Abstract

Ad hoc networks are characterized by multihop wireless connectivity, frequently changing network topology and the need for efficient dynamic routing protocols. We compare the performance of two prominent on-demand routing protocols for mobile ad hoc networks: Dynamic Source Routing (DSR) and Ad Hoc On-Demand Distance Vector Routing (AODV). A detailed simulation model with MAC and physical layer models is used to study interlayer interactions and their performance implications. We demonstrate that even though DSR and AODV share similar on-demand behavior, the differences in the protocol mechanics can lead to significant performance differentials. The performance differentials are analyzed using varying network load, mobility, and network size. Based on the observations, we make recommendations about how the performance of either protocol can be improved.

Performance Comparison of Two On-Demand Routing Protocols for Ad Hoc Networks

CHARLES E. PERKINS, NOKIA RESEARCH CENTER
ELIZABETH M. ROYER, UNIVERSITY OF CALIFORNIA
SAMIR R. DAS AND MAHESH K. MARINA, UNIVERSITY OF CINCINNATI

In an *ad hoc* network, mobile nodes communicate with each other using multihop wireless links. There is no stationary infrastructure; for instance, there are no base stations. Each node in the network also acts as a router, forwarding data packets for other nodes. A central challenge in the design of ad hoc networks is the development of dynamic routing protocols that can efficiently find routes between two communicating nodes. The routing protocol must be able to keep up with the high degree of node mobility that often changes the network topology drastically and unpredictably. Such networks have been studied in the past in relation to defense research, often under the name of *packet radio networks* [1]. Recently there has been a renewed interest in this field due to the common availability of low-cost laptops and palmtops with radio interfaces. Interest is also partly fueled by growing enthusiasm in running common network protocols in dynamic wireless environments without the requirement of specific infrastructures. A *mobile ad hoc networking* (MANET) working group [2] has also been formed within the Internet Engineering Task Force (IETF) to develop a routing framework for IP-based protocols in ad hoc networks.

Our goal is to carry out a systematic performance study of two dynamic routing protocols for ad hoc networks: the *Dynamic Source Routing* protocol (DSR) [3, 4] and the *Ad Hoc On-Demand Distance Vector* protocol (AODV) [5, 6]. DSR and AODV share an interesting common characteristic — they both initiate routing activities on an *on demand* basis. This *reactive* nature of these protocols is a significant departure from more traditional *proactive* protocols, which find routes between all source-destination pairs regardless of the use or need for such routes. The key motivation behind the design of on-demand protocols is the reduction of the routing load. High routing load usually has a significant performance impact in low-bandwidth wireless links.

While DSR and AODV share the on-demand behavior [7] in that they initiate routing activities only in the presence of data packets in need of a route, many of their routing mechanics are very different. In particular, DSR uses source routing, whereas AODV uses a table-driven routing framework and destination sequence numbers. DSR does not rely on any timer-based activities, while AODV does to a certain extent. One of our goals in this study is to extract the relative merits of these mechanisms. The motivation is that a better understanding of the relative merits will serve as a cornerstone for development of more effective routing protocols for mobile ad hoc networks.

The rest of the article is organized as follows. In the following section, we briefly review the DSR and AODV protocols. We present a detailed critique of the two protocols, focusing on the differences in their dynamic behaviors that can lead to performance differences. This lays the foundation for much of the context of the performance study. We describe the simulation environment. We present the simulation results, followed by their interpretations. Related work is presented. We finally draw conclusions and also make recommendations for the improved design of either protocol.

An earlier version of this article appeared at the IEEE INFOCOM 2000 conference.

A Description of the Protocols

DSR

The key distinguishing feature of DSR [3, 4] is the use of *source routing*. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a *route cache*. The data packets carry the source route in the packet header.

When a node in the ad hoc network attempts to send a data packet to a destination for which it does not already

 1070-9916/01/$10.00 © 2001 IEEE

know the route, it uses a *route discovery* process to dynamically determine such a route. Route discovery works by flooding the network with *route request* (RREQ) packets. Each node receiving an RREQ rebroadcasts it, unless it is the destination or it has a route to the destination in its route cache. Such a node replies to the RREQ with a *route reply* (RREP) packet that is routed back to the original source. RREQ and RREP packets are also source routed. The RREQ builds up the path traversed across the network. The RREP routes itself back to the source by traversing this path backward.[1] The route carried back by the RREP packet is cached at the source for future use.

If any link on a source route is broken, the source node is notified using a *route error* (RERR) packet. The source removes any route using this link from its cache. A new route discovery process must be initiated by the source if this route is still needed.

DSR makes very aggressive use of source routing and route caching. No special mechanism to detect routing loops is needed. Also, any forwarding node caches the source route in a packet it forwards for possible future use. Several additional optimizations have been proposed and have been evaluated to be very effective by the authors of the protocol [7], as described in the following:

- *Salvaging*: An intermediate node can use an alternate route from its own cache when a data packet meets a failed link on its source route.
- *Gratuitous route repair*: A source node receiving an RERR packet piggybacks the RERR in the following RREQ. This helps clean up the caches of other nodes in the network that may have the failed link in one of the cached source routes.
- *Promiscuous listening*: When a node overhears a packet not addressed to itself, it checks whether the packet could be routed via itself to gain a shorter route. If so, the node sends a *gratuitous* RREP to the source of the route with this new, better route. Aside from this, promiscuous listening helps a node to learn different routes without directly participating in the routing process.

AODV

AODV [5, 6] shares DSR's on-demand characteristics in that it also discovers routes on an *as needed* basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate an RREP back to the source and, subsequently, to route data packets to the destination. AODV uses sequence numbers maintained at each destination to determine freshness of routing information and to prevent routing loops [5]. These sequence numbers are carried by all routing packets.

An important feature of AODV is the maintenance of timer-based states in each node, regarding utilization of individual routing table entries. A routing table entry is *expired* if not used recently. A set of predecessor nodes is maintained for each routing table entry, indicating the set of neighboring nodes which use that entry to route data packets. These nodes are notified with RERR packets when the next-hop link breaks. Each predecessor node, in turn, forwards the RERR to its own set of predecessors, thus effectively erasing all routes using the broken link. In contrast to DSR, RERR packets in AODV are intended to inform all sources using a link when a failure occurs. Route error propagation in AODV can be visualized conceptually as a tree whose root is the node at the point of failure and all sources using the failed link as the leaves.

The recent specification of AODV [6] includes an optimization technique to control the RREQ flood in the route discovery process. It uses an *expanding ring search* initially to discover routes to an unknown destination. In the expanding ring search, increasingly larger neighborhoods are searched to find the destination. The search is controlled by the Time-To-Live (TTL) field in the IP header of the RREQ packets. If the route to a previously known destination is needed, the prior hop-wise distance is used to optimize the search. This enables computing the TTL value used in the RREQ packets dynamically, by taking into consideration the temporal locality of routes.

A Critique of DSR and AODV

The two on-demand protocols share certain salient characteristics. In particular, they both discover routes only when data packets lack a route to a destination. Route discovery in either protocol is based on query and reply cycles, and route information is stored in all intermediate nodes along the route in the form of route table entries (AODV) or in route caches (DSR). However, there are several important differences in the dynamics of these two protocols, which may give rise to significant performance differentials.

First, by virtue of source routing, DSR has access to a significantly greater amount of routing information than AODV. For example, in DSR, using a single request-reply cycle, the source can learn routes to each intermediate node on the route in addition to the intended destination. Each intermediate node can also learn routes to every other node on the route. Promiscuous listening of data packet transmissions can also give DSR access to a significant amount of routing information. In particular, it can learn routes to every node on the source route of that data packet. In the absence of source routing and promiscuous listening, AODV can gather only a very limited amount of routing information. In particular, route learning is limited only to the source of any routing packets being forwarded. This usually causes AODV to rely on a route discovery flood more often, which may carry significant network overhead.

Second, to make use of route caching aggressively, DSR replies to *all* requests reaching a destination from a single request cycle. Thus, the source learns many alternate routes to the destination, which will be useful in the case that the primary (shortest) route fails. Having access to many alternate routes saves route discovery floods, which is often a performance bottleneck. However, there may be a possibility of a route reply flood. In AODV, on the other hand, the destination replies only once to the request arriving first and ignores the rest. The routing table maintains at most one entry per destination.

Third, the current specification of DSR does not contain any explicit mechanism to expire stale routes in the cache, or prefer "fresher" routes when faced with multiple choices. As noted in [7], stale routes, if used, may start polluting other caches. Some stale entries are indeed deleted by route error packets. But because of promiscuous listening and node mobility, it is possible that more caches are polluted by stale entries than are removed by error packets. In contrast, AODV has a much more conservative approach than DSR. When faced with two choices for routes, the fresher route (based on destination sequence numbers) is always chosen. Also, if a routing table entry is not used recently, the entry is expired. The latter technique is not problem-free, however. It is possible to expire valid routes this way if unused beyond an

[1] A variation of this mechanism is needed for ad hoc networks with unidirectional links. However, here we limit our discussions to bidirectional links.

expiry time. Determination of a suitable expiry time is difficult, because sending rates for sources, as well as node mobility, may differ widely and can change dynamically. In a recent paper [8], the effects of various design choices in caching strategies for on-demand routing protocols are analyzed.

