

# The Performance Impact of Traffic Patterns on Routing Protocols in Mobile Ad Hoc Networks

Himabindu Pucha  
School of Electrical and  
Computer Engineering  
Purdue University  
West Lafayette, IN 47907  
hpucha@purdue.edu

Saumitra M. Das  
School of Electrical and  
Computer Engineering  
Purdue University  
West Lafayette, IN 47907  
smdas@purdue.edu

Y. Charlie Hu  
School of Electrical and  
Computer Engineering  
Purdue University  
West Lafayette, IN 47907  
ychu@purdue.edu

## ABSTRACT

In this paper, we examine the communication model widely used in simulation studies of mobile ad hoc networks (MANETs). We find that the communication model uses an overly simplistic traffic pattern which restricts the number of connections that originate from each source node to be one or two, and thus the communication model may not represent traffic patterns in many potential applications in MANETs. We then propose a new communication model which extends the previous communication model to include a more general traffic pattern that varies the number of connections per source node. We study the performance impact of traffic patterns on various routing protocols via detailed simulations of a MANET of 112 mobile nodes. Our simulation results show that many of the conclusions drawn in previous protocol comparison studies no longer hold under the new communication model. These results motivate the need for performance evaluation of MANETs to not only include rich and diverse mobility models as has been done in the past but also include diverse traffic patterns that stress a wide set of protocol design issues.

## Categories and Subject Descriptors

C.2.2 [Computer Communication Networks]: Network Protocols—*Routing Protocols*; C.2.1 [Computer Communication Networks]: Network Architecture and Design—*Wireless Communication*

## General Terms

Design, Performance

## Keywords

Mobile ad hoc networks, routing protocols, traffic patterns

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MSWiM'04 October 4-6, 2004, Venezia, Italy.

Copyright 2004 ACM 1-58113-953-5/04/0010 ...\$5.00.

## 1. INTRODUCTION

A mobile ad hoc network (MANET) is a collection of wireless mobile nodes dynamically forming a temporary network without the use of existing network infrastructure or centralized administration. In such a network, each node operates not only as a host, but also as a router, forwarding packets for other mobile nodes that may not be within the direct transmission range of each other.

Many different protocols have been proposed to solve the multi-hop routing problem in MANETs, for example, DSDV [13], TORA [11], DSR [10], and AODV [14]. To understand the performance tradeoffs between these protocols, several simulation studies have been performed [3, 6, 9, 4]. These previous comparison studies concentrated on the effects of mobility. Mobility stresses the route maintenance features in routing protocols and poses the challenge to the routing protocols on how to repair broken routes using low overhead maintenance procedures. Mobility in a MANET is typically characterized by the average speed of nodes in the system as well as the duration of the pause time during which nodes do not move. The higher the average speed is, the quicker the routes break. This reduces the effectiveness of route discoveries. In contrast, the communication model adopted in these previous studies used a fairly simplistic traffic pattern: each traffic source node communicates with only one or two other nodes in the MANET.

We argue that the limited number of connections per source node may not reflect the traffic patterns in many potential applications in ad hoc networks, and a more general communication model for comparing different routing protocols should also include a more general traffic pattern that varies the number of connections per source node. Applications developed for MANETs are peer-to-peer in nature. For example, a resource discovery in a battlefield or sharing of information between rescue personnel needs one node to communicate with a large subset of other nodes. In addition, similar to a wired network environment, each node may need to communicate with a few other nodes in order to (i) access various network services, (ii) run multiple applications, each of which may communicate with a few other nodes, (iii) maintain one or more multicast trees to support group communication, and (iv) run peer-to-peer (p2p) applications in which each node connects to multiple other peering nodes in the network.

The contributions of this paper are two-fold. First, we extend the previous communication model to include a more general traffic pattern. By varying the number of destination nodes each source

node communicates with, our proposed communication model encompasses the traffic patterns in a large set of potential applications in MANETs. Second, we revisit the performance tradeoffs between different multi-hop routing protocols for MANETs under the extended model via detailed simulations of a MANET of 112 mobile nodes. Our simulations show that many of the conclusions drawn in previous protocol comparison studies no longer hold under the more general traffic pattern. In particular, our results show that many state-of-the-art multi-hop routing protocols are optimized for the simplistic traffic pattern in which each source node communicates with one or two other nodes only, and perform poorly when communicating with an increasing number of other nodes.

The rest of the paper is organized as follows. Section 2 motivates and presents a new communication model which includes a more general traffic pattern. Section 3 describes a list of representative protocols that are studied in this paper. The methodology of our simulation study is described in Section 4. Section 5 presents the performance results of this study. Section 6 discusses the implications of the new communication model on other evaluation parameters in MANETs and finally, Section 7 concludes the paper.

## 2. COMMUNICATION MODEL

Previous protocol comparison studies have focused on a single communication model. This model consists of the following parameters:

- **Number of CBR sources ( $S$ ):** The number of CBR (Constant Bit Rate) sources is varied to stress the congestion level in the network.
- **Traffic volume ( $V$ ):** The aggregate packet rate from all CBR sources is varied to stress the throughput of the protocol.

Given the traffic volume  $V$  and the number of CBR sources ( $S$ ), the packet rate per CBR source ( $\lambda$ ) is calculated as  $\lambda = V/S$  packets/second.

There is one important parameter missing in this communication model – the number of connections per source node is fixed to one or two, with an average of 1.5. This is an outcome of the way the traffic generator provided in the wireless extensions for ns-2 [2] works. First, the number of CBR sources  $S$  is configured, instructing the traffic generator to create  $S$  random connections in the network. The traffic generator does so by going through the nodes in the network, giving each node a 50% chance of creating one connection and a 25% chance of creating a second one, until the specified number of connections is reached. Thus when  $S$  connections are generated, the number of distinct traffic sources they originate from is approximately  $\frac{2}{3}S$ , each of which has one or two connections originating from it with an equal probability.

