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

Samir R. Das
Division of Computer Science
The University of Texas
at San Antonio
San Antonio, TX 78249-0667
U.S.A.

Charles E. Perkins
Communications Systems
Laboratory
Nokia Research Center
313 Fairchild Drive
Mountain View, CA 94303
U.S.A.

Elizabeth M. Royer
Department of Electrical
and Computer Engineering
University of California,
Santa Barbara
Santa Barbara, CA 93106
U.S.A.

Abstract— **Ad hoc networks are characterized by multi-hop 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 inter-layer interactions and their performance implications. We demonstrate that even though DSR and AODV share a 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.**

Keywords—**Ad hoc networks, wireless networks, mobile networks, routing protocols, simulation, performance evaluation.**

I. Introduction

In an *ad hoc* network, mobile nodes communicate with each other using multi-hop wireless links. There is no stationary infrastructure such as 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* (see, for example, [10]). 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 [11] 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 — *Dynamic Source Routing* protocol (DSR) [3], [9] and Ad Hoc On-Demand Distance Vector protocol (AODV) [13], [14]. 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, that find routes between all source-destination pairs regardless of the use or need of 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 [12] 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 paper is organized as follows. In the following section, we briefly review the DSR and AODV protocols. In Section III, we present a detailed critique of the two protocols, focusing on the differences on their dynamic behaviors that can lead to performance differences. This lays down much of the context of the performance study. Section IV describes the simulation environment. Section V presents the simulation results followed their interpretations in Section VI. Related work is presented in Section VII. We draw our conclusions in Section VIII, where we also make recommendations for improved design of either protocol.

II. Description of Protocols

A. DSR

The key feature of DSR [3], [9] 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 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 a 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 so far. The RREP routes itself back to the source by traversing this path backwards.[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 [12], as described in the following. (i) *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. (ii) *Gratuitous route repair*: A source node receiving a 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. (iii) *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.

B. AODV

AODV [13], [14] 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 a departure from DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate a 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 [13]. These sequence numbers are carried by all routing packets.

An important feature of AODV is 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 that 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.

The recent specification of AODV [14] 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 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.

III. CRITIQUE OF DSR AND AODV

The two on-demand protocols share certain salient characteristics. In particular, they both discover routes only in the presence of data packets in the need for 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 on 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 on 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 a 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 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 [12], 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

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

for routes, the fresher route (based on destination sequence numbers) is always chosen. Also, if a routing table entry is not used recently, this entry is expired. The latter technique is not problem-free, however. It is possible to expire valid routes this way, if unused beyond an expiry time. Also, determination of a suitable expiry time is difficult, as sending rates for sources as well as node mobility may differ widely and can change dynamically.

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 using 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.

IV. Simulation Model

We use a detailed simulation model based on *ns-2* [7] in our evaluation. In a recent work, the Monarch research group in CMU developed support for simulating multi-hop wireless networks complete with physical, data link and MAC layer models [2] on *ns-2*. The distributed coordination function (DCF) of IEEE 802.11 [5] for wireless LANs is used as the MAC layer. The 802.11 DCF uses Request-to-send (RTS) and Clear-to-send (CTS) control packets [1] for "unicast" data transmission to a neighboring node. The RTS/CTS exchange precedes the data packet transmission and implements a form of *virtual carrier sensing* and channel reservation to reduce the impact of the well-known *hidden terminal problem* [15]. Data packet transmission is followed by an ACK. "Broadcast" data packets and the RTS control packets are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance (CSMA/CA) is used to transmit these packets [5]. The radio model uses characteristics similar to a commercial radio interface, Lucent's WaveLAN [6], [16]. WaveLAN is a shared-media radio with a nominal bit-rate of 2 Mb/sec and a nominal radio range of 250 meters. A detailed description of simulation environment and the models is available in [2], [7].

The routing protocol model "sees" 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, RERR and data packets are all unicast packets with a specified neighbor as the MAC destination. Both protocols detect link breakage 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 failure to receive CTS after an RTS, or absence of an ACK following data transmission. No additional network layer mechanism such as *hello messages* [13] is used.

Both protocols maintain a *send buffer* of 64 packets. It buffers all data packets waiting for a route, e.g., 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 sec. 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 is FIFO, with a maximum size of 64. Routing packets are given higher priority than data packets in the interface queue.

