Mobile Networks and Applications 6, 339–349, 2001

Performance of Multipath Routing for On-Demand Protocols in Mobile Ad Hoc Networks *

ASIS NASIPURI and ROBERT CASTAÑEDA
Department of Electrical and Computer Engineering, University of North Carolina, Charlotte, NC 28223-0001, USA

SAMIR R. DAS **
Department of Electrical & Computer Engineering and Computer Science, University of Cincinnati, Cincinnati, OH 45221-0030, USA

Abstract. Mobile ad hoc networks are characterized by multi-hop wireless links, absence of any cellular infrastructure, and frequent host mobility. Design of efficient routing protocols in such networks is a challenging issue. A class of routing protocols called *on-demand* protocols has recently found attention because of their low routing overhead. The on-demand protocols depend on query floods to discover routes whenever a new route is needed. Such floods take up a substantial portion of network bandwidth. We focus on a particular on-demand protocol, called *Dynamic Source Routing*, and show how intelligent use of multipath techniques can reduce the frequency of query floods. We develop an analytic modeling framework to determine the relative frequency of query floods for various techniques. Our modeling effort shows that while multipath routing is significantly better than single path routing, the performance advantage is small beyond a few paths and for long path lengths. It also shows that providing all intermediate nodes in the primary (shortest) route with alternative paths has a significantly better performance than providing only the source with alternate paths. We perform some simulation experiments which validate these findings.

Keywords: ad hoc networks, on-demand routing, route discovery

1. Introduction

A mobile, *ad hoc* network is an autonomous system of mobile hosts connected by wireless links. There is no static infrastructure such as base station. If two hosts are not within radio range, all message communication between them must pass through one or more intermediate hosts that double as routers. The hosts are free to move around randomly, thus changing the network topology dynamically. Thus routing protocols must be adaptive and able to maintain routes in spite of the changing network connectivity. Such networks are very useful in military and other tactical applications such as emergency rescue or exploration missions, where cellular infrastructure is unavailable or unreliable. Commercial applications are also likely where there is a need for ubiquitous communication services without the presence or use of a fixed infrastructure. Examples include home-area wireless networking [6], on-the-fly conferencing applications, networking intelligent devices or sensors, communication between mobile robots, etc.

Design of efficient routing protocols is the central challenge in such dynamic wireless networks. Much work has been done in this area starting from the seventies, when the U.S. Defense Research Agency, DARPA, supported the PRNET (Packet radio Network) [8] and SURAN (Survivable Adaptive Networks) [14] projects. They supported automatic route set up and maintenance in a packet radio network with moderate mobility. Interest in such networks has recently grown due to the common availability of wireless communication devices that can connect laptops and palmtops and operate in license free radio frequency bands (such as the Industrial-Scientific-Military or ISM band in the U.S.). In an interest to run internetworking protocols on ad hoc networks, a working group for Mobile, Ad hoc Networking (MANET) [10] has been formed within the Internet Engineering Task Force (IETF), whose charter includes developing a routing framework for running IP based protocols in ad-hoc networks. Several new routing protocols have been proposed in connection with the MANET working group efforts [10]. Of particular interest is the new class of *on-demand, source-initiated* protocols, that set up and maintain routes from a source to a destination on an "as needed" basis [7,12]. This approach is in sharp contrast with the traditional *shortest path*-based protocols (e.g., link-state and distance vector [9]) that have been used successfully for a long time in dynamic, wireline networks, including the Internet.

1.1. On-demand protocols and multipathing

The motivation behind the on-demand protocols is that the "routing overhead" (typically measured in terms of the number of routing packets transmitted, as opposed to data packets) is typically lower than the shortest path protocols, as only the actively used routes are maintained. However, as some recent performance evaluation work has shown [5], the routing overhead still approaches to that of the shortest path

* This work is partially supported by AFOSR grant no. F49260-96-1-0472 and the following NSF grants, CDA-9633299, ACI-9733836 and ANI-9973147.
** Corresponding author.

protocols, if a moderate to large number of routes needs to be actively maintained (when, for example, there is a moderate to large number of active peer-to-peer conversations). This is because the on-demand protocols discover routes via a *flooding* technique, where the source (or any node seeking the route) floods the entire network with a query packet in search of a route to the destination. Flooding is also necessary for route maintenance activities, when a new route is needed, as the old one breaks because of node mobility. Flooding takes up a substantial amount of network bandwidth, which is at a premium in wireless networks. Efficient control of frequent network-wide flooding is thus important for the efficient performance of on-demand protocols. Some of our prior work was directed to limit the flood within a small region of the network [4] to reduce its impact on the network performance. In this paper, we focus on reducing the frequency of floods, by exploring multiple possible routes from a single flooded query. The goal is to provide enough redundancy at a low cost.

The idea of multipath routing is not new. It always has been a favorable alternative both for circuit switched and packet switched networks, as it provides an easy mechanism to distribute traffic and balance the network load, as well as provide fault tolerance. See, for example, [16] for some prior work on multipath routing on packet switched networks. Ad hoc network community also investigated multipath techniques, *albeit* less vigorously. The Temporally Ordered Routing Algorithm or TORA [11] provides multiple alternate paths by maintaining a "destination-oriented" directed acyclic graph (DAG) from the source. However, TORA does not have any easy mechanism to evaluate the "quality" of these multiple routes. Because of the nature of the 'protocol, it is hard to determine which route is the shortest. Also, TORA did not perform well in comparison to other on-demand protocols in some of the recent simulation studies [2,5], ostensibly because the overhead incurred in maintaining multiple routes overwhelmed their performance benefits. The Dynamic Source Routing (DSR) protocol [3,7] also has an option of maintaining multiple routes, so that an alternate route can be used upon failure of the primary one. In DSR, the quality of routes (i.e., hop-wise lengths) can be easily evaluated and the best (i.e., the shortest) one can be used. But in DSR [3] too many routes are maintained in a trivial manner, without any regard to their ultimate usefulness. In any case, performance benefits of multiple paths have never been evaluated.

In this paper, we propose two multipath techniques for the DSR protocol which use disjoint paths. We develop an analytical modeling framework to evaluate the performance advantage of such multipath techniques. The modeling framework is also useful for performance evaluation of on-demand routing protocols regardless of the use of multiple paths. In our knowledge, this is the first attempt to analytically model on-demand routing protocols in ad hoc networks. Prior evaluations are based solely on simulation studies [2,5]. To confirm the validity of our analytical model, we evaluate the performance of one of the two proposed multipath routing protocols using simulations. Additional parameters evaluated in the simulations provide further insights on the mechanics of the proposed techniques.

The rest of the paper is organized as follows. In the next section, we briefly review the DSR protocol, which we use as the base protocol for our multipath techniques. We then present two multipath extensions for DSR. In section 3, we develop analytical models to demonstrate how the frequency of query floods are reduced with our multipath extensions. In section 4, we present some numerical data obtained from the analytic models. Simulation results are discussed in section 5. We conclude in section 6.