The above communication model has been widely used in previous comparison studies of routing protocols [3, 6, 8]. For example, the studies in [3, 6] provided performance results by varying  $S$ . In doing so, they effectively studied the effect of changing  $V$  on the routing protocols. In [4, 5], the authors performed a preliminary performance analysis of reactive and proactive ad hoc routing protocols using a packet level simulator. In contrast to the other studies, the authors varied the number of connections maintained per source node. However, their evaluation also changed the volume  $V$  as the number of connections was increased. This failed to

separate the effect of increased connections per source from the effect of increased traffic volume. More importantly, their simulator did not capture the effect of multiple access interference which is a major factor in the reduction of network capacity when the number of connections in the network is large. In addition, many protocol specific optimizations such as packet buffering and passive eavesdropping (in DSR) were not implemented.

### 2.1 An Extended Communication Model

In this section, we first motivate the need for a more general traffic pattern in the performance study of routing protocols for MANETs. We then describe how to extend the previous communication model to include such a more general traffic pattern.

#### 2.1.1 Motivation

We argue that the limited number of connections per source node in the previous communication model may not reflect the traffic patterns in many potential applications in MANETs, and a more general communication model for comparing different routing protocols should also include a more general traffic pattern that varies the number of connections per source node. Multiple connections per source node can arise in many potential situations in MANETs:

- **Network Services:** Similar to wired networks, each node in a MANET may communicate with many nodes in order to access various network services such as naming, authentication, and time synchronization.
- **Multiple Applications:** Nodes may be running multiple applications at any given point in time. Even if each application communicates to only one other node, the net degree of connections emanating from a node can be large.
- **P2p Applications:** Applications developed for MANETs are likely to be peer-to-peer in nature, and p2p applications developed for the Internet are likely to be ported to MANETs. In p2p applications, each node communicates with a small number of neighbors (constant in an unstructured p2p protocols, or logarithmic in structured p2p protocols such as Pastry [15]) in the network.
- **Landmark/Hierarchical Protocols:** There has been a growing interest in ad hoc routing using clustering techniques which use landmarks/clusterheads (e.g. [12]). These landmarks potentially communicate with many other landmarks in the network when servicing outgoing packets from nodes in its service region. Conversely, when servicing incoming packets from other landmarks (service regions), a landmark may need to communicate with many nodes in its service region. Under these conditions, multiple connections per source node can arise.
- **Applications with Bursty Traffic:** Many applications such as resource discovery, group messaging, and event notification services exhibit bursty short lived data flows wherein a node may send a burst of data to one destination and the next one to a different destination. Although such applications do not maintain a large number of connections simultaneously, the net effect is to maintain more than a fixed set of 1 or 2 connections.

We note that in the above scenarios with multiple connections per source node, different packets are sent over different connections originated from each source node. This is fundamentally different from multicast communication in which the same packet is sent to multiple receivers.

### 2.1.2 Extended Communication Model

To include more realistic traffic patterns, we extend the previous communication model to include the following parameter:

- **Number of connections per traffic source ( $X$ ):** The number of connections per traffic source is varied to model the simultaneous connections per node in many potential applications in MANETs.

Specifically, a fixed traffic volume  $V$  can now be achieved by different combinations of  $S$ ,  $X$  and  $\lambda$ . Thus the new communication model separates the performance impacts of the traffic volume, the number of traffic sources, and the number of connections per traffic source. Several additional parameters in the communication model can potentially affect the performance of routing protocols. We discuss these parameters below.

**Packet arrival model:** Given an average packet rate  $\lambda$  per connection, the actual packet issue can be generated in at least two ways. In CBR traffic, the inter-arrival time  $\Delta T$  between consecutive packets is constant and a packet is sent over each connection in every interval  $\Delta T$  seconds. CBR traffic has been widely used in previous performance studies of routing protocols [6, 3, 8, 16, 10]. Alternatively,  $\Delta T$  can follow a Poisson distribution with an average value of  $\frac{1}{\lambda}$ . Such a packet arrival model mimics the transient connections in applications such as daemons that interact with network services or p2p applications which typically initiate traffic to random nodes at random instants of time and usually exchange few packets with each selected node at a time.

Although we do not present performance results for the Poisson arrival model due to space limitation, the results obtained were similar to those obtained for the CBR model.

**Connection duration:** In addition to the restricted number of connections per source node, in the previous communication model, each connection, once initiated, lasts for the entire duration of the simulation. This may not accurately model the communication patterns in many applications in which after communicating with a set of nodes for some period of time, a node switches to communicate with a different set of nodes. However, our simulation results have shown that if the duration of the connections is long enough, the relative performance of different routing protocols are similar to in the scenario where all the connections are never changed.

## 3. PROTOCOLS STUDIED

In this section, we briefly describe the key features of the ad hoc routing protocols studied. We study both on-demand (DSR, AODV) and proactive (DSDV) protocols. We also consider several versions of DSR from [8] which differ in their caching strategies to highlight the performance impact of traffic patterns on caching strategies in DSR. Lastly, we propose an improved version of DSR that performs well when the number of connections per source is varied.

## 3.1 Dynamic Source Routing

Dynamic Source Routing (DSR) [10] is a multi-hop routing protocol proposed for MANETs. It is based on the concept of source routing in contrast to hop-by-hop routing. It includes two mechanisms, *route discovery* and *route maintenance*.

Route discovery is the process by which a source node discovers a route to a destination for which it does not already have a route in its cache. The process broadcasts a ROUTE REQUEST packet that is flooded across the network in a controlled manner. In addition to the address of the original initiator of the request and the target of the request, each ROUTE REQUEST packet contains a route record, which records the sequence of hops taken by the ROUTE REQUEST packet as it propagates through the network. ROUTE REQUEST packets use sequence numbers to prevent duplication. The request is answered by a ROUTE REPLY packet either from the destination node or an intermediate node that has a cached route to the destination.

The route maintenance procedure monitors the operation of the route and informs the sender of any routing errors. If a route breaks due to a link failure, the detecting host sends a ROUTE ERROR packet to the source which upon receiving it, removes all routes in the host's cache that use the hop in error.