A. Traffic and mobility models

Traffic and mobility models use similar to previous reported results using this simulator [2], [8]. Traffic sources are CBR (continuous bit-rate). 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 [2] in a rectangular field. Two field configurations are used – (i) 1500 m $\times$ 300 m field with 50 nodes and (ii) 2200 m $\times$ 600 m field with 100 nodes.[2] Here, each node starts its journey from a random location to a random destination with a randomly chosen speed (uniformly distributed between 0–20 m/sec).[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 seconds for 50 nodes, and 500 simulated seconds for 100 nodes. Each data point represents an average of at least five runs with identical traffic models, but different randomly generated mobility scenarios. For fairness, identical mobility and traffic scenarios are used across protocols.

V. Performance Results

A. Performance metrics

Three key performance metrics are evaluated: (i) *Packet delivery fraction* — ratio of the data packets delivered to the destination to those generated by the CBR sources; (ii) *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, propagation and transfer times; (iii) *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.

The first two metrics are the most important metrics for best-effort traffic. The routing load metric evaluates the efficiency of the routing protocol. Note, however, that these metrics are not completely independent. For example, lower packet delivery fraction means that the delay metric is evaluated with fewer number of samples. In the conventional wisdom, the longer the

[2] Slow simulation speed and large memory requirement of the *ns-2* models prevented us from using larger networks at this point. Note that all prior reported simulation results with these *ns-2* models use only 50 nodes. 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 traffi c speeds inside a city.

path lengths, the higher the 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 load impacts both delivery fraction and delay, as it causes less net congestion and multiple-access interference.

B. Varying mobility and number of sources

The first set of experiments uses differing number of sources with a moderate packet rate and changing pause times. For the 50 node experiments we used 10, 20, 30 and 40 traffic sources and a packet rate of 4 packets/sec, except for 40 sources which use 3 packets/sec.[4] Note that the packet delivery fractions for DSR and AODV are very similar for both protocols for 10 and 20 sources (see Fig. 1(a) and (b)). With 30 and 40 sources, however, AODV outperforms DSR (Fig. 1(c) and (d)) except at very high pause times (low mobility). DSR loses about 30–50% more packets than AODV for lower pause times (higher mobility).

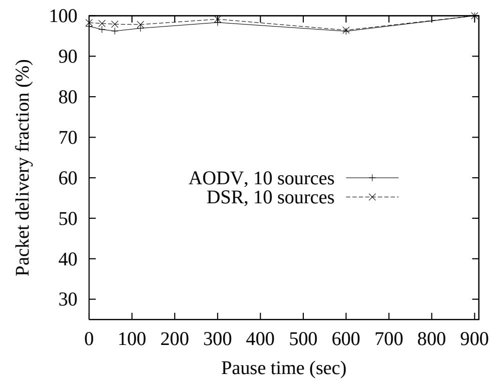
DSR has a better delay than AODV with 10 and 20 sources (see Fig. 2). The differential for 10 sources is large, often more than factor of 4 for lower pause times. The differential reduces for higher pause time (low mobility). With 20 sources, the differential is much smaller. With larger number of sources AODV has a lower delay than DSR for all pause times (Fig. 2(c) and (d)), the difference being large (about half) for lower pause times.

In all cases, DSR demonstrates significantly lower routing load than AODV (Fig. 3), usually by a factor of 4 – 7, with the factor going up somewhat with increasing number of sources. Also, note that relative to AODV, DSR's normalized routing load is fairly stable with increasing number of sources, even though its delivery and delay performance gets increasingly worse. A relatively stable normalized routing load is a desirable property for scalability of the protocols, as this indicates the actual routing load increases linearly with the number of sources.

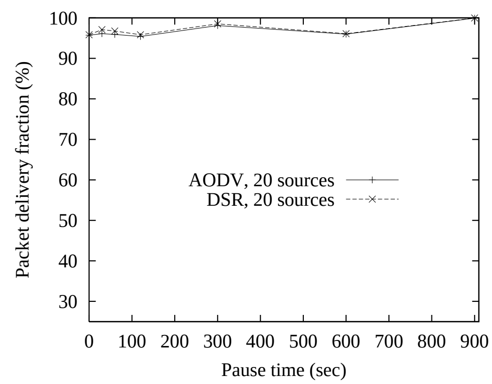
One interesting observation is that the delays for both protocols increase with 40 sources with very low mobility (see Fig. 2(d)). This is due to a high level of network congestion and multiple access interferences at certain regions of the ad hoc network. Neither protocol has any mechanism for load balancing, i.e., for choosing routes in such a way that 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 [8].