2. Multipath Dynamic Source Routing

The Dynamic Source Routing (DSR) protocol [3,7] uses source routing – a technique where the source of a data packet determines the complete sequence of nodes through which to forward the packet; the source explicitly lists this route in the packet's header. DSR builds routes on demand using a technique called *route discovery*. A source initiates route discovery by flooding the network using query messages seeking a route to the destination. Each *query* (or *request*) message carries the sequence of hops it passed through in the message header. Once a query reaches the destination, the destination replies with a *reply* packet that simply copies the route from the query packet and traverses it backwards eventually reaching the source. This route is used for data packets as a source route. Each node maintains a *route cache*, where complete routes to various destinations are stored as learned from replies or any other packets that the node forwards. Route failure is detected by the failure of an attempted message transmission. Such a failure initiates an *error* packet to be sent backward to the source. The *error* packet erases all routes in the route caches of all intermediate nodes on its path, if the route contains the failed link. If a route is still needed a fresh route discovery is initiated.

DSR has a unique advantage by virtue of source routing. As the route is part of the packet itself, routing loops, either short- or long-lived, cannot be formed as they can be immediately detected and eliminated. This property opens up the protocol to a variety of useful optimizations. For example, a flooded query can be quenched early by having any non-destination host reply to the query if that host has a cached route to the intended destination. Ordinarily, this might cause routing loops, which would require elaborate mechanisms to detect or prevent. Also, a node can learn a route to other nodes while forwarding routing packets. Finally, new routes can be learnt or improved by having nodes promiscuously listen to conversations between other nodes in proximity.

2.1. Multipath extension to DSR

We first consider the situation where the destination replies to a selected set of queries. Note that many copies of the

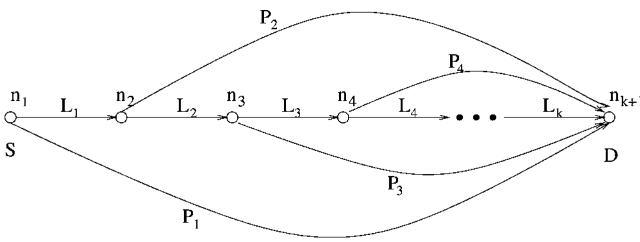

Figure 1. Multipath protocol 2. The primary route is depicted by the sequence of links $L_1 - L_2 - \cdots - L_k$. Each node in the primary route n_i, has an alternate path P_i to the destination.

flooded query message arrive at the destination via different routes. The queries that are replied to are those that carry a source route that is *link-wise disjoint* from the *primary* source route. Primary source route is the route taken by the first query reaching the destination. This usually defines the shortest route between the source and the destination. The destination "remembers" the primary route, in order to figure out which later requests to respond to. Only disjoint routes are chosen, as then a link failure in one route does not affect the others. This also implicitly controls the total number of replies, thus preventing a reply flood. The source keeps all routes received on reply packets in its route cache. When the primary route breaks, the shortest remaining *alternate* route is used. This process continues until all routes break, when a fresh route discovery is initiated. The number of alternate routes used can be a selectable parameter of the protocol. We will later show that only a few routes are actually sufficient. Let us call this technique as protocol 1, for ease of late reference.

Protocol 1 equips only the source with alternate routes. An intermediate link failure on the primary route still sends an error packet back to the source, which will then use an alternate route. This causes a temporary loss of route for the data packets that are already in transit upstream from the failed link. To prevent this, a better alternative is explored. All intermediate nodes are now equipped with a disjoint, alternate route so that in-transit data packets no longer face any route loss. To accomplish this, the destination now replies to *each* intermediate node in the primary route with an alternate disjoint route to the destination. Note that any such reply is in response to a query from the source that has traveled through that intermediate node. The reply is targeted to the intermediate node instead of the source. See figure 1 for an illustration of how the routes are maintained. Note that it may not always be possible for all intermediate nodes to get an alternate disjoint route. This would be particularly true for sparse networks. Thus, this still may result in temporary loss of routes on link failures until an upstream node switches to an alternate route. For the sake of simplicity, we will ignore this possibility in our analytical modeling work.

Refer to figure 1 to see the utility of the multiple routes maintained in each of the intermediate nodes. The source S uses the primary route for transmitting data packets to D until it breaks at some point, say L_i. When the link L_i is broken, the node n_i responds to the situation by replacing the unused portion of the route, $L_i - L_k$, in the data packet header by the alternate route P_i. This will continue until a link on P_i breaks. It will cause an error packet transmitted backward up to node N_{i-1}, which will quench the error packet and switch all later data packets to its own alternate route P_{i-1} by modifying the source route in the packet header as before.

Thus, loss of *all* routes in a node to the destination generates an error packet back to the source. Any intermediate node with an alternate path to the destination quenches the error packet. This node is also responsible for modifying the source route on all later data packets to use its own alternate route. This process continues until the source gets an error packet and has no alternate route to fall back on. Then a new route discovery is initiated. We refer to this technique as protocol 2. Note that unlike protocol 1, we maintain only one alternate path per intermediate node in protocol 2. In principle, any number of such paths can be maintained. But as we will show in our modeling work later, more than one alternate path provides only a very minimal additional advantage.

3. Analytic modeling

We represent the lifetime of a wireless link between a pair of nodes by a random variable. Consider a route from S to D that consists of a sequence of k wireless links over $k - 1$ intermediate nodes. Let L_i be the ith link in the route. The lifetime of L_i is denoted by X_{L_i}. Assume that X_{L_i}, $i = 1, 2, \ldots, k - 1$, are independent and identically distributed (iid) random variables, each with with mean l. We will use upper-case alphabets to denote random variables and the corresponding lower-case alphabets to denote their values.

Since a route fails when *any one* of the wireless links in its path breaks, the lifetime of a route P, consisting of k wireless links, is a random variable X_P that can be expressed as

$$X_P = \min(X_{L_1}, X_{L_2}, \ldots, X_{L_k}). \tag{1}$$

It is well known that X_P is also an exponentially distributed random variable with a mean of l/k [13].

Using the basic assumptions about the link failure behavior as above, we proceed to derive the statistics of the time interval between successive route discoveries for the proposed multipath techniques.

