Trigger-Based Distributed QoS Routing in Mobile Ad Hoc Networks

Swades De[a] **Sajal K. Das**[b] **Hongyi Wu**[c] **Chunming Qiao**[d]
swadesd@eng.buffalo.edu *das@cse.uta.edu* *wu@cacs.louisiana.edu* *qiao@cse.buffalo.edu*
[a]Department of Electrical Engineering, State University of New York at Buffalo, Buffalo, NY 14260
[b]Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019
[c]Center for Advanced Computer Studies, University of Louisiana at Lafayette, Lafayette, LA 70504
[d]Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260

Performance of existing routing protocols in mobile ad hoc networks for real-time applications is limited by high control traffic and database maintenance overhead. We observe that by proper coupling of nodal mobility and location information, real-time applications can be served with limited control traffic and database requirements. In this paper, we investigate a trigger-based *(on-demand)* distributed *routing protocol, called TDR, for supporting real-time applications in mobile ad hoc networks. For increased resource efficiency, the nodal database size is reduced by maintaining only the local neighborhood information. Only one route per session is maintained and the reroute routine is invoked before any active link fails. In addition, by making efficient use of the location information, the control overhead for rerouting is further reduced. Our evaluation shows that the TDR protocol provides better QoS and requires lower control overhead compared to the other existing real-time QoS aware ad hoc routing protocols.*

I. Introduction

A mobile ad hoc network is a collection of mobile hosts which can communicate among themselves via possibly multiple hops. The nodes operate in a self-organized fashion in the sense that all the mobile nodes are responsible for maintaining sessions with no dedicated base stations and controller involved. Lack of a centralized control and dynamism of the network topology make routing in ad hoc networks a unique challenge.

Many authors have addressed the routing issues in ad hoc networks from the best-effort service point of view [7], [11], [12], [14], [18]. While these approaches attempt to minimize the control and database maintenance overhead in serving the traffic, they do not meet *real-time quality of service* (RT-QoS) criteria, such as bandwidth constraint, end-to-end packet delay, and delay jitter. On the other hand, the proposals dealing with RT-QoS provisioning require high control and/or nodal database maintenance overhead [3], [13], [15], [17].

We observe that some form of proactive routing scheme has to be adopted to tackle the delay and bandwidth constraints of real-time applications in ad hoc networks. At the same time one has to see that the buffer and signaling overhead do not go overboard so that the resource utilization is optimized. To address these issues jointly (i.e., QoS support and resource optimization), one needs to have proper mobility and location information about the nodes. As it has been demonstrated in [8], the location information can effectively reduce the route discovery overhead. Likewise, the ability to predict location of nodes with the knowledge of their mobility would help in efficiently rerouting a session. Ideally, with the correct prediction of location of nodes, the alternate path searching algorithm can be triggered at right times and within a suitably limited geographic zone to reduce network control overhead while maintaining the real-time traffic constraints.

In this paper, we present an on-demand and yet proactive routing algorithm, called *trigger-based distributed routing* (TDR), to deal with link failures (induced by, e.g., nodal mobility) in mobile ad hoc networks. Our goal is to provide RT-QoS support while keeping the network overhead low. More specifically, to reduce control traffic, we propose to maintain only the active routes and exploit the GPS (global positioning system)-based location information of the destination to selectively broadcast reroute queries when a link failure is imminent. The proposed algorithm operates in a distributed fashion to reduce the nodal computation and database overhead. Our studies show that the proposed TDR protocol provides better QoS support with lower control overhead in comparison with the schemes in [3] and [15], which operate without link failure prediction capability. The TDR scheme provides QoS support comparable to FORP [17] while incurring substantially lower control overhead.

The rest of the paper is organized as follows. Related previous work is surveyed in Section II. The TDR protocol details are provided in Section III. Section IV provides the analytic performance modeling of the protocol in terms of reduced control overhead due to selective route search. Section V contains simulation-based performance evaluation and comparison results. Section VI concludes the paper.

II. Previous Work

A lot of work have been reported on routing protocols for mobile ad hoc networks. While the *reactive* (or on-demand) algorithms, such as DSR [7], TORA [11], ABR [18], LAR [8], AODV [14], and ZRP [12], operate with limited control and database maintenance overhead, they are suitable only for delay-tolerant applications. On the

other hand, the *proactive* (or table-driven) approaches, such as DSDV [13], WRP [10], GSR [2], and DREAM [1], attempt to minimize the route disruption times, but are encumbered with high control and database maintenance overhead.

Recently, some QoS capable protocols have been reported. For example, a protocol with QoS extension to AODV [15], called E-AODV, addresses the bandwidth and delay guarantee requirements. Route discovery in this protocol is broadcast based. Also, its reactive nature does not help minimize the service disruptions due to nodal mobility. An in-band signaling approach for supporting QoS, called INSIGNIA, is presented in [9], where a route is discovered by the in-flow packets and is maintained at the active nodes by velocity-dependent 'soft state' tags. Since the nodes are not responsible in maintaining the flow state information, in case of route failure duplicate and out-of-order packet delivery can still occur. The Distributed Quality-of-Service Routing (which we call DQoSR) scheme proposed in [3] for meeting bandwidth and/or delay constraints requires that a number of secondary routes be maintained along with the primary (currently in use) route to the destination. The network state information at each node, obtained via periodic beaconing, enables finding routes to the destination by a limited number of 'tickets'. But this costs extra database and bandwidth. In particular, the nodal database will grow at the same rate with network size as in DSDV. In Flow Oriented Routing Protocol (FORP) [17], the flow states are maintained for QoS support, aided by the predicted link expiration times. In this protocol, rerouting is controlled by the destination node and route discovery at any phase is broadcast based. Implementation of proactive routing on top of DSR and AODV for providing QoS support is reported in [5]. The rerouting scheme in this approach is source controlled and route discovery is broadcast-based.

III. Trigger-Based Routing

The proposed *trigger-based distributed routing* (TDR) protocol is designed to support QoS-aware real-time applications. The scheme makes use of on-demand route discovery, as in DSR, AODV, ABR, and TORA, to reduce the control overhead. To maintain the RT-QoS constraints, the flow state for each session is maintained, as in FORP [17], but in a distributed fashion at the active nodes. In case of imminent link failure in the active route, alternate route searching overhead is kept limited by localizing the reroute queries to within certain neighbors of the nodes along the source-to-destination active route. For cost efficiency (quicker search and reduced control overhead), rerouting is attempted from the location of an imminent link failure which we denote as *intermediate node initiated rerouting* (INIR). If INIR fails, to keep the flow state disruption at a minimum, rerouting is attempted from the source node, which is termed as *source initiated rerouting* (SIRR). The TDR scheme keeps the size of the nodal database small, irrespective of the network size, by maintaining only the local neighborhood information. In addition, an activity-based database is maintained at each node whose size is limited by its maximum data handling capacity and interference from the other nearby nodes.