For the 100 node experiments, we have used 10, 20 and 40 sources. The packet rate is fixed at 4 packets/sec for 10 and 20 sources, and 2 packets/sec for 40 sources. In Fig. 4, note that DSR has similar packet delivery performance as AODV for 10 sources, however its performance gets much worse than AODV with larger number of sources. In particular, it loses

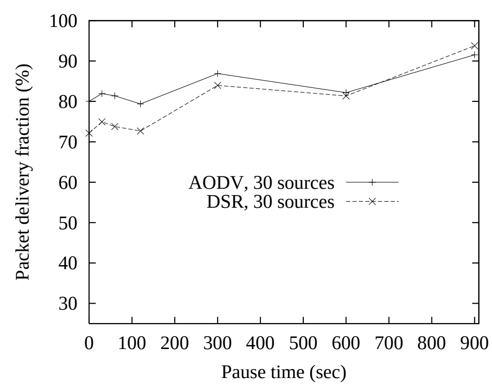
[4] We used a slower rate with 40 sources, as the network congestion was too high otherwise for a meaningful comparison. The higher rates will be considered in the next subsection.



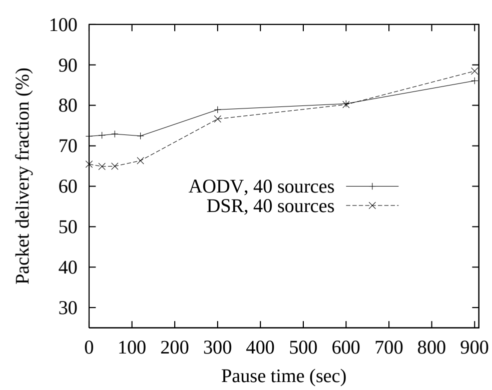
(a) 10 sources



(b) 20 sources



(c) 30 sources



(d) 40 sources

Fig. 1. Packet delivery fractions for the 50 node model with various numbers of sources.

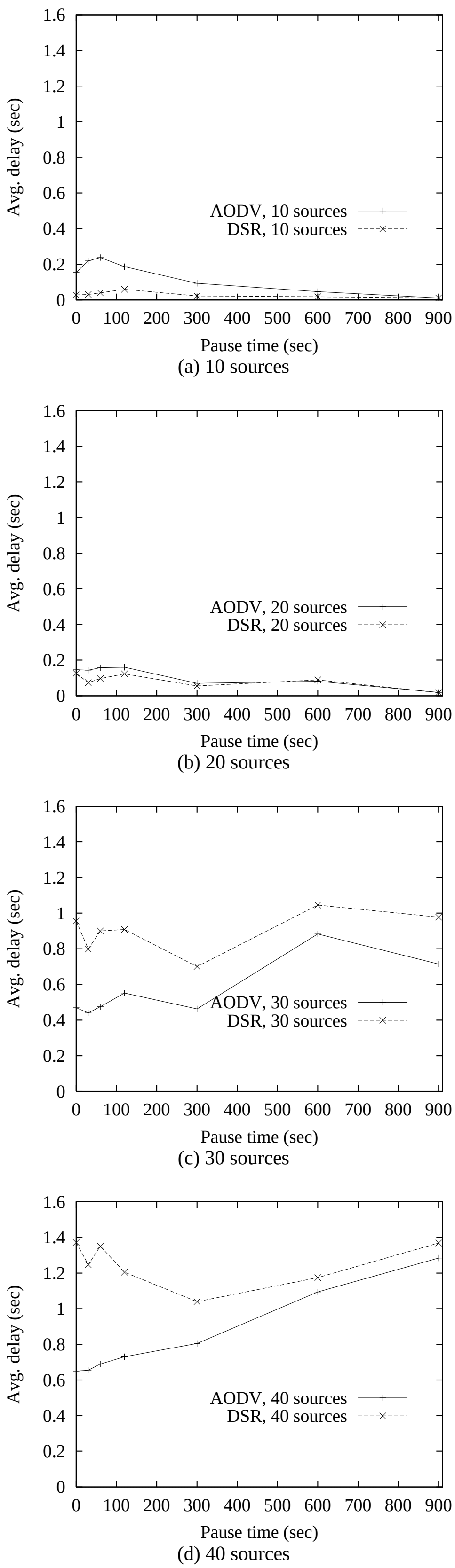


Fig. 2. Average data packet delays for the 50 node model with various numbers of sources.