3.1. Modeling of protocol 1

Assume a source S has N routes to a destination D. The primary route is denoted by P_1 and the $N-1$ alternate routes are denoted by $P_2, P_3, \ldots, P_N$. The length of route P_i is k_i. See figure 2. The time after which none of the routes are useful is a random variable T, where

$$T = \max(X_{P_1}, X_{P_2}, \ldots, X_{P_N}). \tag{2}$$

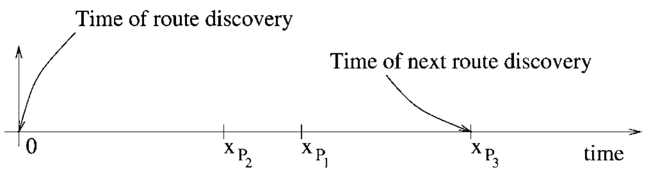

Figure 2. Example of multipath routing where the source has three independent routes P_1, P_2, P_3 to the destination ($N = 3$). The figure represents the lifetimes of the three routes. When the primary route P_1 breaks at time x_{P_1}, S attempts to use P_3, as P_2 is already broken. P_3 breaks at time x_{P_3}, when a new route discovery is initiated.

T represents the time between successive route discoveries. Here, we assume that the end-to-end packet transmission latency in the network is very small compared to the interval between route changes. Thus, the times to discover routes or propagate error packets etc. can be ignored relative to T. These are very reasonable assumptions, as otherwise, routes will fail while discovery or repair is in progress. No routing protocol will perform well in such dynamic conditions.

Lemma 1. The probability density function (pdf) of T, the time between successive route discoveries, is given by

$$\begin{aligned} f_T(t) &= \lambda_1 e^{-\lambda_1 t}\left(1-e^{-\lambda_2 t}\right)\left(1-e^{-\lambda_3 t}\right)\cdots\left(1-e^{-\lambda_N t}\right) \\ &\quad + \lambda_2 e^{-\lambda_2 t}\left(1-e^{-\lambda_1 t}\right)\left(1-e^{-\lambda_3 t}\right)\cdots\left(1-e^{-\lambda_N t}\right)+\cdots \\ &\quad + \lambda_N e^{-\lambda_N t}\left(1-e^{-\lambda_1 t}\right)\left(1-e^{-\lambda_3 t}\right)\cdots\left(1-e^{-\lambda_{N-1} t}\right), \end{aligned} \tag{3}$$

where $\lambda_i = k_i/l = 1/$lifetime of the ith route.

Proof. Consider N iid exponential random variables, $X_{P_1}, X_{P_2}, \ldots, X_{P_N}$, where the pdf of X_{P_i} is $f_{X_{P_i}}(t) = \lambda_i e^{-\lambda_i t}$, $i = 1, 2, \ldots, N$. Note that X_{P_i}'s are independent. Therefore, the cumulative distribution function (cdf) of T, $F_T(t)$, is obtained as

$$\begin{aligned} F_T(t) &= P[T \leqslant t] \\ &= P\left[\max(X_{P_1}, X_{P_2}, \ldots, X_{P_N}) \leqslant t\right] \\ &= P\left[(X_{P_1} \leqslant t) \cap (X_{P_2} \leqslant t) \cap \cdots \cap (X_{P_N} \leqslant t)\right] \\ &= \prod_{i=1}^{N} F_{X_{P_i}}(t), \end{aligned} \tag{4}$$

where $F_{X_{P_i}}(t) = 1 - e^{-\lambda_i t}$ is the cdf of X_{P_i}. By differentiating (4) with respect to t, we get the pdf of T as shown in (3). □

The expected value of T can be derived from (3) by knowing the hop-wise lengths of all the routes, k_i, $i = 1, 2, \ldots, N$. For example, for $N = 2$, the expected value of T, $E[T]$, is

$$E[T] = \frac{\lambda_1^2 + \lambda_2^2 + \lambda_1\lambda_2}{\lambda_1\lambda_2(\lambda_1+\lambda_2)}. \tag{5}$$

For the case that the primary route has $k_1 = 3$ hops, and the alternate route has $k_2 = 4$ hops, we have $E[T] = \frac{9+16+12}{12\times 7} l = \frac{37}{84} l$. Compare this with DSR with only a single route of 3 hops, where $E[T] = l/3$. This represents almost 25% reduction in the frequency of route discoveries compared to the single path case.

Consider also the special case when all the routes are equal, i.e. $k_1 = k_2 = \cdots = k_N = k$. In this case,

$$\begin{aligned} E[T] &= \frac{N}{\lambda} - \frac{N}{2^2\lambda}\binom{N-1}{1} + \frac{N}{3^2\lambda}\binom{N-1}{2} + \cdots \\ &\quad + (-1)^{N-1}\frac{1}{N\lambda}, \end{aligned} \tag{6}$$

where $\lambda = k/l$.

3.2. Modeling of protocol 2

The analysis in this case is a little more involved, as a new route discovery will be initiated only when some intermediate node loses both its downstream link and the alternate route. Let $\overline{L_i}$ denote the event that the link L_i fails, and $\overline{P_i}$ denote the event that the path P_i fails (see figure 1). Then the time until the next route discovery T, is the time before the event E is true, where E is described by the following logical expression:

$$\begin{aligned} E &= \left(\overline{L_1}\,\overline{P_1}\right) + \left(\overline{L_2}\left(\overline{L_1} + \overline{P_2}\right)\overline{P_1}\right) \\ &\quad + \left(\overline{L_3}\left(\overline{L_1} + \left(\overline{L_2} + \overline{P_3}\right)\overline{P_2}\right)\overline{P_1}\right) + \cdots. \end{aligned} \tag{7}$$

The ith term inside the braces of the right-hand side of (7) represents the events starting with the failure of L_i and leading to a new route discovery. For example, the second term represents the following sequence of events:

- L_2 breaks on the primary route, prompting S to form the alternate route $L_1 - P_2$.
- The alternate route breaks when either L_1 or P_2 breaks, prompting S to use the new alternate route P_1.
- This route fails when P_1 breaks, causing S to start a new route discovery.

Hence, starting with the breakage of L_2, the events leading to a new route discovery from S is $\overline{L_2}(\overline{L_1} + \overline{P_2})\overline{P_1}$. The other terms can be derived similarly.

Equation (7) can be simplified to a more compact form:

$$\begin{aligned} E &= \overline{L_1}\,\overline{P_1} + \overline{L_2}\,\overline{P_2}\,\overline{P_1} + \overline{L_3}\,\overline{P_3}\,\overline{P_2}\,\overline{P_1} + \cdots \\ &\quad + \left(\overline{L_k}\,\overline{P_k}\,\overline{P_{k-1}}\cdots\overline{P_1}\right), \end{aligned} \tag{8}$$

which is also quite intuitive. Then T can be expressed as

$$\begin{aligned} T = \min\bigl(&\max(X_{L_1}, X_{P_1}), \max(X_{L_2}, X_{P_2}, X_{P_1}), \\ &\ldots, \max(X_{L_k}, X_{P_k}, X_{P_{k-1}}, \ldots, X_{P_1})\bigr). \end{aligned} \tag{9}$$

Recall that X_{L_i} is an exponential random variable with mean l. If the alternate path P_i consists of k_i hops, then X_{P_i} is also an exponential random variable, with mean l/k_i. Note again that according to our assumptions, X_{L_i} and X_{P_i} are independent.