From the network operations point of view, the proposed TDR scheme is a *reactive* algorithm, as the rerouting routine is *triggered* at an active node based on the level and trend of variation of its receive power from the downstream active node. Hence the name "trigger-based" routing. On the other hand, from the user application point of view, it is a *proactive* algorithm as (ideally) the traffic experiences no break in the logical route during the session, thus making it suitable for dealing with real-time traffic. The routing scheme is also "distributed" in the sense that any active node participating in a session can make its own routing decision, which helps reducing the computational overhead. The protocol details are described below.

III.A. Database Management

All nodes in the network maintain the local neighborhood information. In addition, for an on-going session, depending on its activity a node maintains one of the three information bases: source database, intermediate node database, and destination database.

III.A.1. Local Neighborhood Database

A node can be in either of the two states - idle (when it is not involved in any session) and active (when it participates in a session). In any state (idle or active), a node n periodically broadcasts beacons containing its location and mobility information to its local neighbors. It also listens to the beacons and maintains a local neighborhood database denoted as link table, LT_n, as shown in Table 1. The nodes keep the neighborhood information up-to-date by adjusting the beaconing frequency, depending on the relative mobility of the neighbors.

Table 1: Link table information fi elds at node n for the i-th neighbor

LT_n·	Field description
P_i	Receive power level
X_i, Y_i	Current (X, Y) coordinate
Vel_i, Dir_i	Velocity, direction of motion

Note that unlike TDR (as well as FORP and E-AODV), which maintains only local neighborhood database, DQoSR maintains the global information (delay, bandwidth, and cost to all possible destinations) at each node. Assuming the size of database for each nodal information to be the same in both cases, in an N-node network with n_g neighbors on average, DQoSR would need to maintain a nodal database which is approximately $\left(\frac{N}{n_g}\right)$ times larger than that of TDR. This also indicates that for the same network density, the nodal database size in DQoSR grows linearly with network size.

III.A.2. Activity-Based Database

Besides the neighborhood information, if a node actively participates in a session as either the source (S), the destination (D), or an intermediate node (IN), a corresponding table called a source table ST_n, a destination table DT_n, or an IN table IT_n, is maintained. The fields in the three types of databases are shown in Table 2, where the first three fields (*Session_ID*, *S_ID*, and *D_ID*) uniquely identify a session. The other fields are maintained for routing information exchange, to be explained in the next subsection.

Table 2: Activity-based information fields in different databases at node n

ST_n:	IT_n:	DT_n:	Field description
Session_ID	Session_ID	Session_ID	Session ID
S_ID	S_ID	S_ID	Source ID
D_ID	D_ID	D_ID	Destination ID
.....	S_loc	S_loc	Source location (X, Y)
Max_BW	Max_BW	Max_BW	Maximum bandwidth demand
Max_Del	Max_Del	Max_Del	Maximum acceptable delay
D_loc	D_loc	.....	Destination location (X, Y)
N_ID	N_ID	.....	Next node ID (towards D)
.....	P_ID	P_ID	Previous node ID (towards S)
.....	Dist	Dist	Distance from S (hop count)
Nod_actv	Nod_actv	Nod_actv	Activity flag (0 or 1)

At any time instant, a node n may require to maintain some or all of the tables ST_n, IT_n, and DT_n simultaneously for different on-going sessions. Contrary to the wireline networks, where link capacities (bandwidth) are independent of a node's connectivity, in wireless networks a node's data handling capacity is limited by the node's allocation of bandwidth. For example, if the MAC layer protocol is CDMA-based, then a node's maximum data rate is limited by multiuser interference and the number of available orthogonal codes (if multicoding scheme is used). If the MAC protocol is TDMA-based, then it is limited by the available time slots, frequency spectrum, and co-channel interference. Accordingly, each node n (idle or active) also maintains an updated residual bandwidth (*Resi_BW*$_n$) which indicates its ability to participate in a session. Since the maximum bandwidth resource is limited, the number of sessions that a node can participate in is also limited, irrespective of the network density and size. Therefore, the size of the activity-based database is also limited. The activity-based database is soft-state maintained and requires to be refreshed by in-session data packets. At any time, if at a node (n) the soft-state timer for a session expires (e.g., due to unforeseen route failure), the corresponding nodal database is purged and the *Resi_BW*$_n$ is refreshed.

III.B. Control Traffic Management

To maintain updated routing information (activity-based database) at the nodes, certain information exchange among the active nodes are necessary. The required messages to be exchanged for initiating, maintaining, and terminating a real-time session are discussed below.

III.B.1. Initial Route Discovery

To reduce control traffic, TDR uses GPS-based two-dimensional location information. However, since an idle node keeps only the local neighborhood information, while initiating a session the source node may not have any clue about the location of the destination unless it is a local neighbor, or its location information is cached at the source node among its recently concluded sessions. If the information is available in the source cache, route discovery is performed via selective forwarding. Since the destination's location information in the cache may not be up-to-date (i.e., may be imprecise), the diameter (measured by the number of route request forwarding nodes) of selective broadcast should be larger than that of alternate route search (to be discussed in Section III.B.3). In case of no prior knowledge about the destination, the source initiates flooding-based initial route discovery. To ensure stability of routes and reduce control overhead, only the selected neighbors from where the receive power are more than a threshold level (P_{th1}) are considered for a possible link.

The fields in the initial route discovery control packet are shown in Fig. 1. Description of the fields can be found in Table 2. Each source provides its own *Session_ID*. To reduce the field size, a lowest possible sequence number is picked up, excluding the IDs for the ongoing sessions originated from that node, as a new *Session_ID*.

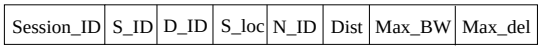

Figure 1: Session initiation route discovery packet structure.

The source (S) checks if it has enough residual bandwidth (*Resi_BW*$_S$) to satisfy the maximum bandwidth[1] requirement (*Max_BW*) for the session. If the demand can be met at S, the required bandwidth is temporarily reserved for a certain lifetime within which it expects to receive the acknowledgment from the destination. The source table ST_S is built with the *Nod_actv* flag still set to '0' (i.e., idle) and the route discovery procedure is initiated. To find a valid route to the destination, a modified breadth first search algorithm is applied abiding by the following rules:

- Upon receiving the first discovery packet for a session, the IN increments the *Dist* tag by one and checks for its residual bandwidth (*Resi_BW*$_{IN}$). If it can meet the maximum bandwidth demand, and the updated *Dist* tag is less than *Max_del* (measured as hop count) , the required bandwidth is temporarily reserved, the activity table IT_{IN} is built with the *Nod_actv* flag '0', and the packet is forwarded to its downstream neighbors with the updated *N_ID* field. If either or both of the *Max_BW* and *Max_del* criteria cannot be satisfied, the discovery packet is simply dropped.

- To ensure loop-free routing, intermediate nodes accept

[1]This is to ensure full QoS support whenever a flow path is ensured. One could instead go for minimum bandwidth criteria for supporting a flexible QoS.

the route discovery packet only once (the one with the minimum *Dist* tag) for a particular session.

