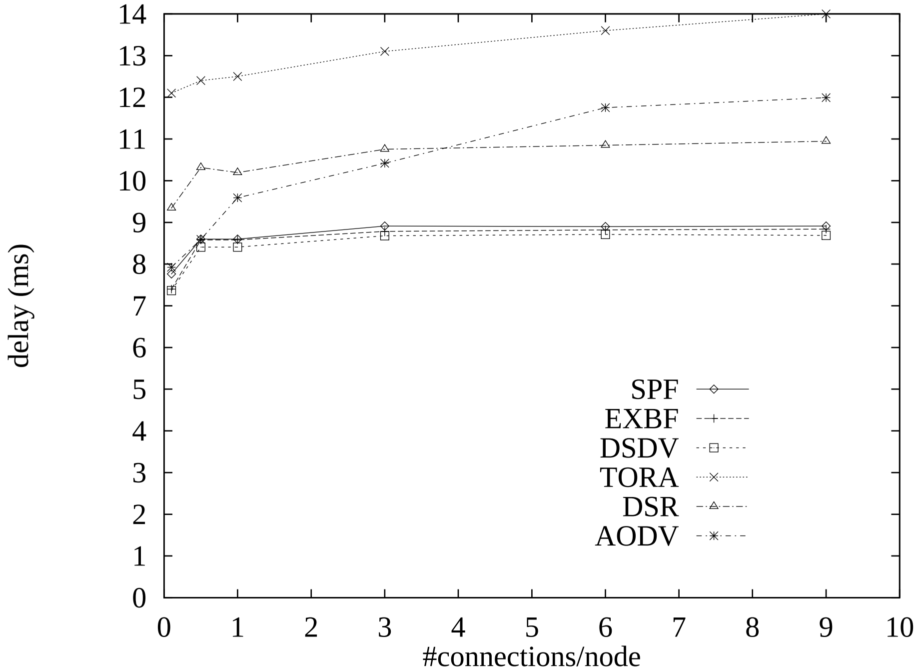

Figure 8: Average end-to-end delay in the 60 node network for all routing protocols for the high mobility case.

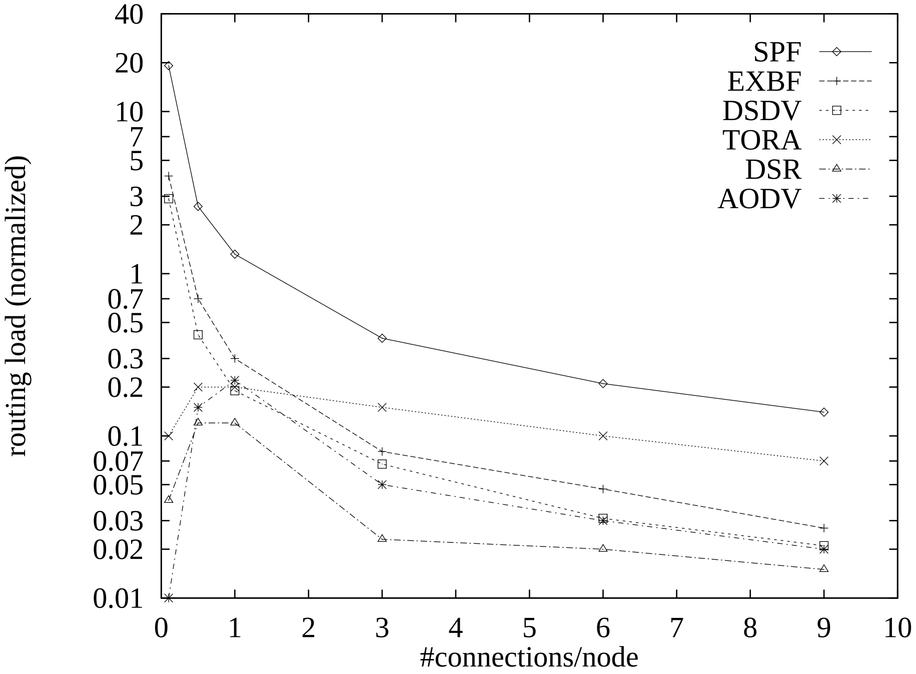

Figure 9: Normalized routing load in the 30 node network for all routing protocols for the low mobility case.

