Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks

Per Johansson
Ericsson Radio Systems AB
SE-126 25 Stockholm, Sweden
+46 8 719 02 90
Per.Johansson@ericsson.com

Tony Larsson
Ericsson Radio Systems AB
SE-126 25 Stockholm, Sweden
+46 8 719 50 33
Tony.Larsson@era-t.ericsson.se

Nicklas Hedman
Ericsson Erisoft AB, Box 920
SE-971 28 Luleå, Sweden
+46 920 20 23 16
Nicklas.Hedman@lu.erisoft.se

Bartosz Mielczarek
Department of Signal and Systems
Chalmers University of Technology
SE-412 96 Göteborg, Sweden
+46 31 772 17 63
bartosz.mielczarek@s2.chalmers.se

Mikael Degermark
Department of CS & EE
Luleå University of Technology
SE-971 87 Luleå, Sweden
micke@sm.luth.se

Swedish Institute of Computer Science
Box 1263
SE-164 29 Kista, Sweden

ABSTRACT

This study is a comparison of three routing protocols proposed for wireless mobile ad-hoc networks. The protocols are: Destination Sequenced Distance Vector (DSDV), Ad-hoc On demand Distance Vector (AODV) and Dynamic Source Routing (DSR). Extensive simulations are made on a scenario where nodes moves randomly. Results are presented as a function of a novel mobility metric designed to reflect the relative speeds of the nodes in a scenario. Furthermore, three realistic scenarios are introduced to test the protocols in more specialized contexts. In most simulations the reactive protocols (AODV and DSR) performed significantly better than DSDV. At moderate traffic load DSR performed better than AODV for all tested mobility values, while AODV performed better than DSR at higher traffic loads. The latter is caused by the source routes in DSR data packets, which increase the load on the network.

1. INTRODUCTION

The notion of a mobile ad-hoc network used in this work is a network formed without any central administration, consisting of mobile nodes that use wireless interfaces to send packet data. The nodes in an ad-hoc network can act as both routers and hosts, thus a node may forward packets between other nodes as well as run user applications.

Mobile ad-hoc networks have been the focus of many recent research and development efforts. Ad-hoc packet radio networks have so far mainly concerned military applications, where a decentralized network configuration is an operative advantage or even a necessity. Networks using ad-hoc configuration concepts can be used in many military applications, ranging from interconnected wireless access points to networks of wireless devices carried by individuals, e.g., digital maps, sensors attached to the body, voice communication, etc. Combinations of wide range and short range ad-hoc networks seek to provide robust, global coverage, even during adverse operating conditions.

In the commercial sector, equipment for wireless, mobile computing has not been available at a price attractive for larger markets. However, as the capacity of mobile computers increases steadily, the need for un-tethered networking is expected to rise as well. Commercial ad-hoc networks could be used in situations where no infrastructure (fixed or cellular) is available. Examples include rescue operations in remote areas, or when local coverage must be deployed quickly at a remote construction site. Ad-hoc networks between notebook or palmtop computers could be used to spread and share information among the participants of a conference. Short range ad-hoc networks can simplify intercommunication of various mobile devices (e.g., a cellular phone and a PDA) by eliminating the tedious need for cables. The latter case could also extend the mobility provided by the fixed network (e.g., Mobile IP) to nodes further out in an ad-hoc network domain.

Since the network nodes are mobile, an ad-hoc network will typically have a dynamic topology which will have a profound effects on network characteristics. Network functions such as routing, address allocation, authentication, and authorization must be designed to cope with a dynamic and volatile network topology. Network nodes will often be battery powered, which limits the capacity of CPU, memory, and bandwidth. This will require network functions that are resource effective. Furthermore, the wireless (radio) media will also affect the behavior of the network due to fluctuating link bandwidths resulting from relatively high error rates.

1.1 Routing protocols for ad-hoc networks

This work focuses on routing protocols for mobile ad-hoc networks. Traditional routing protocols are *proactive* in that they maintain routes to all nodes, including nodes to which no packets are sent. They react to topology changes, even if no traffic is affected by the change. They are based on either link-state or distance vector principles [18] and require periodic control messages to maintain routes to every node in the network. The rate at which these messages are sent must reflect the dynamics of the network in order to maintain valid routes. Hence, the use of scarce resources, e.g., power and link bandwidth, for control traffic will increase with increased node mobility. An alternative approach is *reactive* route establishment, where routes between nodes are determined only when explicitly needed to route packets.

Several routing protocols for ad-hoc networks have been proposed, for instance [1, 3, 4, 6, 8, 9, 10, 13, 15, 17, 19], but few comparisons between the different protocols have been published.

Within the Internet Engineering Task Force (IETF), a working group named Mobile Ad-hoc Networks (MANET) [12] has the charter to standardize an IP routing protocol for mobile ad-hoc networks. All the routing protocols listed above (except for [17]) have been submitted to the MANET group as internet drafts.

The work presented in [2] is the most comprehensive comparison of ad-hoc routing protocols published so far. The study was done in the Monarch[1] project at CMU and aims at a fair evaluation based on quantitative metrics. Examples of other simulation results on individual protocols are [11] and [14], but as these used different metrics the results are difficult to compare.

Three routing protocols are studied in this work, namely Ad-hoc on Demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Destination Sequenced Distance Vector (DSDV). AODV and DSR were selected because they show the best performance in [2], but should be compared and evaluated further using additional metrics and scenarios. As opposed to DSR and AODV, DSDV is a proactive protocol and was included to illustrate the differences between reactive and proactive protocols.

This work has been inspired by the simulations in [2], but extends those results further by introducing a new mobility metric and new network scenarios as well as presenting results on delays and byte overhead. First, a metric called *mobility* is introduced as a means to capture the relative motion of nodes in the network. Second, throughput and delay are measured for the analyzed protocols with mobility and offered traffic load as variables in a random network scenario. Third, three network scenarios are analyzed, denoted Conference, Event Coverage, and Disaster Area, respectively. They are intended to model a set of usage cases believed to be more realistic than a totally random motion pattern. In addition, the simulation tools were modified to include simple obstacles that shadow the coverage of nodes, which add to the realism of the latter scenarios.

1. MObile Networking ARCHitectures

The models of DSDV and DSR used in the study were part of a simulation package from CMU [20], while AODV had to be implemented independently at the time of this work. To clarify the differences to the work made in [2], a discussion on protocol implementations and protocol parameters are presented in conjunction with the protocol descriptions in Section 2.

In all simulations presented herein, the link layer consists of a wireless LAN using a media access control (MAC) function based on the IEEE 802.11 [7] standard. This MAC function uses a random access algorithm denoted CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) that essentially operates as an Ethernet in the air without the collision detection part. The random access concept used in this protocol makes it relatively easy to form ad-hoc networks. The technology is commercially available, and there is an implementation of this link layer in the simulation environment used in this study.

The paper is organized as follows. In Section 2, brief descriptions of the studied protocols are given. Section 3 introduces a mobility metric used throughout the study. In Section 4, simulation results for the random scenario are given and in Section 5 results for the three realistic scenarios are presented and discussed. In Section 6 conclusions are drawn from the study and, finally, in Section 7 planned further work is listed.

2. PROTOCOL DESCRIPTIONS

This section gives short descriptions of the three ad-hoc routing protocols studied in this work.

2.1 Destination Sequenced Distance Vector - DSDV

DSDV [17] is a hop-by-hop distance vector routing protocol. It is proactive; each network node maintains a routing table that contains the next-hop for, and number of hops to, all reachable destinations. Periodical broadcasts of routing updates attempt to keep the routing table completely updated at all times.