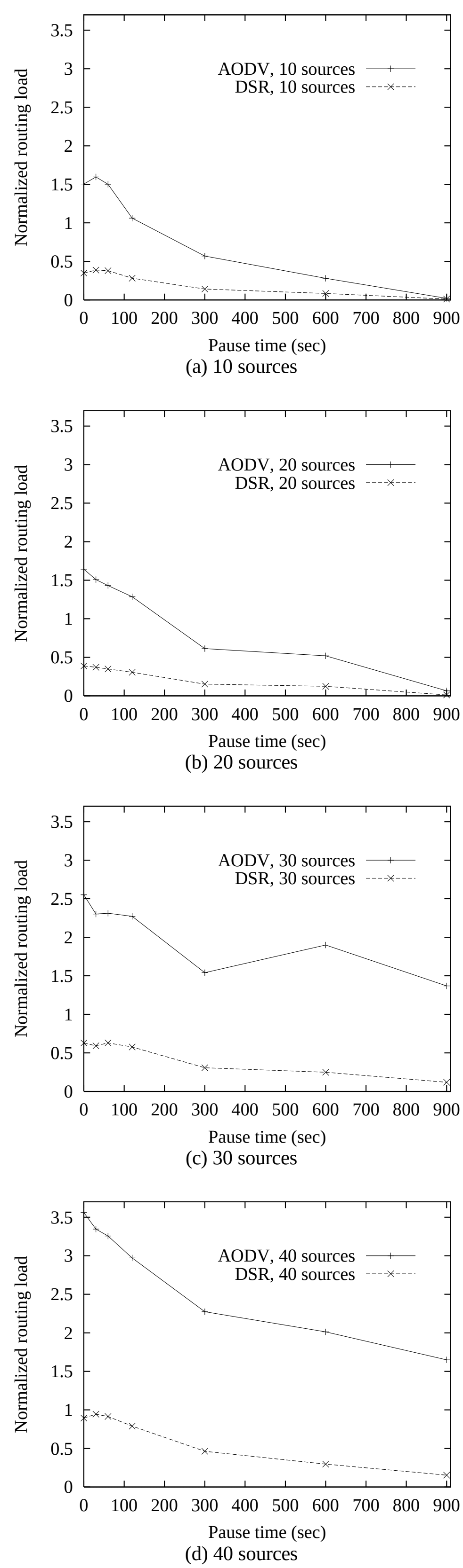
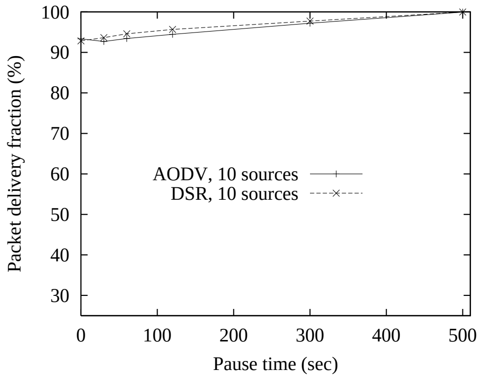
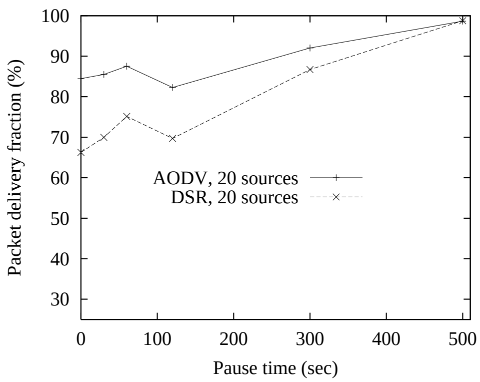


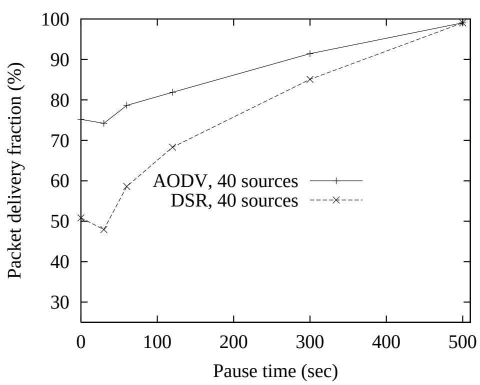
Fig. 3. Normalized routing loads for the 50 node model with various numbers of sources.