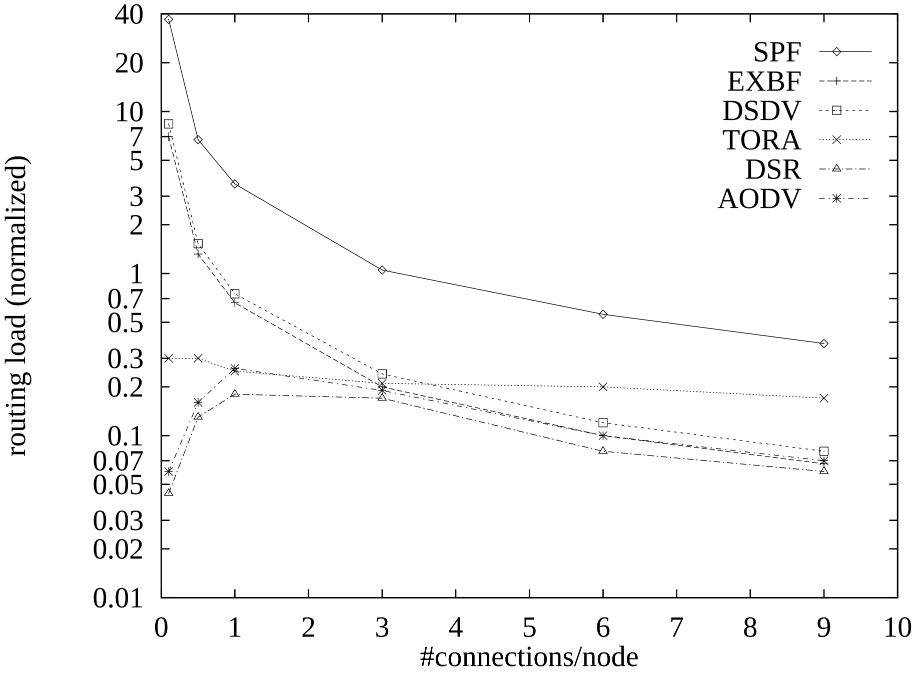

Figure 10: Normalized routing load in the 30 node network for all routing protocols for the high mobility case.

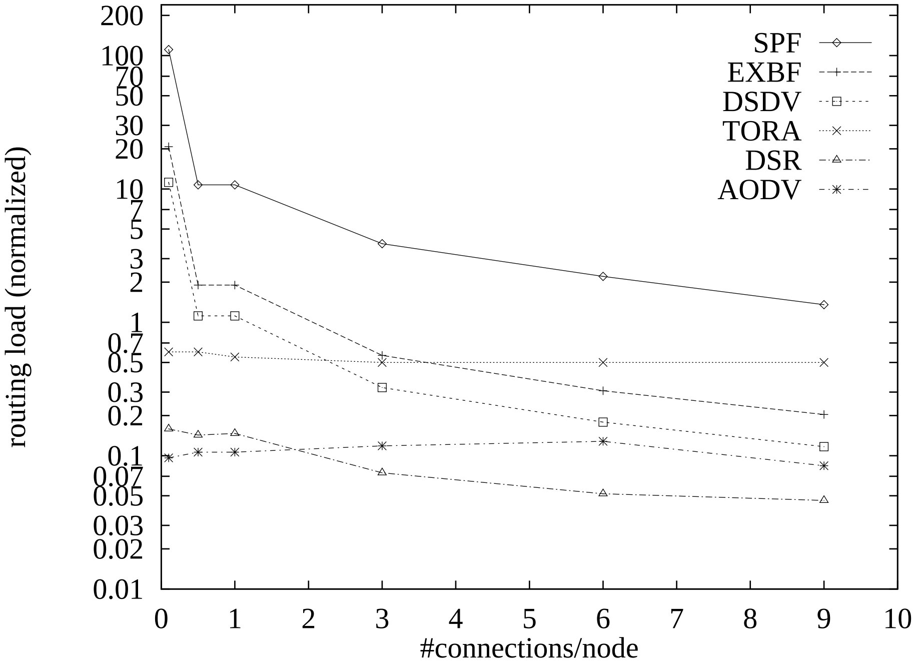

Figure 11: Normalized routing load in the 60 node network for all routing protocols for the low mobility case.