Fourth, the route deletion activity using RERR is also conservative in AODV. By way of a predecessor list, the error packets reach *all* nodes using a failed link on its route to any destination. In DSR, however, a route error simply backtracks the data packet that meets a failed link. Nodes that are not on the upstream route of this data packet but use the failed link are not notified promptly.

The goal of our simulations that follow is to determine the relative merits of the aggressive use of source routing and caching in DSR, and the more conservative routing table and sequence-number-driven approach in AODV.

The Simulation Model

We use a detailed simulation model based on *ns-2* [9] in our evaluation. In a recent paper [10], the Monarch research group at Carnegie-Mellon Unviersity developed support for simulating multihop wireless networks complete with physical, data link, and medium access control (MAC) layer models on *ns-2*. The Distributed Coordination Function (DCF) of IEEE 802.11 [11] for wireless LANs is used as the MAC layer protocol. The 802.11 DCF uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets [12] for "unicast" data transmission to a neighboring node. The RTS/CTS exchange precedes data packet transmission and implements a form of *virtual carrier sensing* and channel reservation to reduce the impact of the well-known *hidden terminal problem* [13]. Data packet transmission is followed by an ACK. "Broadcast" data packets and the RTS control packets are sent using physical carrier sensing. An unslotted carrier sense multiple access (CSMA) technique with collision avoidance (CSMA/CA) is used to transmit these packets [11]. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [14, 15]. WaveLAN is modeled as a shared-media radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 m. A detailed description of the simulation environment and the models is available in [9, 10].

The implementations of AODV and DSR in our simulation environment closely match their specifications ([6, 3], respectively). The routing protocol model "detects" all data packets transmitted or forwarded, and "responds" by invoking routing activities as appropriate. The RREQ packets are treated as broadcast packets in the MAC. RREP and data packets are all unicast packets with a specified neighbor as the MAC destination. RERR packets are treated differently in the two protocols. They are broadcast in AODV and use unicast transmissions in DSR. Both protocols detect link breaks using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop. This is indicated, for example, by the failure to receive a CTS after a specified number of RTS retransmissions, or the absence of an ACK following data transmission. No additional network layer mechanism such as *hello messages* [5] is used.

Both protocols maintain a *send buffer* of 64 packets. It contains all data packets waiting for a route, such as packets for which route discovery has started, but no reply has arrived yet. To prevent buffering of packets indefinitely, packets are dropped if they wait in the send buffer for more than 30 s. All packets (both data and routing) sent by the routing layer are queued at the *interface queue* until the MAC layer can transmit them. The interface queue has a maximum size of 50 packets and is maintained as a priority queue with two priorities each served in FIFO order. Routing packets get higher priority than data packets.

The Traffic and Mobility Models

We use traffic and mobility models similar to those previously reported using the same simulator [10, 16]. Traffic sources are continuous bit rate (CBR). The source-destination pairs are spread randomly over the network. Only 512-byte data packets are used. The number of source-destination pairs and the packet sending rate in each pair is varied to change the offered load in the network.

The mobility model uses the *random waypoint* model [10] in a rectangular field. Two field configurations are used:

- 1500 m x 300 m field with 50 nodes
- 2200 m x 600 m field with 100 nodes[2]

Here, each packet starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0–20 m/s).[3] Once the destination is reached, another random destination is targeted after a pause. We vary the pause time, which affects the relative speeds of the mobiles. Simulations are run for 900 simulated s for 50 nodes, and 500 simulated s for 100 nodes. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. Identical mobility and traffic scenarios are used across protocols.

Performance Results

Performance Metrics

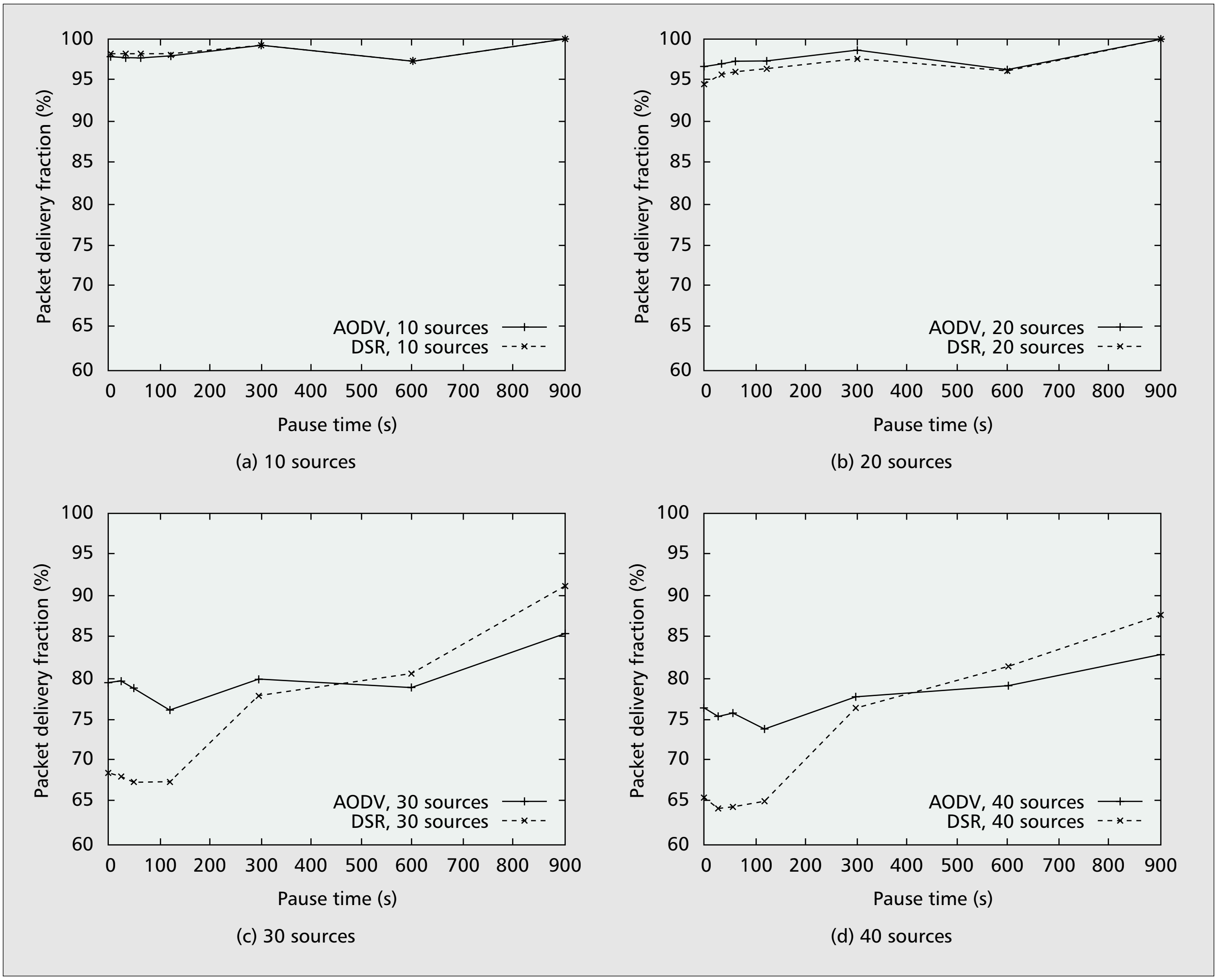
Four important performance metrics are evaluated:

- *Packet delivery fraction* — The ratio of the data packets delivered to the destinations to those generated by the CBR sources; also, a related metric, received throughput (in kilobits per second) at the destination has been evaluated in some cases.
- *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 times.
- *Normalized routing load* — The number of routing packets transmitted per data packet delivered at the destination. Each hop-wise transmission of a routing packet is counted as one transmission.
- *Normalized MAC load* — The number of routing, Address Resolution Protocol (ARP), and control (e.g., RTS, CTS, ACK) packets transmitted by the MAC layer for each delivered data packet. Essentially, it considers both routing overhead and the MAC control overhead. Like normalized routing load, this metric also accounts for transmissions at every hop.

The first two metrics are the most important for best-effort traffic. The routing load metric evaluates the efficiency of the routing protocol. Finally, the MAC load is a measure of effective utilization of the wireless medium by data traffic. Note, however, that these metrics are not completely independent. For example, lower packet delivery fraction means that the delay metric is evaluated with fewer samples. In the conventional wisdom, the longer the path lengths, the higher the

[2] *The slow simulation speed and large memory requirement of the ns-2 models prevented us from using larger networks at this point. We are currently working on optimizing the models to improve scalability.*

[3] *Note that this is a fairly high speed for an ad hoc network, comparable to traffic speeds inside a city.*



■ **Figure 1.** *Packet delivery fractions for the 50-node model with various numbers of sources.*