To guarantee loop-freedom DSDV uses a concept of sequence numbers to indicate the freshness of a route. A route R is considered more favorable than R' if R has a greater sequence number or, if the routes have the same sequence number, R has lower hop-count. The sequence number for a route is set by the destination node and increased by one for every new originating route advertisement. When a node along a path detects a broken route to a destination D, it advertises its route to D with an infinite hop-count and a sequence number increased by one.

Route loops can occur when incorrect routing information is present in the network after a change in the network topology, e.g., a broken link. In this context the use of sequence numbers adapts DSDV to a dynamic network topology such as in an ad-hoc network.

DSDV uses triggered route updates when the topology changes. The transmission of updates is delayed to introduce a damping effect when the topology is changing rapidly. This gives an additional adaptation of DSDV to ad-hoc networks.

The parameter values used for DSDV in the simulations are given in Table 1 and are the same as in [2].

Table 1: DSDV Simulation parameters

Periodic route update interval	*15 s*
Periodic updates missed before link declared broken	*3*
Route advertisement aggregation time	*1 s*
Maximum packets buffered per node per destination	*5*

2.2 Ad-hoc On Demand Distance vector - AODV

AODV [15,16] is a distance vector routing protocol, like DSDV, but it is *reactive* rather than proactive like DSDV. That is, AODV requests a route only when needed and does not require nodes to maintain routes to destinations that are not communicating. The process of finding routes is referred to as the *route acquisition* henceforth. AODV uses sequence numbers in a way similar to DSDV to avoid routing loops and to indicate the freshness of a route.

Whenever a node needs to find a route to another node it broadcasts a *Route Request (RREQ) message* to all its neighbors. The RREQ message is flooded through the network until it reaches the destination or a node with a fresh route to the destination. On its way through the network, the RREQ message initiates creation of temporary route table entries for the reverse route in the nodes it passes. If the destination, or a route to it, is found, the route is made available by unicasting a *Route Reply (RREP) message* back to the source along the temporary reverse path of the received RREQ message. On its way back to the source, the RREP message initiates creation of routing table entries for the destination in intermediate nodes. Routing table entries expire after a certain time-out period.

Neighbors are detected by periodic *HELLO messages* (a special RREP message). If a node *x* does not receive HELLO messages from a neighbor *y* through which it sends traffic, that link is deemed broken and a link failure indication (a *triggered RREP message*) is sent to its *active neighbors*. The latter refers to the neighbors of *x* that were using the broken link between *x* and *y*. When the link failure messages eventually reach the affected sources, these can choose to either stop sending data or to request a new route by sending out new RREQ messages.

The implementation of AODV made within this study combines HELLO messages with information from the MAC layer to detect link failures, which results in quicker failure detection. DSR uses similar methods. The HELLO interval was also increased to 1.5 seconds (1 second in [16]) since the protocol now gets additional information from the link layer. Moreover, the AODV implementation used in this study has a *send buffer* of 64 packets, which is not specified in [16]. The send buffer, located in the sending node, stores outgoing packets until the route acquisition procedure obtains a route to their destination. The AODV specification does not require a send buffer, but it is needed to obtain a fair comparison with DSR which does specify a send buffer. The maximum time to keep packets in the send buffer was set to 8 seconds, which was a heuristically determined value based on a series of initial simulations. Some of the parameters used in the simulation was slightly modified compared to the ones used in [2] and the ones specified by [17]. The *Route reply lifetime* was set to match the *Active route time-out* value. The *Time between retransmitted requests* was set to fit the reverse route life time (3 seconds) since it should be possible to retransmit a request as soon as the reverse route has expired. To save bandwidth, the frequency of *triggered RREP* messages was limited to one every second. The parameter values used in the simulations are given in Table 2.

Table 2: Parameter values for AODV

HELLO interval	*1,5 s*
Active route time-out	*300 s*
Route reply lifetime	*300 s*
Allowed HELLO loss	*2*
Request retries	*3*
Time between retransmitted requests	*3 s*
Time to hold packets awaiting routes	*8 s*
Maximum rate for sending replies for a route	*1/s*

2.3 Dynamic Source Routing - DSR

Dynamic Source Routing (DSR) [2][10][11] is a reactive routing protocol which uses source routing to deliver data packets. Headers of data packets carry the sequence of nodes through which the packet must pass. This means that intermediate nodes only need to keep track of their immediate neighbors in order to forward data packets. The source, on the other hand, needs to know the complete hop sequence to the destination.

As in AODV, the route acquisition procedure in DSR requests a route by flooding a *Route Request* packet. A node receiving a Route Request packet searches its *route cache*, where all its known routes are stored, for a route to the requested destination. If no route is found, it forwards the Route Request packet further on after having added its own address to the hop sequence stored in the Route Request packet. The Route Request packet propagates through the network until it reaches either the destination or a node with a route to the destination. If a route is found, a Route Reply packet containing the proper hop sequence for reaching the destination is unicasted back to the source node. DSR does not rely on bi-directional links since the Route Reply packet is sent to the source node either according to a route already stored in the route cache of the replying node, or by being piggybacked on a Route Request packet for the source node. However, bi-directional links are assumed throughout this study. Then the reverse path in the Route Request packet can be used by the Route Reply message.

The DSR protocol has the advantage of being able to learn routes from the source routes in received packets. When *A* finds a route to *C* through *B*, it will in the process learn a route to *B*, and *C* will learn a route to *A*. When data starts flowing from *A* to *C*, *B* will learn a route *C*. However, if the reverse path from *C* to *A* passes through *B*, *B* will learn a route to *C* already when Route Reply message passes through *B*.

To avoid unnecessarily flooding the network with Route Request messages, the route acquisition procedure first queries the neighboring nodes to see if a route is available in the immediate neighborhood. This is done by sending a first Route Request message with the hop limit set to zero, thus it will not be forwarded by the neighbors. If no response is obtained by this initial request, a new Route Request message is flooded over the entire network.

DSR may use the MAC layer to inform about link failures. Alternatively, it can use the Network Layer Acknowledgment feature as described in [3]. In this study the MAC layer feedback is used only. In case of a link failure, a *route error packet* is sent back to the source node, which then removes the broken link from its route cache and all routes that contain this hop are truncated at the point of the broken link. Furthermore, an intermediate node that forwards the route error packet may also update its route cache in a similar manner.

A DSR node is able to learn routes by overhearing packets not addressed to it (the *promiscuous mode*). However, this feature requires an active receiver in the nodes, which may be rather power consuming. In networks were nodes have limited power the aim is to shut down the transceiver as often as possible to conserve power. In order to investigate how DSR would operate in such an environment the promiscuous mode was not used in the DSR simulations. This decision was also motivated by simulation runs (not presented due to space limitations), comparing DSR with and without the promiscuous mode. In these simulations the use of the promiscuous mode did not give a significant improvement of network performance. However, more exhaustive simulations should be made to confirm this.

The parameter values used in the DSR simulations are taken from [2] (see Table 3).

Table 3: Parameters for DSR.