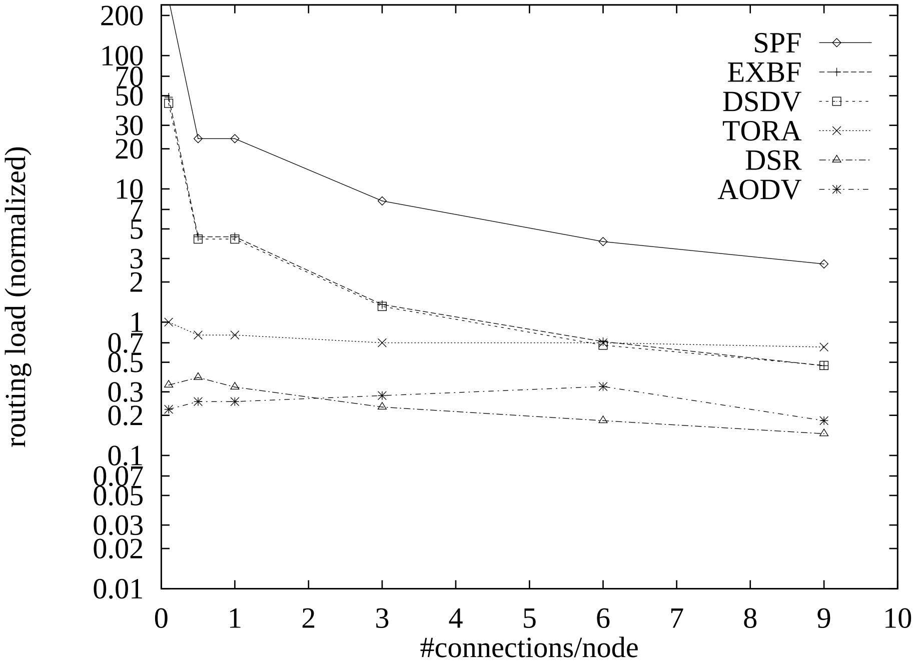

Figure 12: Normalized routing load in the 60 node network for all routing protocols for the high mobility case.

average. However, actual path lengths in hops (and not the end-to-end delay) was measured to determine optimality. Freisleben and Jansen evaluated DSR against DSDV in [10]. They built a more comprehensive simulator complete with a MAC layer model of the IEEE standard 802.11 [8] as well as true mobility. Only packet delivery fraction is evaluated with DSDV performing marginally worse than DSR.

There is a large body of simulation study of shortest-path protocols. Most closely related to our work is the performance study in [33] by Shakar et. al., which used both SPF and EXBF protocols, among others, for a performance comparison using the MaRS simulator. They used a static network with dynamically changing link connectivity with link going up and down according to a stochastic distribution. Various traffic and link failure models were studied to evaluate transient and steady-state performance of the routing protocols. Delay and throughput performance of SPF and EXBF were found to be equivalent in the most part.

6 Conclusions

Our work is the first attempt towards a comprehensive performance evaluation of routing protocols for mobile, ad hoc networks. We evaluated all but one protocol[5] currently (Summer 1998) considered in the IETF MANET working group, in addition to more traditional link state and distance vector protocols. Steady state performance in terms or fraction of packets delivered, delay and routing load have been considered as the performance metrics.