Let us denote the random variable $\max(X_{L_i}, X_{P_i}, X_{P_{i-1}}, \ldots, X_{P_1})$ by Z_i. Hence, following (3), the pdf of Z_i, $f_{Z_i}(t)$, is given by

$$f_{Z_i}(t) = \sum_{j=1}^{i+1}\left(\lambda_j^{(i)} e^{-\lambda_j^{(i)} t} \prod_{k=1,k\neq j}^{i+1}\left(1 - e^{-\lambda_k^{(i)} t}\right)\right), \tag{10}$$

where

$$\lambda_j^{(i)} = \begin{cases} \dfrac{k_j}{l} & \text{for } j = 1, 2, \ldots, i \\ \dfrac{1}{l} & \text{for } j = i+1. \end{cases} \tag{11}$$

We now derive the pdf of $T = \min(Z_1, Z_2, \ldots, Z_k)$. For mathematical tractability we assume that Z_i, $i = 1, 2, \ldots, k$, are statistically independent. Though this may not be strictly true, noting the independence assumption of X_{L_i} and X_{P_i} made earlier, it may be postulated that results obtained under such an assumption will not differ too much from the exact case. Exact results may only be obtained from experimental data or simulations.

Lemma 2. The pdf of T is given by

$$f_T(t) = \sum_{i=1}^{k}\left(f_{Z_i}(t) \prod_{j=1,j\neq i}^{k}\left(1 - F_{Z_i}(t)\right)\right). \tag{12}$$

Proof. From definition, the pdf of T can be written as

$$f_T(t) = \lim_{dt\to 0} \frac{P[t \leqslant T \leqslant t + dt]}{dt}. \tag{13}$$

Consider

$$\begin{aligned} &P[t \leqslant T \leqslant t + dt] \\ &\quad = P\big[t \leqslant \min(Z_1, Z_2, \ldots, Z_k) \leqslant t + dt\big] \\ &\quad = \sum_{i=1}^{k} P\big[t \leqslant Z_i \leqslant t + dt \mid Z_i = \min(Z_1, Z_2, \ldots, Z_k)\big] \\ &\qquad \times P\big[Z_i = \min(Z_1, Z_2, \ldots, Z_k)\big] \\ &\quad = \sum_{i=1}^{k} P[t \leqslant Z_i \leqslant t + dt] \\ &\qquad \times P\big[Z_i = \min(Z_1, Z_2, \ldots, Z_k)\big]. \end{aligned} \tag{14}$$

Dividing both sides by dt and taking the limit as dt approaches zero, we get the desired result:

$$\begin{aligned} f_T(t) &= \sum_{i=1}^{k} f_{Z_i}(t) \prod_{j=1,j\neq i}^{k} P[Z_j > Z_i] \\ &= \sum_{i=1}^{k} f_{Z_i}(t) \prod_{j=1,j\neq i}^{k} \big(1 - F_{Z_i}(t)\big). \end{aligned} \tag{15}$$

□

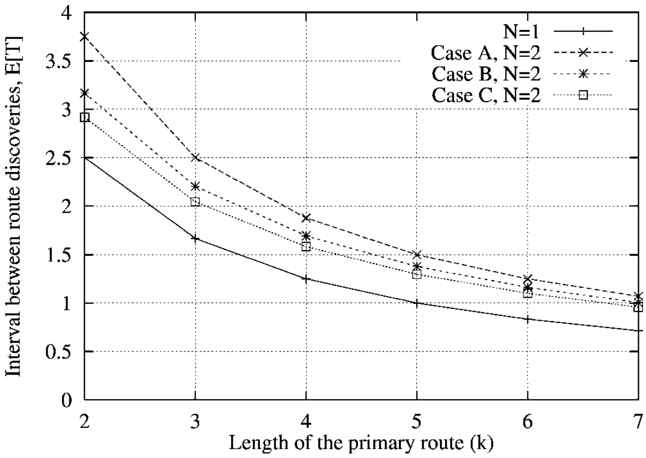

Figure 3. Performance of protocol 1 with different lengths of the primary route. Three cases for two routes (one primary, and one alternate) are compared with the single path case. Mean lifetime of a single link = 5.

4. Numerical results

In this section we present some numerical results showing the performance benefits of the multipath routing protocols using the analysis presented previously. Let us start with the protocol 1. The pdf of the time interval between successive route discoveries is given by equation (3). We evaluate the expected value of this interval using numerical techniques for three special cases. In the first case (*Case A*), we assume that all the N paths from S to D are of the same length (equation (6)). This implies the "best case" scenario for multipath routing, where N disjoint shortest path routes exist between the given source–destination pair. In the second (*Case B*) and third (*Case C*) cases, we assume that the N disjoint routes between the source and destination are of increasing lengths, with the primary route being the shortest. For *Case B* the successive path lengths increase by one, and for *Case C* they increase by two. Thus,

Case A: $k_1 = k_2 = \cdots = k_N = k$;

Case B: $k_1 = k,\ k_2 = k + 1, \ldots, k_N = k + N - 1$;

Case C: $k_1 = k,\ k_2 = k + 2, \ldots, k_N = k + 2(N - 1)$.

The expected time interval between successive route discoveries at source for the three cases are plotted against different values of the path length of the primary route, in figure 3. The mean lifetime of a single link (l) is assumed to be 5, for this plot and all later plots. The interval increases from *Cases C* to *B* to *A*. For all cases the interval is longer than the single path case denoting less frequent route discovery. Also, relative advantage of multipath routing diminishes as the primary route gets longer. This is intuitive, as the alternate routes are at least as long, and longer routes typically break earlier.

In figure 4 the expected interval between route discoveries in protocol 1 is plotted against the number of routes (N) maintained, all for the same primary route length. Two primary route lengths are used ($k = 3$ and 6). As expected performance always improves with increasing number of alter-

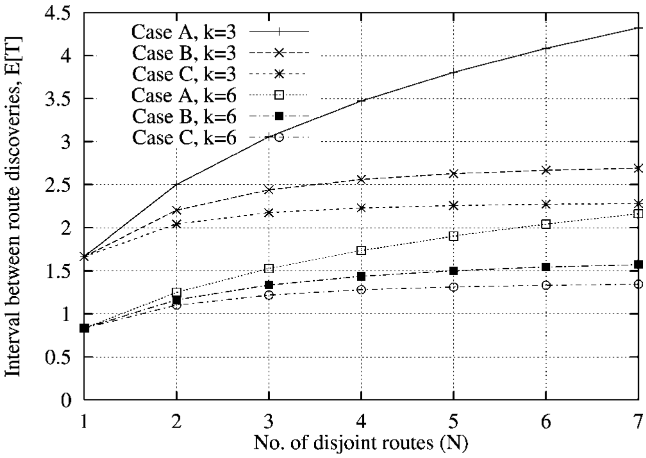

Figure 4. Performance of protocol 1 with varying number of routes (N). One is the primary route and the rest $N-1$ are alternate routes. Two different lengths of the primary route are used, $k=3$ and 6.