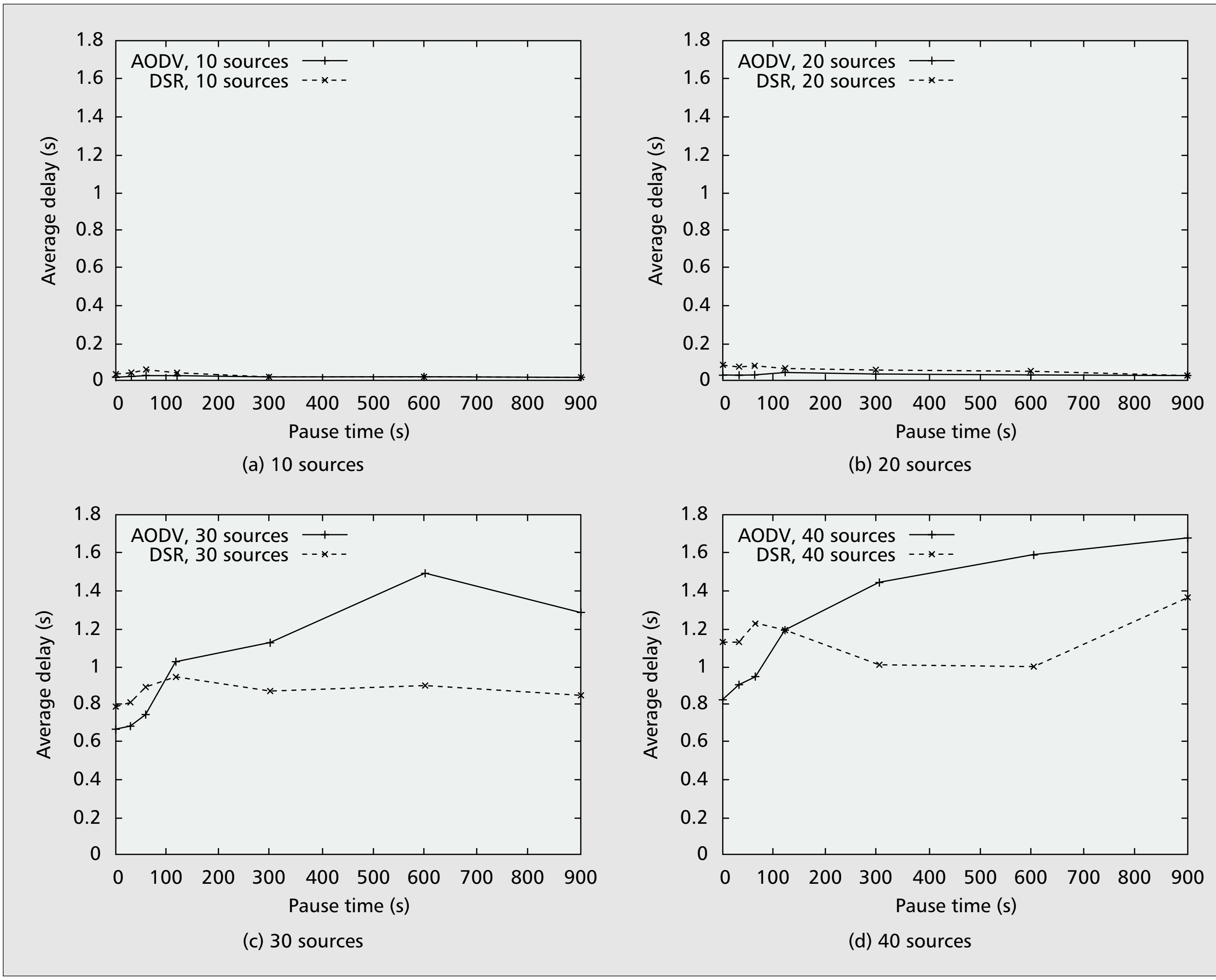
probability of a packet drop. Thus, with a lower delivery fraction, samples are usually biased in favor of smaller path lengths and thus have less delay. Also, low routing and MAC load impact both delivery fraction and delay, since network congestion and multiple access interference are reduced. Finally, MAC load also includes routing load.

Varying Mobility and Number of Sources

The first set of experiments uses differing numbers of sources with a moderate packet rate and varying pause times. For the 50 node experiments we used 10, 20, 30, and 40 traffic sources and a packet rate of 4 packets/s, except for 40 sources, which use 3 packets/s. We used a slower rate with 40 sources because the network congestion was too high otherwise for a meaningful comparison. The higher rates will be considered in the next subsection. The packet delivery fractions for DSR and AODV are very similar with 10 and 20 sources (Fig. 1a and b). With 30 and 40 sources, AODV outperforms DSR by about 15 percent (Fig. 1c and d) at lower pause times (higher mobility). For higher pause times (low mobility), however, DSR has a better delivery fraction than AODV. The relative performance of both protocols with respect to delays is similar to that with delivery fractions. DSR and AODV have almost identical delays with 10 and 20 sources (Fig. 2a and b). With 30 and 40 sources, AODV has about 25 percent lower delay than DSR (Fig. 2c and d) for lower pause times. But for higher pause times, DSR has better (30–40 percent lower) delay than AODV. Detailed interpretations of the results presented in this section are provided later.

In all cases, DSR demonstrates significantly lower routing load than AODV (Fig. 3), usually by a factor of 2–3, with the factor increasing with a growing number of sources. Also, note that relative to AODV, DSR's normalized routing load is fairly stable with an increasing number of sources, even though its delivery and delay performance get increasingly worse. A relatively stable normalized routing load is a desirable property for scalability of the protocols, since this indicates that the actual routing load increases linearly with the number of sources. In contrast to the routing load comparison, AODV has similar or slightly lower MAC load than DSR (Fig. 4) for lower pause times. As the pause time is increased, the MAC load comparison goes against AODV. With increase in pause time, MAC load remains almost steady for AODV, while it decreases significantly for DSR. This trend is seen regardless of the number of sources even though the margin of difference gets bigger for more sources.

One interesting observation is that the delays for both protocols increase with 40 sources with very low mobility (Fig. 2d). This is due to a high level of network congestion and multiple access interferences at certain regions of the ad hoc network. Neither protocol has any mechanism for load balancing, that is, choosing routes in such a way that



■ **Figure 2.** *Average data packet delays for the 50-node model with various numbers of sources.*

the data traffic can be more evenly distributed in the network. This phenomenon is less visible with higher mobility where traffic automatically gets more evenly distributed due to source movements. A similar phenomenon was also observed in [16].

For the 100-node experiments, we have used 10, 20, and 40 sources. The packet rate is fixed at 4 packets/s for 10 and 20 sources, and 2 packets/s for 40 sources. In Fig. 5a, c, and e, note that DSR has similar packet delivery performance to AODV for 10 sources; however, its performance gets much worse than AODV with larger numbers of sources. In particular, AODV has 22–41 percent higher packet delivery fractions than DSR for higher mobility scenarios. For 10 sources, DSR and AODV have similar delays (Fig. 5b). However, DSR's delay performance again worsens with larger numbers of sources (Fig. 5d and f). The delays for DSR are larger than AODV by a factor of about 2–6 for high mobility, with the factor increasing with the number of sources. Unlike in the 50-node networks, the relative performance (both delivery fraction and delay) of AODV and DSR is consistent across almost all pause times.

The difference in routing load for 100 nodes (Fig. 6a, c, and e) is not as pronounced as for 50 nodes. In high-mobility scenarios, the routing load of AODV is about twice as much as DSR with 10 and 20 sources and about 15 percent higher than DSR for 40 sources. For both protocols, routing load drops with increase in pause time (decrease in mobility). Note that the routing load performance of DSR is no longer as stable as with 50 nodes. For 100 nodes, comparing MAC load for the two protocols presents a different picture from 50 nodes (Fig. 6b, d and f). DSR has significantly higher MAC load than AODV for all cases (different number of sources), except at very high pause times.

In summary, when the number of sources is low, the performance (delivery fraction and delay) of DSR and AODV is similar regardless of mobility. With large numbers of sources, DSR delivers better performance under low-mobility conditions. However, AODV starts outperforming DSR for high-mobility scenarios. The point where AODV begins performing better than DSR seems to depend on the size of the network. As the data for 20 sources demonstrate, AODV starts outperforming DSR at a lower load with a larger number of nodes. This hypothesis is further reinforced in the following subsection with a load test.

DSR always demonstrates a lower routing load than AODV. We found that the major contribution to AODV's routing overhead is from route requests, while route replies constitute a large fraction of DSR's routing overhead. Furthermore, AODV has more route requests than DSR, and the converse is true for route replies. Note also that we have represented routing load in terms of packets and not in terms of bytes, since the cost to gain access to the radio medium dominates with the 802.11 MAC relative to per-byte transmission cost. The relative routing load differences will be much smaller if the comparison is made in terms of bytes, the reasons being:

