Available online at www.sciencedirect.com
ScienceDirect
Computer Networks 51 (2007) 3595–3616
Computer Networks
www.elsevier.com/locate/comnet

# The performance impact of traffic patterns on routing protocols in mobile ad hoc networks

Himabindu Pucha, Saumitra M. Das, Y. Charlie Hu *

*Center for Wireless Systems and Applications in the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907, United States*

Received 26 March 2006; received in revised form 19 October 2006; accepted 27 February 2007
Available online 15 March 2007

Responsible Editor: V.R. Syrotiuk

**Abstract**

As mobile ad hoc network (MANET) systems research has matured and several testbeds have been built to study MANETs, research has focused on developing new MANET applications such as collaborative games, collaborative computing, messaging systems, distributed security schemes, MANET middleware, peer-to-peer file sharing systems, voting systems, resource management and discovery, vehicular computing and collaborative education systems. The growing set of diverse applications developed for MANETs pose far more complex traffic patterns than the simple one-to-one traffic pattern, and hence the one-to-one traffic pattern widely used in previous protocol studies has become inadequate in reflecting the relative performance of these protocols when deployed to support these emerging applications.

As a first step towards effectively supporting newly developed and future diverse MANET applications, this paper studies the performance impact of diverse traffic patterns on routing protocols in MANETs. Specifically, we propose a new communication model that 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 an ad hoc network 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 traffic patterns. These results motivate 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.


*Keywords:* Traffic patterns; Routing protocols; Mobile ad hoc networks; Route caching

## 1. Introduction

A mobile ad hoc network 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,

* Corresponding author. Tel.: +1 765 494 9143.
*E-mail addresses:* hpucha@purdue.edu (H. Pucha), smdas@purdue.edu (S.M. Das), ychu@purdue.edu (Y.C. Hu).

1389-1286/$ - see front matter 
doi:10.1016/j.comnet.2007.02.009

but also as a router, forwarding packets for other mobile nodes that may not be within the direct transmission range of each other. Example applications of ad hoc networks include disaster relief personnel coordinating efforts after an earthquake or soldiers relaying information on the battlefield.

A fundamental problem in mobile ad hoc networks (MANETs) is to develop a routing protocol that delivers applications packets from source nodes to the destination nodes in a scalable manner and without incurring high control overhead. Over the past decade, many different protocols have been proposed to solve the multi-hop routing problem in ad hoc networks, for example, DSDV [32], TORA [29], DSR [22], and AODV [33]. To understand the performance tradeoffs between these protocols, several simulation studies have been performed [8,12,21,10]. Since the node mobility is the most important factor that contributes to the challenge of scalable and efficient routing in MANETs, as it directly contributes to the route breakage, these previous comparison studies have concentrated on the effects of mobility, such as varying the node speed and pause time. In contrast, the communication model used in these previous studies has received much less attention and has largely used a simplistic traffic pattern: each traffic source node communicates with only one or two other nodes in the ad hoc network.

As mobile ad hoc network systems research has matured and several testbeds have been built to study MANETs, many new MANET applications have been proposed and developed. These include collaborative games [2], collaborative computing [35,26], messaging systems [28,41], distributed security schemes, MANET middleware [23,25,5,4], peer-to-peer file sharing systems [6,40,37], voting systems [13,15], resource management and discovery [27,31], vehicular computing [14] and collaborative education systems [16].

Such a diverse and mature set of applications present far more complex traffic patterns as a workload to the routing protocols than the simple one-to-one traffic pattern that has been assumed in previous routing protocol studies. One of the most apparent departure from the simple one-to-one traffic is that similar to a wired network environment, each node may need to communicate with many other nodes in order (i) to access various basic network services such as naming, authentication, and time servers, (ii) to run multiple applications (e.g., both resource discovery and file sharing), each of which may communicate with a few other nodes, (iii) to maintain one or more multi-cast trees to support group communication, (iv) to perform application tasks in collaborative applications, for example, a resource discovery applications in a battlefield or an information sharing application for rescue personnel needs one node to communicate with a large subset of other nodes.

To demonstrate the significant impact of traffic patterns in comparing different routing protocols, we show the performance of two representative on-demand routing protocols, DSR and AODV, and a proactive protocol, DSDV, under varying number of connections per source node in a network of 112 nodes. The same protocol parameters and the same random-way point model as specified in [8] are used except that the number of traffic sources are 90. The number of connections every traffic source maintains is varied between 1 and 8, while keeping the overall volume of packets in the network constant at 60 packets/s. The results for the pause time 0 are plotted in Fig. 1a and b. The results show that the performance of DSR and AODV degenerates drastically as the number of

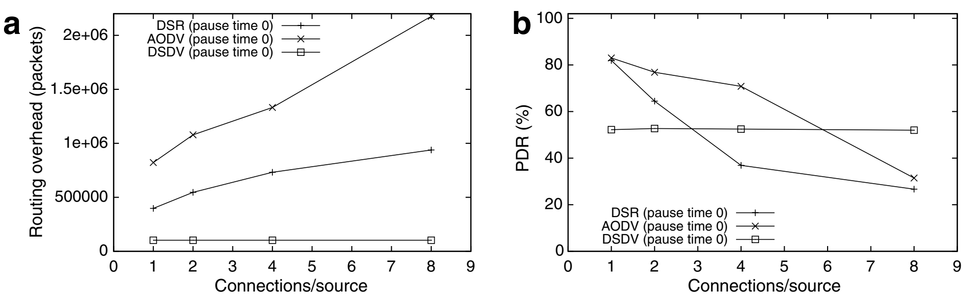


Fig. 1. Need for a general traffic pattern. (a) Routing overhead as connections/source increases for pause time 0 s. (b) Packet delivery fraction as connections/source increases for pause time 0 s.

connections per source node increases, although the total packet rate per source node is kept constant. For example, the routing overhead of DSR increases from below 400,000 packets to over 900,000 packets as the number of connections per source ($X$) increases from 1 to 8. This effect is also evident in AODV but in an even larger scale; the routing overhead increases from 800,000 to up to 2 million packets. In contrast, the proactive protocol DSDV, though inferior for small values of $X$, delivers more packets with a constant and lower overhead than both DSR and AODV for larger values of $X$. These results suggest that an in-depth study of the effects of traffic patterns on the performance of routing protocols is required.

As a first step towards effectively supporting newly developed and future diverse MANET applications, this paper studies the performance impact of diverse traffic patterns that potentially exist in these applications on a set of contemporary routing protocols in MANETs. The contributions of this paper are twofold. 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 ad hoc networks.

Second, we revisit the performance tradeoffs between different multi-hop routing protocols for ad hoc networks under the new traffic patterns via detailed simulations of an ad hoc network 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. These results demonstrate the need for performance evaluation of ad hoc network routing protocols 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.

While this work (based on [34]) is the first devoted to studying data traffic patterns and its impact on protocol performance on MANET routing protocols, there are certain limitations of this study. Despite the flexibility of varying different parameters via simulation, its not clear how well such parameters correspond to the traffic patterns we are likely to see in deployed mobile ad hoc networks. However, our study provides a first order approximation of the significant impact of traffic patterns and draws the community's attention towards this effect. We anticipate more accurate studies of this subject will be enabled when real traffic traces for such networks become available in the future. There has been some recent progress towards making available real traffic traces of wireless networks [1]. However, these traces concentrate on Internet usage of one-hop wireless LAN networks whose traffic patterns and mobility information of users are likely to be different from ad hoc network applications. We hope with the increasing interest in ad hoc network implementations and deployment, we will soon have real user traffic traces from events such as large conferences. We plan to work with the community to make progress on collecting real ad hoc network traces which will enable further study on the subject of this paper.

This paper extends an earlier conference version of our work on this topic [34]. This paper includes an expanded introduction and a new background section in order to convey the motivation of the paper better. The communication models have been described in more detail as well. We have also included several new results on transient connections, traffic concentration, network size and traffic volume in this paper that provide a more in-depth treatment to the topic and have also summarized the findings and conclusions as simple guidelines for future MANET routing protocol design.