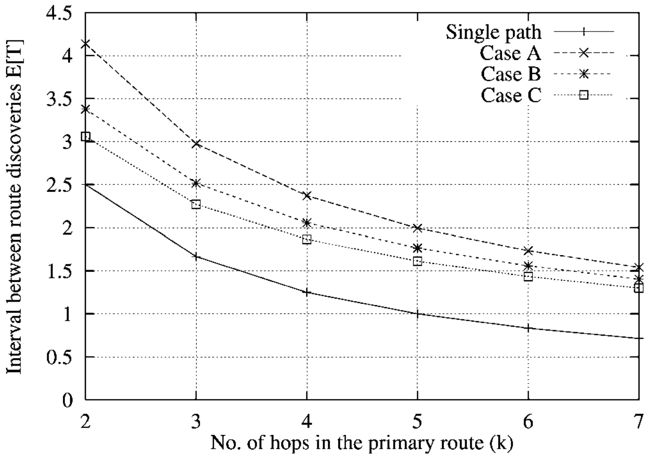

Figure 5. Performance of multipath routing protocol 2 with different lengths of the primary route. Performance of the single path routing is shown as a reference.

nate routes. However, the incremental improvement is very small for $N > 3$, except when the paths are of the same length (*Case A*). Note that this case is very unlikely to occur in practice. This indicates that usually only one or two additional routes will be sufficient.

In protocol 2 the performance is dependent on the values of k_i, $i = 1, 2, \ldots, k$, i.e., the path lengths in the alternate routes from each intermediate node. The actual values of these parameters are dependent on the dynamic conditions of the network. To get an idea of the performance improvement with multipath routing, we consider three different estimates of these parameters. In *Case A*, we assume that the alternate routes from each intermediate node n_i are of the same length as that of the primary route from n_i to the destination. In *Cases B* and *C*, we assume that the alternate route is larger than the primary route by one and two, respectively. Note that we consider only a single alternate route per node in protocol 2. Since the primary route is of length k, the path lengths of P_i, $i = 1, 2, \ldots, k$, are:

Case A: $k_i = k - i + 1$;

Case B: $k_i = k - i + 2$;

Case C: $k_i = k - i + 3$.

For these three cases, we again use numerical techniques to determine the expected time interval between route discoveries ($E[T]$) from the pdf of T as given by equation (12). We plot the expected time interval for some special cases in figure 5. As expected, the protocol 2 performs significantly better than single path routing.

To compare the relative performance of protocols 1 and 2, we look at *Case A* for both protocols, where the "best" alternate routes are assumed to be available for both protocols. The expected time between successive route discoveries are compared in table 1. Also shown in table 1 are frequencies of route discoveries for multipath routing protocols relative to the single path routing. As expected protocol 2 provides more substantial performance advantage compared to protocol 1.

Table 1
Comparison between the two variations of the multipath protocols. All frequencies are relative to single path routing.

	Protocol 1 ($N=2$)		Protocol 2	
k	$E[T]$	Frequency of route discovery	$E[T]$	Frequency of route discovery
3	2.5	0.668	2.67	0.560
4	1.875	0.667	2.03	0.527
5	1.50	0.667	1.64	0.502
6	1.25	0.667	1.37	0.481
7	1.07	0.667	1.18	0.462

5. Performance evaluation via simulations

The analytic model presented in the previous sections provides a basic understanding of the performance impact of the multipath routing. However, the model uses somewhat idealized assumptions such as iid exponential link lifetimes. It does not model the routing load, which may impact the overall performance. Also, the model does not consider the end-to-end delay which tends to get longer as longer alternate routes are used. In this section, we use a routing simulator with more realistic model of the protocol and the mobile network to validate the general observations made in the previous sections.

We simulated the DSR protocol with single path as well as multipath routing for a mobile, ad hoc network. The multipath protocol 2 was used with only one alternate route, though some experiments were also performed with more alternate routes. Note that in the simulations, the protocol attempts to discover alternate routes for each intermediate node as well as the source. But there may not be any such route present in the network at that instant. This is an important difference between the analytic modeling assumptions and the realities in the simulations. An event-driven, packet-level routing simulator, called MaRS [1] is used for the simulation study. MaRS has been used before for studying dynamic routing protocols in wired networks with dynamic traffic conditions and faulty links [15]. We extended

MaRS to make the nodes mobile. The links fail and connect as nodes go away or come within the radio range of one another. We have used this extended version of MaRS earlier for evaluations of ad hoc routing protocols [4,5]. MaRS simulates all relevant network layer details including queuing delays and packet processing times in each node. Link failures and reconnects automatically generate link-layer events in the simulation model, to which the routing protocol responds. No MAC or physical layer is simulated and error-free wireless transmission without any multiple-access interference is assumed. Even though this is a limitation of the simulation model, our earlier experience [5] with MaRS shows that it is able to make a very good qualitative evaluation at the network-layer comparable to more elaborate simulation models [2].

In the model we have simulated, 60 mobile hosts move around a rectangular region of size 1000 m $\times$ 1000 m according to the following mobility model. Each node chooses a direction, speed and distance of a "move" based on a predefined distribution and then computes its next position P and the time instant T of reaching that position. Similarly, a new move is again computed at simulation time T. A node computes its neighborhood after each such move, thus generating link failure and link repair events that in turn drive the routing protocol. Each node is assumed to have a radio range of 350 m. For the experiments reported first, the speed of each move is uniformly distributed between 14 and 18 m/s (average 16 m/sec). Distance is exponentially distributed with a mean of 5 m, and the direction is uniformly distributed within [+90°, −90°] with respect to the direction of the previous move. Note that in the chosen mobility model, the nodes are always moving (*albeit* in discrete time) without stationary intervals. This presents a stress case for the routing protocol.

A simple datagram workload model is used. All data packets are 512 bytes long, and interarrival times are exponentially distributed with a mean of 300 ms. There is no acknowledgment, or flow or congestion control in the workload model. Workload traffic is always between a pair of source and sink nodes, called a *conversation*. The number of such pairs or conversations is varied over a wide range in the simulation experiments. In the performance plots, it is presented in terms of the number of conversations per node in the network. All simulations are run for 300 simulated seconds and each point in a plot represents an average of five runs with different random number streams.