(a) 10 sources

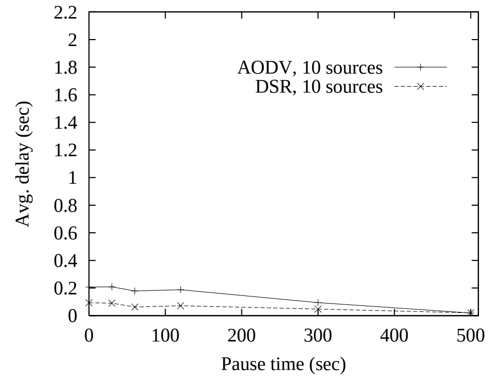


(b) 20 sources

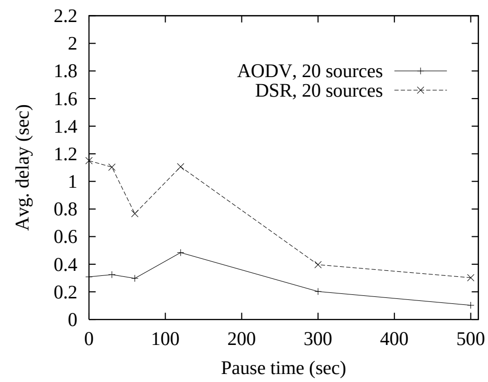


(c) 40 sources

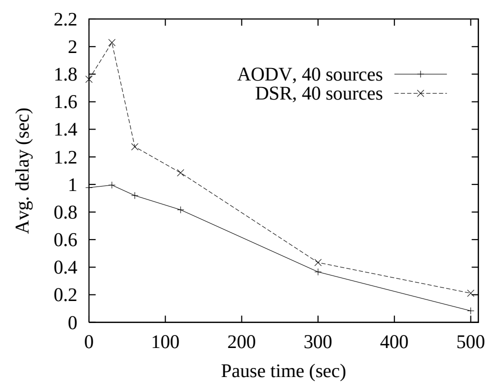
Fig. 4. Packet delivery fractions for the 100 node model with various numbers of sources.



(a) 10 sources



(b) 20 sources



(c) 40 sources

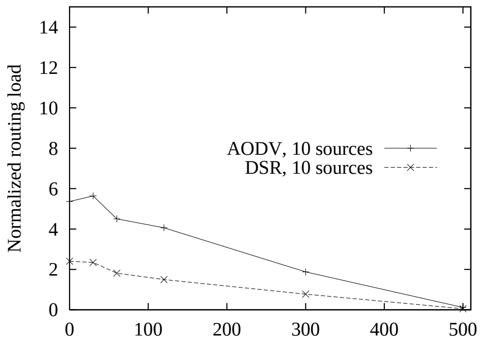
Fig. 5. Average data packet delays for the 100 node model with various numbers of sources.

about twice as many packets as AODV for higher mobility scenarios.

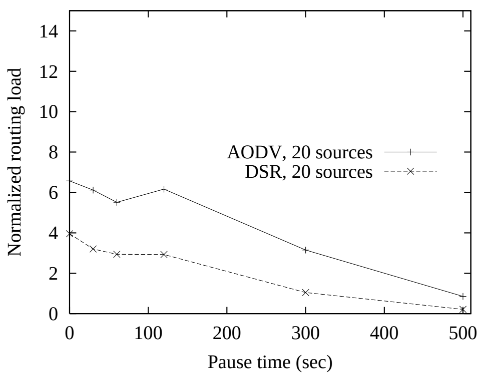
For 10 sources, DSR has lower delay than AODV (Fig. 5), often by a factor of 2. However, DSR's delay performance again worsens with larger number of sources. It gives about twice as much delay. The routing load differentials for 100 nodes (Fig. 6) are not as pronounced as 50 nodes. DSR always performs better by a factor of about 1 – 3. Note that the routing load performance of DSR is no longer as stable as with 50 nodes.

As a general observation, AODV outperforms DSR except under low load (i.e., when the number of sources is low) for the application-oriented metrics (delivery fraction and delay). The point where AODV begins outperforming DSR seems to depend on the number of nodes. As the 20 source data 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.