The rest of the paper is organized as follows. Section 2 reviews the three major parameters that have been used in previous comparison studies of routing protocols for MANETs. Section 3 motivates and presents a new communication model which includes a more general traffic pattern. Section 4 describes a list of representative protocols that are studied in this paper. Section 5 describes the methodology of our simulation study. Section 6 presents the performance results of this study and Section 7 concludes the paper.

## 2. Background: Performance evaluation of routing protocols in MANETs

In this section, we review the three major parameters that have been used in previous performance studies of various routing protocols proposed for

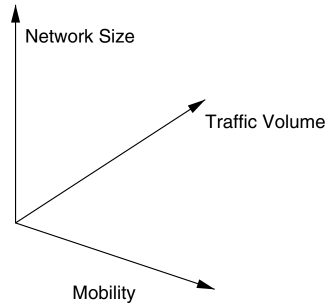


Fig. 2. Major parameters in simulation studies of routing protocols for MANETs.

mobile ad hoc networks. The three major parameters are depicted in Fig. 2, and are discussed in detail in the following.

### 2.1. Network size

The network size of an ad hoc network refers to the total number of nodes in the ad hoc network.

As the network size $N$ increases, the average hop length of routes increases as $\theta(\sqrt{N})$ [17]. This increase in hop length increases the probability that a data packet will experience an error while being routed along the path. Although this probability increases irrespective of the routing protocol used, protocols that use source routing such as DSR have an additional penalty from carrying longer routes in the packet header. For example, in [12], the authors showed that AODV performs better than DSR as the network size is increased.

### 2.2. Mobility model

The mobility in an ad hoc network can be characterized by the speed of nodes in the systems as well as the duration of the pause time during which nodes do not move.

The average speed of nodes in the system determines the rate at which links break and consequently the overhead consumed by the route maintenance in on-demand protocols. The increased routing overhead in turn can affect the packet delivery ratio of routing protocols from increased multi-access interference.

One of the most widely used mobility models is the *random waypoint* model first proposed in [22] and refined in [8]. In this model, nodes in a large "room" choose some destination, and move towards it at a random speed uniformly chosen from $(0, V_{\max}]$, where $V_{\max}$ is the maximal speed of the simulation. Upon reaching the destination, a node pauses for a system-wide constant time before moving towards the next chosen destination at a newly chosen speed.

The random waypoint model has been widely used in performance comparison studies of routing protocols. In particular, authors of [8,12] compared several routing protocols using the random waypoint model varying the pause time. They also evaluated the effect of different average speeds on the performance of routing protocols.

In [42], the authors showed that random waypoint model fails to attain a steady state average speed for its nodes when the speed of the nodes is chosen randomly between $(0, V_{\max}]$. Consequently, the results obtained under these conditions become a function of the simulation time. They also proposed a simple fix of setting a non-zero minimum speed of nodes which allows the simulated mobile network to approach within 10% of the steady-state average speed quickly. In [43], the same authors proposed a general technique for transforming any given mobility model with decay into a stationary one.

The work in [19,9] studied the performance impact of several other mobility models such as the Brownian, the Column, the Pursue and the Random Gauss–Markov models on routing protocols.

Several studies (e.g., [21]) emphasized the evaluation of routing protocols based on specific mobility scenarios. The motivation was to measure the performance of protocols in real world situations (e.g., a conference). The scenario-specific communication models had arbitrarily chosen sources and receivers based on the scenario under consideration.

More recently, the authors of [20] proposed the use of obstacles and node movements using the Voronoi diagram of obstacle vertices as an aid to the evaluation of performance tradeoffs among routing protocols.

### 2.3. Communication model

A specific communication model included with the random waypoint mobility model [8] in the ns-2 [7] network simulator has been widely used in previous protocol comparison studies. 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 routing 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/s.

There is one important parameter missing in the previous communication model – the number of connections per source node is fixed to one or two, with an average of 1.5. This fixation is an outcome of the way the traffic generator 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 source 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 communication model has been widely used in previous comparison studies of routing protocols [8,12,19]. For example, the studies in [8,12] provided performance results by varying $S$. In doing so, they effectively studied the effect of changing $V$ on the routing protocols.

In [10,11], 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 were 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.

## 3. 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 (Section 2.3) to include such a more general traffic pattern.

### 3.1. 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:

- *New MANET applications*: New applications are being rapidly developed for MANETs [2,35,26,28,41,23,25,5,4,6,40,37,13,15,27,31,14,16] and most such applications involve peer-to-peer communication to multiple nodes due to the decentralized environment of MANETs. For example, resource discovery involves probing multiple nodes, file sharing involves searching multiple nodes, voting systems involves contacting multiple nodes, and collaborative games involves collecting multiple player attributes.
- *P2p applications*: Mature P2p applications developed for the wired Internet are likely to be ported to MANETs for users who are used to these applications' functionalities regardless of the physical medium (wired or wireless). In most such 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 [38]).
- *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. For example, some applications on a node could be accessing file servers while a browser could be opening concurrent TCP connections to servers in the Internet.
- *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.

- *Landmark/hierarchical protocols*: There has been a growing interest in ad hoc routing using clustering techniques which use landmarks/clusterheads (e.g., [30]). 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 multi-cast communication in which the same packet is sent to multiple receivers.

### 3.2. Extended communication model

To include more realistic traffic patterns, we extend the previous communication model (Section 2.3) 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 per connection rate $\lambda$. 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 [12,8,19,42,22]. 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. The two packet arrival models place different stress on routing protocols that use static and adaptive timeouts. For example, a sequence of packets with short inter-arrival times can benefit from the reuse of cached routes while a sequence of packets with longer inter-arrival times can cause timeouts of valid routes and links as in AODV [33] which uses an active route timeout or DSR with a link cache [19]. Our simulation studies in Section 6 consider both the CBR traffic and the Poisson arrival model.

*Packet issue model for CBR connections*: By allowing multiple connections out of each traffic source, our new model introduces the freedom on the timing at which CBR connections out of each traffic source are initiated. In particular, a node that newly joins the network has the choices of starting all its connections (i) at random times (*random issue model*) or (ii) within a short period of time (*clustered issue model*).

Assume there are $S$ traffic sources in an $N$-node network, and each traffic source initiates $X$ connections to randomly chosen destination nodes. Thus the inter-arrival time between two consecutive packets on each connection is $\Delta T = \frac{S \cdot X}{V}$ seconds. In the random issue model, the traffic source then starts all its $X$ connections at random times within each interval of $\Delta T$ seconds. In the clustered issue model, each source initiates all its $X$ connections within a short period ($t_{\text{startup}}$) within each $\Delta T$ interval to model the behavior of a node initiating all of its network connections right after a system booting.

Fig. 3 depicts the timing of packet issuing in the random issue model (Fig. 3a) and the clustered issue model (Fig. 3b). The horizontal axis denotes the time. We denote the $j$th connection of the $i$th traffic source as $C_{i,j}$. Each connection $C_{i,j}$ is depicted by an arc whose intercept on the horizontal axis denotes the transmission of a packet on that connection.

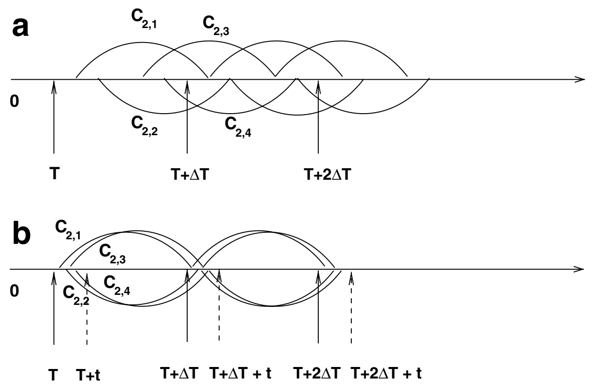


Fig. 3. Packet issue under (a) the random issue model and (b) the clustered issue model, $X = 4$.

Fig. 3 shows that in both models, within each interval of $[T + i \cdot \Delta T, T + (i+1) \cdot \Delta T]$ seconds, one packet is sent by traffic source node 2 over each of its four connections ($C_{2,1}, C_{2,2}, C_{2,3}, C_{2,4}$). Under the random issue model, a data packet for all $X$ connections may be sent at any point in each interval $[T + i \cdot \Delta T, T + (i+1) \cdot \Delta T]$. However, under the clustered issue model, one data packet for each of the $X$ connections has to be sent within each interval of $[T + i \cdot \Delta T, T + i \cdot \Delta T + \min(t_{\text{startup}}, \Delta T)]$.