- Upon reception of the first discovery packet, if the destination satisfies the *Max_del* requirement (after incrementing the *Dist* tag) and has at least *Max_BW* available, the discovery packet and the corresponding route are accepted. This also ensures the shortest route from the source satisfying the bandwidth and delay criteria.

The concept of temporary reservation of bandwidth in the route discovery phase in TDR is similar to that in resource reservation protocol (RSVP) [6], but differs in implementation. More specifically, unlike in RSVP, to minimize the resource holding, the reservation time in TDR is varied depending on the node's location which is approximately known from the *Max_del* tag and the updated *Dist* tag in the discovery packet. The closer the *Dist* tag to the *Max_del* value, the lesser the reservation time. Let T_d be the current *Dist* tag value at a node and T_M be the *Max_del* requirement for the session. Then the maximum temporary bandwidth reservation time at that node is $2(T_M - T_d)\tau_h$, where τ_h is the maximum time required for a discovery packet to proceed from one node to another which includes packet processing and propagation time.

III.B.2. Route/Reroute Acknowledgment

Once a route is accepted, the destination node builds the DT_D table with the *Nod_actv* flag set to '1' (i.e., active) and initiates a route acknowledgment (ACK) message towards the source along the selected route. On receiving the ACK packet, all intermediate nodes and the source node update the fields in their respective *IT* and *ST* tables (i.e., set their *Nod_actv* flags to '1') and refresh their *Resi_BW* status. Once the logical flow path is set up, the packet transmission for the session can follow immediately. The fields in a route/reroute acknowledgment packet are shown in Fig. 2.

Session_ID	S_ID	D_ID	S_ID/ IN_ID	D_loc	P_ID	Dist	Max_BW	Max_del

Figure 2: Route/reroute acknowledgment packet structure.

Besides acknowledging the route/reroute queries, the destination node also sends its location update to the active nodes via the ACK packet whenever there is appreciable change in its location (based on its own GPS information). This reduces the chance of using stale location information for rerouting purposes.

III.B.3. Alternate Route Discovery

Rerouting a QoS session is necessary when an active node notifies its imminent shut down state or its receive power from its local active neighbor reduces beyond a certain critical limit. In any case, the upstream active node (closer to the source) initiates the rerouting process. We denote this as *link degradation triggered rerouting*[2].

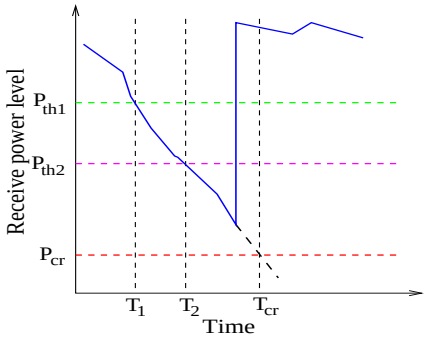

Figure 3: A pictorial representation of rerouting process.

Rerouting process can be either source initiated, called SIRR, or intermediate node initiated, called INIR. An intermediate active node (IN) monitors its downstream receive power level. In SIRR, when the receive power level at an IN decreases to the threshold P_{th2} (see Fig. 3), the IN sends a rerouting indication via a 'status query' packet to the source node with the call identification fields (*Session_ID*, *S_ID*, *D_ID*) and the *RR_flag* set to '1'. Henceforth the source takes control of the rerouting process. This rerouting approach is similar to that in [5], but differs in selective forwarding of route requests.

On the other hand, in INIR, when the downstream receive power level at an IN falls below a threshold P_{th1} with a negative rate of change, it initiates a 'status query' packet towards the source with appropriate call identification fields, filling the *QN_ID* (querying node ID) and *N_ID* fields with its own ID, the *P_ID* field with its previous node in the active route, and with the *RR_stat* flag set to '0'. If any upstream node is in the rerouting process, upon reception of the 'status query' packet it sets the *RR_stat* flag to '1' and returns the packet (as a 'status reply') to the querying node (*QN_ID*). On arrival at the source, the 'status query' packet is discarded (implying that the querying node can initiate the rerouting process). If the query initiating node receives no reply before its power level from the downstream node goes below second threshold, P_{th2}, and further tends to decrease, it triggers the alternate route discovery process. Otherwise, it relinquishes the control of rerouting. This query/reply process eliminates the chance of duplicate reroute discovery for a session. If the downstream receive power at any active intermediate node goes below a critical limit P_{cr}, the source-destination route gets disrupted until the source is able to set up an alternate route. As in hand-off in cellular systems [16], selection of thresholds P_{th1} and P_{th2} have to be judicious so that unnecessary rerouting is avoided and at the same time a successful rerouting is done in case of a genuine link failure. The status query/reply packet structure is shown in Fig. 4.

[2]Other than for increased inter-nodal distance, link degradation can also occur due to channel fading effects caused by the inherent nature of wireless medium. The slow fading problem can be tackled by this scheme. For fast fading, conventional protection mechanism at the data link layer has to be incorporated.

Session_ID	S_ID	D_ID	QN_ID	P_ID	N_ID	RR_stat

Figure 4: Route status query/reply control packet structure.

An example of rerouting due to link degradation in the active route is shown in Fig. 5, where it depicts the INIR. The size of a node indicates the level of bandwidth usage at that node and the thickness of a link denotes the amount of traffic carried along that link (possibly belong to multiple sessions). Since TDR has distributed control, it inherently adopts the INIR scheme. If INIR fails, to avoid/minimize route disruption SIRR is also attempted. It may be noted here that the preemptive routing in [5] follows SIRR. Due to this, and also since it does not use the GPS-based location information, the routing/rerouting control overhead in this approach is expected to be more control overhead intensive.

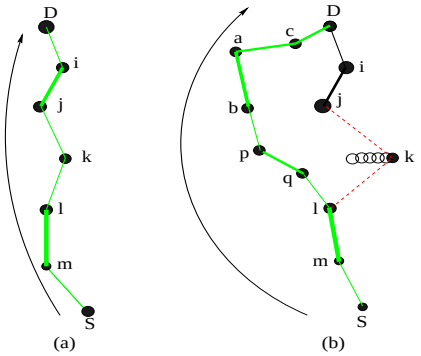

Figure 5: An example of link degradation-based rerouting.

In either rerouting approach (SIRR, INIR), the alternate route discovery packet structure as shown in Fig. 6 can be used. The process is similar to the initial route discovery except that the packet forwarding from a node in this case is done more selectively. Particularly, the rerouting process takes advantage of location information of the local neighbors and the approximate location of the destination, and forwards the rerouting requests to only selected neighbors closest to the destination satisfying the delay and bandwidth constraints.

The members of this selective broadcast group can change due to nodal mobility, network density, and traffic intensity. For highly mobile scenarios, link degradation occurs fast. In such cases as well as due to outdated GPS information, the membership count can be increased to ensure an alternate route at appropriate time.

Session_ID	S_ID	D_ID	S_ID/ IN_ID	D_loc	N_ID	Dist	Max_BW	Max_del

Figure 6: Reroute discovery packet structure.