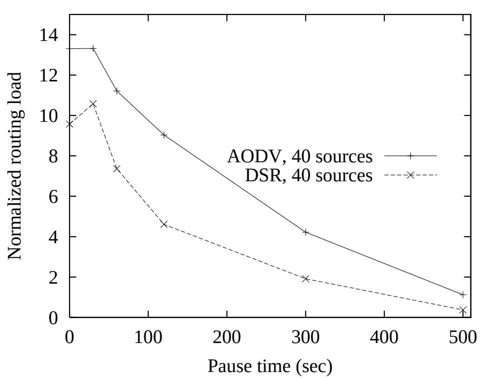
DSR always demonstrates a lower routing load than AODV. Note also that we have represented routing load in terms of packets and not in terms of bytes, as the cost to gain access to the radio medium dominates with the 802.11 MAC relative to per-byte transmission cost. The relative routing load differences will be much smaller if comparison is made in terms of



(a) 10 sources



(b) 20 sources



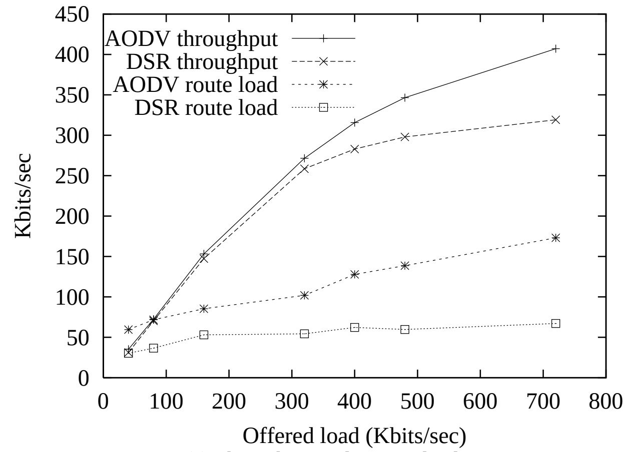
(c) 40 sources

Fig. 6. Normalized routing loads for the 100 node model with various numbers of sources.

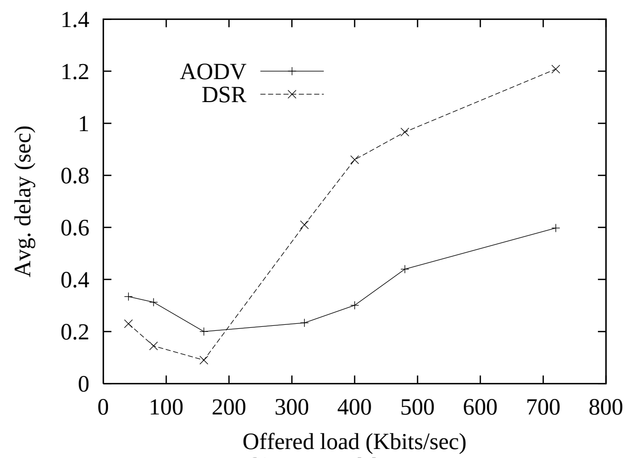
bytes, the reason being – (i) DSR typically uses larger routing packets because of source routing, and (ii) DSR data packets carry routing information in form of source routes and these could be counted as a part of routing load. A byte-wise routing load metric will be presented in the next subsection.

C. Varying offered load

These 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



(a) Throughput and routing load



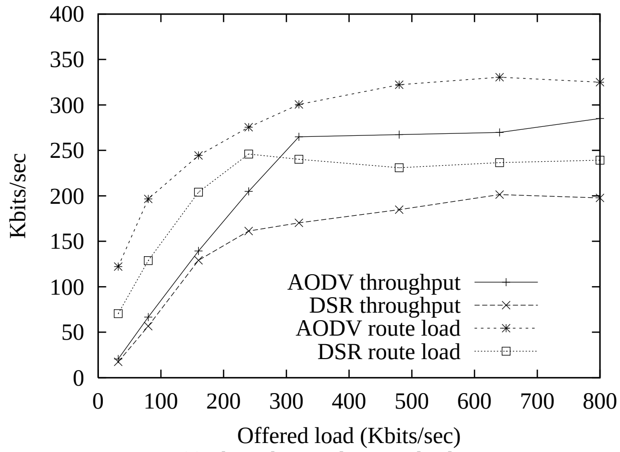
(b) Average delay

Fig. 7. Performance with increasing offered load with 100 nodes and 10 sources.