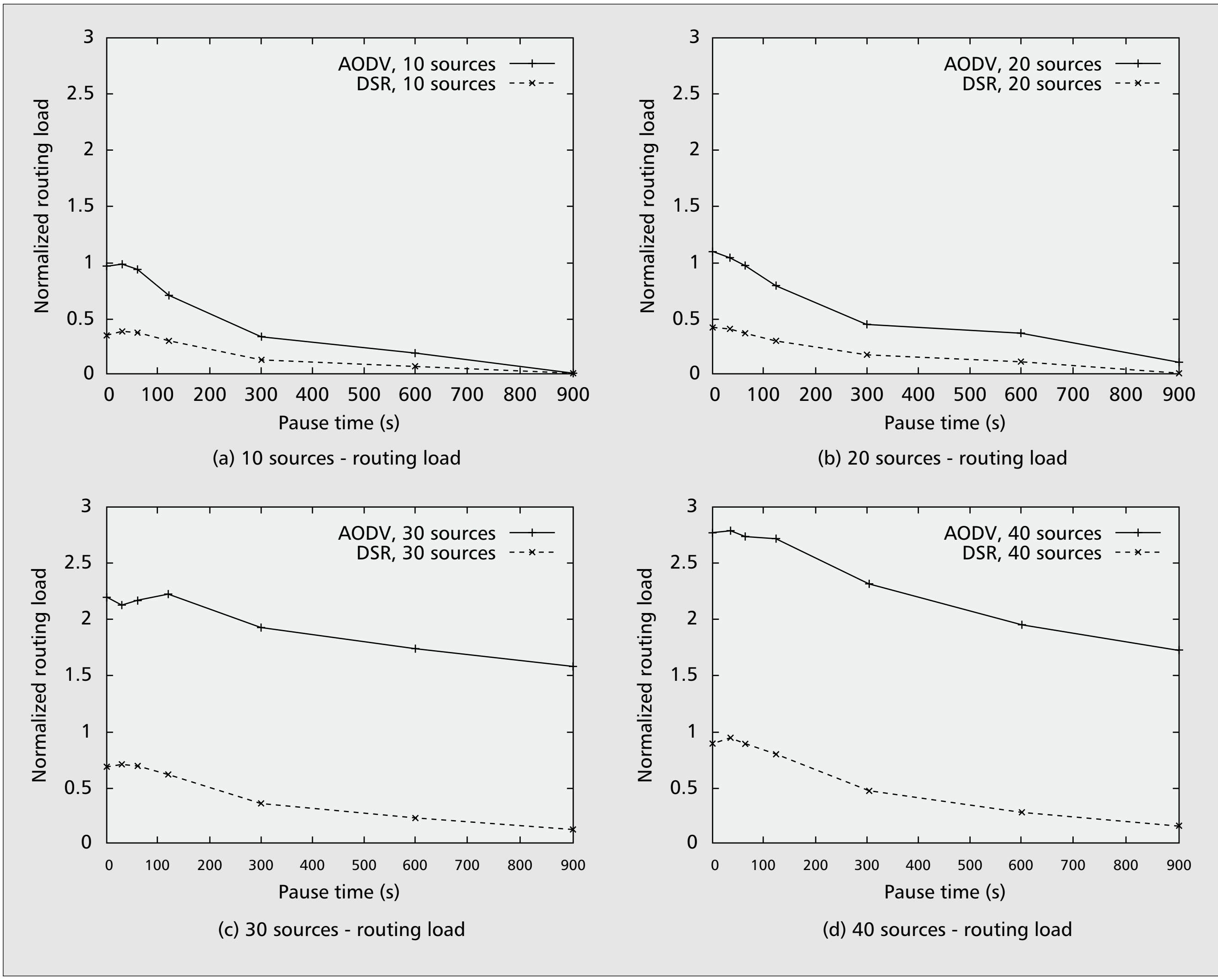


Figure 3. *Normalized routing loads for the 50 node model with various numbers of sources.*

- DSR typically uses larger routing packets due to source routing.
- DSR data packets carry routing information in the form of source routes, and these could be counted as a part of routing load.

A byte-wise routing load metric will be presented in the next subsection. Comparison of MAC load goes against DSR except under low-mobility conditions. Notice that MAC load computation takes into account both the routing and control packets at the MAC layer. When only control packets were considered, we have seen that AODV always has lower load than DSR.

Varying Offered Load

The next set of experiments (Figs. 7 and 8) demonstrate the effect of loading the network. We choose the highest mobility (i.e., zero pause time) to make the situation fairly challenging for the routing protocols. We use the 100-node model and keep the number of sources fixed (we use 10 or 40 sources). The packet rate is slowly increased until the throughput saturates. The throughput here represents the combined received throughput at the destinations of the data sources. The "offered load" in the performance plots indicate the combined sending rate of all data sources. Note that without any retransmission, the ratio of throughput and offered load is simply the packet delivery fraction. Here, we choose the units to be kilobits per second (instead of packets per second) to measure the simulated network capacity used. In order to see how routing load compares with received throughput, we also show the routing load in kilobits per second in the throughput plots.[4] In addition to throughput and average delay, we present normalized routing load and normalized MAC load for the two protocols to reason further about their performance differences with varying network load.

With 10 sources, DSR's throughput starts saturating only at an offered load of around 400 kb/s (Fig. 7a). This is due to a poor packet delivery fraction. AODV's throughput, however, increases further along, before finally starting to saturate around 700 kb/s. AODV always has lower average delay than DSR (Fig. 7c), until the point where DSR begins to saturate (around 400 kb/s). The comparison of delays beyond that point does not provide any useful insight since DSR loses more than half the packets. As expected, AODV generates higher routing load in kilobits per second (Fig. 7a) than DSR. The routing load comparison in packets after normalization (Fig. 8a) also show similar behavior. However, the MAC load comparison shows a complete reversal of trends. AODV has, in fact, lower MAC load than DSR (Fig. 8c).

The qualitative scenario is similar with 40 sources (Fig. 7b and d), but the quantitative picture is very different. Both AODV and DSR now saturate much earlier, AODV around 300 kb/s and DSR around 200 kb/s. DSR again performs

[4] *Here, DSR's routing load does not include the bits in the data packets used for source routes.*

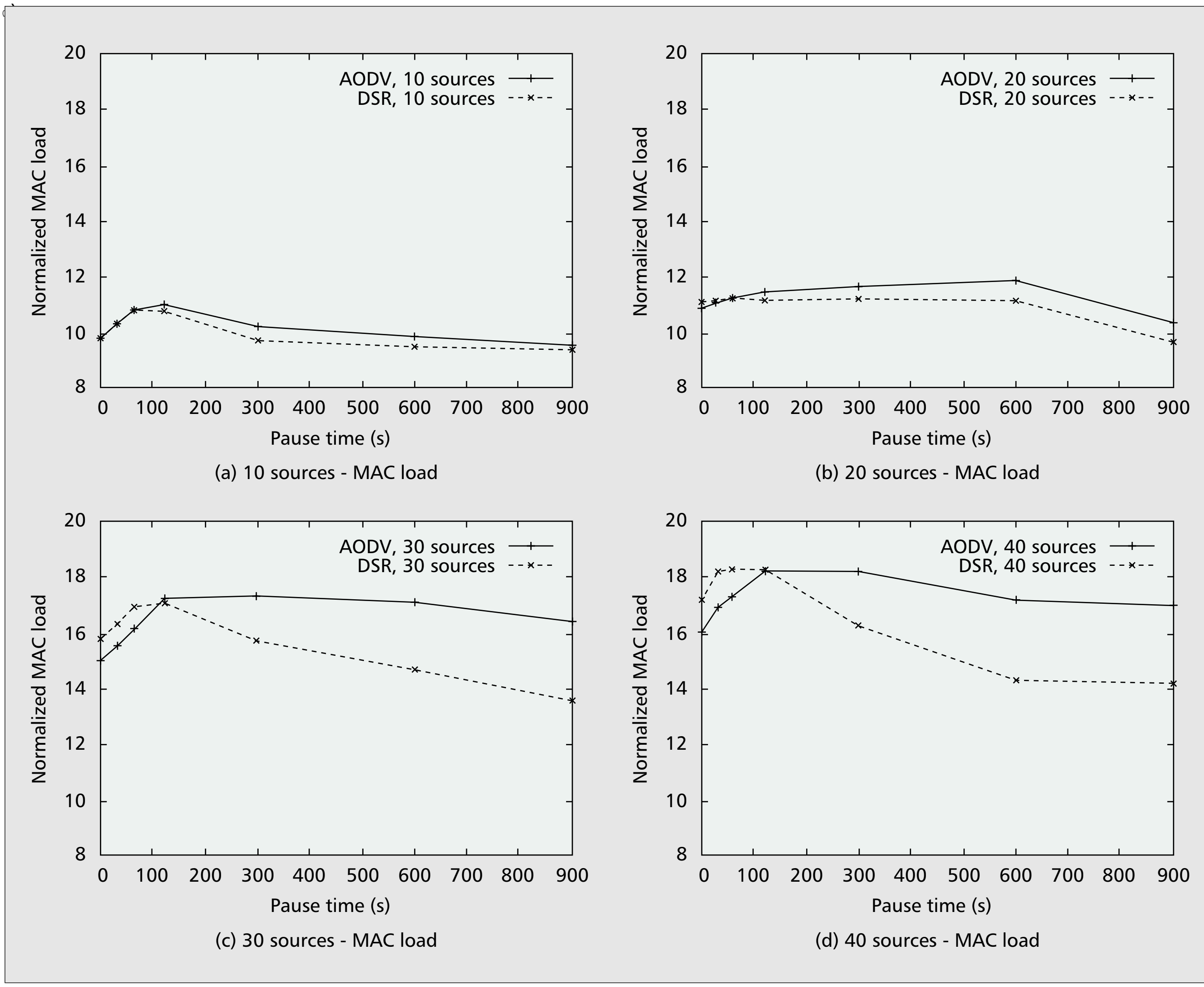


Figure 4. *Normalized MAC loads for the 50-node model with various numbers of sources.*

poorly relative to AODV, saturating at a much lower offered load. As with 10 sources, AODV has a better delay characteristic than DSR.

One interesting difference for 40 sources is that now the routing load is much higher for both protocols. This is, however, expected since four times as many sources will produce about four times as much routing load in an on-demand protocol if the sources and destinations are widely distributed in the network. As before, AODV has a higher normalized routing load and lower normalized MAC load than DSR (Fig. 8b and d).

Observations

The simulation results bring out several important characteristic differences in the two on-demand protocols. We categorize and discuss them in this section.