*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. Our simulation studies in Section 6 consider both persistent connections as well as transient connections.

### 3.3. Summary

In summary, the different parameters that affect the performance of routing protocols can be divided into two categories: (1) parameters that are directly related to the workload and composition of the traffic pattern such as the overall traffic volume $V$, number of sources $S$, number of connections per source $X$, and packet rate per connection $\lambda$, and (2) parameters that are related to the timing of the traffic in a specific traffic workload, including packet arrival pattern, connection durations, and packet issue pattern.

Fundamentally, $V$ is most important as it determines the overall load imposed on the network. Next in importance is "how" this overall load or volume of traffic is composed. The number of sources can change the relative performance. However, even for a given number of sources, a small $X$ can impose a different workload from a large $X$ and a proportionally smaller $\lambda$ (i.e., keeping $X \times \lambda$ constant). While previous studies have only varied $V$ by changing $S$ and $\lambda$, the focus of our work is to introduce to the communication model, and study the impact of, $X$ which provides a more general description of a traffic pattern. Finally, the packet arrival pattern, packet issue pattern and connection duration are concerned with the timing aspects of packet issues that may occur in practice. However, due to the fine-grained nature of the differences caused by these last three parameters, their impact is expected to be relatively less prominent compared to that of the other parameters. Our studies in the following sections evaluate the performance of routing protocols under a variety of workloads composed from adjusting these parameters.

## 4. Ad hoc routing 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) as well as proactive (DSDV) protocols. In addition, we consider several versions of DSR from [19] which differ in their caching strategies to highlight the 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 are varied.

### 4.1. Dynamic source routing (DSR)

DSR [22] is a representative multi-hop routing protocol for ad hoc networks. 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 with each node adding its address. When the ROUTE REQUEST reaches the destination or an intermediate node with a cached route, a ROUTE REPLY is unicast back to the source. The route maintenance procedure informs the sender of any routing errors using a ROUTE ERROR packet.

DSR uses aggressive caching to reduce the frequency and propagation of route discoveries. The original design of DSR [8] uses a *path cache* which stores whole source routes. Another design, called a *link cache* proposed in [19], stores individual links of routes to build a topological graph of the network. This 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* uses the original DSR cache as in [8]. It is a generational cache in which discovered and overheard routes are maintained in two separate caches. The size of the primary cache in which discovered routes are stored is fixed at 30. However, the study in [19] showed that using a secondary cache of size 64 (*Path-Gen*-64) gives better performance than using one of size 32 (*Path-Gen*-32). We thus use *Path-Gen*-64 as the representative DSR design with 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 [19]. 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 algorithm on the topological graph.

### 4.2. 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 DSR 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 less frequently used but still valid routes for another. Moreover, each minicache is a unified cache which 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 $k$ affects the tradeoff between route availability and route freshness.

### 4.3. Destination-sequenced distance-vector (DSDV)

DSDV [32] is a proactive routing protocol based on a modified form of the Bellman-Ford [3] algorithm. Each node maintains routing table entries for all reachable destinations (tagged with a destination-specific sequence number) each having a next hop and hop count to the destination. A node exchanges routing tables (fully or partially) with its neighbors, periodically or whenever a change in topology is detected. 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 [8], as listed in Table 1.

### 4.4. Ad hoc on-demand distant vector (AODV)

AODV [33] 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. Timers are used to expire routes that have not been used recently. AODV

Table 1
Parameters used in the DSDV simulation

| | |
|---|---|
| Periodic update interval | 15 s |
| Missed updates before link declared broken | 3 |
| WST weighting factor | 7/8 |
| Route advertisement aggregation time | 1 s |
| Max. packets buffered per node per destination | 5 |

Table 2
Parameters used in the AODV simulation

| | |
|---|---|
| Active route timeout | 10 s |
| Request retries | 2 |
| Time to hold packets awaiting routes | 10 s |
| Link failure detection | MAC layer only |
| Time before broken link removal | 3 s |

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* [8]. The set of parameters used in the simulation are based on the AODV implementation for ns-2 [7] (version 2.1b8a) provided by the authors and are listed in Table 2.

## 5. Methodology

We evaluate the performance of the various routing protocols using ns-2 [7] (version 2.1b8a). The link cache implementation was ported from the ns-2.1b3 release provided by the authors of [19]. The *AODV-LL* implementation used is an updated version for ns-2.1b8a provided by the authors of the protocol. We implemented *DSR-NCache* in ns-2.1b8a. The rest of the protocols evaluated were part of the ns-2.1b8a release.

### 5.1. Mobility models

The mobility scenarios used in the simulations are generated using the "random waypoint" model [8,12]. We use the modified version of this model due to the problems associated with the original model as described in [42]. In the modified model, nodes move at a speed uniformly distributed between 1 and 19 m/s.

### 5.2. Communication model

Connections initiated by traffic sources are assumed to be constant bit rate (CBR), same as in previous studies. TCP sources would provide a conforming network load which does not allow us to evaluate all the protocols under similar traffic conditions. A packet size of 64 bytes is used. The random issue model as proposed in Section 3 is used for all experiments while the clustered issue model is evaluated separately in Section 6.5. Each traffic source selects $X$ other nodes as the destinations for the duration of the simulation. These $X$ nodes are chosen uniformly from the set of $N$ nodes in the network. The traffic volume in the network is kept constant at 60 packets/s unless otherwise stated. All the experiments assume the network has 112 nodes with 80% or 90 nodes as traffic sources.

### 5.3. Simulation parameters

We perform simulations for a network size of 112, spread out in a rectangular area of 2250 m × 450 m. The area is chosen such that the node density is at 1/9000 $m^2$, same as in previous studies [8]. A radio model with a nominal bit-rate of 2 Mbps and a nominal transmission range of 250 m is used. The effect of multiple access interference, capture, RF propagation, signal strengths, and propagation delays are modeled. The link layer used is based on the IEEE 802.11 DCF (distributed coordination function). Both DSR and AODV maintain a send buffer of 64 packets. All the results are obtained by averaging over 10 random mobility scenarios for each pause time.

An important difference between our study and previous studies is that we aim to evaluate the *steady state* behavior of the routing protocols. Specifically, the total simulation time is 1200 s, and the cutoff time of 300 s is used after which the performance statistics are collected. This cutoff ensures that all connections have been initiated and the instantaneous average speed of nodes in the network has stabilized to a steady state value. It was shown in [42] that performance metrics evaluated after the instantaneous average node speed has stabilized are such that time averages remain steady over the course of the simulation.

### 5.4. Metrics

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.

## 6. Simulation results

This section compares the performance of the protocols under the new communication model. In the following section, we use the same notations as in Section 3: Let $C$ be the total number of source–destination pairs (CBR connections) in the network, $\lambda$ be the packet rate per connection in packets/s, $S$ be the number of traffic sources, and $X$ be the number of connections/source. 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, $N$ is fixed at 112, $S$ is fixed at 90, and $V$ is constant at 60 packets/s.

We initially compare the performance of protocols with respect to parameters directly related to the composition of the traffic workload. To this end, we first compare the effect of the number of connections $X$ per source on the performance of routing protocols and also evaluate the joint impact of $X$ and mobility as well as $X$ and network size. We then study the impact of $S$ and $X$ together and finally the effect of changing $V$.

We then compare the performance of protocols under different timing parameters by changing the packet arrival pattern and connection durations as well as the impact of packet issue pattern.

### 6.1. Results: 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 [8,12]. Fig. 4 presents the results for three different pause times: a completely static network, a moderately mobile network with a pause time of 120 s, and a completely mobile network with a pause time of 0 s.

#### 6.1.1. DSR performance

Fig. 4a 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 Fig. 4b. This is expected, since a static network does not pose any challenge to their route maintenance mechanisms.