DSR uses aggressive caching to reduce the frequency and propagation of route discoveries. The original design of DSR [3] uses a *path cache* which stores whole source routes. Another design, called a *link cache* proposed in [8], stores individual links of routes to build a topological graph of the network. This potentially increases the effectiveness of the cache since it enables DSR to construct routes (using the graph) that were neither overheard nor discovered. The various versions of DSR evaluated in this paper are:

- **DSR-Path** proposed in [3] uses a generational path cache in which discovered and overheard routes are maintained in two separate caches. The size of the primary cache for discovered routes is 30 and the secondary cache is 64 which are the values that provide better performance as shown in [8] for a path cache.
- **DSR-PathInf** uses a path cache with no capacity limit.
- **DSR-Link** is the best performing version of DSR, called *Link-MaxLife* in [8]. It uses a link cache that has an adaptive timeout mechanism for expiring links based on the stability of the endpoint nodes of that link. The stabilities of the endpoint nodes of a link are increased by an additive factor whenever the link is used as a part of source route, and are decreased by a multiplicative factor whenever a link breaks. The timeout of any link is chosen as the minimum of the stability values of its endpoints. Routes that are of the shortest length and have the highest minimum timeout value of any of their contained links (largest lifetime) are constructed using Dijkstra's [1] algorithm on the topological graph.

## 3.2 DSR-NCache

We propose a new version of DSR, *DSR-NCache*, that performs more effective caching than *DSR-Path* when each traffic source maintains multiple connections. In *DSR-NCache*, the route cache is effectively an array of *minicaches*, one for each destination node. This structure prevents more frequently discovered or overheard

routes for one destination from unnecessarily evicting the less frequently used but still valid routes for another. Moreover, each minicache is a unified cache that stores up to  $k$  discovered and overheard routes. This design decision is prompted by the lower cycling rates of routes per minicache that occurs in such a cache organization. Each minicache replaces source routes in a FIFO order and uses a route with the minimum hop count. In this implementation, a value of 5 was chosen for  $k$ . The value of  $k$  controls the tradeoff between route availability and route freshness.

### 3.3 Destination-Sequenced Distance-Vector

Destination-Sequenced Distance-Vector (DSDV) [13] is a proactive routing protocol based on a modified form of the Bellman-Ford [1] algorithm. It overcomes some of the weaknesses of the Bellman-Ford algorithm such as poor looping properties in case of broken links. Each node maintains a routing table which lists the next hop and the hop count for all reachable destinations. Each routing table entry is tagged with a destination based sequence number to reduce propagation of stale routes. A node exchanges routing tables with its neighbors periodically or whenever a change in topology is detected. The routing table updates can be sent in two ways: (i) a full dump sends the full routing table to the neighbors and can span many packets, and (ii) an incremental update only sends those entries with a metric change since the last update.

In this paper, we use the version *DSDV-SQ* in which triggered updates are caused by the receipt of a new sequence number for a destination. We use the same set of parameters for *DSDV-SQ* as in [3].

### 3.4 Ad Hoc On-Demand Distance Vector

Ad Hoc On-Demand Distance Vector (AODV) [14] is a routing protocol that shares on-demand behavior with DSR and the use of hop-by-hop routing and destination based sequence numbers with DSDV. Routes are obtained via a discovery process similar to DSR. However, AODV stores routing information as one entry per destination in contrast to DSR which caches multiple entries per destination. A node satisfies the ROUTE REQUEST by sending a ROUTE REPLY back to the source or by increasing the hop count and re-broadcasting to its neighbors. As the ROUTE REQUEST propagates from the source to various nodes, a reverse path is set up from these nodes back to the source.

The destination sequence numbers in control packets ensure loop freedom and freshness of routing information. Timers are used to expire routes that have not been used recently. AODV ensures wider propagation of ROUTE ERRORS, achieved using a per destination predecessor list at each node, than DSR. Optimizations such as expanding ring search are used to contain the propagation of ROUTE REQUESTS. The AODV version in this study uses link layer feedback for detection of broken links. We refer to this as *AODV-LL* [3]. The set of parameters used in the simulation are based on the AODV implementation for ns-2 [2] (version 2.1b8a) provided by the authors.

## 4. METHODOLOGY

We evaluate the performance of the various routing protocols using ns-2 [2] (version 2.1b8a). We implemented *DSR-NCache* and *DSR-Link* in ns-2.1b8a. The rest of the protocols evaluated were part of the ns-2.1b8a release.

The mobility scenarios used in the simulations are generated using the modified random waypoint model [16], with nodes moving at a speed uniformly distributed between 1-19 m/s.

In all simulations, the network has 112 nodes with 80% or 90 nodes as traffic sources. A packet size of 64 bytes is used. The traffic volume in the network is kept constant at 60 packets/second. Connections initiated by traffic sources are assumed to be constant bit rate (CBR) and persist throughout the simulation, same as in previous studies. Each traffic source initiates connections to  $X$  other destination nodes selected uniformly from the set of 112 nodes in the network.

We perform simulations for a rectangular area of 2250m x 450m. The area is chosen such that the node density is at  $1/9000m^2$ , same as in previous studies [3]. A 2Mbps 802.11 radio with a transmission range of 250m is used. Each simulation lasts for 900 seconds.

The following metrics are evaluated for the routing protocols: (1) Routing overhead – The number of control packets transmitted, with each hop-wise transmission of a control packet counted as one transmission; (2) Packet delivery ratio (PDR) – The ratio of the data packets delivered to the destinations over the data packets generated by the traffic sources; and (3) Average delay – The end-to-end delay of packet routing which accounts for all possible delays caused by buffering during route discovery process, queuing at the interface queue, retransmissions at the MAC, and propagation and transfer through channel.

## 5. SIMULATION RESULTS

This section compares the performance of the protocols under the new traffic pattern. In the following, we use the same notations as in Section 2: Let  $C$  be the total number of source-destination pairs (CBR connections) in the network,  $S$  be the number of traffic sources,  $X$  be the number of connections per source, and  $\lambda$  be the packet rate per connection in packets/second. Then the total number of connections is  $C = S \cdot X$ , and the total traffic volume is  $V = S \cdot X \cdot \lambda$ . In all experiments,  $S$  is fixed at 90, and  $V$  is fixed at 60 packets/second.