Note that, LAR [8] uses the location information in a different way. Based on the destination's approximate location, it defines a conical region from the source, and all nodes within the cone are responsible in forwarding the route query. In case of route search failure, the cone angle is expanded. Clearly, depending on nodal density, latency in route search in this approach can vary widely. Route searching control overhead in LAR is also a function of nodal density. In contrast, our local neighborhood information based selective forwarding approach does not have this dependency.

III.B.4. Route Deactivation

When a session is either finished, terminated, or rerouted, the old route has to be released. In case of a session completion or termination, the source node purges its corresponding *ST* table and sends a route deactivation packet through the old route to the destination. The packet structure is shown in Fig. 7. Upon receiving a route deactivation packet, a node updates its *Resi_BW* (by releasing the reserved bandwidth) purges the activity database (*IT* or *DT*) for that session. No explicit deactivation packet is sent in case of rerouting, as the new route could consist of some old active nodes. The departed nodes refresh their activity databases and residual bandwidths after a certain fixed 'soft state' interval (as in E-AODV [15] or RSVP [6]). Also, if for some reason (e.g., fast link failure) an old route could not be released, the associated nodes refresh their *Resi_BW* and clears their respective activity-based tables after a fixed 'soft state' interval.

Session_ID	S_ID	D_ID	S_ID / IN_ID	N_ID

Figure 7: Route deactivation packet structure.

IV. Performance Analysis

In this section, we analyze the effect of selective forwarding on rerouting success.

IV.A. Average Number of Neighbors

Consider N nodes distributed uniformly over a mobility space of area A. The approximate number of neighbors (n_g) of a node is given by

$$\begin{aligned} n_g &\approx \sum_{i=1}^{N-1} i \times \Pr\{\text{the node has } i \text{ neighbors}\} \\ &= \sum_{i=1}^{N-1} i C_i^{N-1} \left(\frac{a}{A}\right)^i \left(1-\frac{a}{A}\right)^{N-1-i} \end{aligned} \quad (1)$$

where a is the coverage area of a mobile node (considered equal for all nodes).

Note that Eq. (1) does not consider the 'boundary effects' where the nodes near the boundaries will have lesser region covered within the rectangle and hence there would be less than the predicted number of nodes around them. However, the error in the estimate (without considering the 'boundary effects') and in subsequent analysis would be negligibly small for smaller R, larger N, and larger A.

IV.B. Route Search Failure Probability

We begin with the case of only one forwarding neighbor. Subsequently, we obtain the failure probability for more relaxed cases, with more than one forwarding neighbors. In selecting one or more forwarding neighbors of a node out of its all neighbors, it is assumed that the ones closest to the destination (based upon the GPS information) qualify first. It may be recalled that the purpose of selective route request forwarding is to reduce the control traffic without sacrificing the QoS of the application.

In general, a rerouting request can fail if

1) the requesting node has only one neighbor (which is the upstream node from where rerouting request has been received), or,

2) the requesting node has more than one neighbors, however, all but the upstream neighbor are busy and hence unable to take the request from that node.

A node can be found busy either due to its lack of sufficient residual bandwidth or because it has been booked earlier by a routing request for the same session. Note that the case where a node has no neighbor is excluded because it implies occurrence of a network partition. As will be seen in Section V, appropriate measures are taken in simulation experiments to ensure that network partition does not occur during runtime.

Case 1: *Only one forwarding neighbor :*
In this case a k-hop route discovery can fail at any stage with equal probability. The probability of route search failure at any stage is given by

$$\begin{aligned} P(1) &= \Pr\{\text{a node has only one neighbor}\} \\ &\quad + \sum_{i=2}^{N-1} \Pr\{\text{a node has } i \text{ neighbors}\} \\ &\quad \times \Pr\{\text{only the upstream node is free}\} \end{aligned} \tag{2}$$

Considering that a node can serve one call at a time, if the call arrival at each node is a Poisson process with rate λ and the average call holding time is $\overline{x}$, then

$$\Pr\{\text{a node is busy acting as a source}\},\ P_s = \lambda\overline{x}$$

Considering equiprobable source-destination pairs, for a call from any other node, a node can act as a destination with probability $\frac{1}{N-1}$. There are $N-1$ such potential nodes that could choose it as a destination. Therefore,

$$\Pr\{\text{a node is busy acting as a destination}\},\ P_d = \lambda\overline{x}$$

Also, for a call from any other node, a node can act as an IN. If the average route length is h-hop long, there would be on average $(h-1)$ nodes acting as INs for each call. Hence,

$$\Pr\{\text{a node is busy acting as an IN}\},\ P_r = (h-1)\lambda\overline{x}$$

Summing up all these, $\Pr\{\text{a node is busy}\}$, $P_B = P_s + P_d + P_r$ is given by

$$P_B = (h+1)\lambda\overline{x} \tag{3}$$

Assuming each session takes a fixed (same) amount of bandwidth, if a node can support c such real-time sessions simultaneously, then Eq. (3) will be modified as $P_B = (h+1)\left(\frac{\lambda\overline{x}}{c}\right)$. In any case, as a stability criteria the values of λ, h, c, and $\overline{x}$ should be able to satisfy the condition $P_B < 1$.

Continuing with Eq. (3), we have

$$\begin{aligned} &\Pr\{\text{only the upstream node out of } i \text{ neighbors is free}\} \\ &\qquad = (1-(h+1)\lambda\overline{x})\left((h+1)\lambda\overline{x}\right)^{i-1} \end{aligned} \tag{4}$$

Substituting Eq. (4) in Eq. (2) and using Eq. (1), we obtain

$$\begin{aligned} P(1) &= (N-1)\left(\frac{a}{A}\right)\left(1-\frac{a}{A}\right)^{N-2} \\ &\quad + \sum_{i=2}^{N-1} C_i^{N-1}\left(\frac{a}{A}\right)^i\left(1-\frac{a}{A}\right)^{N-1-i} \\ &\quad \times (1-(h+1)\lambda\overline{x})\left((h+1)\lambda\overline{x}\right)^{i-1} \end{aligned} \tag{5}$$

Hence, the overall k-hop route search failure probability with only one forwarding node is obtained as

$$P^{(k)}(1) = \sum_{i=0}^{k-1}\left[1-P(1)\right]^i P(1) \tag{6}$$

Case 2: *More than one forwarding neighbors :*
Consider the route search failure probability with maximum two forwarding nodes. An example of reroute discovery with maximum two forwarding nodes at each stage is shown in Fig. 8. Note that at the last hop to the destination only one forwarding path is shown. In our simulation, route forwarding process stops immediately after a successful route is obtained. In practice, an upper limit of *Max_del* can be used to stop forwarding the routing request after a certain expected number of hops.

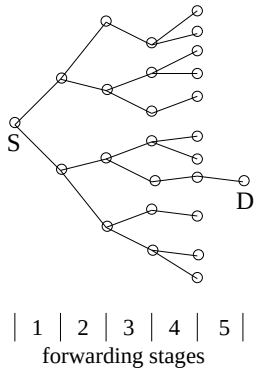

Figure 8: An example of route request branching process with maximum two forwarding nodes.