Time between retransmitted requests	*500 ms*
Size of source route header carrying n addresses	*4n + 4 bytes*
Time-out for non propagating search	*30 ms*
Time to hold packets awaiting routes	*30 s*
Maximum rate for sending replies for a route	*1/s*

3. MOBILITY METRIC

This section defines a mobility metric, henceforth referred to as *mobility*, intended to capture and quantify the kind of node motion relevant for an ad-hoc routing protocol. Ad-hoc routing protocols must take action when the relative motion of nodes causes links to break or form, and a mobility metric should thus be proportional to the number of such events. The metric should also, if possible, be independent of the particular network technology used. Therefore a mobility metric is proposed which is *geometric* in the sense that the speed of a node in relation to other nodes is measured, while it is independent of any links formed between nodes in the network.

The study in [2] uses the pause time at waypoints in a random motion model as a mobility metric. This makes sense for the particular motion model used in that study but is too ad-hoc[1] to be useful for generic motion models. For instance, the pause time metric is ill-defined when node motion is continuous or when nodes use different pause times. Moreover, the speed at which nodes move between waypoints is also relevant for how often links break and form.

The mobility metric proposed here describes the mobility of a scenario with a single value M which is a function of the relative motion of the nodes taking part in a scenario. If $l(n,t)$ is the position of node n at time t, the relative velocity $v(x,y,t)$ between nodes x and y at time t is

$$v(x, y, t) = \frac{d}{dt}(l(x, t) - l(y, t)) \ . \qquad (1)$$

The mobility measure, M_{xy}, between any pair (x, y) of nodes is defined as their absolute relative speed taken as an average over the time, T, the mobility is measured. The formula for obtaining M_{xy} is given below.

$$M_{xy} = \frac{1}{T} \int_{t_0 \le t \le t_0 + T} |v(x, y, t)| dt \qquad (2)$$

In order to arrive at the total mobility metric, M, for a scenario, the mobility measure in (2) is averaged over all node pairs, resulting in the following definition

$$M = \frac{1}{|x, y|} \sum_{x, y} M_{xy} = \frac{2}{n(n-1)} \sum_{x=1}^{n} \sum_{y=x+1}^{n} M_{xy} \ , \qquad (3)$$

where $|x,y|$ is the number of distinct node pairs (x,y) and n is the number of nodes in the scenario. (Note that the second relation in (3) assumes nodes being numbered from 1 to n.) Hence, the mobility expresses the average relative speed between all nodes in the network. Consequently, the mobility for a group of nodes standing still, or moving in parallel at the same speed, is zero.

For practical reasons a discrete version of the mobility formula is used when computing the mobility for the network scenarios in this study. M is approximated by summing the

1. No pun intended.

relative speeds over small time steps, *0.1* seconds. Decreasing the time increment below *0.1* seconds did not improve the accuracy significantly for the scenarios under study.

The distances are measured in meters which gives the mobility measure in meters per second. Alternatively, the distance could be normalized with the transmitting range of the nodes to compare systems with different radio coverage. However, this modification is left for evaluation in future studies.

The mobility metric *M* appears to capture something relevant for the routing protocols. The diagram in Figure 1 is offered as evidence for this claim. The diagram shows the number of times links break or form as a function of the mobility when the nodes move in a random model as described in Section 4. The diagram gives average values, based on data from all the random scenario simulations.

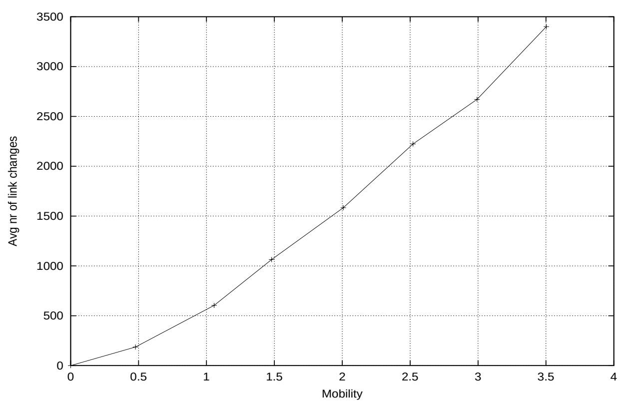

Figure 1. Mobility vs. link changes for a random scenario.

4. SIMULATIONS - RANDOM SCENARIOS

The simulation study was conducted in the Network Simulator (ns2) [5] environment and used the ad-hoc networking extensions provided by CMU [20]. All simulations were performed on a PC (Pentium-2, 400 MHz, 128 MB of RAM) running FreeBSD 2.2.6.

In the random scenario, each node randomly selects waypoints in a square environment space (1 km x 1 km). At each waypoint a node pauses for a predefined time and picks the speed to the next waypoint from a uniformly distributed interval [$0..v_{max}$]. The simulations of random scenarios are similar to the approach in [2], where the area was instead rectangular, 1500m x 300m.

A square area does not “discriminate” one direction of motion like a rectangular area do. On the other hand, it limits the number of hops (from 6 to 4 for a transmitting range of 250m). Since Section 5 analyzes scenarios with many hops, the square area was chosen for this part of the study.

Delay and throughput were measured. In addition, to understand the protocol efficiency, the overhead imposed by the routing protocols was measured both in terms of packets and bytes. Two sets of simulations were run. First, the mobility was varied and the offered load was held constant. In the second set of simulations the offered load was varied as well as the mobility. Table 4 provides all the simulation parameters.

Table 4: Simulation Parameter Values

Transmitter range	*250 m*
Bandwidth	*2 Mbps*
Simulation time	*250 s*
Number of nodes	*50*
Pause time	*1 s*
Environment size	*1000x1000 m*
Traffic type	Constant Bit Rate
Packet rate	*5 packets/s*
Packet size	*64 byte*
Number of flows	*15*

In all random scenario simulations the implicit mobility value is controlled through the explicit maximum speed parameter, $v_{max.}$. The mobility value is difficult to set exactly, so an interval of ±*0.1* for each point was allowed. The mobility values used in the simulations are: *0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0,* and *3.5*, where a mobility factor of *3.5* corresponds to a $v_{max.}$ of *20 m/s.*

In all the simulations the traffic was generated by *15* continuous bit rate (CBR) sources spreading the traffic randomly among all nodes. The packet size was *64* bytes and the packet rate was *5 packets/s* in the first set of simulations. In the second set of simulations the rate ranged from *5 packets/s* to *20 packet/s.*

4.1 Delay

4.1.1 First set of simulations - Varied Mobility

The average packet delay increases with mobility for all three protocols, as shown in Figure 2. However, DSR has a lower delay than AODV at higher mobility values due to the way routes are detected in DSR. The route acquisition procedure in DSR allows more routes to be detected and cached than in AODV, which obtains a single route per RREQ. With DSR, packets wait less during route acquisition than with AODV.

DSDV exhibits a low delay because only packets belonging to valid routes at the sending instant get through. A lot of packets are lost until new (valid) route table entries have been propagated through the network by the route update messages in DSDV. For DSR and AODV, on the other hand, the reactive route acquisition procedures manage to provide new routes with a low packet loss.

4.1.2 Second set of simulations - Varied Load

The results for AODV, DSR, and DSDV are shown in Figure 3, Figure 4, and Figure 5, respectively.