Routing Load and MAC Overhead

DSR almost always has a lower routing load than AODV. The difference is often significant (by a factor of up to 3) if the routing load is presented in terms of packet counts. Presenting routing loads in terms of bytes is, however, less impressive (at most about a factor of 2). By virtue of aggressive caching, DSR is more likely to find a route in the cache, and hence resorts to route discovery less frequently than AODV; but DSR generates more replies and errors (gratuitous or otherwise). Thus, even with a carefully optimized route discovery process, we found that AODV's routing load was dominated by RREQ packets (often as much as 90 percent of all routing packets). DSR's routing load, on the other hand, was dominated by RREP packets, primarily due to multiple replies from the destination or potentially many cache replies. Roughly half of all routing packets in DSR were RREPs in many scenarios. In terms of absolute numbers, DSR always generated more RREP and RERR packets (usually by a factor of 2–4) than AODV, but significantly fewer RREQ packets (up to an order of magnitude for high mobilities). Thus, all the routing load savings for DSR came from a large saving in RREQs.

However, this did not typically translate to a real decrease in network load. The higher MAC load for DSR in more challenging situations (high mobility and/or high traffic load) is evidence of this fact. Our simulation results show that MAC load is a good measure for predicting application performance. Recall that RREPs are unicast in AODV and DSR, and use the RTS/CTS/Data/ACK exchanges in the 802.11 MAC. RREQs, on the other hand, do not use any additional MAC control packets and thus have much less overhead. RERRs are handled differently in each protocol. RERRs are unicast in DSR, and, therefore contribute to additional MAC overhead like RREPs. In AODV, RERRs are broadcast like RREQs, and hence are less expensive. Consequently, when the MAC overhead was factored

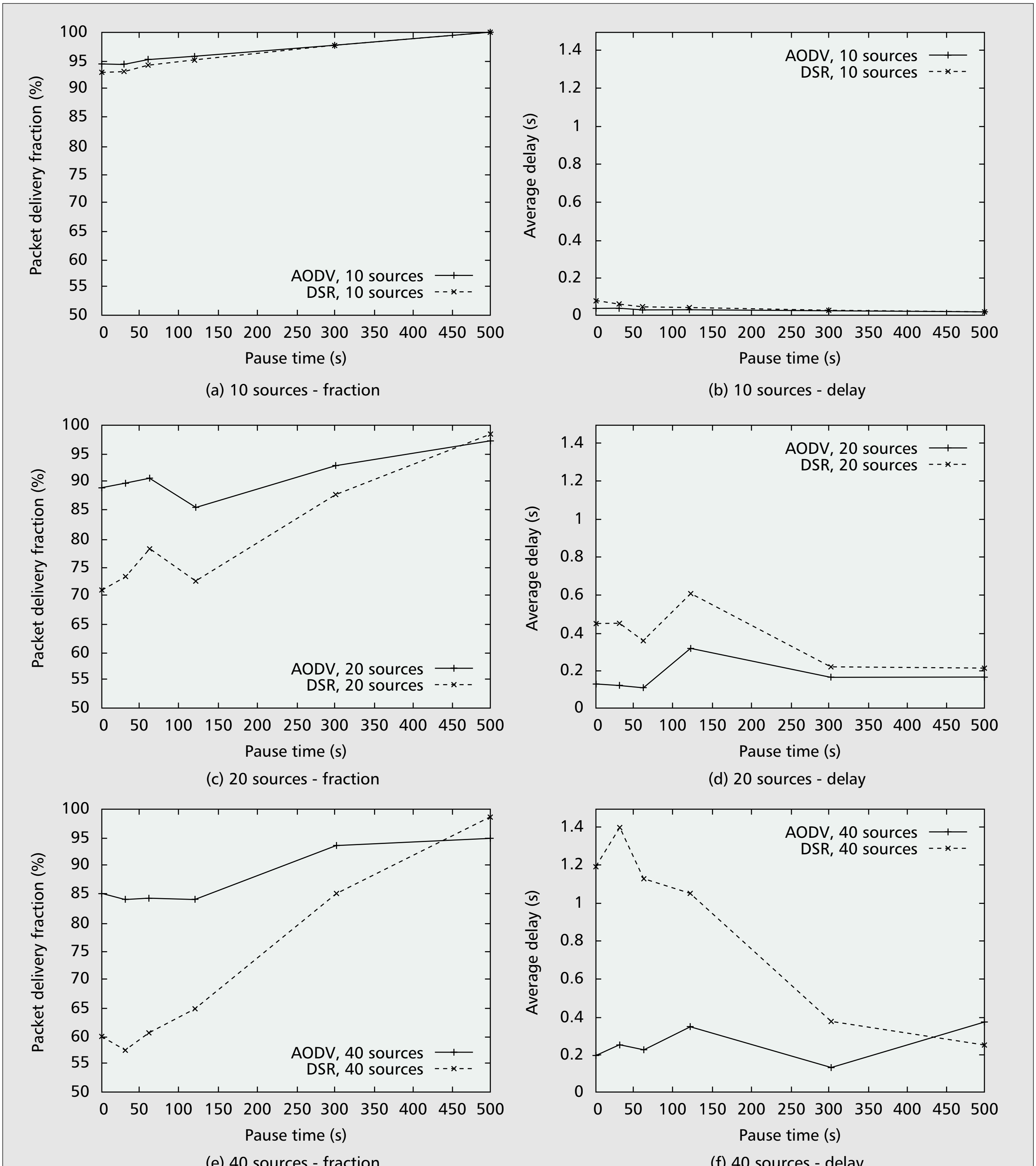


Figure 5. *Packet delivery fractions and average data packet delays for the 100-node model with various numbers of sources.*

in, DSR was found to generate higher overall network load than AODV in all interesting scenarios (high mobility or high traffic), despite having far less routing overhead.

To further establish this point, we consider an example scenario and show detailed statistics at the application layer (Fig. 9a), the routing layer (Fig. 9b), and the MAC layer (Fig. 9c). This scenario corresponds to a network of 100 nodes with zero pause time (constant mobility). Traffic in this example involves 40 CBR sources each generating packets at the rate of 2/s, each of size 512 bytes. For this example, the application-oriented metrics (Fig. 9a) point out that AODV outperforms DSR by large margins. In particular, DSR has a nearly 32 percent lower delivery fraction than AODV and five times higher delay.

Routing overhead (Fig. 9b) conforms to the general trend, that is, RREQs dominate AODV's routing load while RREPs do so for DSR. Overall, routing overhead is substantially higher (about 75 percent more) in AODV than in DSR. Interest-

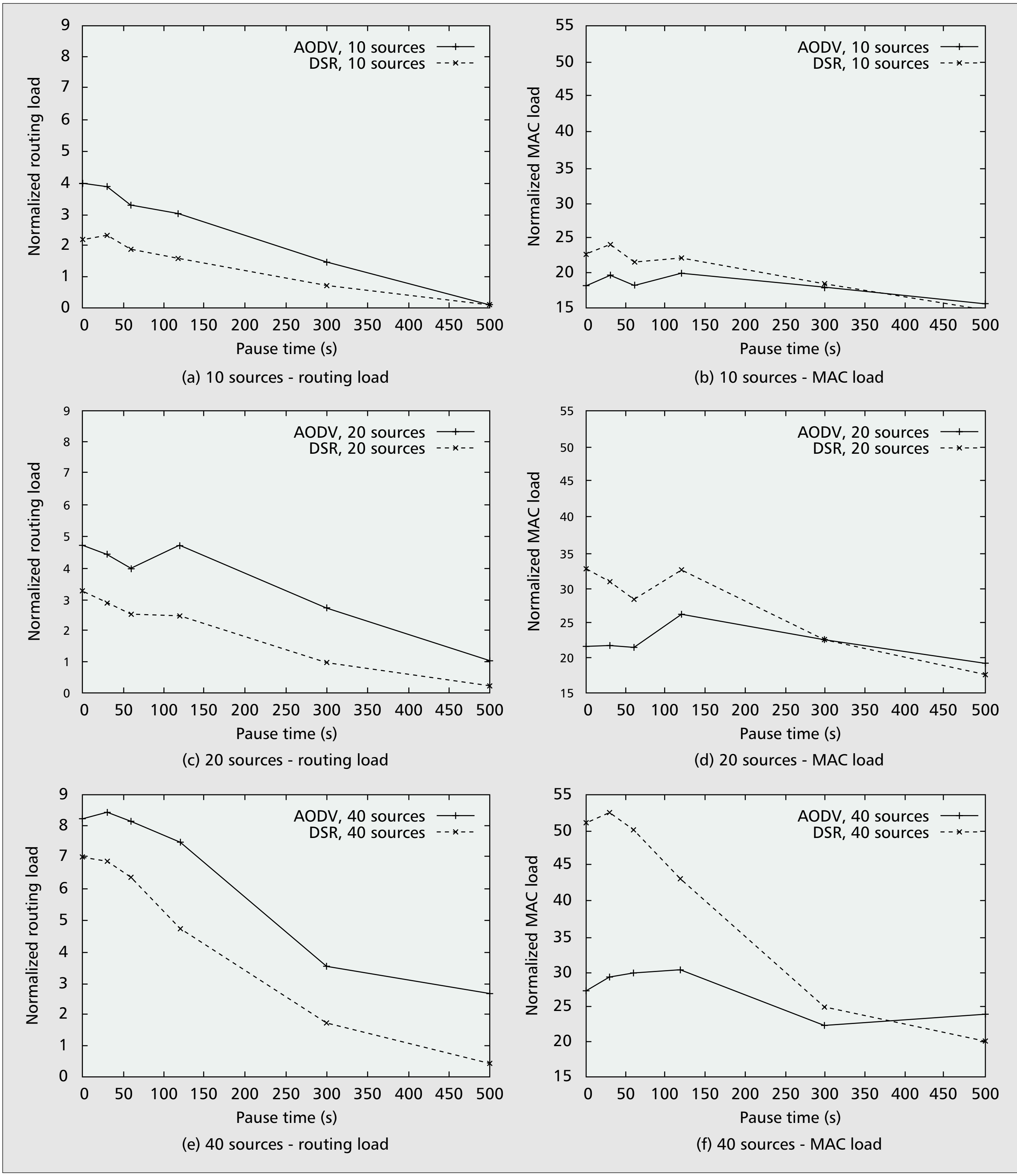


Figure 6. *Normalized routing and MAC loads for the 100-node model with various numbers of sources.*

ingly, the number of RREQs in AODV itself far exceeds the total number of routing packets in DSR.