Probability of route search failure at stage 1 is

$$P^{(1)}(2) = \left[P(1)\right]^2$$

where in $P^{(i)}(j)$, i indicates the failure stage and j indicates the maximum number of forwarding neighbors, and $P(1)$ is given by Eq. (5).

Considering one stage further, probability of failure within up to stage 2 is

$$P^{(2)}(2) = P^{(1)}(2) + C_1^2 \overline{P}(1) \left[P(1)\right]^3 + C_2^2 \left[\overline{P}(1)\right]^2 \left[P(1)\right]^4$$

where $\overline{P}(1) = 1 - P(1)$.

Likewise, probability of failure within up to 3 stages is given by

$$\begin{aligned} P^{(3)}(2) &= P^{(2)}(2) + C_1^4 \left[\overline{P}(1)\right]^2 \left[P(1)\right]^4 \\ &\quad + C_2^4 \left[\overline{P}(1)\right]^3 \left[P(1)\right]^5 + C_3^4 \left[\overline{P}(1)\right]^5 \left[P(1)\right]^7 \\ &\quad + C_4^4 \left[\overline{P}(1)\right]^6 \left[P(1)\right]^8 \end{aligned}$$

Similarly, route search failure at higher stages can be obtained.

Note that in the expression for $P^{(2)}(2)$, the dominating term is $P^{(1)}(2)$. The other terms are lesser than the first term by a few orders of magnitude. For example, if $P(1) = 10^{-2}$, $P^{(1)}(2) = 10^{-4}$ and the second term of $P^{(2)}(2)$ is $\sim 10^{-6}$. Similar is the case for $P^{(3)}(2)$, where $P^{(2)}(2)$ dominates. In other words, route search failure with maximum two forwarding nodes is dominated by the failure at the first stage. Therefore, a k-hop route search failure probability with maximum two forwarding nodes can be approximated as

$$P^{(k)}(2) \triangleq P(2) \approx \left[P(1)\right]^2 \tag{7}$$

With the same argument, in general, route search failure for the case of maximum i ($i > 1$) forwarding nodes will be

$$P(i) \approx \left[P(1)\right]^i \tag{8}$$

From Eq. (8) it can be noted that for reasonably small $P(1)$, say, $P(1) = 10^{-2}$, practically not much gain in terms of route search success rate is achieved beyond maximum two forwarding nodes.

Table 3: Route search failure probability as a function of traffic intensity and nodal density. Area of mobility space, $A = 1500 \times 1000\ m^2$; range of of circular coverage of an MH, $R = 300\ m$; number of hops, $k = 3$.

$N = 60$			$P_B = 0.6$		
P_B	$P^{(k)}(1)$	$P(2)$	N	$P^{(k)}(1)$	$P(2)$
0.1	3×10^{-4}	10^{-8}	20	0.2816	0.0109
0.3	0.0015	2.5×10^{-7}	40	0.0876	9.4×10^{-4}
0.5	0.0087	8.4×10^{-6}	60	0.0194	4.2×10^{-5}
0.7	0.0408	1.9×10^{-4}	80	0.0042	2×10^{-6}
0.9	0.1044	0.0013	100	9×10^{-4}	9×10^{-8}

Route search failure probability as a function of traffic intensity and nodal density is shown in Table 3. Observe that at higher traffic intensity and for lower nodal density, maximum number of reroute request forwarding nodes has to be increased for better rerouting success.

V. Simulation Studies

We evaluate the performance of the proposed TDR protocol via C-based discrete event simulation. In the simulation model, we are primarily interested in studying the *effect of mobility* on selective forwarding and prediction-based distributed routing, and comparing them with broadcast-based as well as reactive routing schemes. For simplicity, channel fading effects are not included in our current simulation, which will affect all the routing schemes discussed here, but not the general performance trends.

As observed in [19] and [20], the mobile hosts (also called users or nodes) are assumed 'well behaved' such that their movement patterns are not completely random. In our simulation a node's average velocity in an epoch[3] is constant along a specific direction. At the end of an epoch, the velocity and movement direction of the node randomly changes only within certain limits. To trigger a reroute search, in addition to the current receive power (based on relative distance), we take into account the rate of change of receive power. This is to ensure some priority to the active nodes with degrading link condition [4]. Only when the current receive power is below a predefined lower threshold and its rate of change is negative, the reroute discovery process is initiated.

The following assumptions on network condition are made in the simulation:

- Poisson arrival process;
- exponentially distributed session duration;
- uniformly distributed mobility;
- equiprobable source-destination pairs;
- a node can handle more than one session simultaneously;
- only real-time applications served.

Since only real-time sessions are considered, an in-session data flow is always along a pre-set path. Because of this, in-session MAC conflict is assumed non-existent. It is also ensured that no network partition occurs during run time. Since the fading channel effects are not included, the receive power is considered in terms of equivalent internodal distance. The simulation parameters are listed in Table 4. Based on the above assumptions and parameter values, we study the network performance with the proposed TDR protocol and compare it with three existing QoS routing protocols (e.g., FORP, DQoSR, and E-AODV).

In evaluating and comparing the TDR protocol performance, it is assumed that in case of resource unavailability an attempted session could be either lost (loss model) or delayed (delay model). In the loss model, the session acceptance performance is measured by *grade of service* (GoS) which is the ratio of number of sessions lost to the number of attempts. In the delay model, the session acceptance performance is measured by *queueing delay* which is the average waiting time of an attempted session in the input buffer before it is accepted.

Because the network topology and mobility pattern vary widely for different SEED values, for each protocol we simulate six different scenarios for each average velocity.

[3] An epoch is specified by the session interarrival time in the network.

Table 4: Simulation parameters

Parameters	Values
Location area (A)	$1500 \times 1000\ m^2$
Default number of nodes (N)	60
Coverage range of a mobile host (equal) (R)	$300\ m$
End threshold distance (Th_2)	$270\ m$
Average velocity (V_{avg})	1 *m/s* to 10 *m/s*
Maximum velocity change per epoch	10% of average
Maximum direction change per epoch	90°
Maximum data handling capacity of a node	$10\ kbps$
Maximum data rate per session (uniform RV)	$2\ kbps$
Average session interarrival time per node ($1/\lambda$)	$6\ min$
Default average session duration ($\overline{x}$)	3 *min*
Average epoch length	6 *s*
Default maximum reroute request forwarding nodes	2
Total number of sessions attempted per run	20,000

V.A. TDR Protocol Evaluation

In this subsection, performance of the proposed TDR protocol is studied in terms of grade of service (GoS).

Fig. 9 shows the respective effective control overhead plots for full broadcast-based and selective forwarding-based route discovery processes. Effective control overhead is defined as the average number of rerouting packets generated per successful session. Observe that the selective forwarding-based alternate route search has much less overhead and slower increment rate with mobility compared to that in the broadcast-based approach.