Four important performance metrics are evaluated: (i) *fraction of packets dropped* – measured as the ratio of the number of data packets dropped *en route* to the number of data packets sent by the source; (ii) *end-to-end delay* – measured as the average end-to-end latency of data packets; (iii) *number of route discoveries* – measured as the number of times a route discovery is initiated by any node during the simulation run (it is inversely proportional to the time interval between route discoveries); (iv) *routing load* – measured in terms of the total number of routing packets transmitted (broadcast transmissions are counted as a single transmission).

5.1. Simulation results

Figure 6 shows significant improvements in the number of route discoveries for different number of conversations demonstrating that the multipath protocol is able to reduce

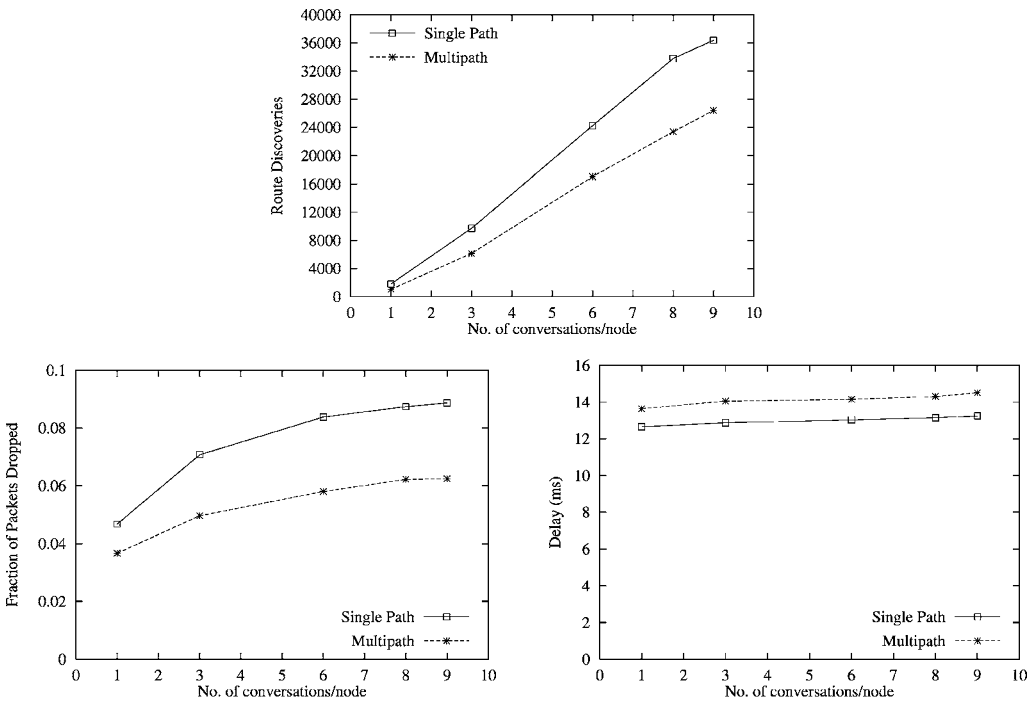

Figure 6. Number of route discoveries, fraction of data packets dropped, and end-to-end delay for the single path and multipath DSR protocols. The multipath protocol (protocol 2) uses one disjoint alternate route from each intermediate node (including the source) to the destination.

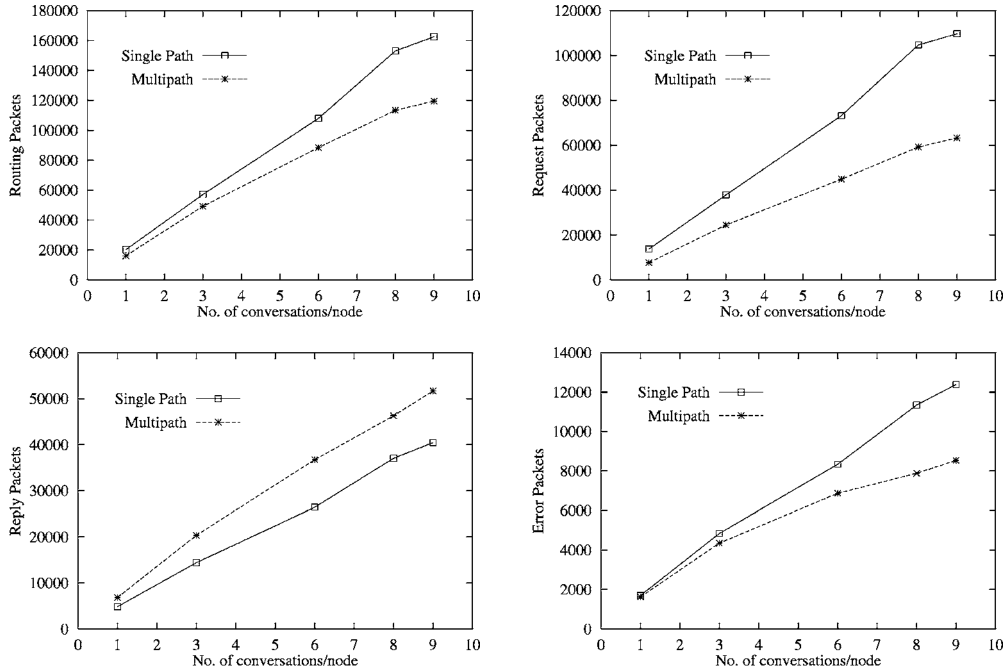

Figure 7. Total number of routing packets and its components, consisting of the request, reply and error packets, transmitted during the simulation run for the same set of simulations as in figure 6.