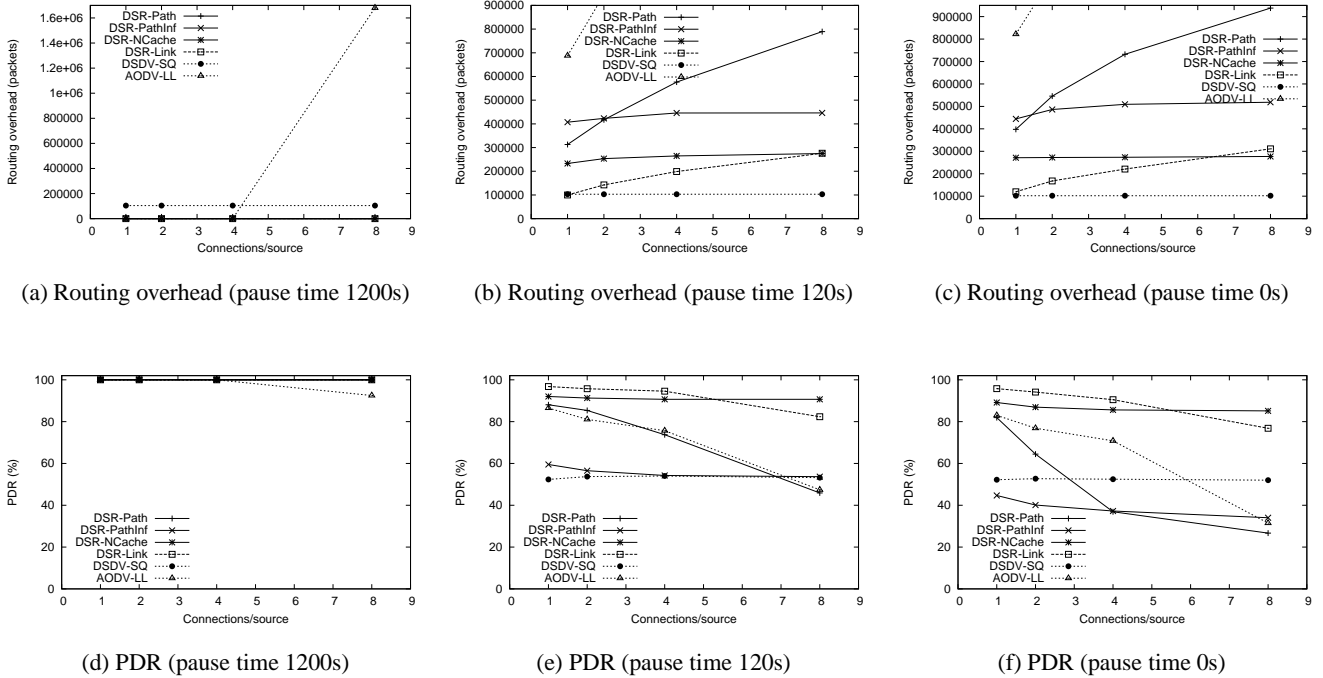
### 5.1 Effects of Number of Connections

In this experiment, we vary  $X$  from 1 to 8 and reduce  $\lambda$  proportionally, so that the total packet rate and consequently the multi access interference due to data packets at each source node remain constant. Note that an  $X$  value of 1 or 2 is similar to the traffic pattern that has been used in previous simulation studies [3, 6]. Figure 1 presents the results for 3 different pause times: a completely static network, a moderately mobile network with a pause time of 120 seconds, and a completely mobile network with a pause time of 0 second.

#### 5.1.1 DSR Performance

Figure 1(a) shows the routing overhead of all the DSR versions as  $X$  increases in a static network. Since there is no mobility, the various implementations of DSR have very low routing overhead. All of them successfully deliver almost all of the packets transmitted as shown in Figure 1(d). This is expected, since a static network does not pose any challenge to their route maintenance mechanisms.

Figures 1(b) and 1(c) show that in a network with moderate to high mobility, the routing overhead of *DSR-Path* increases almost



**Figure 1: Routing overhead and PDR comparison as  $X$  increases for 112 nodes. Routing overhead of AODV-LL is truncated for clarity of other protocols.**

linearly with  $X$ . Also at these mobilities, the PDR of *DSR-Path* drops from 81% to 26% (Figures 1(e) and 1(f)). A careful analysis of the cache structure reveals the following: In *DSR-Path*, routes are evicted from the cache either due to the capacity limitation or on receipt of a ROUTE ERROR. On one hand, a limited capacity ensures that stale routes do not stay in the cache for too long. On the other hand, a limited capacity may cause a still fresh route to a particular destination A to be evicted from the cache upon receiving a flurry of routes to a different destination B. If a route to A is required in the future for data delivery, a rediscovery of the same route becomes necessary. We refer to the eviction of routes to one destination by routes to another as *route competition*. The traffic patterns used in previous studies had on an average 1.5 connections per node, and consequently did not cause information from one route request to evict information from another route request very often. However, maintaining multiple connections at each node will result in route competition which can cause otherwise unnecessary route discoveries.

In view of this problem, the next logical step was to develop an infinite path cache version (*DSR-PathInf*) so that this *route competition* would not adversely affect the performance of DSR. However, our results show that merely increasing the cache size only solves part of the problem. Figures 1(c) and 1(f) show that for high network mobility and low values of  $X$ , although *DSR-PathInf* has lower routing overhead than *DSR-Path*, its PDR is also consistently lower than that of *DSR-Path*. This is because an increase in the cache size leads to an increase in stale cached routes, which causes an increased number of ROUTE ERRORS. Figure 2(c) shows that, for high mobility and  $X=1$ , the ROUTE ERRORS in *DSR-PathInf* are 256% more than in *DSR-Path*. Interestingly, Figure 1(f) shows

as  $X$  increases from 4 to 8, *DSR-PathInf* outperforms *DSR-Path* by delivering 8% more packets than *DSR-Path*. This occurs because the benefit of reduced route competition in *DSR-PathInf* outweighs the increased staleness of its entries.