The statistics at the MAC layer (Fig. 9c), however, present a different picture. DSR has a larger number of MAC packets than AODV in this example. The number of RTS packets for DSR is also very high, about three times the number of CTS packets. This is a result of the large number of RTS retransmissions due to collisions or link failures. The ratio of RTS to CTS packets is better (about 2) with AODV. The number of ACKs closely matches the sum of the data and unicast routing packets. Note that unicast routing packets for DSR comprise both RREPs and RERRs as opposed to only RREPs for AODV. As expected, the relative number of routing packets at the MAC layer resembles the routing layer statistics. Most important, AODV transmits 40 percent more data packets than DSR at the MAC layer. The above observations are not just true for this specific scenario, but are typically applicable for all stressful situations, where AODV outperforms DSR.

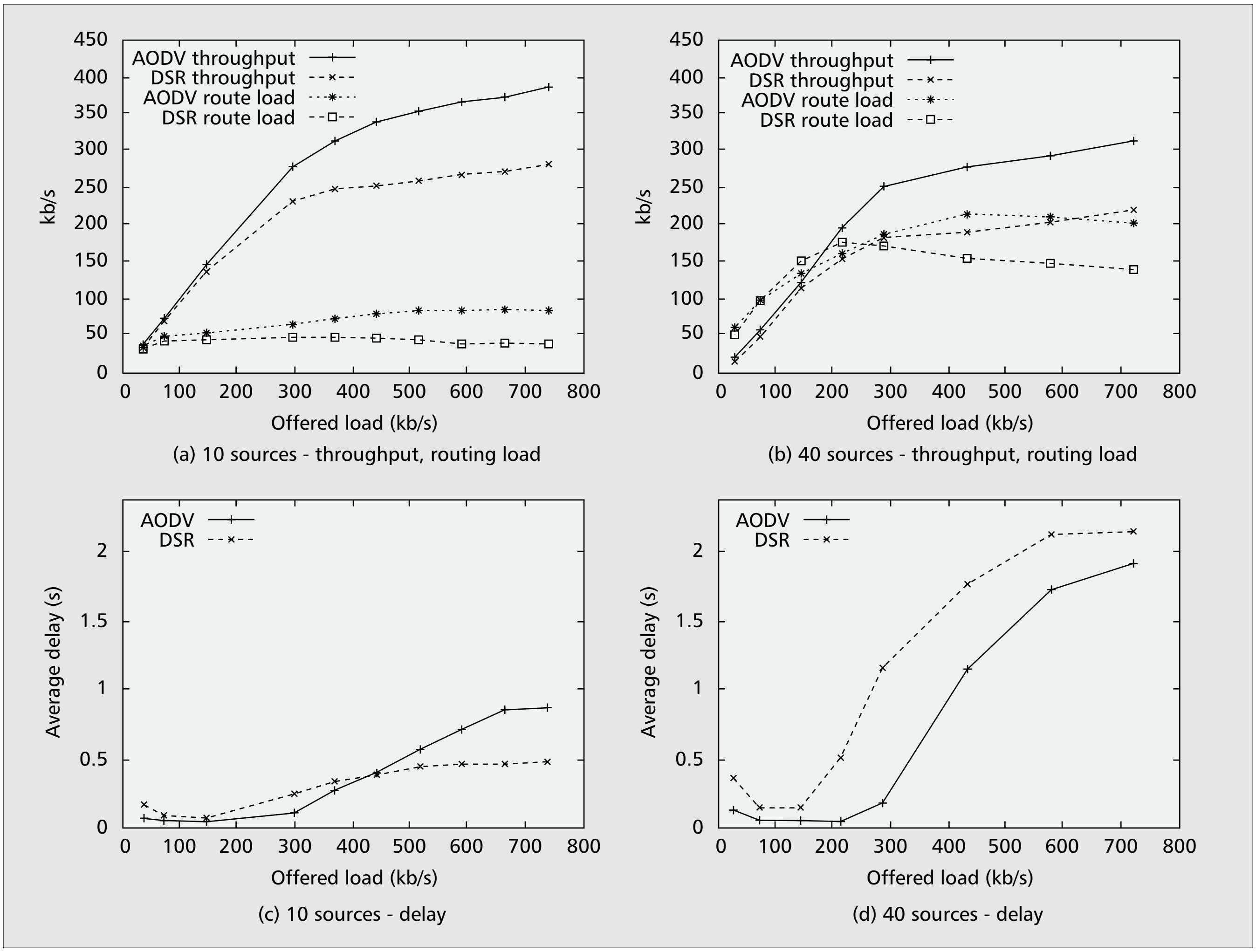


Figure 7. *Performance with increasing offered load for 100 nodes with 10 and 40 sources.*

Hence, even if routing protocols are of the same nature (e.g., AODV and DSR are both on demand) and appear very attractive in isolation, their actual performance is highly dependent on their interaction with lower layers. This indicates that careful attention must be paid to the interlayer interactions when designing protocols for wireless ad hoc networks.

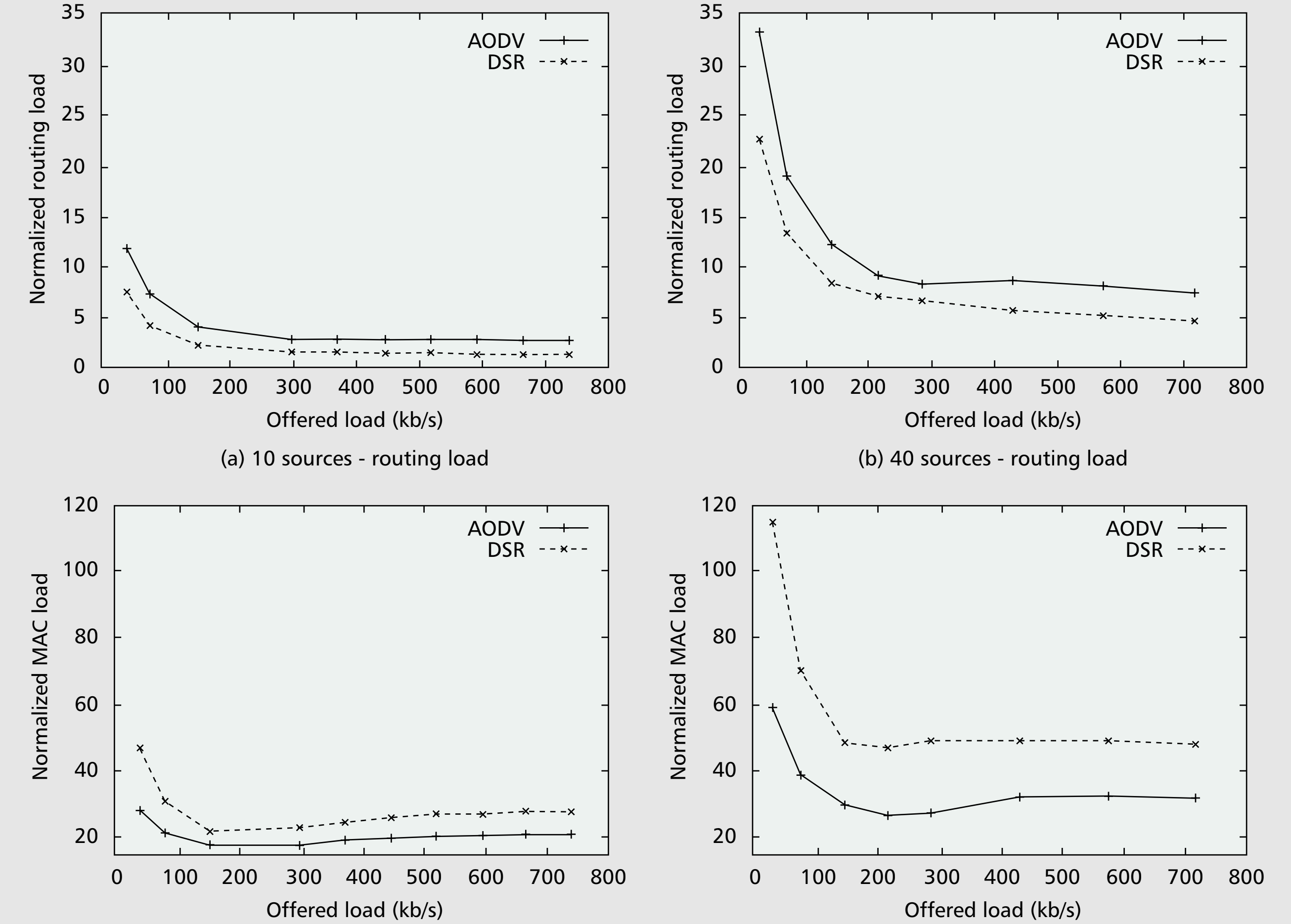
The Effect of Mobility

Our simulation results show that mobility affects the performance of AODV and DSR differently. In the presence of high mobility, link failures can happen very frequently. Link failures trigger new route discoveries in AODV since it has at most one route per destination in its routing table. Thus, the frequency of route discoveries in AODV is directly proportional to the number of route breaks. The reaction of DSR to link failures in comparison is mild and causes route discovery less often. The reason is the abundance of cached routes at each node. Thus, the route discovery is delayed in DSR until all cached routes fail. But with high mobility, the chances of the caches being stale is quite high in DSR. Eventually when a route discovery is initiated, the large number of replies received in response are associated with high MAC overhead and cause increased interference to data traffic. Hence, the cache staleness and high MAC overhead together result in significant degradation in performance for DSR in high mobility scenarios. Our simulation results show that this effect is more severe with large numbers of sources and for larger networks.