All protocols exhibit higher delays with increased load. This is because send buffers become more filled with increasing load. At the highest mobility, AODV shows the highest de-

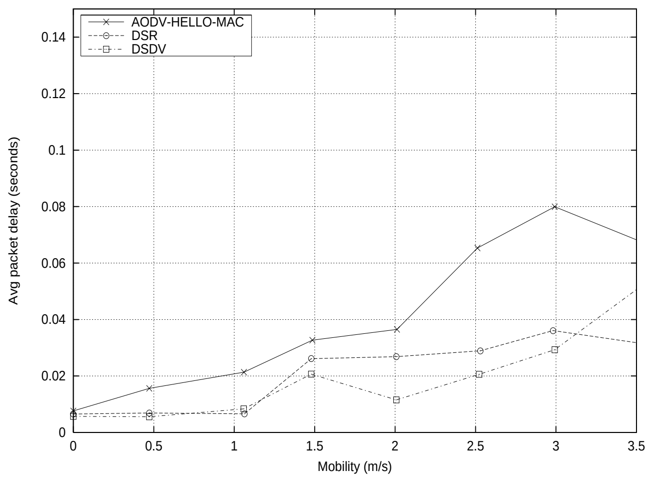

Figure 2. All - Average delay with varied mobility.

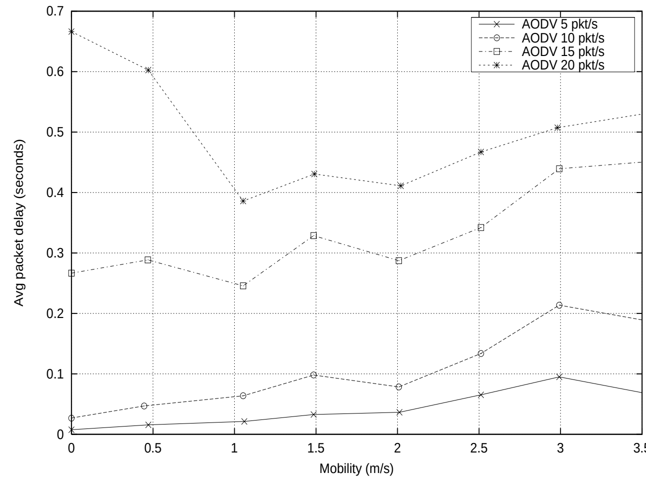

Figure 3. AODV- Average delay with varied offered traffic.

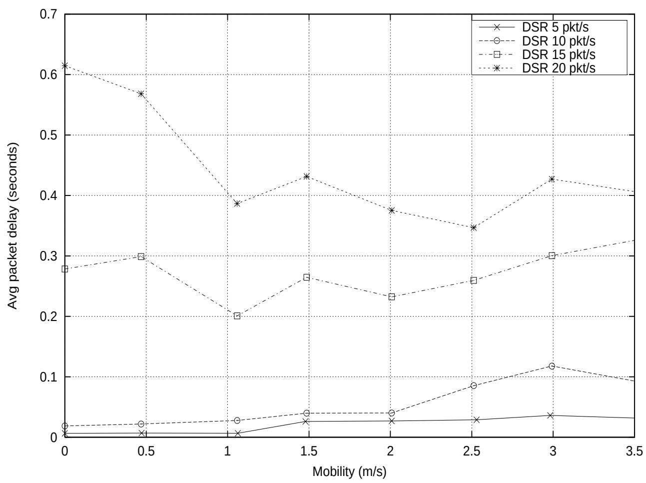

Figure 4. DSR - Average delay with varied offered traffic.

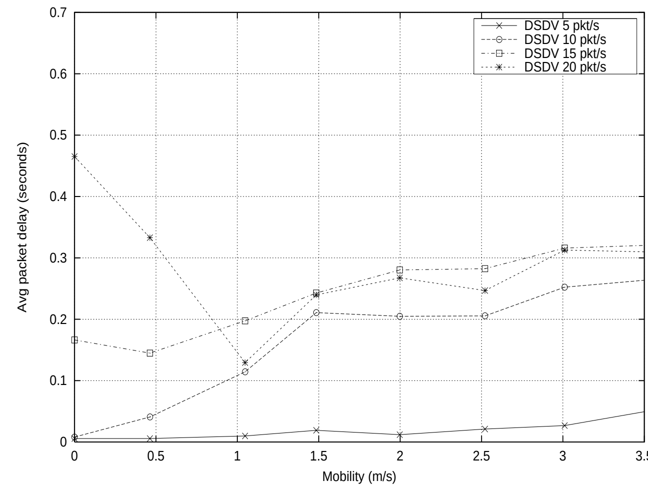

Figure 5. DSDV - Average delay with varied offered traffic.

lay, but also delivers more packets than DSR and DSDV (see Section 4.2).

DSDVs inability to converge when the mobility is high becomes increasingly evident at high loads. More traffic is offered but the route update interval remains unchanged. In Figure 5 this can be seen as a high delay increase when the packet rate goes from *5 to 10 packets/s* at *1.5 m/s* mobility.

At the highest load all protocols exhibit a somewhat surprising property; the delay is higher at low mobility than at moderate mobility (*0 - 1 m/s*). The explanation is that at low mobility routes are relatively long lived. More traffic is carried over the same paths during longer times, so longer queues will form and incur higher delays. At higher mobility routes are reestablished occasionally, so the traffic is spread out over a larger number of nodes, one might say that the result is a form of load balancing. At the highest mobility the delay for AODV and DSR increases again due to longer route acquisition procedures, while it saturates for DSDV due to the short send buffer.

4.2 Throughput

4.2.1 First set of simulations - Varied Mobility

The average throughput for the network is shown in Figure 6. With an offered load of *5 packets/s* the maximum throughput is approximately *2.5 kbps*. Throughput decreases only slightly for AODV and DSR with increased mobility (about *4-5* percent packet loss at the highest mobility). DSDV on the other hand has difficulties in finding routes when mobility increases. This is clear from Figure 6, where the throughput drops with about *40* percent at high mobility.

The slightly lower throughput for DSDV at zero mobility is caused by packets that are sent (and lost) before routes have converged initially in the network. Note that all simulations are started without any established routes.

4.2.2 Second set of simulations - Varied Load

At higher offered loads, DSDV exhibits the highest drop in throughput (*50* percent at *20 packets/s*). This is due to packets being dropped along outdated routes (see Figure 9). DSR, in Figure 8, also shows a big drop in throughput at higher loads, e.g., *40* percent at *20 packets/s*. This is an ef-

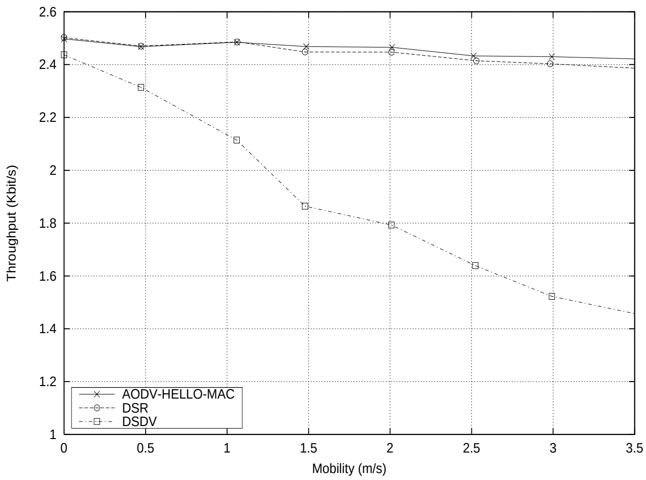

Figure 6. All - Throughput with varied mobility.