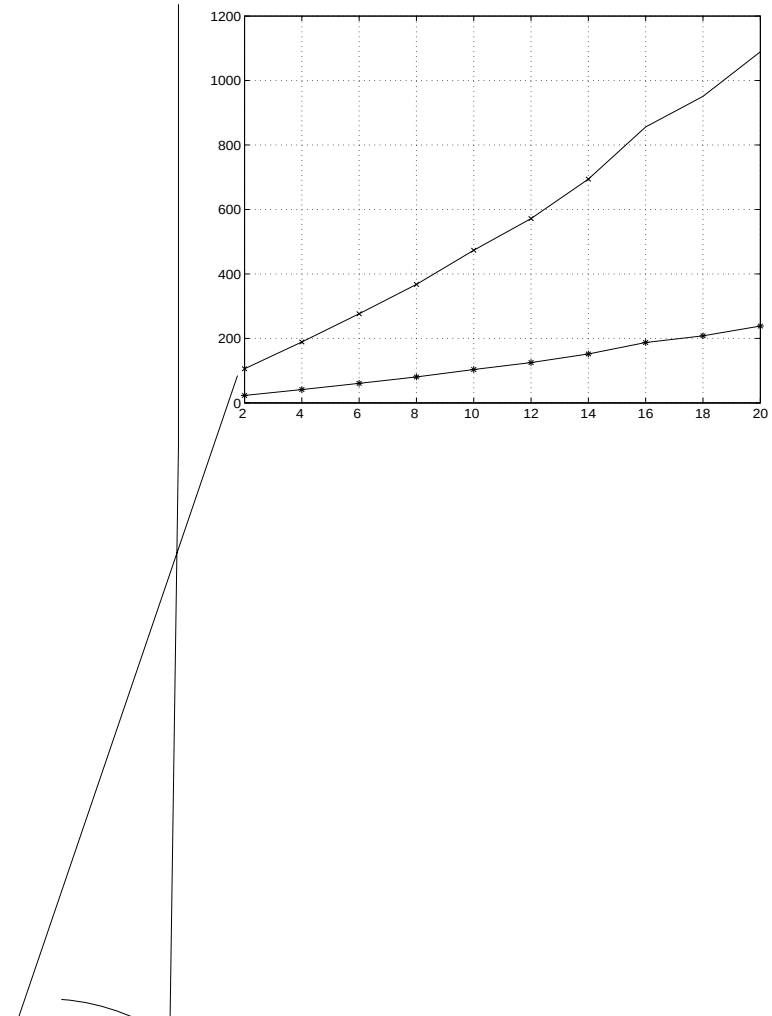

Table 5: Dependence of maximum number of reroute request forwarding nodes (*MaxRR*) on traffic intensity and nodal density. QoSR: QoS ratio; ROH: rerouting control overhead. Maximum velocity 20 *m/s*; $A = 1500 \times 1000\ m^2$; $\frac{1}{\lambda} = 6$ *min*.

N=100				$\overline{x} = 3$ *min*			
$\overline{x}$	MaxRR	QoSR	ROH	N	MaxRR	QoSR	ROH
2	1	0.997	54	50	1	0.956	101
	2	0.998	158		2	0.964	242
	3	0.998	289		3	0.964	304
10	1	0.937	1370	100	1	0.994	112
	2	0.946	2757		2	0.995	291
	3	0.947	3373		3	0.995	483

mance (GoS, QoS ratio) can be achieved is shown in Table 5. Traffic intensity is varied by changing the session duration ($\overline{x}$) while the session arrival rate is kept constant. In agreement with the analytic data (Table 3), we note that as the traffic intensity increases, higher value of *MaxRR* is required to achieve better system performance. For example, at $\overline{x} = 2$, *MaxRR*=1 nearly achieves the best possible system performance, whereas at $\overline{x} = 10$, *MaxRR* value should be at least 2. Similarly, for higher nodal density, lesser value of *MaxRR* can achieve nearly the best possible system performance. The degradation of overall performance (lesser QoS ratio and more control overhead) can also be noted at higher traffic intensities and for lower nodal densities.

V.B. Comparison Results

In the following discussions, comparative performance results of TDR with respect to FORP, DQoSR, and E-AODV are presented. In comparing protocol performances, once a session is successfully initiated, it is not dropped prematurely even if there is intermittent route failure. The packets during the route failure intervals are dropped. Protocol performance in such cases are measured in terms of *QoS ratio* which is defined as the fractional successful packet transmissions per session, or alternatively, packet dropping probability. Note that since in all protocols neighborhood/network information is maintained by periodic beaconing, this common overhead is not taken into account for comparison of control overhead; rather only the rerouting overheads are considered.

In FORP [17], only one active route is maintained. Based on the predicted route failure time, the destination initiates *broadcast-based* alternate route discovery up to the source. From rerouting control point of view, this scheme is similar to TDR with SIRR.

In simulating DQoSR protocol [3] we go for up to two disjoint routes (the primary and one secondary). A session is accepted even if only one (primary) route could be secured. At any stage, if a session has only the primary route, the source tries for a secondary route at every status update epoch. In case of primary route failure, if there is a secondary route available, then it immediately takes over the session and is treated as the current primary route. There is no QoS degradation in this case. On the other hand, during primary route failure, if no secondary route exists, the packets are dropped as long as the route failure persists.

In E-AODV protocol [15], only the active routes are maintained (soft state concept). No attempt is made to maintain the source-to-destination logical connection. If the route fails, *broadcast-based* route discovery process is re-initiated. The packets during the route failure intervals are dropped.

We provide the comparative performance results of the these four protocols (TDR, FORP, DQoSR, and E-AODV) with blocked call delayed assumption. The results for loss model are not shown as they follow similar trends.

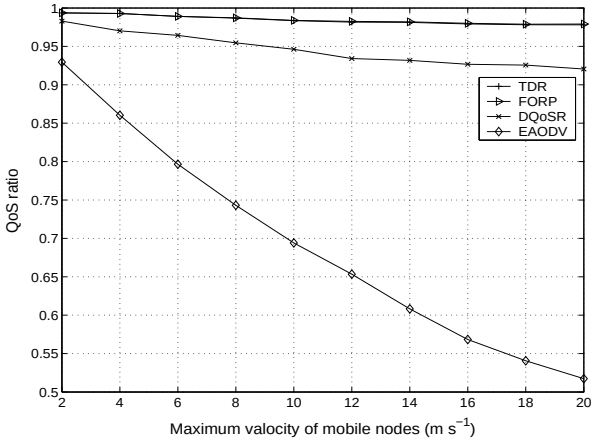

Figure 12: Variation of QoS ratio with mobility.

Fig. 12 shows the QoS performances of different protocols, where it is observed that the E-AODV performs poorly at higher velocity as it has neither route prediction capability nor does it maintain alternate routes. TDR and FORP perform nearly the same as both the protocols operate under the same prediction capability. DQoSR performs a little poorer than TDR and FORP, since it has to allocate more resources to support the ongoing sessions and also becuase of its reactive nature.