With low mobility, the possibility of link failures is low. However, nodes usually get clustered with low mobility, an artifact of our node movement (random waypoint) model. This leads to network congestion in certain regions in the presence of high traffic. Congestion in turn causes link layer feedback to report link failures even when the nodes are relatively static and the physical link exists between nodes. Such spurious link failures lead to new route discoveries in AODV. DSR, in contrast, is largely unaffected by this problem at low mobility. DSR caches are nearly up to date in low-mobility cases. Thus, even when a spurious link failure is reported, DSR benefits from caching considerably by salvaging at intermediate nodes and using alternate routes at the sources. Also, the AODV timer-based route expiry mechanism could result in unnecessary route invalidations since the spacing between data packets using a route is critical to refreshing timers associated with that route at different nodes. The above effects of mobility are particularly visible for high traffic scenarios. Also, in reality, a combination of nodes with different mobility (different speeds and different pause times) can form an ad hoc network. In that case, it is hard to predict the relative performance of AODV and DSR.

Packet Delivery and Choice of Routes

DSR fared comparatively poorly in our application-oriented metrics (delivery fraction and delay) in more “stressful” situations (i.e., larger numbers of nodes, sources, and/or higher mobility). However, DSR performed better in less stressful situations. The reason for both of these phenomena is the



(a) 10 sources - routing load

(b) 40 sources - routing load

(c) 10 sources - MAC load

(d) 40 sources - MAC load

Figure 8. *Normalized routing and MAC loads with increasing offered load for 100 nodes with 10 and 40 sources.*

aggressive use of route caching in DSR. In our observation, such caching provides a significant benefit up to a certain extent. With higher loads the extent of caching is deemed too large to benefit performance. Often, stale routes are chosen since route length (and not any freshness criterion) is the only metric used to pick routes from the cache when faced with multiple choices. Picking stale routes causes two problems:

- Consumption of additional network bandwidth and interface queue slots even though the packet is eventually dropped or delayed
- Possible pollution of caches in other nodes

Additional analysis of the performance data illustrates this point. Degradation of TCP performance due to stale routes in DSR was reported by Holland *et al.* [17]. The performance impact of various caching mechanisms for on-demand protocols was evaluated recently in [9], using DSR as a case study. We have also independently observed that cache expiry using suitable timeouts and wider propagation of route errors can improve the performance of DSR significantly. When compared to AODV, a much smaller number of packets was dropped in DSR for lack of route (e.g., indicating a high cache hit ratio); however, significantly more packets were dropped due to the interface queue being full. An efficient mechanism to "age" packets and drop aged packets from the network will improve delays in both protocols, particularly DSR. This could be achieved by decrementing the TTL field of a data packet at suitable intervals, when the packet waits in an interface queue.

Delay and Choice of Routes

We found that the correlation between the end-to-end delay and number of hops is usually small (with the *correlation coefficient* often less than 0.1), except at very low load. Further analysis of the simulation traces reveals that various buffering and queuing delays and time to gain access to the radio medium in a single congested node are often very large compared to the same delays in other nodes in a multihop route. Note that any route discovery latency is also included in the end-to-end delay. Even though more latency often indicates worse congestion, both protocols solely use hop-wise path length as the metric to choose between alternate routes. AODV has a somewhat better technique in this regard, since the destination replies only to the first arriving RREQ. This automatically favors the least congested route instead of the shortest route. In DSR, the destination replies to all RREQs, making it difficult to determine the least congested route. We found that DSR always had a shorter average path length than AODV (often 15–30 percent shorter), even though AODV often has less delay. In both protocols, careful use of congestion related metrics, such as interface queue lengths, could provide better performance.

The Effect of Loading the Network

In addition to the characteristic differences, our load tests in Fig. 7 show that network capacity is poorly utilized by the combination of the 802.11 MAC and on-demand routing. We found, via a separate measurement, that the time average of the instantaneous network capacity is roughly seven times the nominal channel bandwidth (2 Mb/s) for the highly mobile (zero pause) scenario with 100 nodes. This measurement provides an upper bound on the capacity, assuming that each node is transmitting and is able to get a $1/(n + 1)$ fraction of the nominal channel bandwidth, where n is the number of neighbors of the node in the ad hoc network. This means that the delivered throughput to the application was at most about 2–3 percent of the network capacity. This figure may seem low, but is justified given that:

- Bandwidth consumed by the delivered data packets is in fact equal to delivered throughput times the average number of hops traversed (between 3–4 in these simulations).
- Additional bandwidth is consumed by the data packets that are dropped, depending on the number of hops they travel before being dropped.
- Routing load consumes a significant portion of the bandwidth in addition to MAC control packets (e.g., RTS, CTS).
- RTS/CTS/data/ACK exchanges for reliable delivery of unicast packets often slow down packet transmissions.

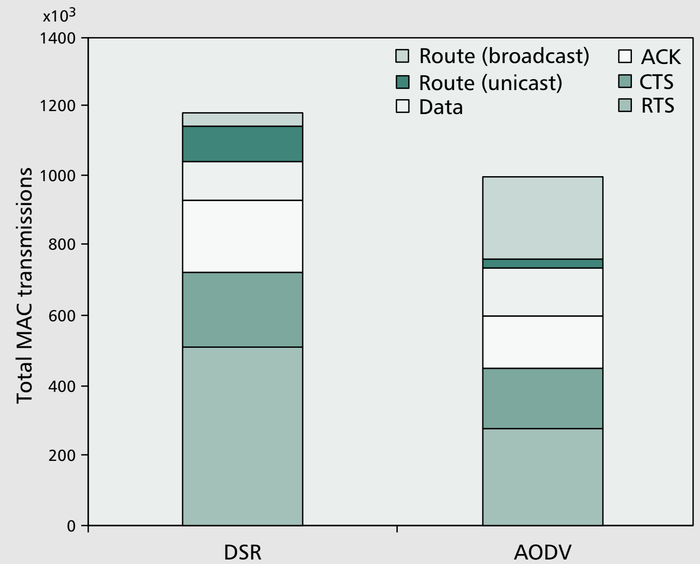
In particular, we found that in stressful situations (high mobility and/or load) the number of RTSs sent is often twice the number of CTSs received. This is due to frequent RTS retransmissions for errors due to collisions or link loss. Note that RTS packets themselves are exposed to the hidden terminal problem. As discussed before, with more unicast routing packets, DSR suffers from this phenomenon more than AODV.

Performance metrics	DSR	AODV
Packet delivery fraction (%)	56.88	83.66
Average delay (s)	1.36	0.26

(a) Applications

Routing packets	DSR	AODV
Route requests	37,774	228,094
Route replies	82,710	17,753
Route errors	26,591	9808
Total	147,075	255,655

(b) Routing



(c) MAC

Figure 9. *Application, routing, and MAC layer statistics for an example scenario for a network of 100 nodes with zero pause time (constant mobility) and 40 CBR sources.*

Related Work

Two recent efforts are the most closely related to our work, since they use the same *ns-2*-based simulation environment. Broch, Maltz, Johnson, Hu, and Jetcheva, the original authors of the simulation model, evaluated four ad hoc routing protocols including AODV and DSR [10]. They used only 50-node models with mobility and traffic scenarios similar to ours. Traffic loads are kept low (4 packets/s, 10–30 sources, 64-byte packets). Packet delivery fraction, number of routing packets, and distribution of path lengths were used as performance metrics. An earlier version of AODV was used without the query control optimizations. DSR demonstrated vastly superior routing load performance, and somewhat superior packet delivery and route length performance. This is along the lines of our observations for the loads that were considered. Routing load performance and packet delivery ratio has improved, however, in the current AODV model for comparable loads, although DSR remains a superior protocol for low loads with small numbers of nodes.