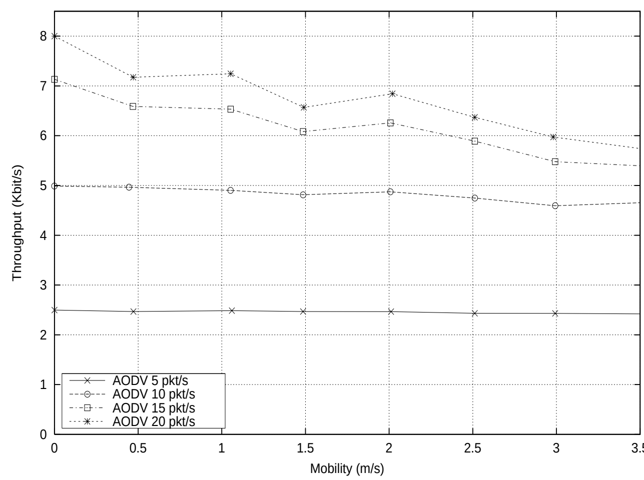

Figure 7. AODV - Throughput with varied offered traffic.

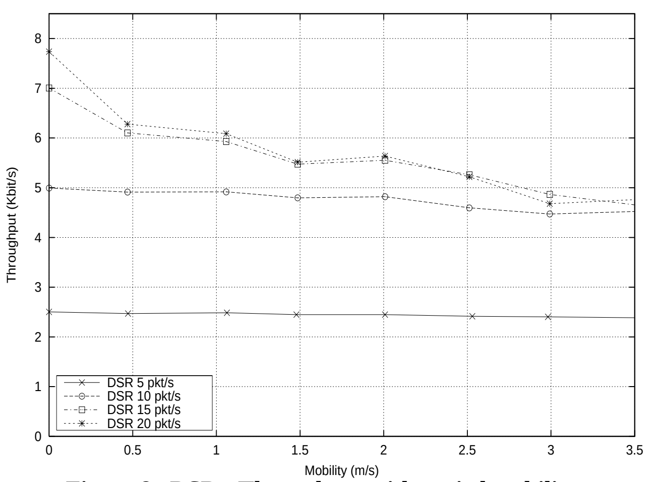

Figure 8. DSR - Throughput with varied mobility.

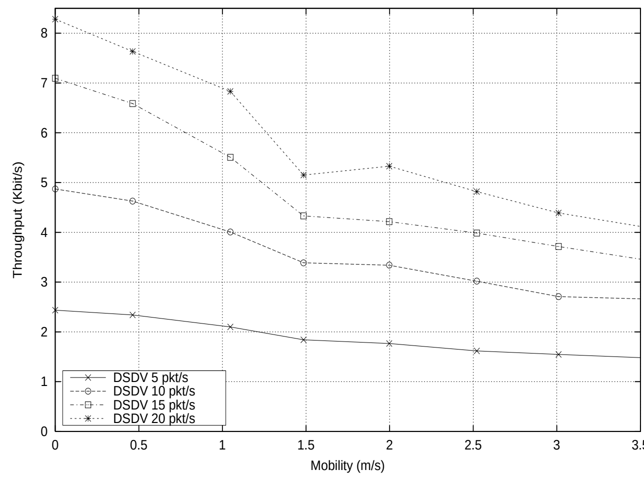

Figure 9. DSDV - Throughput with varied offered traffic.

fect of the higher network load caused by the source routes carried in all data packets. Thus, DSR will face a higher packet loss than AODV at higher loads. AODV is more robust and drops about *28* percent in throughput at *20 packets/s* (see Figure 7).

Note that at the highest packet rate, *20 packets/s*, and zero mobility, all protocols still only deliver about *80* percent of the offered packets. At this load the network drops a rather large number of packets due to buffer overflow in some congested nodes. This congestion is caused by an increase in MAC layer packet collisions, giving less capacity to drain queues, combined with a higher aggregated packet rate in some forwarding nodes.

4.3 Routing protocol overhead

The overhead was measured as number of control packets and as byte overhead. The latter includes overhead in data packets, e.g. source routes as well as the entire control packets. The total number of packets (or bytes) sent during the entire simulation is reported. Only overhead stemming from the IP layer is included, i.e. link layer or physical layer overhead is not.

The packet overhead shown in Figure 10 clearly exposes the characteristics of the three protocols. DSDV does not adapt to increased mobility; the update intervals remain constant. AODV and DSR on the other hand detect and react to more link failures when mobility increases, resulting in an increased number of control packets. Moreover, AODV sends HELLO packets periodically which gives it a higher packet overhead.

Figure 11 shows the byte overhead. It reveals the impact of overhead for source routes as used by DSR. DSDV has the highest byte overhead of all protocols because the routing table updates often contain the entire routing table. This is accentuated when the mobility increases since more routes need to be fully updated in each update. At high offered loads the byte overhead becomes large for DSR, as can be seen from the upper plots in Figure 13. The high byte overhead at low mobility is caused by many packets being delivered and thus also many source routes carried in data packet headers. The byte overhead decreases for DSR when the mobility increases to moderate values and causes lower throughput. At the highest mobility the overhead increases again as more control packets are needed to acquire routes.

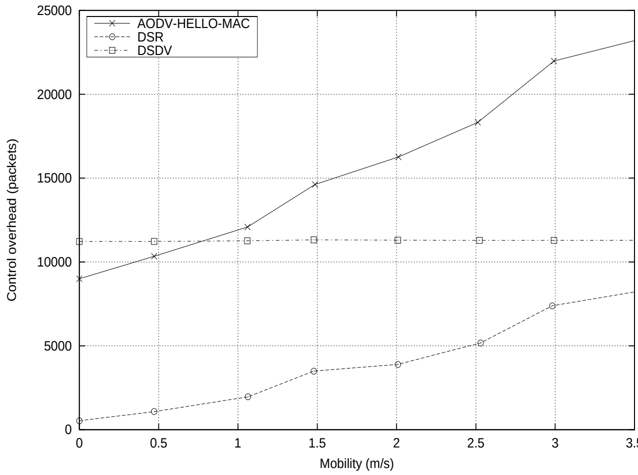

Figure 10. All - Packet overhead with varied mobility.

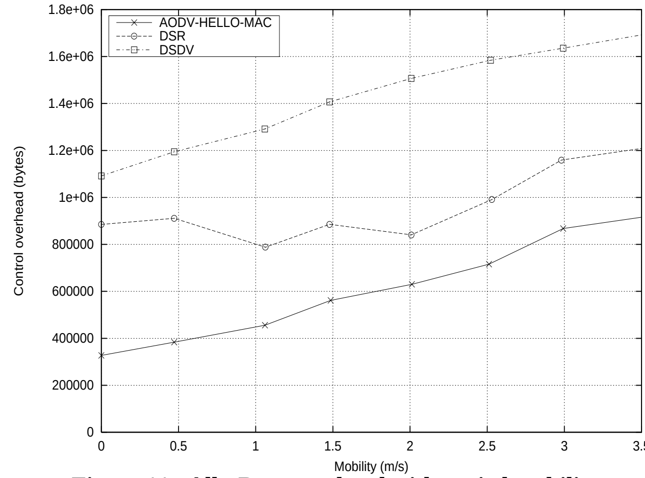

Figure 11. All - Byte overhead with varied mobility.