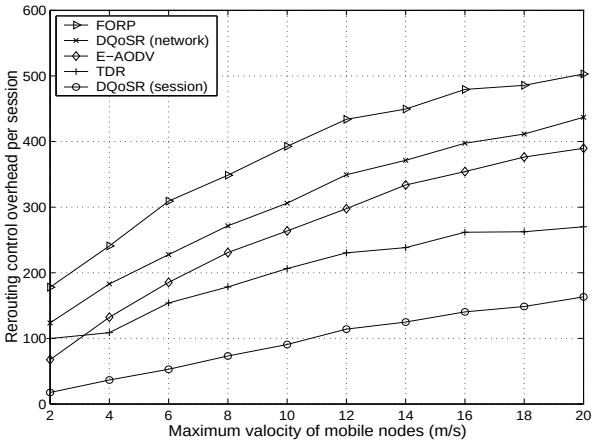

Figure 13: Rerouting control overhead at different mobility.

Fig. 13 shows the average control overhead per session (average number of rerouting control packets generated per

successful session) associated with the protocols. Since DQoSR maintains secondary resources for the on-going sessions, the sessions experience the minimum overhead, but the total overhead experienced by the network is much higher. Note that DQoSR has an additional nodal database overhead for maintaining network-wide delay and bandwidth information, which is not captured in our simulation. E-AODV has higher control overhead than that seen by a session in DQoSR because in this case every time the route fails, the session is interrupted and it (E-AODV) has to immediately start a reroute discovery process. Distributed rerouting control and selective forwarding based route discovery causes lesser rerouting overhead in TDR than that in FORP, which adopts localized control and broadcast-based route discovery. Although both FORP and E-AODV follow broadcast-based route search, FORP being proactive protocol it requires more frequent invocation of rerouting routine, leading to higher overhead compared to E-AODV.

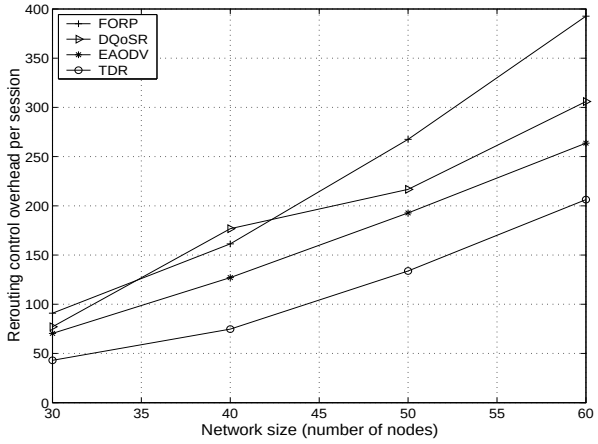

Figure 14: Rerouting control overhead versus network size. Maximum velocity 10 *m/s*.

Variation of average rerouting control overhead per session with network size (for nearly the same nodal density, by varying the area of mobility space with the number of nodes) is shown in Fig. 14. Call arrival rate at each node is kept constant for different network size. Obvious general trend is that the average route length increases with increase in network size, causing increase in route maintenance overhead. It also shows that TDR has low rate of overhead increment. Although FORP maintains only the active route, its broadcast-based route discovery causes higher overhead increment rate. The control overhead in DQoSR is lower than the case of FORP, as the route discovery is controlled by number of tickets. Having poor QoS support in E-AODV, its overall control overhead is also low and increment is slower.

The preemptive routing approach in [5] was not explicitly considered for comparison as it is basically similar to FORP. More specifically, both of these protocols follow end node controlled rerouting (FORP is destination controlled whereas preemptive routing is source controlled) and both of them do not take location advantage in the alternate route discovery process.

VI. Conclusions

In this paper, we have presented a routing scheme called trigger-based distributed routing (TDR) for supporting RT-QoS traffic in mobile ad hoc networks. The proposed TDR scheme has failure prediction-based alternate route discovery and avoids maintenance of additional routes. This reduces control traffic as well as the size of nodal database. In addition, TDR makes use of selective forwarding of routing requests based on GPS information, and as a result its route discovery overhead is further reduced. As an added cost, this protocol requires some extra nodal computation for selecting appropriate nodes to forward route requests.

The effect of selective forwarding on the rerouting success is quantified via analysis. The TDR protocol performance has been studied and compared with the existing QoS protocols for ad hoc networks such as FORP, DQoSR, and E-AODV, via simulations. Significant superiority in the QoS performance of 'prediction-based' TDR over these 'prediction-less' QoS routing protocols (E-AODV, DQoSR) have been noted. Although both TDR and FORP are 'prediction-based' protocols, and have a comparable QoS performance in terms of queueing delay and QoS ratio, TDR is more scalable because of its distributed control and selective forwarding-based rerouting.

In the simulation, to ensure full QoS support whenever logical flow paths were available, resource reservations were done on maximum bandwidth demand for a session. This model can be extended to study the QoS performance based on the minimum bandwidth demand (for flexible QoS support) and with heterogeneous traffic. In such cases, however, even if a flow path exists, there can be QoS degradation in terms of QoS ratio and end-to-end delay variation due to burstiness of packet arrivals. Our simulated model did not take fading channel effects into account as we were primarily interested in studying the benefit of proactive and selective forwarding-based rerouting over comparing them with the reactive and broadcast-based rerouting strategies. Since fading channel will affect the performance of all the protocols, we expect that the trends of performance results will remain valid.

References

[1] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," in *Proc. of ACM MobiCom*, 1998.

[2] T.-W. Chen and M. Gerla, "Global State Routing: A New Routing Scheme for Ad-hoc Wireless Networks," in *Proc. of IEEE ICC*, 1998, pp. 171-175.

[3] S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in Ad Hoc Networks," *IEEE J. Sel. Areas in Comm.*, vol. 17(8), pp. 1488-1505, Aug. 1999.

[4] H. G. Ebersman and O. K. Tonguz, "Handoff ordering using signal prediction priority queueing in personal communication systems," *IEEE Trans. Vehicular Tech.*, vol. 48(1), pp. 20-35, Jan. 1999.

[5] T. Goff, N. Abu-Ghazaleh, D. Pathak, and R. Kahvecioglu, "Preemptive Routing in Ad Hoc Networks," in *Proc. of ACM MobiCom*, 2001.

[6] *http://www.ietf.org/rfc/rfc2205.txt*, Sep. 1997.

[7] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, Ed. T. Imielinski and H. Korth, Ch. 5, pp. 153-181, Kluwer Academic Publishers, 1996.

[8] Y.-B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," in *Proc. of ACM MobiCom*, 1998.

[9] S.-B. Lee, G.-S. Ahn, X. Zhang, and A. T. Campbell, "INSIGNIA: An IP-Based Quality of Service Framework for Mobile ad Hoc Networks," *J. Parallel and Distributed Comp.*, vol. 60, pp. 374-406, 2000.

[10] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *Mobile Networks and Appl.*, vol. 1(2), pp. 183-197, Oct. 1996.