Based on the above observations, we designed a new version of DSR, *DSR-NCache* (Section 3.2), that efficiently maintains fresh routes to multiple destinations. Figure 1(c) shows that *DSR-NCache* achieves a lower overhead than *DSR-PathInf* while delivering close to 90% of the data packets. This is because *DSR-NCache* removes route competition while maintaining the freshness of routes by implementing independent FIFO queues for source routes to different destinations, each with a smaller capacity than *DSR-Path*. As a result, it incurs far fewer ROUTE ERRORS than *DSR-PathInf* and *DSR-Path*. Figure 2(c) shows that at  $X=8$ , *DSR-NCache* has 37% and 78% fewer ROUTE ERRORS than *DSR-Path* and *DSR-PathInf*, respectively.

The routing overhead of *DSR-NCache* and *DSR-PathInf* remains largely constant with increased  $X$ . This can be explained as follows. First, since both cache structures avoid route competition, there exists more diversity in the routes stored. This reduces the incremental penalty of additional connections since ROUTE REPLYs for one destination can be used to deliver packets to another destination. This happens because a source route of length  $P$  can be potentially used to deliver packets to  $P - 1$  destinations using the  $P - 1$  partial paths. We term this as *reuse*. Specifically, we term the usage of source routes discovered for other destinations and stored in the local cache to deliver packets to the current destination as *local reuse*. Note that local reuse increases as  $X$  increases due to the increased number of source routes stored in each local cache and the increased opportunities for using them. Second, this diverse

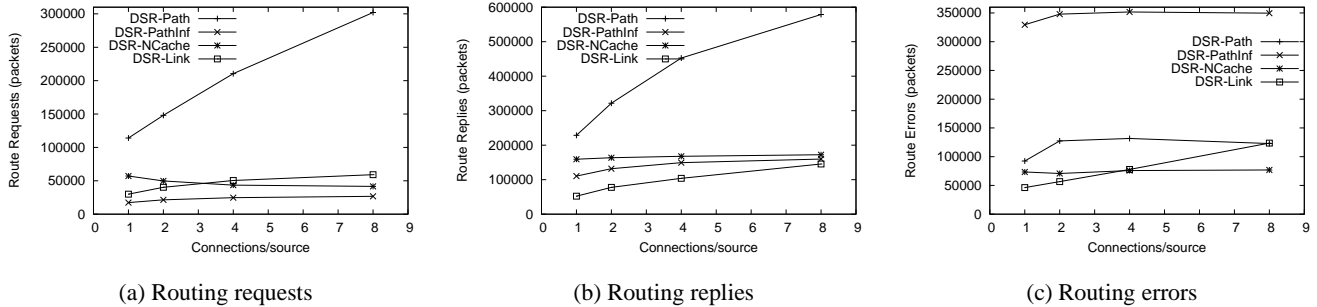


Figure 2: Routing message breakdown of DSR versions as  $X$  increases (pause time 0s).

cache at each node increases the confinement of the propagation of ROUTE REQUESTS and thus reduces the number of ROUTE REPLYs per ROUTE REQUEST. We term this as *remote reuse* where the cached routes of intermediate nodes are used to reply to ROUTE REQUESTS and thus used for sending packets by a source node. Remote reuse also increases with increased  $X$ .

Another design of DSR is based on a graph-based cache structure as in [8]. This graph-based cache has an inherent advantage in that route competition does not exist. Though this fact was not mentioned in [8], we believe this is a subtle but important feature of this design. The disadvantage of a graph based cache is that the protocol needs to specify timeouts for links since there is no longer a capacity limitation that is used to ensure freshness of routes.

*DSR-Link* uses an adaptive timeout mechanism which considerably improves its performance. For low values of  $X$ , *DSR-Link* has a lower routing overhead than *DSR-Path* and *DSR-NCache* since it has the ability to construct routes it has not explicitly discovered. However, Figures 1(c) and 1(f) also show that at high mobility, as  $X$  increases, the routing overhead of *DSR-Link* increases and at  $X=8$  it exceeds that of *DSR-NCache*. At  $X=8$ , *DSR-Link* delivers 9% fewer packets than *DSR-NCache*. This is due to the higher number of ROUTE ERRORS compared to *DSR-NCache* at  $X=8$  as seen in Figure 2.

The reason that the routing overhead of *DSR-Link* increases with  $X$  can be explained as follows. The adaptive timeout mechanisms in the design of *DSR-Link* help to adapt the timeout of each newly added link to the past history of the stability of the endpoint nodes of that link. This basically is a form of adaptation based on continuous feedback from the traffic pattern. The adaptation works as follows: Every time a source route is used to deliver a packet, the stability of each endpoint in all the links of the source route is increased by an additive factor (4); every time the link containing these endpoints is found to be broken, the stability of these endpoints is reduced by a multiplicative factor (2). Therefore, the positive feedback for the stability of the endpoints depends on the inter-arrival time of the requests. If the inter-arrival time of requests decreases as is the case when  $X$  increases, there will be less positive feedback to increase the stability of the endpoints of links. This in turn reduces the timeout chosen for a future addition of a link between those endpoints. The reduction in timeout causes the links to expire even before they are actually broken thereby increasing the ROUTE REQUESTS and the corresponding ROUTE REPLYs as  $X$  increases as seen in Figure 2. Also, as  $X$  increases, *DSR-*

*Link* tries to reuse as many links discovered for one connection as possible by constructing routes to the remaining destinations using Dijkstra’s algorithm on the graph of the network. Since these constructed routes are likely to be less accurate than explicitly discovered routes, they contribute to the increase in the number of ROUTE ERRORS as  $X$  increases. This in turn causes the overall increase in routing overhead and reduction in PDR.

We summarize the performance of the three main versions of DSR: *DSR-Path*, *DSR-NCache*, and *DSR-Link*. The value of  $X$  adversely affects *DSR-Path* causing its routing overhead to grow and PDR to fall with an increase in  $X$ . This is primarily due to its cache design, as the improved caching design in *DSR-NCache* achieves a stable routing overhead and PDR with increased  $X$ . Although *DSR-Link* has a lower routing overhead than and comparable PDR to *DSR-NCache* for small values of  $X$ , its performance degrades as  $X$  increases. Note that we only modified the caching structures in *DSR-NCache* to improve its performance while preserving its basic nature of source routing.