Even with a packet-level simulation model the essential aspects of the routing protocols are exposed. The key observations are as follows. The proactive, shortest path protocols provide excellent performance in terms of end-to-end delays and packet delivery fraction, however, at the cost of higher routing load. The on-demand protocols suffer from sub-optimal routes as well as worse packet delivery fraction because of more dropped data packets. However, they are significantly more efficient in terms of the routing load. The multipath protocol, TORA, did not perform well in spite of maintaining multiple redundant path. The overhead of finding and maintaining multiple paths seems to outweigh the benefits. Also, the end-to-end delay performance is poor because of the loss of distance information. The routing load differentials between all routing protocols reduce with large number of peer-to-peer conversations in the network. However, the other performance differentials are not affected conclusively. Rate of mobility and network size do not seem to affect the performance beyond what is normally expected – such as higher routing load, more delay and dropped packets.

---

[5] As mentioned earlier the Zone Routing Protocol (ZRP) [15] has not been evaluated.

It is important to note the limitations of the study. First, a packet level simulation has its own limitations. No MAC protocol or multiple access interference is modeled. Thus a high routing packet load does not interfere as much with the data transmissions (except for queueing delays) as it would in reality. Also, there are no transmission errors. The current study best reflects the performance when all active links in the network are on a separate frequency band. Second, only a moderate size network has been studied. Though it is unclear what sizes will be realistic for an ad hoc network running IP based protocols, using a few other sizes, going upto a few hundreds, will provide more maturity to the study. Third, a few different traffic models, for example, dynamically changing peers for conversations and introduction of hot-spots should be studied to evaluate the sensitivity to traffic models. Fourth, fine tuning of certain protocol parameters (e.g., various timeout periods for the on-demand protocols) is possible with changing mobility and traffic characteristics. We have used reasonable values that work well, but have not changed the values for different traffic and mobility. Also, certain protocol specific optimizations (e.g., passive eavesdropping in DSR) as well as more general optimizations (e.g., buffering of data packets on route loss until route is repaired) are possible. They may impact relative performance. Fifth, impact of memory usage by the protocols have been ignored. This may be important as the computing nodes deployed in a mobile, ad hoc environment can be low power and small size devices. In spite of these limitations, we have gained valuable insight into the behavior of routing algorithms in an ad hoc network. Our future work will address these limitations.

Acknowledgments

This work is partially supported by AFOSR grant no. F49260-96-1-0472, Texas Advanced Technology Program grant no. 010115-248b, NSF MII grant no. CDA-9633299 and NSF CAREER award no. ACI-9733836.

References

[1] C. Alaettinoglu, A. U. Shankar, K. Dussa-Zieger, and I. Matta. Design and implementation of MaRS: A routing testbed. *Journal of Internetworking: Research and Experience*, 5(1):17–41, 1994.

[2] D. Bertsekas and R. Gallager. *Data Networks*. Prentice-Hall, 1987.

[3] 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. In *Proceedings of the 4th International*

Conference on Mobile Computing and Networking (ACM MOBICOM'98), October 1998. to apper.

[4] C. Cheng, R. Riley, and S. P. R. Kumar. A loop-free extended Bellman-Ford routing protocol without bouncing effect. In *Proc. of ACM SIGCOMM Conf.*, pages 224–236, 1989.

[5] M.S. Corson, S. Batsell, and J. Macker. Architectural considerations for mobile mesh networking. `http://tonnant.itd.nrl.navy.mil/mmnet/ mmnetRFC.txt`, May 1996. Request for Comments Draft.

[6] M.S. Corson and A. Ephremides. A distributed routing algorithm for mobile wireless networks. *Wireless Networks*, 1(1):61–81, 1995.

[7] S. Corson, S. Papademetriou, P. Papadopoulos, V. Park, and A. Qayyum. An internet MANET encapsulation protocol (IMEP) specification. `http://www.ietf.org/internet-drafts/draft-ietf-manet-imep-spec-01.txt`, 1998. Internet Draft.

[8] IEEE Standards Department. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE standard 802.11–1997, 1997.

[9] E. W. Dijkstra. A note on two problems in connection with graphs. *Numerical Mathematics*, 1:269–271, Oct. 1959.

[10] B. Freisleben and R. Jansen. Analysis of routing protocols for ad hoc networks of mobile computers. In *Proceedings of the 15th IASTED International Conference on Applied Informatics*, pages 133–136, Innsbruck, Austria, 1997. Iasted-Acta Press.