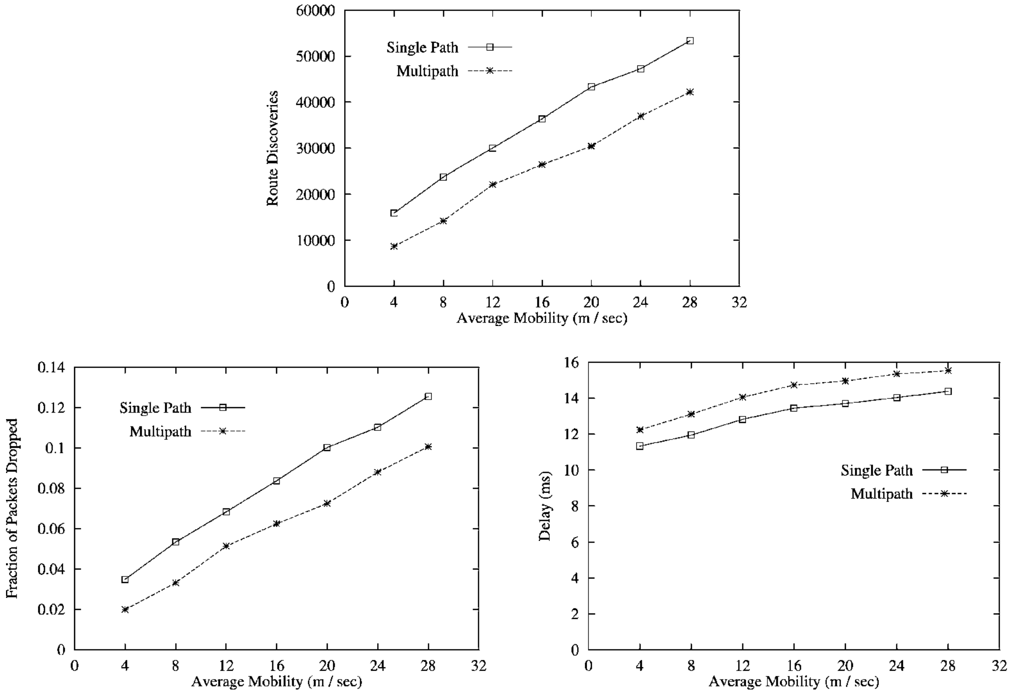

Figure 8. Number of route discoveries, fraction of data packets dropped, and end-to-end delay for the same network with varying mobility. Now the traffic load is fixed at 9 conversations per node.

the number of route discoveries by roughly onethird. Note that it also translates to improvements in the fraction of packets lost, by again about onethird, as many of the packets lost during a route discovery period can now be saved. Note that in the simulation model of the protocols, data packets are not buffered when the route discovery is in progress. The end-to-end delay, however, has increased a little in the multipath protocol. This is due to the fact that the alternate routes are typically longer than the primary. Detailed instrumentation of the simulator reveals that 30–40% of delivered data packets use alternate routes. The alternate routes are on an average about 1 hop longer than the primary route. The primary route length was found to be about 4, on an average. Note that according to our numerical results earlier (figure 5), the improvement in the number of route discoveries was about one-half for these parameter values, while in the simulations

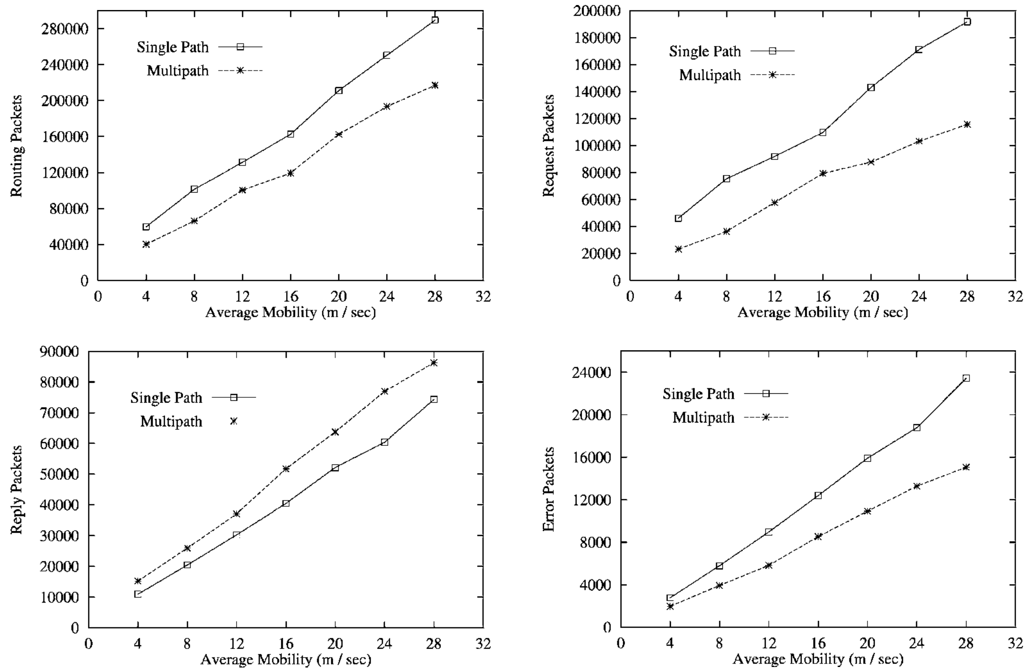

Figure 9. Total number of routing packets and its components, consisting of the request, reply and error packets, that are transmitted during the simulation run for the same set of simulations as in figure 8.

it is in the ballpark of onethird. We attribute this quantitative difference to the idealistic assumptions in the analytic model, such as exponential lifetime of links, availability of alternate routes of specific lengths etc. These are hardly true in the simulation model. Note also that it is reasonable for the simulation model to underestimate performance as compared to the analytic model. Availability of alternate routes is not always guaranteed in the simulated network.

Figure 7 shows the routing load and its breakdown into request, reply and error packets. Note that the number of route request packets come down for the multipath protocol almost proportionate to the number of route discoveries. The replies, on the other hand, increase somewhat. This is expected because of multiple replies per route discovery cycle. The error packets also come down because of fewer route errors generated. Overall these translate to savings on the routing load. The savings are more significant on high loads, close to a quarter.

Figures 8 and 9 show results from a similar set of experiments with varying mobility but constant traffic load at 9 conversations per node. Very similar results were also obtained from other values of traffic loads and are not presented here. General observations are same as above, except that we find a smaller fractional improvement of routing load and number of route discoveries for higher mobilities. This is due to the fact that for higher mobilities it is more likely that a breakage of an existing route is accompanied by a breakage of its alternative route. The breakages of the existing route and the alternate route may be correlated, for instance, when the existing route breaks due to the movement of the destination node. Note that this effect is not captured in our analytical results since they were obtained under the assumption that all individual link lifetimes (and hence lifetimes of the routes) are statistically independent regardless of the actual value of the link lifetime. Thus, the relative performance results there (figures 3–5) are independent of link lifetime and hence node mobility.

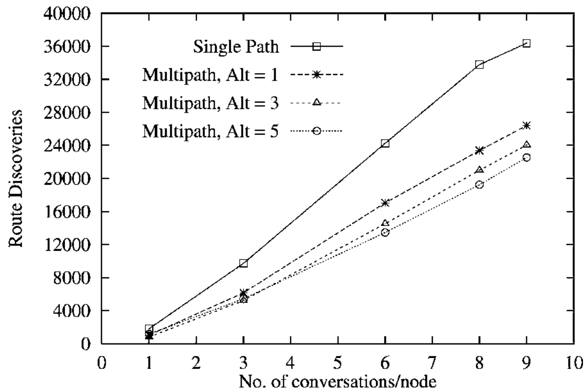

Figure 10. Number of route discoveries for the same set of simulations as in figure 6. Now three different versions of the multipath protocol are shown, with 1, 3 and 5 alternate routes.

To verify the hypothesis formed with analytical modeling that more than just a few alternate route is not often very helpful, we ran some simulations with 3 and 5 alternate routes in addition to 1 (as presented before). Figure 10 presents data for route discoveries. Notice that the improvement beyond 1 alternate route is minimal.