### 5.1.2 DSDV Performance

In a static network, even though on-demand protocols have close to zero overhead, *DSDV-SQ* incurs a constant overhead of approximately 100,800 packets as it uses periodic table exchanges for route maintenance. In fact, *DSDV-SQ* maintains a constant routing overhead for all values of mobility and  $X$ . This happens because the routing overhead depends only on the periodic update interval (15s) and the size of the network. As the route aggregation time used is 1s, every node in the network on average triggers an update every second. This results in a constant routing overhead of simulation time (900s) \* network size (112) = 100,800 packets. For moderate to high mobility, Figures 1(b) and 1(c) show that this constant routing overhead is in fact less than all the versions of DSR for all values of  $X$ . However, the low overhead advantage of *DSDV-SQ* is accompanied by its lower PDR. This is because the routing tables do not converge when the network is highly dynamic, resulting in packets being routed using stale entries. Interestingly, at high mobility and high  $X$  values, the PDR of *DSDV-SQ* is superior to those of *DSR-Path* and *AODV-LL*.

### 5.1.3 AODV Performance

Figure 1 also shows the effect of increasing  $X$  on the performance of *AODV-LL*. First, in the no mobility scenario, Figures 1(d) and 1(a) show that the PDR and routing overhead of *AODV-LL* are

comparable to in other protocols for up to  $X=4$ . At  $X=8$ , however, *AODV-LL* suffers a large overhead and a slight reduction in its PDR. This can be explained as follows. The active timeout of routes in *AODV-LL* is refreshed every time a packet is transmitted using that route. Thus, if the inter-arrival time of the packets is larger than the active timeout, these routes will be invalidated before being used despite them being still valid. In this scenario, every packet attempted to be sent may potentially result in a ROUTE REQUEST. At  $X=8$ , the inter-arrival time of traffic for a particular destination is  $\frac{90 \times 8}{60} = 12s$ , which is greater than the active timeout (10s) used in the simulations. As a result, a static network yielded a routing overhead of 1680418 packets. This high routing overhead is further responsible for the reduction in PDR. The same scenario with a timeout of 20s resulted in much lower overhead. This exemplifies the crucial role of this timeout value in the performance of the protocol. To avoid this mismatch between the timeout value and inter-arrival time, an adaptive timeout approach for AODV similar to that for *DSR-Link* may prove useful.

Results for the medium mobility scenario as depicted in Figures 1(b) and 1(e) indicate that the routing overhead of *AODV-LL* is the highest as compared to all the other protocols and increases with  $X$ . *AODV-LL* also shows a reduction in the PDR as  $X$  increases. A similar trend is observed for the high mobility scenario (Figures 1(c) and 1(f)).

The higher routing overhead of *AODV-LL* at medium to high mobility is best explained by comparing it to *DSR-Path*. Table 1 shows that for  $X=2$ , *DSR-Path* and *AODV-LL* initiate comparable numbers of route discoveries (10348 and 10953). This is caused by frequent route breakages due to the medium to high mobility. However, these comparable route discoveries translate into 524% more ROUTE REQUEST transmissions in *AODV-LL* than in *DSR-Path*. This is because *DSR-Path* uses aggressive caching and overhearing, due to which it has fewer propagating ROUTE REQUESTS and better confinement of these propagating ROUTE REQUESTS.

When  $X$  is increased to 8, the cost of maintaining more connections (i.e., number of route discoveries) in both *AODV-LL* and *DSR-Path* increases. *DSR-Path* is adversely affected because the limited few cached routes in *DSR-Path*'s cache (due to the limited capacity) can be used only for other nodes along those routes. *AODV-LL* is adversely affected because its distributed routes set up for one connection cannot be used to deliver packets for another connection, and because of the mismatch between the active timeout (10s) and the packet inter-arrival time (12s). To isolate the effect of this mismatch, we ran *AODV-LL* with an active timeout of 20s. Table 1 shows that with a pause time of 0s, adjusting the active timeout reduces the number of route discoveries by 53% and the number of route request packets by 32%. This improves the PDR from 31% to 60% as shown in Figure 3.

Table 1 also shows that as  $X$  is increased from 2 to 8, while *AODV-LL* and *DSR-Path* have more than a two-fold increase in the number of discoveries, *DSR-NCache* has less than 15% increase in the number of discoveries. This is because of its better reuse of source routes already cached and not yet evicted. In particular, *DSR-NCache* (and similarly *DSR-Link*) benefits from local reuse (using routes discovered for one destination for another destination) and improved remote reuse (using routes cached by other nodes) as compared to *DSR-Path* and *AODV-LL*. As a result, it incurs lower overhead than *AODV-LL* and *DSR-Path* and achieves a higher PDR.

	Route Dis.		Route Req.	
	X=2	X=8	X=2	X=8
<i>DSR-Path</i>	10348	24395	148046	302026
<i>DSR-NCache</i>	7844	9078	49732	41604
<i>AODV-LL</i>	10953	26345	924133	1759796
<i>AODV-LL (20s)</i>	10953	17321	924133	1328189

**Table 1: Routing overhead comparison between DSR and AODV simulations for pause time 0s. *AODV-LL (20s)* refers to *AODV-LL* with an active timeout of 20s.**

The performance of *AODV-LL* under the new traffic pattern can potentially be improved with new design choices. For example, *AODV-PA* [7], an AODV variant, proposes the accumulation of paths in ROUTE REQUESTS and ROUTE REPLYs and uses such source route information gathered in the control packets for route reuse among multiple connections. Such route reuse reduces the overall routing overhead. In other words, the performance of AODV under the new traffic pattern can potentially be improved by adopting a mechanism similar to source routing of DSR. We could not include *AODV-PA* in our study due to the lack of a publicly available implementation.