Fig. 4c and e show that in a network with moderate to high mobility, the routing overhead of *DSR-Path* increases almost linearly with $X$. Also at these mobilities, the PDR of *DSR-Path* drops from 81% to 26% (Fig. 4d and 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 a larger number of 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. Fig. 4e and 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. Fig. 5 shows that, for high mobility and $X = 1$, the Route Errors in *DSR-PathInf* are 256% more than in *DSR-Path*. Interestingly, Fig. 4f 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*, that efficiently

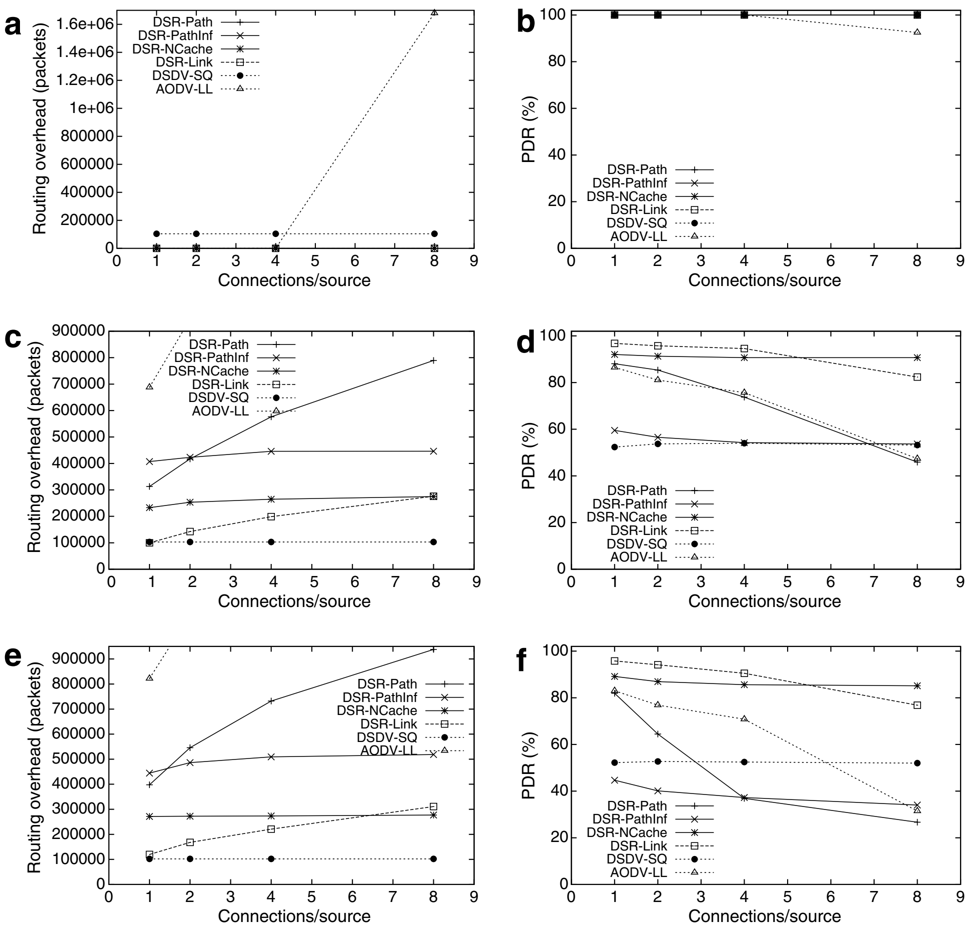


Fig. 4. Routing overhead and PDR comparison as *X* increases for 112 nodes. Routing overhead of *AODV-LL* is truncated for clarity of other protocols. (a) Routing overhead (pause time 1200 s). (b) PDR (pause time 1200 s). (c) Routing overhead (pause time 120 s). (d) PDR (pause time 120 s). (e) Routing overhead (pause time 0 s). (f) PDR (pause time 0 s).

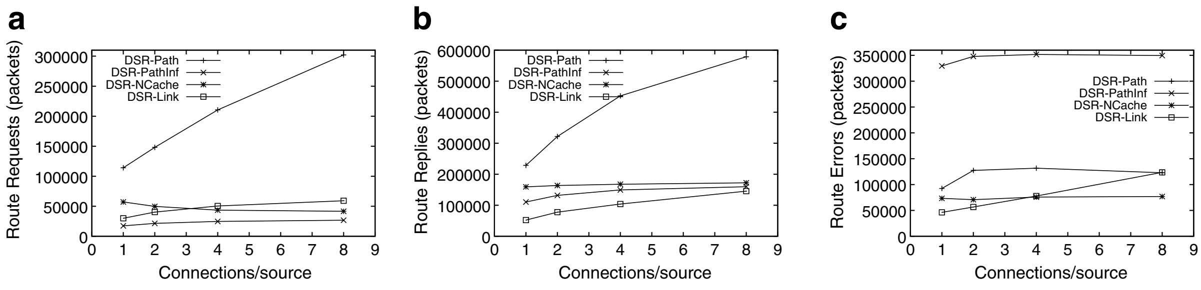


Fig. 5. Routing message breakdown of DSR versions as *X* increases (pause time 0 s). (a) Routing requests. (b) Routing replies. (c) Routing errors.

maintains fresh routes to multiple destinations. Fig. 4e 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 limit than *DSR-Path*. As a result, it incurs far fewer Route Errors than *DSR-PathInf* and *DSR-Path*.

Fig. 5 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* is largely constant with increasing *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 previous 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. Second, this diverse 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 for sending packets by a source node. Remote reuse also increases with increasing *X*.

Another design of DSR is based on a graph-based cache structure as in [19]. This graph-based cache has an inherent advantage in that route competition does not exist. Though this fact was not mentioned in [19], 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, Fig. 4e and 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 Fig. 5.

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 interarrival time of the requests. If the interarrival 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 assigned to a link added in the future, which is calculated as the minimum of the stability of the two 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 Fig. 5. Also, as *X* increases, the number of unique links added to the link cache increases as the node discovers routes to unique destinations. Using these increased link information in the graph, *DSR-Link* may construct longer routes to a given destination. These long routes may be unreliable since they have not been explicitly discovered. This explains the increase in number of Route Errors as *X* increases. This causes the overall increase in routing overhead and reduction in the PDR of *DSR-Link* as *X* increases.

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 increasing *X*. Although *DSR-Link* has a lower routing overhead and comparable PDR to *DSR-NCache* for small values of *X*, its performance degrades as *X* increases.

#### 6.1.2. 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 (15 s) and the size of the network. As the route aggregation time used is 1 s, every node in the network on average triggers an update every second. This results in a constant routing overhead of simulation time (900 s) $*$ network size (112) = 100,800 packets. For moderate to high mobility, Fig. 4c and e show that this constant routing overhead is in fact less than all the versions of DSR for all $X$. However, the low overhead advantage of *DSDV-SQ* is negated by the 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*.

#### 6.1.3. AODV performance

Fig. 4 also shows the effect of increasing $X$ on the performance of *AODV-LL*. However, the curves for routing overhead are truncated for clarity of other protocols. First, in the no mobility scenario, Fig. 4b and 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 interarrival 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 interarrival time of traffic for a particular destination is $\frac{90*8}{60} = 12s$, which is greater than the active timeout (10 s) used in the simulations. As a result, a static network yielded a routing overhead of 1,680,418 packets. This high routing overhead is further responsible for the reduction in PDR. The same scenario with a timeout of 20 s resulted in an 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 interarrival 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 Fig. 4c and d 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 (Fig. 4e and f). Complete curves for *AODV-LL* for a pause time of 0 s are shown in Fig. 1a and b, in which the three protocols correspond to *DSR-Path*, *AODV-LL*, and *DSDV-SQ*, respectively. In Fig. 1, for $X = 1$ and 2, the routing overhead of *AODV-LL* is higher compared to *DSR-Path* and *DSDV-SQ*, but its PDR remains at nearly 80%. As $X$ increases, the routing overhead of *AODV-LL* increases almost linearly to about two million packets. This increase of overhead is accompanied by a corresponding decrease in PDR to about 31% at $X = 8$.