[11] V. Park and M.S. Corson, "Temporally-Ordered Routing Algorithm (TORA) Version 1 Functional Specification," *IETF Internet draft*, `draft-ietf-ma-net-tora-spec-00.txt`, Dec 1997.

[12] M. R. Pearlman and Z. J. Haas, "Determining the Optimal Configuration for the Zone Routing Protocol," *IEEE J. Sel. Areas in Comm.*, vol. 17(8), Aug. 1999.

[13] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," in *Proc. of ACM SIGCOMM*, pp. 234-244, Aug. 1994.

[14] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90-100, Feb. 1999.

[15] C. E. Perkins, E. M. Royer, and S. R. Das, "Quality of Service for Ad Hoc On-Demand Distance Vector Routing," *IETF Internet draft*, `draft-ietf-manet-aodvqos-00.txt`, July 2000.

[16] T. Rappaport, *Wireless Communications: Principles and Practice*. Prentice Hall, 1996.

[17] W. Su and M. Gerla, "IPv6 Flow Handoff in Ad Hoc Wireless Networks Using Mobility Prediction," in *Proc. of IEEE GLOBECOM*, 1999.

[18] C.-K. Toh, "Associativity-Based Routing For Ad Hoc Mobile Networks," *Wireless Personal Comm. Journal*, vol. 4(2), Mar. 1997.

[19] K. Yeung and S. Nanda, "Channel Management in Microcell/Macrocell Cellular Radio Systems," *IEEE Trans. Vehicular Tech.*, vol. 45(4), pp. 601-612, Sep. 1996.

[20] M. Zonoozi and P. Dassanayake, "User Mobility Modeling and Characterization of Mobility Patterns," *IEEE J. Sel. Areas in Comm.*, vol. 15(7), pp. 1239-1252, Sep. 1997.

Biography

Swades De received his B.Tech degree in Radiophysics and Electronics from University of Calcutta in 1993 and his M.Tech degree in Optoelectronics and Optical Communication from Indian Institute of Technology Delhi in 1998. During 1993-1997 he was a hardware development engineer and in the first half of 1999 he was a software engineer in different telecommunication companies in India. He is a Ph.D candidate in the Electrical Engineering Department, State University of New York at Buffalo. His current research interests include performance study, QoS routing and resource optimization in mobile ad hoc networks and wireless sensor networks, integrated wireless technologies, dynamic routing in high-speed networks, and communication and systems issues in optical networks.

Sajal K. Das received the Ph.D degree in Computer Science in 1988 from the University of Central Florida, Orlando. Currently, he is a full professor of Computer Science and Engineering and the founding director of the Center for Research in Wireless Mobility and Networking CReWMaN) at the University of Texas at Arlington (UTA). Prior to 1999, he was a professor of Computer Science at the University of North Texas (UNT), Denton, where he founded the Center for Research in Wireless Computing (CReW) in 1997 and served as the director of the Center for Research in Parallel and Distributed Computing (CRPDC) during 1995-1997. He is a recipient of the UNT Student Association's Honor Professor Award in 1991 and 1997 for best teaching and scholarly research, UNT's Developing Scholars Award in 1996 for outstanding research, and UTA's Outstanding Senior Faculty Research Award in Computer Science in 2001. He has visited numerous universities, research organizations, and industry research labs for collaborative research and invited seminar talks. He was a visiting scientist at the Council of National Research in Pisa, Italy, and Slovak Academy of Sciences in Bratislava, and was also a visiting professor at the Indian Statistical Institute, Calcutta. He is frequently invited as a keynote speaker at international conferences and symposia. His current research interests include resource and mobility management in wireless networks, mobile computing, QoS provisioning and wireless multimedia, mobile Internet, network architectures and protocols, distributed/parallel processing, performance modeling, and simulation. He has published more than 185 research papers in these areas, directed several projects funded by industry and government, and filed four US patents in wireless mobile networks. He received the Best Paper Awards for significant research contributions at the ACM Fifth International Conference on Mobile Computing and Networking (MobiCom'99), the Third ACM International Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM 2000), and the ACM/IEEE International Workshop on Parallel and Distributed Simulation (PADS'97). He serves on the editorial boards of the *Journal of Parallel and Distributed Computing*, *Parallel Processing Letters*, *Journal of Parallel Algorithms and Applications*, and *Computer Networks*. He serves on numerous

IEEE and ACM conferences as a technical program committee member, program chair, or general chair. He is a member of the IEEE TCPP Executive Committee and advisory boards of several cutting-edge companies. He a member of the IEEE and the IEEE Computer Society.

Hongyi Wu is currently a tenure-track Assistant Professor at the Center for Advanced Computer Studies (CACS), University of Louisiana (UL) at Lafayette. He finished his Ph.D degree in the Department of Computer Science and Engineering at State University of New York (SUNY) at Buffalo in 2002, and received his M.S. degree from SUNY at Buffalo in 2000 and B.S. degree from Zhejiang University in 1996, respectively. He worked in Nokia Research Center in the summer of 2001 and 2000. His research interests include wireless mobile ad hoc networks, 2G/3G cellular systems, integrated heterogeneous wireless systems, routing protocols, and the Internet. He has published more than a dozen technical papers in leading journals and conference proceedings, as well as a book chapter.

Chunming Qiao earned his B.S. in Computer Engineering from University of Science and Technology of China. He received the Andrew-Mellon Distinguished doctoral fellowship award, and subsequently earned his Ph.D in Computer Science from University of Pittsburgh. Dr. Qiao is currently an Associate Professor at the Computer and Science Engineering Department, University at Buffalo (SUNY), where he directs the Lab for Advanced Network Design, Evaluation and Research (LANDER), and conducts cutting-edge research related to optical networks, wireless/mobile networks, and the Internet. His research in optical networks and mobile/wireless networks has been supported by a number of NSF grants including an Research Initiation Award (IRA) and Information Technology Research (ITR) award, and by Alcatel USA, Nokia Research Center, Nortel Networks, and Telcordia.

Dr. Qiao has published more than one hundred papers in leading technical journals and conference proceedings, authored several book chapters, and given several keynote speeches, tutorials and invited talks. His contributions to the next generation Optical Internet, and in particular, his pioneering research on optical burst switching (OBS) are internationally acclaimed. His research on integrated cellular and ad hoc relaying systems (iCAR) has been featured in magazines such as BusinessWeeks and Wireless Europe. He has filed several patent applications on these subjects.

Dr. Qiao is the IEEE Communication Society's Editor-at-Large for optical networking and computing, an editor of several journals and magazines including IEEE/ACM Transactions on Networking (ToN), as well as guest editor for two JSAC issues. Dr. Qiao has chaired or co-chaired several conferences and workshops including the Symposium on Optical Networking at ICC'03, Opticomm'02, and the High Speed Network Workshops (formerly GBN) at Infocom 2002 and 2001. He is also the founding chair of the Technical Group on Optical Networks (TGON) sponsored by SPIE, and a Vice Chair of the IEEE Technical Committee on Gigabit Networking (TCGN).