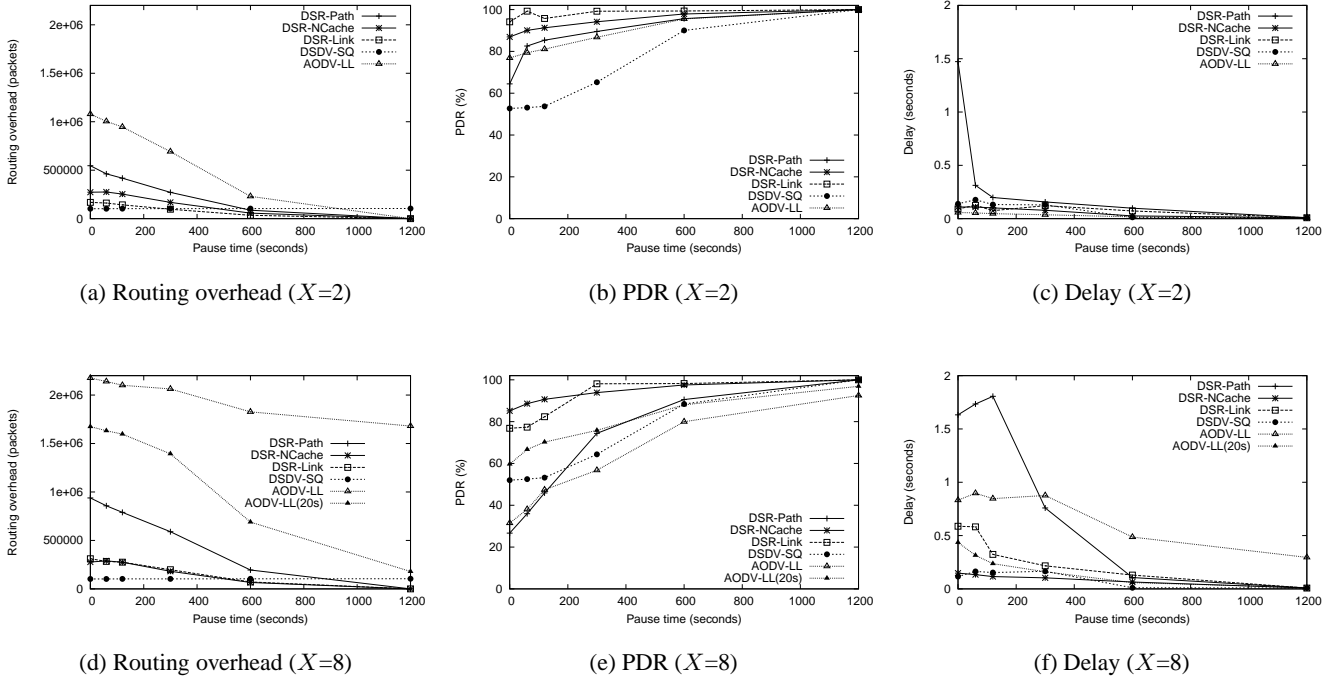
## 5.2 Effects of Mobility

In this section, we revisit the simulation results of Section 5.1 to study the effects of mobility on different routing protocols for two values of  $X$ , 2 and 8. In particular, the mobility in the network was successively decreased from using a pause time of 0 second to a completely static network.

Figure 3 compares the performance of all the protocols as the mobility changes. For  $X=2$ , the results comparing *DSR-Path* and *AODV-LL* are similar to those reported in [3] which used an average value of  $X=1.5$ . In particular, *AODV-LL* has a higher routing overhead than *DSR-Path* for all mobilities. The PDR of both on-demand protocols are close to 100% at low mobilities. For higher mobilities, the PDRs of both protocols drop below 80%. However, *AODV-LL* achieves a higher PDR than *DSR-Path*. This is consistent with results in [6], which used a 100 node network.

However, when  $X$  is increased to 8, the routing overhead of both *DSR-Path* and *AODV-LL* increases more quickly and the PDR decreases more quickly with the mobility compared to  $X=2$ . The increase in routing overhead is especially large (beyond the range of the y-axis) for *AODV-LL* because of the mismatch between its active timeout and the inter-arrival time of the traffic as well as wider propagation of ROUTE REQUESTS. To isolate the effect of the non-optimal active timeout, we reran *AODV-LL* with an active timeout of 20s for  $X=8$  and the results are shown in Figures 3(b)(d)(f). The results show that an optimal active timeout can drastically reduce the routing overhead and latency and improve the PDR.

*DSDV-SQ* delivers fewer packets as the mobility in the network increases. This is because the routing tables fail to converge in such a dynamic environment. Note that the lower constant routing overhead of *DSDV-SQ* regardless of both  $X$  and mobility is an attractive feature. When  $X=2$ , *DSDV-SQ* incurs lower overhead but also has lower PDR compared to *DSR-Path* and *AODV-LL* at medium to high mobilities. A significant observation is that when  $X=8$ , *DSDV-SQ* achieves higher PDR while incurring lower overhead than *DSR-Path* and *AODV-LL* (with active timeout 10s) at high mobilities.



**Figure 3: Routing overhead, PDR and delay for varying mobility and  $X$ . Routing overhead of AODV-LL is truncated for clarity of other protocols. AODV-LL (20s) refers to AODV-LL with an active timeout of 20s.**

Compared to *DSR-Path* and *AODV-LL*, both *DSR-Link* and *DSR-NCache* incur almost the same amount of routing overhead as  $X$  is increased from 2 to 8, and consequently achieving almost the same high PDR across all mobilities. A close look shows that *DSR-Link* suffers a higher routing overhead and achieves a lower PDR at high mobilities with  $X=8$  than with  $X=2$ . This is due to the reliance of its adaptive timeout on the traffic pattern as explained in Section 5.1.

*DSR-NCache* exhibits superior delay performance regardless of  $X$ , significantly reducing the latency in highly mobile networks with rich connectivity. The delay of *DSR-Path* sharply rises for high mobility. This high delay in *DSR-Path* occurs even for medium mobilities when  $X=8$ . Although *AODV-LL* and *DSR-Link* have good delay performance at  $X=2$ , their delay increases significantly with the mobility when  $X=8$ . Note that in [6], a similar result where *AODV-LL* has lower delay than *DSR-Path* for a network size of 100 nodes and  $X=1.5$  was observed.

In summary, the effects of mobility on different routing protocols for  $X=8$  are significantly different from those for  $X=2$ , reaffirming the importance of using diverse traffic patterns to evaluate different routing protocols.