The higher routing overhead of *AODV-LL* at medium to high mobility is best explained by comparing it to *DSR-Path*. Table 3 shows that for $X = 2$ and with a pause time of 0 s, *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, the comparable numbers of 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 promiscuous overhearing, and as a result 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

Table 3
Routing overhead comparison between DSR and AODV simulations for pause time 0 s

| | Route discovery | | Route request | |
|---|---|---|---|---|
| | $X = 2$ | $X = 8$ | $X = 2$ | $X = 8$ |
| DSR-Path | 10,348 | 24,395 | 148,046 | 302,026 |
| DSR-NCache | 7844 | 9078 | 49,732 | 41,604 |
| AODV-LL | 10,953 | 26,345 | 924,133 | 1,759,796 |
| AODV-LL (20 s) | 10,953 | 17,321 | 924,133 | 1,328,189 |

AODV-LL (20 s) refers to AODV-LL with an active timeout of 20 s.

connection cannot be used to deliver packets for another connection, and because of the mismatch between the active timeout (10 s) and the packet interarrival time (12 s). To isolate the effect of this mismatch, we ran *AODV-LL* with an active timeout of 20 s. Table 3 shows that with a pause time of 0 s, 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 later in Fig. 6).

Table 3 also shows that as $X$ is increased from 2 to 8, while *AODV-LL* and *DSR-Path* have more than a twofold 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 and achieves a higher PDR than *AODV-LL* and *DSR-Path*.

The optimizations proposed for *AODV-LL* such as maintaining multiple alternate routes to each destination (*AODV-BR* [24]) or accumulating paths from Route Requests and Route Replys (*AODV-PA* [18]) can improve the local and remote reuse for *AODV-LL*. Thus, the performance of *AODV-LL* under the new communication model exposes design choices that are beneficial to the operation of the protocol.

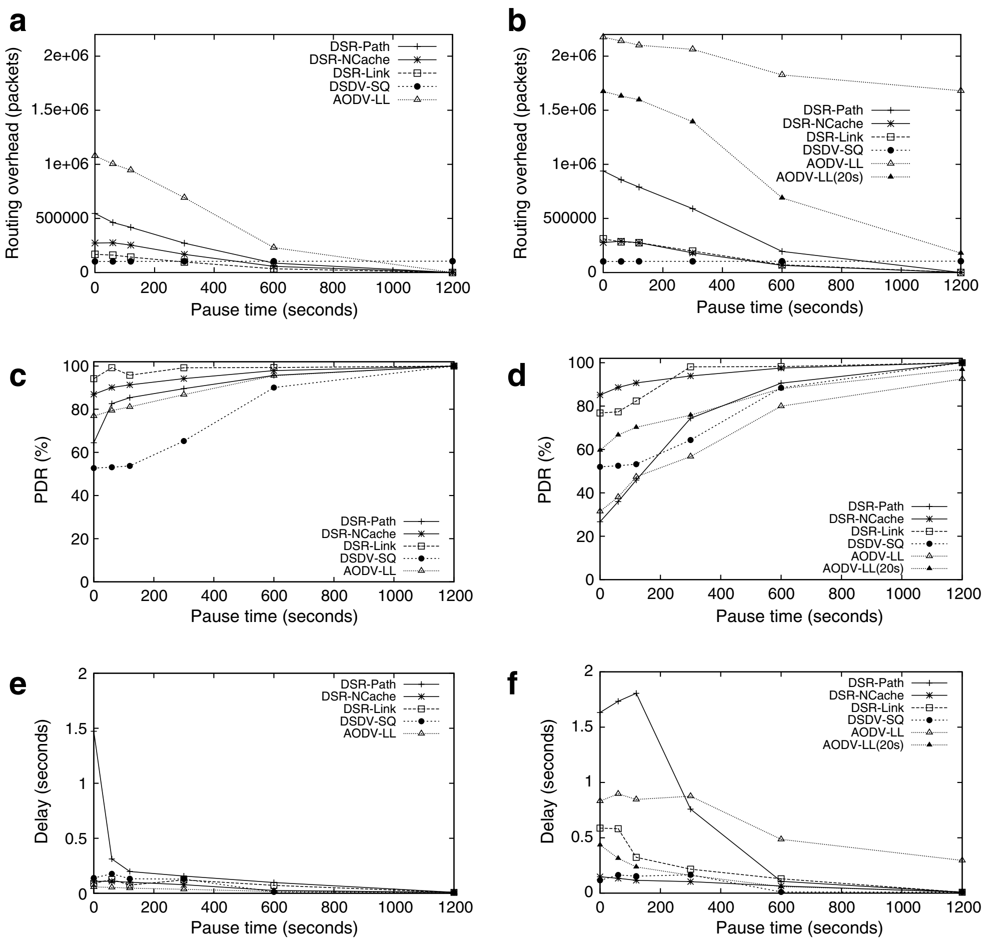


Fig. 6. Routing overhead, PDR and delay for varying mobility and $X$. Routing overhead of *AODV-LL* is truncated for clarity of other protocols. (a) Routing overhead ($X = 2$). (b) Routing overhead ($X = 8$). (c) PDR ($X = 2$). (d) PDR ($X = 8$). (e) Delay ($X = 2$). (f) Delay ($X = 8$).

### 6.1.4. Results: Effects of mobility together with the number of connections

In this section, we revisit the simulation results of Section 6.1 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 s to a completely static network.

Fig. 6 compares the routing overhead, PDR, and delay 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 [8] which used an average value of 1.5 for *X*. 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 [12], which used a network of 100 nodes.

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 interarrival 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 20 s for $X = 8$ and the results are shown in Fig. 6b, d, and f. The results show that an optimal active timeout can drastically improve routing overhead, PDR, and latency.

*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* has lower routing overhead and lower PDRs than *DSR-Path* and *AODV-LL* for all mobilities. A significant observation is that when $X = 8$, *DSDV-SQ* incurs lower routing overhead and higher‘ PDRs than *DSR-Path* and *AODV-LL* (with active timeout 10 s) at high mobilities.

Unlike *DSR-Path* and *AODV-LL*, *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 when *X* is increased from 2 to 8. This is due to the reliance of its adaptive timeout on the traffic pattern as explained in Section 6.1.

*DSR-NCache* exhibits superior delay performance regardless of *X*, significantly reducing the latency in highly mobile networks with complex traffic patterns. 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 [12], 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 across all pause times for similar speeds as those used in our evaluation.

In summary, the effects of mobility on different routing protocols for $X = 8$ are significantly different from those for $X = 2$. In general, mobility imposes stress on routing protocols due to route breakage and subsequent rediscovery cost, increased data packet retransmission due to the use of stale links, etc. On top of this, if the routing protocol does not deal well with different traffic patterns (such as increasing *X*), there is an additional burden which causes performance to degrade significantly when both mobility and traffic patterns are challenging (e.g., *DSR-Path* and *AODV-LL*). Protocols that deal well with changes in *X* such as *DSR-NCache* only have to deal with mobility rather than both traffic patterns and mobility and thus fare much better. Finally, protocols whose routing overhead is independent of mobility such as *DSDV-SQ* are not at all impacted and thus can be relatively more useful in highly mobile scenarios. Overall, these results reaffirm the importance of using a general communication model to evaluate different routing protocols.

### 6.1.5. Results: Effects of network size together with the number of connections

In this section, we study the effect of varying the network size on routing protocols with a general traffic pattern workload.

The results in Fig. 7 show that the impact of *X* exists both for a small 50-node network and a large 112-node network. For example, as *X* increases from 2 to 8, the routing overhead of *DSR-Path* increases for both network sizes.

Thus, the impact of using a general traffic patterns exists for varying network sizes. However, the magnitude of the impact increases with the network size. For example, as the network size is increased, while the PDR of *DSR-Path* drops from 97% to 65% for $X = 2$, it drops from 95% to 28% for $X = 8$. Thus, considering a general traffic pattern becomes increasingly important as the network size increases. This is because the penalty paid for bad route caching decisions is higher in larger networks due to the high costs of route discovery.

### 6.2. Results: Effects of traffic concentration (keeping $C = S \cdot X$ constant)

In this section, we evaluate the performance of the protocols by increasing the traffic sources $S$ in the network and simultaneously decreasing the $X$ value to keep the total number of connections in the network $C$ constant. While similar results are observed for medium and high mobility scenarios, we only show the results for high mobility scenarios due to space limitation.