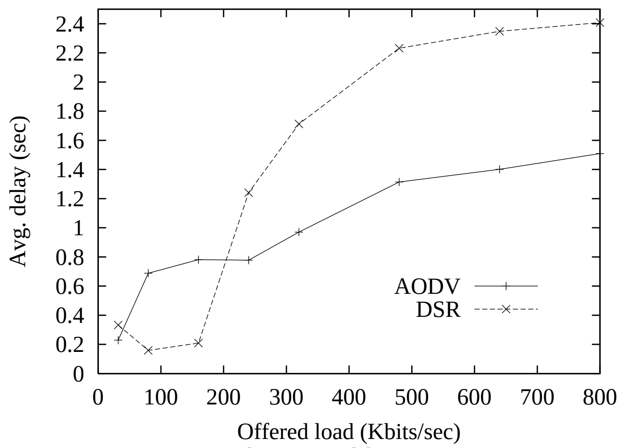
the number of sources fixed (we use 10 or 40 sources). Packet rate is slowly increased until the throughput saturates. The throughput here represents the combined "received" throughput at the destinations of the data sources. The "offered load" in the performance plots indicate the combined sending rate of all data sources. Note that without any retransmission, the ratio of throughput and offered load is simply the packet delivery fraction. Here, we chose the units to be Kbits/sec (instead of packets/sec) to measure the simulated network capacity being used. To make comparisons easier, the routing load is also shown using the same metric instead of using normalization.[5]

With 10 sources, DSR's throughput starts saturating only at an offered load of around 400 Kbits/sec (Fig. 7(a)). This is due to a poor packet delivery fraction. AODV's throughput, however, increases further along, before finally starting to saturate around 700 Kbits/sec. The average delay with DSR is smaller at low load, but much higher at high loads (Fig. 7(b)). As expected, AODV generates higher routing load than DSR.

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



(a) Throughput and routing load



(b) Average delay

Fig. 8. Performance with increasing offered load with 100 nodes and 40 sources.

The qualitative scenario is similar with 40 sources (Fig. 8), but the quantitative picture is very different. Both AODV and DSR now saturate much earlier, AODV around 300 Kbits/sec and DSR around 200 Kbits/sec. DSR again performs poorly relative to AODV, saturating at a much lower offered load. DSR again has a much higher delay compared to AODV except at a very low load. One interesting difference for 40 sources is that now the routing load is much higher for both protocols, more than the throughput! This is, however, expected, as 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. AODV has a higher routing load than DSR as before.

VI. Observations

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

A. 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 5), if 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 the virtue of aggressive caching DSR's cache hit ratio is high (an observation also made in [12]). Thus DSR rarely resorts to a route discovery process unlike AODV, but 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% of all routing packets). DSR's routing load, on the other hand, was dominated by RREP packets, primarily due to multiple replies from destination. 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 less RREQ packets (up to an order of magnitude for high mobilities). Thus, all the routing load savings for DSR came from a large saving on RREQs.

But, this did not typically translate to a real saving on the network load. Recall that RREPs and RERRs 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. Consequently, when the MAC overhead was factored in, DSR was found to generate about as much overall network load as AODV, even in the scenarios where DSR was doing particularly better than AODV. This indicates that careful attention must be paid to the interlayer interactions when designing protocols for wireless ad hoc networks.

B. Packet delivery and choice of routes

DSR fared comparatively poorly for our application-oriented metrics (delivery fraction and delay) in more “stressful” situations (i.e., larger number of nodes, sources and/or higher mobility). However, DSR performed better in less stressful situations. The reason for both of these phenomena is the 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 as the route length (and not any freshness criterion) is the only metric used to pick routes from cache when faced with multiple choices. Picking stale routes causes two problems – (i) consumption of additional network bandwidth and interface queue slots even though the packet is eventually dropped or delayed, and (ii) possible pollution of caches in other nodes. Additional analysis of the performance data illustrates this point. 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 dropping aged packets from the network will improve delays in both protocols, particularly DSR. This could be achived by decrementing the TTL field of a data packet at suitable intervals, when the packet waits in an interface queue.

C. 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, as the destination replies only to the first arriving RREQ. This automatically favors the least congested route instead of the shortest route. DSR replies to all RREQs, making it difficult to determine the least congested route. We found that DSR always had a shorter average path length compared to AODV (often 15% – 30% 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.