A more recent work, by Johansson, Larssson, Hedman and Mielczarek [16], extended the above work by using new mobility models. To characterize these models, a *new mobility* metric is introduced that measures mobility in terms of relative speeds of the nodes rather than absolute speeds and pause times. Again, only 50 nodes were used. A limited amount of load test was performed, but the number of sources was always small (15). Throughput, delay, and routing load (in numbers of both packets and bytes) were measured. The AODV model used hello messages for neighborhood detection in addition to the link layer feedback. The DSR model did *not* use promiscuous listening, thus losing some of its advantages. In spite of the differences in the model implementations, the overall observation was similar to ours. In low loads DSR was more effective, while AODV was more effective at higher loads. The packet-wise routing load of DSR was almost always significantly lower than AODV; however, the byte-wise routing load was often comparable. The authors attributed the comparative poor performance of DSR to the source routing overheads in data packets. They used small data packets (64 bytes), thus making things somewhat unfavorable for DSR. With 512-byte packets, we didn't find source routing overhead to be a very significant performance issue for the node populations we studied.

Other papers have compared the performance of these two on-demand protocols, including [18]. The performance of the two protocols was found to be similar. However, the simulation environment was rather limited, with no link or physical layer models. The routing protocol models also did not include many useful optimizations.

Comparisons aside, several recent papers have dealt with DSR's caching performance, an important performance determinant in our experience as presented in this article. In [7], the authors concluded that even though many cache replies carried stale routes, route maintenance in DSR is able to adapt and deliver good performance. However, Holland *et al.* [17] have shown that the stale caches in DSR have a harmful effect on TCP performance, and observed that performance could be improved by switching off replies from caches. More recently, the effects of cache structure, cache capacity, cache timeouts, and mobility patterns on the performance of DSR were studied [8]. It was observed that, in general, expiration of cached routes improved performance.

Conclusions

We have compared the performance of DSR and AODV, two prominent on-demand routing protocols for ad hoc networks. DSR and AODV both use on-demand route discovery, but with different routing mechanics. In particular, DSR uses source routing and route caches, and does not depend on any periodic or timer-based activities. DSR exploits caching aggressively and maintains multiple routes per destination. AODV, on the other hand, uses routing tables, one route per destination, and destination sequence numbers, a mechanism to prevent loops and to determine freshness of routes. We used a detailed simulation model to demonstrate the performance characteristics of the two protocols. The general observation from the simulation is that for application-oriented metrics such as delay and throughput, DSR outperforms AODV in less "stressful" situations (i.e., smaller number of nodes and lower load and/or mobility). AODV, however, outperforms DSR in more stressful situations, with widening performance gaps with increasing stress (e.g., more load, higher mobility). DSR, however, consistently generates less routing load than AODV.

The poor delay and throughput performances of DSR are mainly attributed to aggressive use of caching, and lack of any mechanism to expire stale routes or determine the freshness of routes when multiple choices are available. Aggressive caching, however, seems to help DSR at low loads and also keeps its routing load down. We believe that mechanisms to expire routes and/or determine freshness of routes in the route cache will benefit DSR's performance significantly. Concurrently with our work, the performance effects of various route caching strategies have been recently explored in [8]. On the other hand, AODV's routing loads can be reduced considerably by source routing the request and reply packets in the route discovery process. Since AODV keeps track of actively used routes, multiple actively used destinations also can be searched using a single route discovery flood to control routing load. In general, it was observed that both protocols could benefit:

- From using congestion-related metrics (e.g., queue lengths) to evaluate routes instead of emphasizing the hop-wise shortest routes.
- By removing "aged" packets from the network. The aged packets are typically not important for the upper layer protocol, because they will probably be retransmitted. These stale packets do contribute unnecessarily to the load in the routing layer.

We also observed that the interplay between the routing and MAC layers could affect performance significantly. For example, even though DSR generated much fewer routing packets overall, it generated more unicast routing packets, which were expensive in the 802.11 MAC layer we used. Thus, DSR's apparent savings on routing load did not translate to an expected reduction on the real load on the network. This observation also emphasizes the critical need for studying interactions between protocol layers when designing wireless network protocols.

References

[1] J. Jubin and J. D. Tornow, "The DARPA Packet Radio Network Protocols," *Proc. IEEE*, vol. 75, no. 1, Jan. 1987, pp. 21–32.

[2] J. Macker and S. Corson, "Mobile Ad Hoc Networks (MANET)," IETF WG Charter., http://www.ietf.org/html.charters/manet-charter.html, 1997.

[3] J.Broch, D. Johnson, and D. Maltz. "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt, IETF Internet draft, Oct. 1999, work in progress.

[4] D. Johnson and D. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," T. Imielinski and H. Korth, Eds. *Mobile Computing*, Ch. 5, Kluwer, 1996.

[5] C. E. Perkins and E. M. Royer, "Ad Hoc On-demand Distance Vector Routing," *Proc. 2nd IEEE Wksp. Mobile Comp. Sys. and Apps.*, Feb. 1999, pp. 90–100.

[6] C. E. Perkins, E. M. Royer, and S. R. Das, "Ad Hoc on Demand Distance Vector (AODV) Routing, http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-06.txt, IETF Internet Draft, July 2000, work in progress.

[7] D. Maltz *et al.*, "The Effects of On-demand Behavior in Routing Protocols for Multihop Wireless Ad Hoc Networks," *IEEE JSAC*, vol. 17, no. 8, Aug. 1999.

[8] Y. C. Hu and D. Johnson, "Caching Strategies in On-demand Routing Protocols for Wireless Ad Hoc Networks," *Proc. IEEE/ACM MOBICOM '00*, Aug. 2000, pp. 231–42.

[9] K. Fall and K. Varadhan, Eds., ns notes and documentation, 1999; available from http://www-mash.cs.berkeley.edu/ns/.

[10] J. Broch *et al.*, "A Performance Comparison of Multihop Wireless Ad Hoc Network Routing Protocols," *Proc. IEEE/ACM MOBICOM '98*, Oct. 1998, pp. 85–97.

[11] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std. 802.11-1997, 1997.

[12] V. Bharghavan *et al.*, "MACAW: A Media Access Protocol for Wireless LANs," *Proc. ACM SIGCOMM '94*, Aug. 1994, pp. 212–25.

[13] F. A. Tobagi and L. Kleinrock, "Packet Switching in Radio Channels: Part-II — The Hidden Terminal Problem in Carrier Sense Multiple-Access Models and the Busy-Tone Solution," *IEEE Trans. Commun.*, vol. COM-23, no. 12, 1975, pp. 1417–33.

[14] D. Eckhardt and P. Steenkiste, "Measurement and Analysis of the Error Characteristics of an In-building Wireless Network," *Proc. ACM SIGCOMM '96*, Oct. 1996, pp. 243–54.

[15] B. Tuch, "Development of WaveLAN, an ISM Band Wireless LAN," *AT&T Tech. J.*, vol. 72, no. 4, July/Aug 1993. pp. 27–33.

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

[17] G. Holland and N. H. Vaidya, Analysis of TCP Performance Over Mobile Ad Hoc Networks," *Proc. IEEE/ACM MOBICOM '99*, Seattle, Aug. 1999, pp. 219–30.

[18] S. R. Das *et al.*, "Comparative Performance Evaluation of Routing Protocols for Mobile Ad Hoc Networks," *7th Int'l Conf. Comp. Commun. and Networks*, Oct. 1998, pp. 153–61.

Biographies

CHARLES E. PERKINS is a research fellow at Nokia Research Center, investigating mobile wireless networking and dynamic configuration protocols. He is the editor for several ACM and IEEE journals in areas related to wireless networking. He is serving as document editor for the mobile-IP working group of the IETF, and is author or co-author of standards-track documents in the mobileip, svrloc, dhc (dynamic host configuration), and IPng working groups. He has served on the Internet Architecture Board (IAB) of the IETF and on various committees for the National Research Council. He is also associate editor for *Mobile Communications and Computing Review*, the official publication of ACM SIGMOBILE, and is on the editorial staff of *IEEE Internet Computing*. He has authored and edited books on Mobile IP and ad hoc networking, and has published a number of papers and award winning articles in the areas of mobile networking, ad hoc networking, route optimization for mobile networking, resource discovery, and automatic configuration for mobile computers.

ELIZABETH M. ROYER completed her Master's and Ph.D. degrees in computer engineering at the University of California, Santa Barbara in 1997 and 2000, respectively. Beginning in January 2001, she is an assistant professor with the Computer Science Department at the University of California at Santa Barbara. Her research interests focus on mobile wireless networks, and include such topics as routing, multicast, address autoconfiguration, quality of service, and security.

SAMIR R. DAS (sdas@ececs.uc.edu) is currently an associate professor in the Department of Electrical and Computer Engineering and Computer Science at the University of Cincinnati. Prior to this he was a faculty member in the Division of Computer Science at the University of Texas at San Antonio. He received his B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, India, in 1986; M.E. degree in computer science and engineering from the Indian Institute of Science, Bangalore, in 1988; and Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1994. He also held research positions at the Indian Statistical Institute, Calcutta, and Sun Microsystems Laboratory, Palo Alto, California. His research interests include mobile/wireless networking, performance evaluation, and parallel discrete event simulation.

MAHESH K. MARINA [StM]is a Ph.D. student in computer science at the University of Cincinnati. He received his B.Tech. degree in computer science and engineering from the Regional Engineering College, Warangal, India, in 1998 and his M.S. degree in computer science from the University of Texas at San Antonio in 1999. His research interests are in the areas of mobile/wireless networking, performance analysis, and distributed algorithms. The focus of his current work is on protocols for ad hoc networks. He is a student member of ACM.