We vary the $S$ as 10, 20, 40, 80 with the corresponding $X$ values as 8, 4, 2, 1. In each case $C = S \cdot X$ remains constant at 80 connections. Volume ($V$) is also kept constant at 60 packets/s, and the interarrival time $\Delta T$ remains constant (1.33 s) for all the connections in all the scenarios. We define *traffic concentration* as the spatial distribution of traffic sources in the network. Thus when $S = 80$ and $X = 1$, we say that the *traffic concentration* is low and the connections are evenly distributed. On the other hand, when $S = 10$ and $X = 8$, we say that the *traffic concentration* is high as all the connections in the network originate

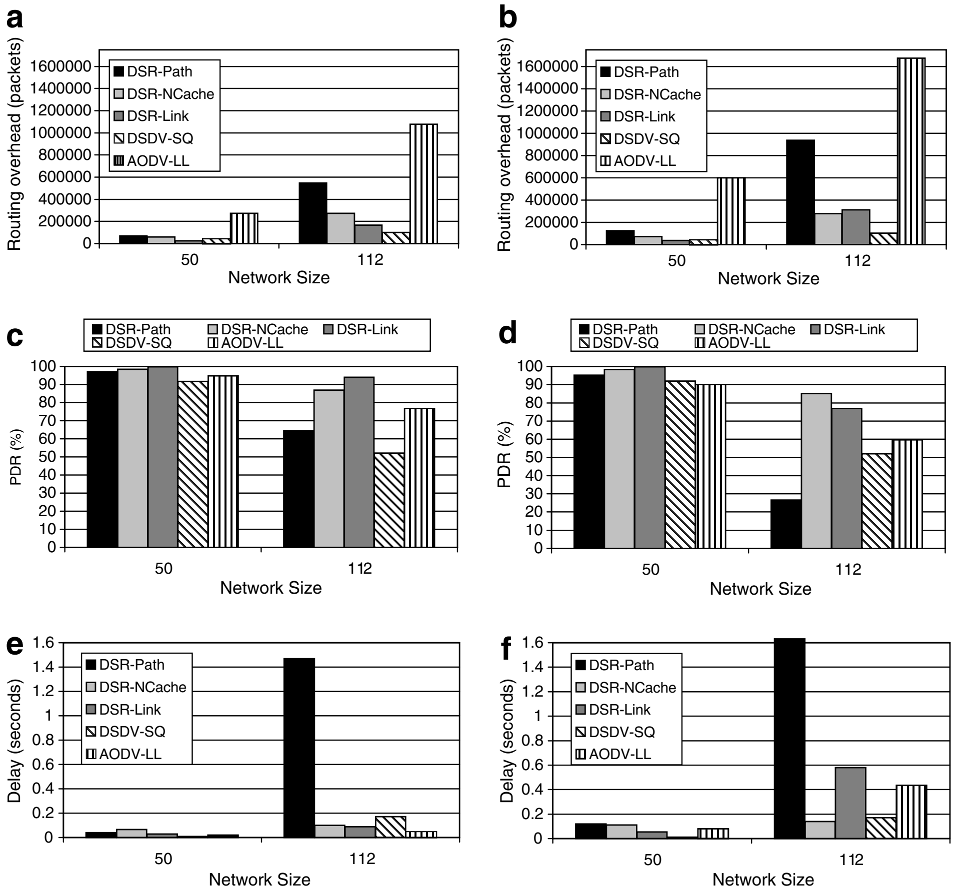


Fig. 7. Performance comparison for varying network size. Both $X = 2$ and $X = 8$ are considered. Pause time is 0 s. (a) Routing overhead ($X = 2$). (b) Routing overhead ($X = 8$). (c) PDR ($X = 2$). (d) PDR ($X = 8$). (e) Delay ($X = 2$). (f) Delay ($X = 8$).

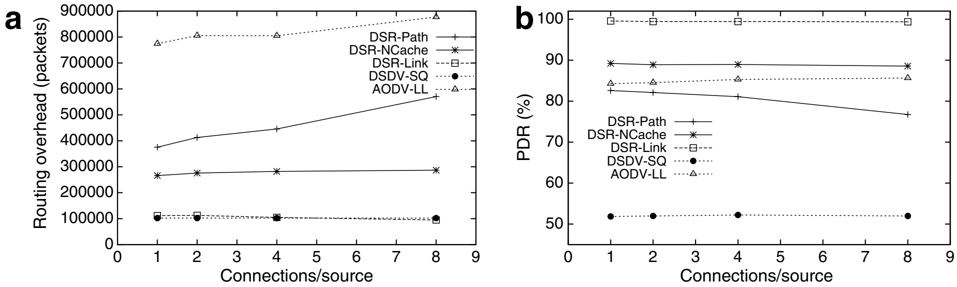


Fig. 8. Routing overhead and PDR comparison as $X$ increases keeping $C$ constant for 112 nodes. Routing overhead of *AODV-LL* is truncated for clarity of other protocols. (a) Routing overhead (pause time 0 s). (b) PDR (pause time 0 s).

from 10 out of the 112 nodes. We expect the routing overhead and PDR of all the protocols to be largely unaffected by the traffic concentration as the number of connections remains a constant. Fig. 8 depicts the routing overhead and PDR of the protocols evaluated as the traffic concentration is increased.

First, the routing overhead and PDR of *DSDV-SQ* of *DSR-NCache* remains constant irrespective of the traffic concentration similar to the behavior observed in Section 6.1.

Second, the routing overhead of *DSR-Path* increases as the traffic concentration increases with a corresponding decreased in PDR. As described in Section 6.1, this degradation in the performance of *DSR-Path* is due to the route competition in the cache that occurs as the value of $X$ increases. Thus, the routing overhead of *DSR-Path* increases with increased traffic concentration for a constant number of connections.

Third, for *DSR-Link*, the routing overhead and PDR are independent of traffic concentration. This is intuitive since the number of connections and the interarrival time of packets over each connection ($\Delta T$) are identical across all scenarios which results in similar feedback to the timeout mechanisms in *DSR-Link*.

Fourth, similar to *DSR-Link* and *DSR-NCache*, *AODV-LL* has a fairly constant routing overhead and PDR for a given mobility at all levels of traffic concentration. As noted in Section 6.1, in *AODV-LL* there is no reuse between connections as each connection needs an explicit path set up. Thus the routing overhead of *AODV-LL* is directly affected by the number of connections in the network. In this scenario, as the numbers of connections are the same, *AODV-LL* performs the same amount of work to set up these connections and consequently performs better than *DSR-Path*.

### 6.3. Results: Effects of traffic volume

In this section, we revisit the impact of varying traffic volume on the performance of various protocols under our communication model, i.e., with varying number of connections per source node. We do not depict these results due to lack of space but discuss the main findings.

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 overall network load imposed by data packets remains similar.[1] Increased traffic volume for a fixed $X$ affects the performance of DSR more adversely than AODV because DSR has been shown to have a high MAC load [12]. This is because the routing overhead of DSR is primarily composed of unicast packets which incur the overhead 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 [12]. However, as $X$ increases, the large number of Route Requests in AODV outweigh the less link stress from fewer unicasts. This implies that an isolated argument about the impact of MAC load on the routing performance may not be applicable for high values of $X$.

[1] We note the medium access interference caused by the same network load may still vary because it also depends on other factors such as the traffic distribution (sources and destinations) and the physical layer characteristics at the time of data transmissions.

Interestingly, in simulations run for high values of $X$ we found that as the traffic volume is increased, 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.

### 6.4. Results: Effects of transient connections

In this section, we examine the effect of applications with transient connections on the performance of routing protocols. We define such an application to be one that at any point in time is equally likely to choose any destination in the network to initiate a connection with and these connections are short lived. Examples of such applications are daemons that interact with network services as well as applications built on top of structured p2p systems such as CAN, Chord Pastry, and Tapestry [36,39,38,44]. Note that such applications essentially generate a traffic pattern with large values of $X$ though these $X$ connections are not simultaneously maintained. As an example, we consider a generic object storage/retrieval application (e.g., resource discovery). At each node, we assume a Poisson arrival (using an Exponential traffic generator) of requests to store and retrieve information with each request directed to a randomly chosen node. The net arrival rate of requests from each node was matched to the traffic volumes used for Section 6.1.