D. Effect of loading the network

In addition to the characteristic differences, our load tests in Figs. 7 and 8 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, the time average of the instantaneous network capacity is roughly 7 times the nominal channel bandwidth (2 Mbits/sec) 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% of the network capacity. This figure may seem low, but is justified given that (i) 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), (ii) additional bandwidth is consumed by the data packets that are dropped, depending on the number of hops they travel before being dropped, (iii) routing load consumes a significant portion of the bandwidth in addition to MAC control packets (e.g., RTS, CTS etc.), and (iv) 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 as much as 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.

VII. Related Work

Two recent efforts are the most related to our work, as 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 [2]. They used only 50 node models with similar mobility and traffic scenarios that we used. Traffic loads are kept low (4 packets/sec, 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 line 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, though DSR remains a superior protocol for low loads with small number of nodes.

A more recent work, Johansson, Larssson, Hedman and Mielczarek [8] 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 were always small (15). Throughput, delay and routing load (both in number of 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, though 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 overheads to be a very significant performance issue for the node populations we studied.

Other papers have compared performance of these two on-demand protocols, including [4]. 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.

VIII. Conclusions

We have compared 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 to 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, will benefit DSR's performance significantly. 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 (i) from using congestion-related metrics (such as queue lengths) to evaluate routes instead of emphasizing the hop-wise shortest routes, and (ii) 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.

ACKNOWLEDGEMENT

We would like to acknowledge the support of Sun Microsystems for providing the equipment and the opportunity for us to carry out the research program leading to these results. We also thank the CMU Monarch group for making the wireless extension of the *ns-2* simulator available for public use. The first author also acknowledges a faculty development leave from UTSA that enabled him to carry out this research.

REFERENCES

[1] Vaduvur Bharghavan, Alan Demers, Scott Shenker, and Lixia Zhang. MACAW: A media access protocol for wireless LAN's. In *Proceedings of the SIGCOMM'94*, pages 212–225, August 1994.

[2] J. Broch, D. A. Maltz, D. B. Johnson, Y-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th International Conference on Mobile Computing and Networking (ACM MOBICOM'98)*, pages 85–97, October 1998.

[3] Josh Broch, David Johnson, and David Maltz. The dynamic source routing protocol for mobile ad hoc networks. http://www.ietf.org/internet-drafts/ draft-ietf-manet-dsr-01.txt, Dec 1998. IETF Internet Draft (work in progress).

[4] S.R. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile, ad hoc networks. In *7th Int. Conf. on Computer Communications and Networks (IC3N)*, pages 153–161, October 1998.

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

[6] David Eckhardt and Peter Steenkiste. Measurement and analysis of the error characteristics of an in-building wireless network. In *Proceedings of the ACM SIGCOMM '96 Conference*, pages 243–254, Oct. 1996.

[7] Kevin Fall and Kannan Varadhan (Eds.). *ns* notes and documentation, 1999. available from http://www-mash.cs.berkeley.edu/ns/.

[8] Per Johansson, Tony Larsson, Nicklas Hedman, and Bartosz Mielczarek. Routing protocols for mobile ad-hoc networks - a comparative performance analysis. In *Proceedings of the 5th International Conference on Mobile Computing and Networking (ACM MOBICOM'99)*, pages 195–206, August 1999.

[9] D. Johnson and D. Maltz. Dynamic source routing in ad hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile computing*, chapter 5. Kluwer Academic, 1996.

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

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

[12] David Maltz, Josh Broch, Jorjeta Jetcheva, and David Johnson. The effects of on-demand behavior in routing protocols for multi-hop wireless ad hoc networks. *IEEE Journal on Selected Areas in Communication*, 1999. to appear.

[13] Charles Perkins and Elizabeth Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb 1999.

[14] Charles Perkins, Elizabeth Royer, and Samir Das. Ad hoc on demand distance vector (AODV) routing. http:// www.ietf.org/ internet-drafts/ draft-ietf-manet-aodv-03.txt, June 1999. IETF Internet Draft (work in progress).

[15] 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 Transactions in Communications*, COM-23(12):1417–1433, 1975.

[16] Bruce Tuch. Development of WaveLAN, an ISM band wireless LAN. *AT&T Technical Journal*, 72(4):27–33, July/Aug 1993.