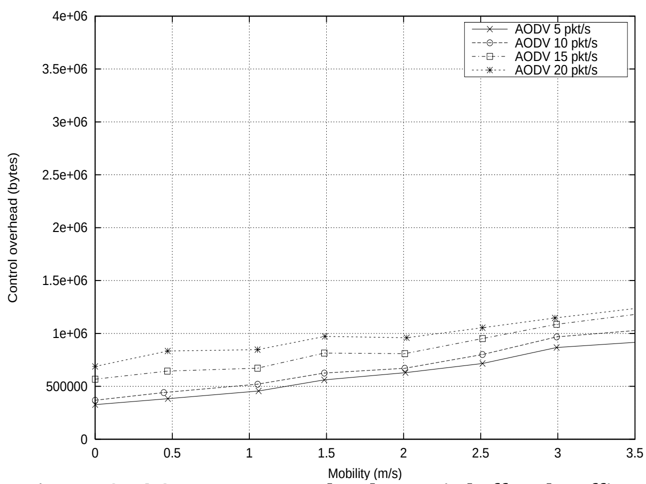

Figure 12. AODV - Byte overhead at varied offered traffic.

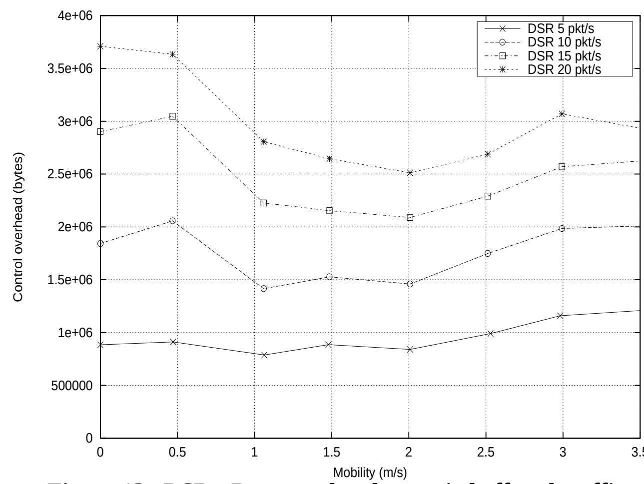

Figure 13. DSR - Byte overhead at varied offered traffic.

AODV has a more robust byte overhead than DSR at higher loads since the overhead is caused by control packets only (Figure 12). It needs to be pointed ut, however, that the small packets used (64 bytes) penalizes DSR because the source routes are large compared to the payload.

5. SIMULATIONS - REALISTIC SCENARIOS

In order to investigate how the routing protocols perform in less artificial scenarios than random movement, three "realistic" scenarios were designed and simulated. The scenarios are

- *Conference*, with low mobility
- *Event Coverage*, with fairly high mobility.
- *Disaster Area*, with some relatively slow nodes and some very fast nodes (vehicles).

The names of the scenarios attempt to categorize them and should not be construed as precise definitions. The parameters common to all simulations are given in Table 5.

Table 5: Parameters used during realistic simulations.

Parameter	Conference	Event Coverage & Disaster area
Transmitter range	25 m	250 m
Bandwidth	2 Mbps	2 Mbps
Simulation time	900 s	900 s
Number of nodes	50	50
Environment size	150x90m	1500x900 m
Traffic type	Constant Bit Rate	Constant Bit Rate
Packet rate	4 packets/s	4 packets/s
Packet size	512 byte	512 byte
Speed of a person	1 m/s	1 m/s
Speed of a vehicle	(not used)	20[a] m/s

a. Disaster Area only

Low-power radios used for indoor communication typically cannot propagate signals through walls, doors, and other obstacles in a building, without severe attenuation. Similar conditions may exist in an outdoor scenario, where objects in the terrain, such as buildings, cars, etc. may shadow radio transceivers. In order to get significant results in a simulation claiming to be realistic, obstacles to radio propagation should be modeled. Consequently, the capability to model obstacles[1] was added to the simulation tool. This feature allows the placement of obstacles in the form of boxes among the moving nodes. If the straight line between any two nodes are crossed by an obstacle, a link between these nodes is considered broken until the nodes move out of the shadowed area (the straight line is not crossed). A more realistic model would include radio signals penetrating some of the objects only partly absorbed as well as reflected radio signals. However, this simple model is a first approximation only, which assumes fully absorbing objects.

5.1 Conference scenario

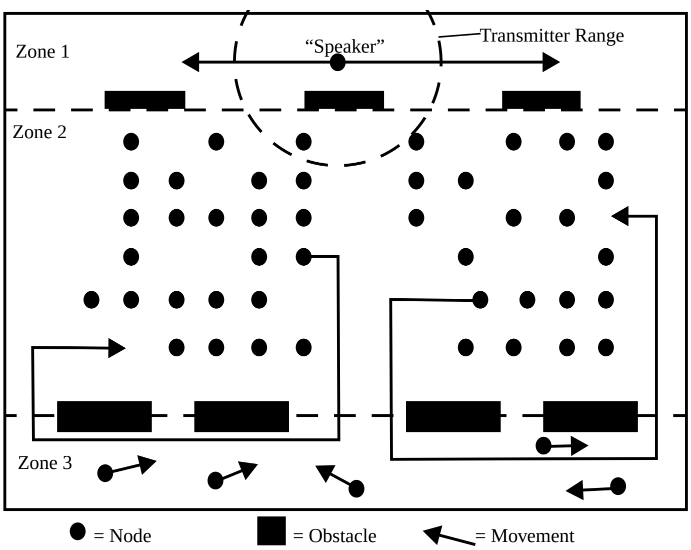

Figure 14. Conference scenario.

The conference scenario models *50* people attending a conference, seminar session, or a similar activity as illustrated in Figure 14. It includes *2* CBR sources and *6* receivers resulting in *6* CBR flows. Three zones can be distinguished in the scenario: 1) the speaker zone where the speaker moves sideways and constantly changes her/his closest neighbor in the audience, 2) the audience zone where people are rather static, when someone moves a long-lived route might break, 3) the entrance zone where curious people outside the room establish routes to the speaker to try to decide, based on the retrieved information, if they should join the session or not.

The conference scenario has rather low mobility as only *10* percent of the nodes are moving at any moment in time. The routes typically involve many hops and the traffic is concentrated to the speaker. Due to high node density, there will be relatively high radio interference.

1. Obstacles could be placed out in the original version of the mobility extensions from CMU, but these were transparent to radio signals.

The purpose of this scenario is to test responsiveness to local changes of long-lived routes. Furthermore, the low mobility in combination with the traffic concentration will stress congestion properties.

The results fare shown in Table 6. The calculated mobility for this scenario is very low. AODV and DSR perform quite well, they deliver *94* to *98* percent of the packets with an average throughput of *15.0 – 15.7 kbps*. DSDV delivers only *75.6* percent of the packets with an average throughput of *12.1 kbps*. This indicates that an ad-hoc routing protocol must adapt quickly to topology changes even for long-lived routes.

Table 6: Conference simulation results.

	DSDV	DSR	AODV
Mobility factor	0.04	0.04	0.04
Received	75.6%	98.0%	94.0%
Throughput [kbps]	12.10	15.70	15.00
Sent	21510	21510	21510
Average delay [s]	0.052	0.230	0.390
Dropped	5250	422	1376
Received packets	$16.3 \cdot 10^3$	$21.1 \cdot 10^3$	$20.1 \cdot 10^3$
Packet overhead	$44.0 \cdot 10^3$	$4.11 \cdot 10^3$	$54.7 \cdot 10^3$
Byte overhead [MB]	6.41	4.10	2.11
Average hop-count	5.32	5.79	6.45

5.2 Event coverage

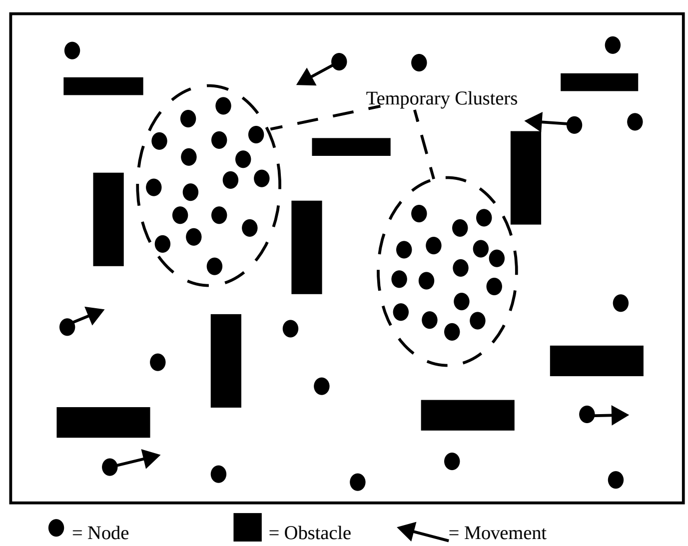

Figure 15. Event coverage scenario.