Fig. 9 compares the routing overhead, PDR, and delay of each of the protocols considered as the mobility is varied. We omit *DSR-PathInf* from this comparison due to inherent weaknesses in its design as discussed in Section 6.1.

Although transient connectivity shares the similar effect of increasing $X$ for each node as in Section 6.1.4 (e.g., from 2 to 8), there are two key differences between these two traffic patterns. In Section 6.1.4, $\Delta T$ is constant and each node sends a packet over each of its $X$ connections in every interval $\Delta T$ whereas here $\Delta T$ is exponentially distributed and the number of packets sent by a node in every interval $\Delta T$ is random. Additionally, in Section 6.1.4, connections once initiated last for the duration of the simulation whereas here each connection is transient (short lived). These differences in the traffic pattern have the following key implications.

First, *AODV-LL* has a higher routing overhead and lower PDR than what was observed in Section 6.1.4. Since the total unique connections initiated in the network are potentially large, the routing overhead of *AODV-LL* grows faster since each unique connection requires a route to be set up. Additionally, the interval between the repeated selection of a particular node (e.g., to store or retrieve content in the storage application) is random and typically of the order of a few seconds. Thus routes discovered by *AODV-LL* for one connection time out frequently before they can be used again.

Second, compared to in Section 6.1.4, *DSR-Link* has a higher drop in PDR as the mobility increases and approaches that of *DSDV-SQ* at 0 pause time. The higher routing overhead of *DSR-Link* as compared to *DSR-NCache* is due to the timeout associated with its links as discussed in Section 6.1. When connections are transient, the positive feedback to link timeouts reduce further, resulting in frequent timeouts for valid links. Compared to *DSR-NCache*, the drop in PDR of *DSR-Link* is due to the increased number of Route Errors.

An additional observation is that now *DSDV-SQ* outperforms both *DSR-Path* and *AODV-LL* at high mobilities by delivering more packets with lesser routing overhead.

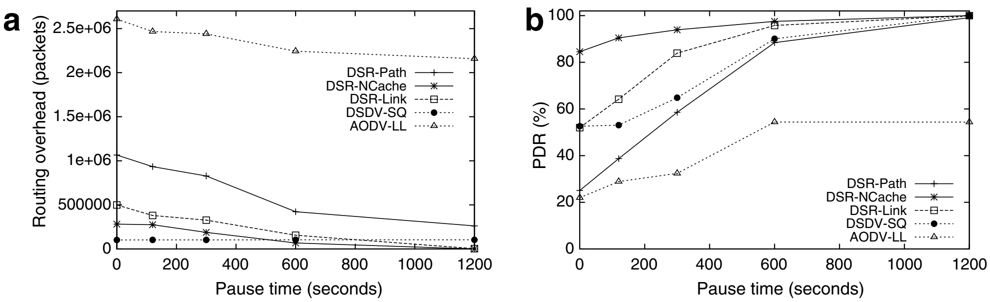


Fig. 9. Routing overhead and PDR varying mobility and X for p2p traffic pattern in 112 nodes. (a) Routing overhead. (b) PDR.

In summary, application with transient connections affect protocols with static or adaptive timeouts. The interarrival time between packets over the same connection is an important factor that affects performance. Large interarrival times with short flows reduce the caching efficiency and cause timeouts of valid routes and links.

### 6.5. Results: Effects of clustered issue

In this section, we study the effect of clustered issue on the performance of protocols by comparing the performance of each protocol under the clustered issue model with that under the random issue model. The clustered issue model may better reflect the behavior of a node for certain applications, for example, upon starting up, a node in a deployed ad hoc network typically initiates all its network connections to network services within a short span of time. Intuitively, the resulting clustered packet issuing would cause increased network congestion and consequently multi-access interference which would then degrade the routing performance.

We compare the performance of individual protocols under the two models as we keep the traffic volume $V$ constant and vary the total number of connections $C$ by varying $X$ from 1 to 8. In the clustered issue model, $t_{\text{startup}}$ is chosen to be 1.33 s. Traffic patterns were generated such that the $X$ connections initiated by each node are identical in both models, and only the time of initiation are different. Keeping the mobility and all other parameters of the communication model identical under the two issue models enables us to isolate the effects of clustered issue and random issue.

We do not depict performance results for this section because we found that the performance of all the protocols is largely similar for both models independent of the value of $X$. Our findings are summarized as follows: When connections are initiated together within a short span, Route Requests for these connections are also initiated within that short span. For all DSR based protocols, these simultaneous discoveries can result in the reuse of one discovered route for another connection. In contrast, in the random issue model, since the time at which different connections send packets are independent of each other, and thus spaced out from each other, Route Replys received for the $i$th connection are more likely to become invalid by the time a route needs to be discovered for the $(i+1)$th connection. The performance comparison suggests that this better reuse under the clustered issue model balances out the increased congestion due to clustered packet issuing.

*AODV-LL* cannot reuse the routes learned for one destination to deliver a packet to another destination. However, since *AODV-LL* has an on-demand nature, a clustered issue of data packets within $[T, T+t_{\text{startup}}]$ does not imply that those data packets are actually sent out within that interval. Many data packets could be buffered awaiting the discovery of routes, resulting in some fraction of packets being sent out in the interval $[T+t_{\text{startup}}, T+t_{\text{startup}}+\Delta T]$. This balances out the increased congestion due to clustered issue, resulting in similar performance in *AODV-LL* under the two models. This effect also occurs for DSR based protocols although at a smaller scale due to the use of aggressive caching and consequently a lower probability of buffering a packet pending the discovery of a route.

*DSDV-SQ* showed a slightly lower PDR for high values of $X$ in the clustered issue model as compared to the random issue model. The congestions due to control overhead for *DSDV-SQ* under the two traffic patterns are identical, since the periodical control packets are independent of the data packets. The increased interference due to clustering of data packets cannot be offset by reuse (as in DSR) or spreading out the transmission time of data packets (as in AODV) which causes the PDR to drop slightly.

In summary, although clustered packet issuing may generate more congestion and consequently higher multi-access interference than random packet issuing, our simulation results show that the performance of all protocols are largely similar under the two models. This suggests that although the clustered issue model may better reflect the traffic patterns generated by many nodes, it is sufficient to just use the random issue model in the study of ad hoc routing protocols.

## 7. Conclusions

As mobile ad hoc network (MANET) systems research has matured and several testbeds have been built to study MANETs, research has focused on developing new MANET applications which can present much more complex traffic patterns as a workload to routing protocols. In this paper, we first pointed out that the communication model typically used to evaluate MANET routing protocols 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 new applications developed for MANETs. We 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. We list the main findings and conclusions of this study in Table 4 which highlight the need for protocol designers in MANETs to take into account general traffic patterns.

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 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 larger interarrival 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.

Table 4
Summary of findings

| Under restricted traffic pattern | Under general traffic pattern |
|---|---|
| At high mobility, DSDV delivers fewer packets than DSR-Path and AODV | Previous conclusion valid for small values of $X$. At high $X$, DSDV can outperform DSR-Path and AODV |
| DSR-Path is the best path-cache-based DSR version | Poor cache organization was not exposed in using simplistic traffic pattern. NCache structure is better than capacity-limited path cache structure |
| Adaptive timeouts in DSR-Link reduces route errors | Timeouts adapted to traffic pattern can evict valid routes |
| AODV performs better than DSR at high mobility and large volume | Previous conclusion valid for small $X$. At large $X$, AODV may not reach volumes at which it outperforms DSR |
| AODV is more scalable than DSR | Previous conclusion valid for small $X$. At large $X$, the gain due to hop-by-hop routing can be offset by lack of extensive caching |

The table first lists the findings of previous studies under the previously used simple restricted traffic pattern along with the corresponding new findings under a general traffic pattern.

## Acknowledgment

This work was supported in part by NSF grant ANI-0338856.