[11] E. Gafni and D. Bertsekas. Distributed algorithms for generating loop-free routes in networks with frequently changing topology. *IEEE Trans. on Communication*, 29(1):11–18, January 1981.

[12] J. J. Garcia-Luna-Aceves. A unified approach to loop free routing using distance vectors or link states. In *Proc. of ACM SIGCOMM'89 Conf.*, pages 212–223, Sept. 1989.

[13] M. Gerla and J.T.-C. Tsai. Multicluster, mobile, multimedia radio networks. *Wireless Networks*, 1(3):255–265, 1995.

[14] Z. Haas. A new routing protocol for the reconfigurable wireless networks. In *Proc. of the IEEE Int. Conf. on Universal Personal Communications.*, Oct 1997.

[15] Z. J. Haas and M. R. Pearlman. The zone routing protocol (ZRP) for ad hoc networks. `http://www. ietf.org/internet-drafts/draft-ietf- manet-zone-zrp-00.txt`, 1997. IETF Internet Draft.

[16] Z. J. Haas and M. R. Pearlman. The performance of query control schemes for the zone routing protocol. In *Proceedings of ACM SIGCOMM'98 Conference*, pages 167–177, Vancouver, Sept. 1998.

[17] C. Hedrick. Routing information protocol. RFC 1058, June 1988.

[18] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile computing*. Kluwer Academic, 1996.

[19] John Jubin and Janet D. Tornow. The DARPA packet radio network protocols. *Proceedings of the IEEE*, 75(1):21–32, January 1987.

[20] S. Keshav. *An Engineering Approach to Computer Networking : ATM Networks, the Internet, and the Telephone Network*, chapter 11. Addison-Wesley, 1997.

[21] A. Khanna and J. Zinky. A revised ARPANET routing metric. In *Proc. of the ACM SIGCOMM'89 Conf.*, pages 45–56, Sept. 1989.

[22] P. Krishna, N. H. Vaidya, M. Chatterjee, and D. K. Pradhan. A cluster-based approach for routing in dynamic networks. *ACM SIGCOMM Computer Communication Review*, pages 49–65, April 1997.

[23] J. Macker and S. Corson. Mobile ad hoc networks (MANET). `http://www.ietf.org/html.charters/manet-charter.html`, 1997. IETF Working Group Charter.

[24] J. Moy. OSPF version 2. RFC 1247, July 1991.

[25] S. Murthy and J. J. Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. *Mobile Networks and Applications*, 1(2):183–197, Oct. 1996.

[26] V. Park and S. Corson. A performance comparison of TORA and ideal link state routing. In *Proceedings of IEEE Symposium of Computers and Communication*, June 1998.

[27] V. Park and S. Corson. Temporally ordered routing algorithm (TORA) version 1, functional specification. `http://www.ietf.org/ internet-drafts/ draft-ietf-manet -tora-spec-01. txt`, 1998. IETF Internet Draft.

[28] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM'97 Conf.*, April 1997.

[29] Charles Perkins. Ad hoc on demand distance vector (AODV) routing. `http://www.ietf.org/internet-drafts/ draft-ietf-manet-aodv-00.txt`, 1997. IETF Internet Draft.

[30] Charles E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the ACM SIGCOMM '94 Conference*, pages 234–244, August 1994.

[31] Y. Rekhter and T. Li. A Border Gateway Protocol-4. RFC 827, March 1995.

[32] N. Schacham and J. Westcott. Future directions in packet radio architectures and protocols. *Proceedings of the IEEE*, 75(1):83–99, Jan 1987.

[33] A. U. Shankar, C. Alaettinoglu, K. Dussa-Zieger, and I. Matta. Transient and steady-state performance of routing protocols: Distance-vector versus link-state. *Journal of Internetworking: Research and Experience*, 6:59–87, 1995. (Preliminary version appeared in Proc. ACM SIGMETRICS/PERFORMANCE Conf., 1992, pages 181-192.).