6. Conclusions

We proposed a multipath extension for the popular on-demand routing protocol DSR. The extension explores alternate, disjoint routes that can be useful in case the primary route breaks. Two variations are explored. In the first, only the source gets multiple alternate routes. In the second, each intermediate node on the primary route gets an alternate route. The key advantage is the reduction in the frequency of route discovery flood, which is recognized as a major overhead in on-demand protocols. We also provide a framework for analytical modeling of the time interval between successive route discoveries for on-demand protocols based on simple assumptions on the lifetime of a single wireless link. Evaluation of the multipath routing extension in this framework demonstrates that there are definite advantages to be gained from providing alternate routes on intermediate nodes, in addition to the source node. In any case, any form of multipath technique always performs substantially better than single path routing. The modeling effort also shows that longer alternate paths are less advantageous, as they tend to break too early. This indicates that it is only useful to explore alternate routes with some bound on the hop-wise path length. Also, the performance advantage from using more than one or two alternate routes is usually minimal.

These observations are also validated using a packet-level routing simulator working on a model of a mobile ad hoc network. Qualitative obervations are similar to analytic modeling. There are significant reductions in the number of route discoveries and hence in routing load from the use of multipath routing. This also improves the fraction of packets dropped. Quantitatively however, the simulation results underestimate performance somewhat compared to analytic modeling because of the difference in modeling assumptions. The simulation study also demonstrates that even though the multipath routing brings down routing load, it also increases end-to-end delay, as alternate routes tend to be longer than primary routes. This presents a subtle trade-off between delay and routing load. However, our simulation does not model the radio link layer and thus no multiple access interference is captured. We believe that in a real network, reduction in routing load will also contribute to reduced end-to-end delay because of reduced interference. Thus, the trade-off will be more on the side of reduction of the routing load.

References

[1] C. Alaettinoglu, A.U. Shankar, K. Dussa-Zieger and I. Matta, Design and implementation of MaRS: A routing testbed, Journal of Internetworking: Research and Experience 5(1) (1994) 17–41.

[2] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu and J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: *Proc. 4th Int. Conf. on Mobile Computing and Networking (ACM MOBICOM'98)* (October 1998) pp. 85–97.

[3] J. Broch, D. Johnson and D. Maltz, The dynamic source routing protocol for mobile ad hoc networks, IETF Internet Draft (work in progress), `http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-03.txt` (October 1999).

[4] R. Castañeda and S.R. Das, Query localization techniques for on-demand protocols in ad hoc networks, in: *Proc. 5th Int. Conf. on Mobile Computing and Networking (ACM MOBICOM'99)* (1999) pp. 186–194.

[5] S.R. Das, R. Castañeda, J. Yan and R. Sengupta, Comparative performance evaluation of routing protocols for mobile, ad hoc networks, in: *Proc. 7th Int. Conf. on Computer Communications and Networks (IC3N)* (October 1998) pp. 153–161.

[6] K.J. Negus et al., HomeRF and SWAP: Wireless networking for the connected home, ACM SIGMOBILE Mobile Computing and Communications Review 2(4) (October 1998) 28–37.

[7] D. Johnson and D. Maltz, Dynamic source routing in ad hoc wireless networks, in: *Mobile Computing*, eds. T. Imielinski and H. Korth (Kluwer Academic, 1996) chapter 5.

[8] J. Jubin and J.D. Tornow, The DARPA packet radio network protocols, Proceedings of the IEEE 75(1) (January 1987) 21–32.

[9] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network* (Addison-Wesley, 1997) chapter 11.

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

[11] V.D. Park and M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: *Proc. IEEE INFOCOM'97 Conf.* (April 1997) pp. 1405–1413.

[12] C. Perkins and E. Royer, Ad hoc on-demand distance vector routing, in: *Proc. 2nd IEEE Workshop on Mobile Computing Systems and Applications* (February 1999) pp. 90–100.

[13] S. Ross, *Introduction to Probability Models* (Academic Press, 1989).

[14] N. Schacham and J. Westcott, Future directions in packet radio architectures and protocols, Proceedings of the IEEE 75(1) (January 1987) 83–99.

[15] A.U. Shankar, C. Alaettinoglu, K. Dussa-Zieger and I. Matta, Transient and steady-state performance of routing protocols: Distance-vector versus link-state, Journal of Internetworking: Research and Experience 6 (1995) 59–87. (Preliminary version appeared in: *Proc. ACM SIGMETRICS/PERFORMANCE Conf.* (1992) pp. 181–192.)

[16] W.T. Zaumen and J.J. Garcia-Luna-Aceves, Loop-free multipath routing using generalized diffusing computations, in: *Proc. IEEE INFOCOM'98 Conf.* (March 1998) pp. 1408–1417.

Asis Nasipuri is currently an Assistant Professor of Electrical and Computer Engineering at the University of North Carolina at Charlotte. He received his B.Tech. (Hons.) degree in electronics and electrical communication engineering from the Indian Institute of Technology, Kharagpur, in 1987, and his M.S. and Ph.D. degrees in electrical and computer engineering from the University of Massachusetts at Amherst, in 1990 and 1993, respectively. He has been a member of the faculty of the Department of Electronics and Electrical Communication Engineering at the Indian Institute of Technology, Kharagpur, from 1993 to 1998. From 1998 to 2000, he has been a postdoctoral fellow at the Division of Computer Science at the University of Texas at San Antonio. His research interests include wireless communications and mobile networks. He is a member of the IEEE.
E-mail: anasipur@uncc.edu

Robert Castañeda got his Ph.D degree in computer science from the University of Texas at San Antonio in 2000. His research interests were in the field of wireless ad hoc network protocols and simulation. He also obtained a M.S. and B.S. degrees at the same university in 1994 and 1990, respectively, in computer science. He is currently a software engineer at Symmetry Communications Systems in San Jose, California.
E-mail: robert_castaneda@symmetrycomm.com

Samir R. Das is currently an Associate Professor in the Department of Electrical and Computer Engineering and Computer Science at University of Cincinnati. Prior to this he was a faculty member at The University of Texas at San Antonio. He received his B.E. degree in electronics and telecommunication engineering from Jadavpur University, Calcutta, in 1986; M.E. degree in computer science and engineering from the Indian Institute of Science, Bangalore, in 1988; and Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, in 1994. His research interests include mobile/wireless networking and parallel discrete event simulation. He has published over 30 research articles on these topics. He received the U.S. National Science Foundation's Faculty Early CAREER award in 1998. Dr. Das served on the program committees of several recent conference and workshops related to parallel/distributed simulation, mobile/wireless networking and performance evaluation; and guest-edited a special issue of the *Computer Communication* journal. He is a member of the IEEE, IEEE Computer Society and ACM.
E-mail: sdas@ececs.uc.edu