## References

[1] A Community Resource for Archiving Wireless Data at Dartmouth, http://crawdad.cs.dartmouth.edu.
[2] L. Barkhuus, M. Chalmers, M. Hall, P. Tennent, M. Bell, S. Sherwood, B. Brown, Picking pockets on the lawn: the development of tactics and strategies in a mobile game, in: Proc. of UbiComp, 2005.
[3] D. Bertsekas, R. Gallager, Data Networks, Prentice Hall, 1987, pp. 297–333.
[4] M. Bisignano, A. Calvagna, G.D. Modica, G. Tomarchio, Experience: a jxta middleware for mobile ad-hoc networks, in: Proc. of P2P, 2003.
[5] D. Bottazzi, A. Corradi, R. Montanari, A context-aware group management middleware to support resource sharing in manet environments, in: Proc. of MDM, 2005.
[6] M. Boulkenafed, V. Issarny, Adhocfs: sharing files in wlans, in: Proc. of IEEE NCA, 2003.
[7] L. Breslau et al., Advances in network simulation, IEEE Computer 33 (5) (2000) 59–67.

[8] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proc. of ACM MobiCom, October 1998.
[9] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Communications and Mobile Computing 2 (5) (2002) 483–502.
[10] S.R. Das, R. Castaneda, J. Yan, R. Sengupta, Comparative performance evaluation of routing protocols for mobile ad hoc networks, in: Proc. of ICCCN, October 1998.
[11] S.R. Das, R. Castaeda, J. Yan, Simulation-based performance evaluation of routing protocols for mobile ad hoc networks, Mobile Networks and Applications 5 (3) (2000) 179–189.
[12] S.R. Das, C.E. Perkins, E.M. Royer, Performance comparison of two on-demand routing protocols for ad hoc networks, in: Proc. of IEEE INFOCOM, March 2000.
[13] A. Deshpande, A. Hegde, A. Shetty, Cvs: Collaborative voting system to detect routing misbehavior in wireless ad hoc networks, in: Proc. of ICENCO, 2004.
[14] M.D. Dikaiakos, S. Iqbal, T. Nadeem, L. Iftode, Vitp: a vehicular information transfer protocol, in: Proc. of ACM VANET, 2005.
[15] J.M. DiMicco, Mobile ad-hoc voting, in: Proc. of CHI Workshop on Mobile Ad-Hoc Collaboration, 2002.
[16] A. Fuller, P. McFarlane, D. Saffioti, Distributed, collaborative learning environments using ad hoc networks, in: Proc. of IEEE ICALT, 2004.
[17] P. Gupta, P.R. Kumar, The capacity of wireless networks, IEEE Transactions on Information Theory 46 (2) (2000) 388–404.
[18] S. Gwalani, E.M. Belding-Royer, C.E. Perkins, AODV-PA: AODV with path accumulation, in: Proc. of Next Generation Internet Symposium, May 2003.
[19] Y.-C. Hu, D.B. Johnson, Caching strategies in on-demand routing protocols for wireless ad hoc networks, in: Proc. of ACM MobiCom, August 2000.
[20] A. Jardosh, E.M. Belding-Royer, K.C. Almeroth, S. Suri, Towards realistic mobility models for mobile ad hoc networks, in: Proc. of ACM MobiCom, September 2003.
[21] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, M. Degermark, Scenario-based performance analysis of routing protocols for mobile ad-hoc networks, in: Proc. of ACM MobiCom, August 1999.
[22] D.B. Johnson, D.A. Maltz, Dynamic Source Routing in Ad Hoc Wireless Networks, Kluwer Academic, 1996.
[23] M.-O. Killijian, D. Powell, M. Bantre, P. Couderc, Y. Roudier, Collaborative backup for dependable mobile applications, in: Proc. of ACM MPAC, 2004.
[24] S.-J. Lee, M. Gerla, AODV-BR: backup routing in ad hoc networks, in: Proc. of IEEE WCNC, September 2000.
[25] Z. Li, B. Li, D. Xu, X. Zhou, iflow: middleware-assisted rendezvous-based information access for mobile ad hoc applications, in: Proc. of USENIX MobiSys, 2003.
[26] H. Liu, T. Roeder, K. Walsh, R. Barr, E.G. Sirer, Design and implementation of a single system image operating system for ad hoc networks, in: Proc. of USENIX MobiSys, 2005.
[27] J. Liu, D. Sacchetti, F. Sailhan, V. Issarny, Group management for mobile ad hoc networks: design, implementation and experiment, in: Proc. of MDM, 2005.
[28] M. Musolesi, C. Mascolo, S. Hailes, Adapting asynchronous messaging middleware to ad hoc networking, in: Proc. of ACM MPAC, 2004.
[29] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proc. of IEEE INFOCOM, April 1997.
[30] G. Pei, M. Gerla, X. Hong, LANMAR: landmark routing for large scale wireless ad hoc networks with group mobility, in: Proc. of IEEE/ACM Mobihoc, August 2000.
[31] S. Penz, Slp-based service management for dynamic ad-hoc networks, in: Proc. of ACM MPAC, 2005.
[32] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proc. of ACM SIGCOMM, August 1994.
[33] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: Proc. of IEEE WMCSA, February 1999.
[34] H. Pucha, S.M. Das, Y.C. Hu, The performance impact of traffic patterns on routing protocols in mobile ad hoc networks, in: Proc. of ACM MSWiM, October 2004.
[35] A. Raniwala, G. Zhang, J. Zheng, A. Sridhar, T. Chiueh, Wshare: an instant secure collaboration workspace over ad hoc wireless lan, IJWMC: Special Issue on Group Communications in Ad hoc Networks (3) (2005).
[36] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Schenker, A scalable content-addressable network, in: Proc. of ACM SIGCOMM, August 2001.
[37] T. Repantis, V. Kalogeraki, Data dissemination in mobile peer-to-peer networks, in: Proc. of MDM, 2005.
[38] A. Rowstron, P. Druschel, Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems, in: Proc. of Middleware, November 2001.
[39] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, Chord: a scalable peer-to-peer lookup service for internet applications, in: Proc. of ACM SIGCOMM, August 2001.
[40] P. Tennent, M. Hall, B. Brown, M. Chalmersa, S. Sherwood, Three applications for mobile epidemic algorithms, in: Proc. of Mobile HCI, 2005.
[41] E. Vollset, D.B. Ingham, P.D. Ezhilchelvan, Jms on mobile ad hoc networks, in: Proc. of PWC, 2003.
[42] J. Yoon, M. Liu, B. Noble, Random waypoint considered harmful, in: Proc. of IEEE INFOCOM, April 2003.
[43] J. Yoon, M. Liu, B. Noble, Sound mobility models, in: Proc. of ACM MobiCom, September 2003.
[44] B.Y. Zhao, J.D. Kubiatowicz, A.D. Joseph, Tapestry: an infrastructure for fault-resilient wide-area location and routing. Technical report UCB//CSD-01-1141, UC Berkeley, April 2001.

**Himabindu Pucha** is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously she received a MSEE degree from Purdue University, USA and a B.Engg. degree from the University of Bombay, India. Her research interests include Internet routing and overlay networks, peer-to-peer systems and applications, and mobile computing.

**Saumitra M. Das** is currently a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, USA. Previously, he received a M.S. degree from Carnegie Mellon University, USA and a B.Engg. degree from the University of Bombay, India. His research interests include cross-layer system design for multi-hop wireless networks, scalable routing strategies in wireless ad hoc networks, and mobile robotics.

**Y. Charlie Hu** is an Assistant Professor of Electrical and Computer Engineering and Computer Science at Purdue University. He received his M.S. and M.Phil. degrees from Yale University in 1992 and his Ph.D. degree in Computer Science from Harvard University in 1997. From 1997 to 2001, he was a research scientist at Rice University. His research interests include operating systems, distributed systems, networking, and parallel computing. He has published over 90 papers in these areas. He received the NSF CAREER Award in 2003. He served as a TPC vice chair for IEEE ICDCS 2007 and the 2004 International Conference on Parallel Processing, and a co-founder and TPC co-chair for the International Workshop on Mobile Peer-to-Peer Computing. He is a member of USENIX, ACM, and IEEE.