The event coverage scenario, depicted in Figure 15, models a group of *50* highly mobile people which are frequently changing position. It may represent a group of reporters that are covering a political event, a sport event, or stockbrokers negotiating at a stock exchange.

There are *9* CBR sources and *45* receivers, giving *45* CBR flows. The scenario has a rather high mobility in that at any moment *50* percent of the nodes move with a speed of *1 m/s*. Clusters consisting of around 10 nodes are formed spontaneously in the network as the nodes move. The routes consist of relatively few hops and are generally short lived. Since the simulation area has many obstacles, interference is rather low unless clusters are formed

The objective with this scenario was to test the ability to respond to fast topology changes and fluctuating traffic. Moreover, the overhead due to frequent topology changes was also of interest. The traffic was intentionally spread out all over the area to avoid congested nodes in this scenario.

The results from the simulations are presented in Table 7. All protocols have fairly high throughput, with DSR and AODV performing best. The event coverage scenario has a fairly low mobility (*0.72*) due to the low speed (*1 m/s*) of the moving nodes. The traffic is generally traversing only a few hops (on average *1.5)*. The short paths result in low byte overhead for DSR since the source routes in data packets are short (*160 kB* overhead compared to over *4 MB* for the conference scenario).

AODV gives a delay almost a magnitude lower than DSR with roughly the same throughput. This is a positive effect of the HELLO message mechanism in AODV, which gives an a priori knowledge of the neighbors. It fits nicely in this scenario since the destination of a packet sent in a cluster is often a neighbor. The route acquisition procedure need not be invoked, which saves time.

An entirely proactive protocol like DSDV may have large overhead due to frequent full topology updates, which also add extra load to the network. In this scenario the offered traffic load was low so DSDV had a fairly high throughput and low delay.

Table 7: Event coverage simulation results.

	DSDV	DSR	AODV
Mobility factor	0.72	0.72	0.72
Received	91.4%	97.7%	95.1%
Throughput [kbps]	14.75	15.71	15.36
Sent	4500	4500	4500
Average delay [s]	0.075	0.140	0.015
Dropped	385	102	219
Received packets	4115	4398	4281
Packet overhead	$42.4 \cdot 10^3$	$1.35 \cdot 10^3$	$31.4 \cdot 10^3$
Byte overhead [MB]	10.6	0.158	1.14
Average hop-count	1.46 hops	1.57 hops	1,55 hops

5.3 Disaster area

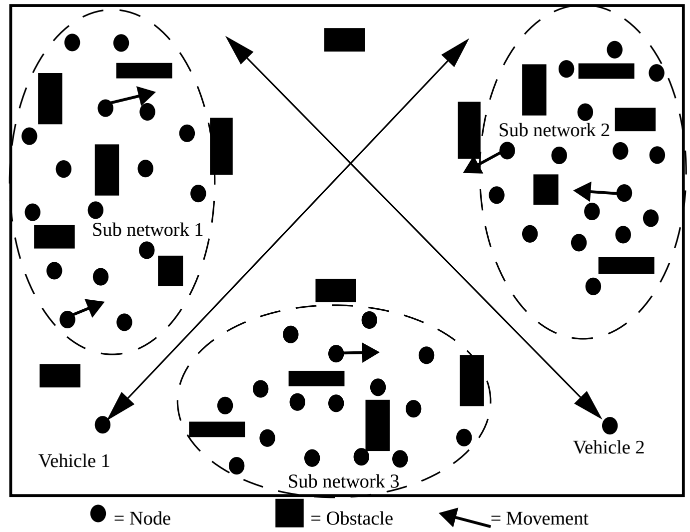

Figure 16. Disaster area scenarios

The disaster area scenario aims at representing a rescue operation at a natural disaster area. Members of the rescue team have personal communicators with ad-hoc network capability. The scene, depicted in Figure 16, consists of three groups that can intercommunicate only via the nodes mounted on vehicles *1* and *2* (helicopters, cars etc.). The vehicles are moving back and forth at *20 m/s*, while the other nodes (people) move more slowly (*1 m/s*) and randomly within each group. There are *38* CBR sources with *87* receivers for a total of *87* CBR flows.

The characteristics of this scenario include diverse mobilities (*95* percent of the nodes have low mobility and *5* percent very high) and several network partitioning events. Thus it provides a way to study how the protocols behave when node speeds are diverse and when the network partitions and heals.

Throughput is measured only when the CBR flows are actually being received in order to show the performance when the network is not partitioned. This explains the seeming discrepancy between throughput and the fraction of received packets.

Results are shown in Table 8. Due to the network partitioning events, less than *55* percent of the deliverable offered traffic is delivered. DSDV only delivers about *30* percent of the traffic, which is a clear indication that proactive protocols should not be used under these conditions.

DSDV has the lowest delay, mainly due to its low delivery ratio; packets are dropped instead of queued. AODV has slightly lower delay than DSR because the HELLO mechanism provides routes to neighbor nodes immediately.

The rather large hop-count result in substantial overhead for DSR because the source routes become relatively large.

DSDV finds the shortest paths, just like in the other realistic scenarios, but the difference is more accentuated here. However, DSDV drops a large number of packets due to invalid routes, which must be taken into account. The rapidly changing routes through the fast (vehicle) nodes are required for inter-group traffic and are fairly long. DSDV cannot adapt well to such fast route changes and thus the routes found by DSDV are relatively short.

Table 8: Disaster area simulation results.

	DSDV	DSR	AODV
Mobility factor	1.16	1.16	1.16
Received	29.5%	54.5%	54.0%
Throughput [kbps]	12.42	14.43	14.09
Sent	$29.6 \cdot 10^3$	$29.6 \cdot 10^3$	$29.6 \cdot 10^3$
Average delay [s]	0.196	1.187	0.988
Dropped	$20.9 \cdot 10^3$	$13.5 \cdot 10^3$	$13.6 \cdot 10^3$
Received packets	$8.8 \cdot 10^3$	$16.2 \cdot 10^3$	$16.0 \cdot 10^3$
Packet overhead	$41.4 \cdot 10^3$	$30.7 \cdot 10^3$	$77.3 \cdot 10^3$
Byte overhead [MB]	6.50	5.14	3.10
Average hop-count	3.42 hops	5.16 hops	5.26 hops

6. CONCLUSIONS

The simulations presented here clearly show that there is a need for routing protocols specifically tuned to the characteristics of ad-hoc networks. The mobility metric used throughout the study explicitly shows how the examined protocols behave for various degrees of relative node motion. The mobility metric is explicitly designed to capture the kind of motion important for an ad-hoc network - the relative motion of nodes. It can be used for any continuous node motion.