## 6. DISCUSSIONS

### 6.1 Effects of Traffic Volume

The traffic volume  $V$  in our model can be increased by increasing the packet rate per connection  $\lambda$  for a fixed number of connections per source  $X$  and a fixed number of traffic sources  $S$  ( $V = \lambda \cdot X \cdot S$ ). Note that varying the traffic volume  $V$  is very different from varying  $X$  while keeping  $V$  constant because in the latter, the contribu-

tion to medium access interference from data packets remains the same. Increased traffic volume for a fixed  $X$  affects DSR more adversely than AODV because DSR has been shown to have a high MAC load [6]. In DSR, a large fraction of overhead is composed of unicast packets which incur the costs of RTS/CTS exchanges whereas AODV has a higher fraction of broadcast packets in its routing overhead. This implies that for  $X=1$ , if the traffic volume is increased, AODV will increasingly outperform DSR as was shown in [6]. However, as  $X$  increases, the large number of ROUTE REQUESTS in AODV outweighs the lower link stress from fewer unicasts. Thus an isolated argument about the impact of MAC load on the routing performance may not be applicable for high values of  $X$ . In simulation runs for high values of  $X$  (not depicted in the paper due to lack of space), we found that as the  $V$  increases, the routing overhead of DSDV essentially remains constant and it is able to deliver more packets than either *AODV-LL* or *DSR* (*DSR-Path* and *DSR-Link*). In summary, while the increased traffic volume in previous studies only stressed the effect of MAC load due to data traffic, our model stresses the effect of MAC load due to both data traffic and the maintenance of multiple connections.

### 6.2 Effects of Network Size

As the network size  $N$  increases, the average hop length of routes increases as  $\theta(\sqrt{N})$ . This increase in hop length increases the probability that a data packet will experience an error along the route. Although this probability increases irrespective of the routing protocol used, protocols that use source routing like DSR have an additional penalty. Since carrying source routes increases the average packet size, DSR has an increased probability of route error compared to DSDV and AODV. Since DSDV has been shown



to not converge at high mobilities, AODV seems the more scalable of the three protocols. However, using a more general traffic pattern, a wider set of tradeoffs emerge. Our simulations show that for a fixed network size, the routing overhead of *AODV-LL* and *DSR-Path* can increase significantly when the number of connections per source node increases, while the routing overhead of DSDV and *DSR-NCache* is largely unaffected. Therefore, when both  $X$  and  $N$  increase, the scalability of *AODV-LL* and *DSR-Path* is adversely affected by the increased routing overhead associated with both  $X$  and the increased hop length, while other protocols such as *DSR-NCache* are only affected by the overhead from the latter. This suggests that scalability studies of routing protocols should also take traffic patterns into consideration.

## 7. CONCLUSIONS

In this paper, we first pointed out that the communication model uses an overly simplistic traffic pattern which restricts the number of connections that originate from each source node to be 1.5 on average, and thus may not represent traffic patterns in many potential applications in ad hoc networks. We then proposed a more general communication model that varies the number of connections per source node and thus enables performance studies to decouple the performance impact of the traffic volume, the number of traffic source nodes, and the number of connections per traffic source node. We then presented a detailed study of the effects of varying the number of connections per source ( $X$ ) on the performance of various routing protocols.

Our simulations showed that many of the conclusions drawn in previous comparison studies no longer hold true under the new traffic pattern. First, we showed that the performance of two popular protocols DSR and AODV degenerate when a node on average maintains connections to a larger subset of nodes than was previously used while keeping the total traffic volume per traffic source constant. More importantly, we showed that the degeneration was not due to an increase in medium access interference but rather the design of the protocols. Second, our results showed that as  $X$  is increased, the routing overhead of DSR and AODV increases which implies that the volume of data packets that can be delivered reduces. Thus, the value of  $X$  is also an important factor in deciding the maximum traffic volume a protocol can withstand. Third, we showed that in addition to the route discovery and maintenance mechanisms and the overhead of carrying the source route in each data packet, the value of  $X$  is also an important factor in deciding the network size to which a protocol can be scaled. Fourth, we showed that a proactive protocol such as DSDV can be attractive in networks with rich connections and moderate to low mobility. Fifth, our results indicate that the lower inter-arrival rates between data packets on the same connection affect protocols with static and adaptive timeout mechanisms in addition to reducing caching efficiency. This indicates that applications such as those that interact with network services and consequently have short lived flows pose a challenge to routing protocols. Lastly, we proposed a new caching structure for DSR that performs well disregarding the number of connections per source node.

Our work motivates the need for performance evaluation of ad hoc networks to not only include rich and diverse mobility models as has been done in the past but also include diverse traffic patterns that stress a wide set of protocol design issues.

## Acknowledgment

Support for this research has been provided in part by NSF award ANI-0338856.

## 8. REFERENCES

- [1] D. Bertsekas and R. Gallager. *Data Networks*. pages 297-333, Prentice Hall, 1987.
- [2] L. Breslau. et al., Advances in network simulation. *IEEE Computer*, 33(5):59-67, May 2000.
- [3] J. Broch. et al., A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM MobiCom*, October 1998.
- [4] S. R. Das, R. Castaneda, J. Yan, and R. Sengupta. Comparative performance evaluation of routing protocols for mobile ad hoc networks. In *Proc. of ICCCN*, October 1998.
- [5] S. R. Das, R. Castañeda, and J. Yan. Simulation-based performance evaluation of routing protocols for mobile ad hoc networks. *Mobile Networks and Applications*, 5(3):179-189, 2000.
- [6] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM*, March 2000.
- [7] S. Gwalani, E. M. Belding-Royer, and C. E. Perkins. AODV-PA: AODV with path accumulation. In *Proc. of Next Generation Internet Symposium*, May 2003.
- [8] Y.-C. Hu and D. B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proc. of ACM MobiCom*, August 2000.
- [9] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proc. of ACM MobiCom*, August 1999.
- [10] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.
- [11] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. of IEEE INFOCOM*, April 1997.
- [12] G. Pei, M. Gerla, and X. Hong. LANMAR: Landmark routing for large scale wireless ad hoc networks with group mobility. In *Proc. of IEEE/ACM Mobihoc*, August 2000.
- [13] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM*, August 1994.
- [14] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of IEEE WMCSA*, February 1999.
- [15] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. of Middleware*, November 2001.
- [16] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proc. of IEEE INFOCOM*, April 2003.