In networks with a dynamic topology, proactive protocols such as DSDV have considerable difficulties in maintaining valid routes, and loses many packets because of that. With increasing mobility, its strive to continuously maintain routes to every node increases network load as updates become larger.

This study clearly indicates that a reactive routing protocol is superior to a proactive one. The principle of focusing only on explicitly needed connectivity, and not all connectivity, seems to be excellent when the network consists of moving nodes. In addition, the protocol should be able to detect link failures as quickly as possible to avoid use of invalid routes.

Overall, the proactive protocols under study (AODV and DSR) behaved similarly in terms of delay and throughput. On the basis of this study both should be considered suitable for mobile ad-hoc networks. However, a number of differences between the protocols do exist.

The source routes used by DSR give increased byte overhead compared to AODV when routes have many hops and packet rates are high. DSR is, on the other hand, efficient in finding (learning) routes in terms of the number of control packets used, and does not use periodic control messages.

Data packets in AODV carry the destination address only, and not source routes. Therefore, the byte overhead for AODV is the lowest of the examined protocols. The overhead is however high in terms of packets since AODV broadcasts periodic HELLO messages to its neighbors, and needs to send control messages more frequently than DSR to find and repair routes.

The simulations in this work show that DSR performs better than AODV for low traffic loads, since it discovers routes more efficiently. At higher traffic loads, however, AODV performs better than DSR due to less additional load being imposed by source routes in data packets.

The realistic scenarios were examined to get an understanding on how the protocols would behave in an environment more realistic than the random scenarios. The results confirm most of the properties found in the random scenarios. DSDV had considerable difficulties in handling most scenarios even though the mobility was kept rather low. The conference scenario and event coverage scenarios were handled very well by both DSR and AODV, with DSR generally providing slightly better performance. The loads were rather low and did not bring out the byte overhead disadvantage of DSR. The disaster area scenario was a challenge for all protocols since most routes passed through fast nodes and links were often obscured by objects. DSR and AODV managed to deliver about 55 percent of the traffic while DSDV only delivered 30 percent. It should be noted, however, that the disaster scenario exhibited frequent partitioning of the network.

Both DSR and AODV performed quite well for almost all examined scenarios, while DSDV had serious performance problems. As a preliminary recommendation, DSR should be considered for ad-hoc networks where paths have a limited number of hops and where it is crucial to limit packet overhead. AODV on the other hand appears to perform better in networks where paths have many hops and low byte overhead is preferred over low packet overhead.

7. FURTHER WORK

The work presented herein is the first of a series of simulation studies within the area of mobile ad-hoc networking. These studies will include

- additional analysis of other proposed protocols (e.g. TORA, ZRP and CBRP),
- measurements and estimation of power consumption and processing costs,
- other traffic than CBR (e.g., TCP transfers),
- inclusion of QoS mechanisms for real-time and non real-time traffic,
- evaluation of proposed multicast routing protocols,
- analysis of interworking functions for Mobile IP.

8. ACKNOWLEDGEMENT

The authors would like to thank the students and faculty members within the Monarch Project at CMU. Their work on the ad-hoc network extensions for the ns2 simulator has to a large extent contributed to this work.

9. REFERENCES

[1] Bommaiah, McAuley and Talpade. AMRoute, “Adhoc Multicast Routing Protocol”, Internet draft, draft-talpade-manet-amroute-00.txt, August 1998. Work in progress.

[2] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva, “A performance Comparison of Multi-hop Wireless Ad Hoc Network Routing Protocols”. *Mobicom'98, Dallas Texas, 25–30 October, 1998*.

[3] Josh Broch, David B. Johnsson, David A. Maltz, “The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks”. Internet Draft, draft-ietf-manet-dsr-00.txt, March 1998. Work in progress.

[4] M.Scott Corson, S. Papademetriou, Philip Papadopolous, Vincent D. Park and Amir Qayyum, “An Internet MANET Encapsulation Protocol (IMEP) Specification”. Internet draft, draft-ietf-manet-imep-spec01.txt, August 1998. Work in progress.

[5] Kevin Fall and Kannan Varadhan, “ns notes and documentation”. The VINT project, UC Berkeley, LBL, USC/ISI, and Xerox PARC, May 1998. Work in progress.

[6] Zygmunt J. Haas and Marc R. Pearlman, “The Zone Routing Protocol (ZRP) for Ad Hoc Networks”, Internet draft, draft-ietf-manet-zone-zrp-01.txt, August 1998. Work in progress.

[7] IEEE Computer Society LAN MAN Standards Committee, “*Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*”, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York.

[8] Philippe Jacquet, Paul Muhlethaler and Amir Qayyum, “Optimized Link State Routing Protocol”. Internet draft, draft-ietf-manet-olsr-00.txt, November 1998. Work in progress.

[9] Mingliang Jiang, Jinyang Li and Yong Chiang Tay, “Cluster Based Routing Protocol (CBRP) Functional specification”. Internet draft, draft-ietf-manet-cbrp-spec-00.txt, August 1998. Work in progress.

[10] David B. Johnson and David A.Maltz, “Dynamic source routing in ad hoc wireless networks”. In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, chapter 5, pages 153-181. Kluwer Academic Publishers, 1996.

[11] David B. Johnson and David A. Maltz, “Protocols for adaptive wireless and mobile computing”. *In IEEE Personal Communications*, 3(1), February 1996.

[12] Mobile Ad-hoc Networks (MANET). URL: http://www.ietf.org/html.charters/manet-charter.html. (1998-11-29). Work in progress.

[13] Vincent D. Park and M. Scott Corson, “Temporally-Ordered Routing Algorithm (TORA) Version 1: Functional specification”. Internet draft, draft-ietf-manet-tora-spec-01.txt, August 1998. Work in progress.

[14] Vincent D. Park and M. Scott Corson, “A performance comparison of the Temporally-Ordered Routing Algorithm and Ideal Link-state routing”. In *Proceedings of IEEE Symposium on Computers and Communication ’98*, June 1998.

[15] Charles E. Perkins, “Ad Hoc On Demand Distance Vector (AODV) Routing”. Internet draft, draft-ietf-manet-aodv-01.txt, August 1998. Work in progress.

[16] Charles E. Perkins, “Ad Hoc On Demand Distance Vector (AODV) Routing”. Internet draft, draft-ietf-manet-aodv-02.txt, November 1998. Work in progress.

[17] Charles E. Perkins and Pravin Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”. In *Proceedings of the* SIGCOM '94 *Conference on Communications Architecture, protocols and Applications*, pages234-244, August 1994. A revised version of the paper is available from http://www.cs.umd.edu/projects/mcml/papers/Sigcomm94.ps. (1998-11-29)

[18] Larry L. Peterson and Bruce S. Davie, “*Computer Networks - A Systems Approach”*. San Francisco, Morgan Kaufmann Publishers Inc. ISBN 1-55860-368-9, 1996.

[19] Raghupathy Sivakumar, Prasun Sinha and Vaduvur Bharghavan, “Core Extraction Distributed Ad hoc Routing (CEDAR) Specification”, Internet draft, draft-ietf-manet-cedar-spec-00.txt, October 1998. Work in progress.

[20] The CMU Monarch Project. “The CMU Monarch Projects Wireless and Mobility Extensions to *ns”*. URL: http://www.monarch.cs.cmu.edu/. (1998-11-29). Work in progress.