A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks

Elizabeth M. Royer
Dept. of Electrical & Computer Engineering
University of California, Santa Barbara
eroyer@alpha.ece.ucsb.edu

C-K Toh
Dept. of Electrical & Computer Engineering
Georgia Institute of Technology, Atlanta
cktoh@ee.gatech.edu

Abstract

An ad-hoc mobile network is a collection of mobile nodes that are dynamically and arbitrarily located in such a manner that the interconnections between nodes are capable of changing on a continual basis. In order to facilitate communication within the network, a routing protocol is used to discover routes between nodes. The primary goal of such an ad-hoc network routing protocol is correct and efficient route establishment between a pair of nodes so that messages may be delivered in a timely manner. Route construction should be done with a minimum of overhead and bandwidth consumption. This paper examines routing protocols for ad-hoc networks and evaluates these protocols based on a given set of parameters. The paper provides an overview of eight different protocols by presenting their characteristics and functionality, and then provides a comparison and discussion of their respective merits and drawbacks.

1 Introduction

Since their emergence in the 1970s, wireless networks have become increasingly popular in the computing industry. This is particularly true within the past decade which has seen wireless networks being adapted to enable mobility. There are currently two variations of mobile wireless networks. The first is known as infrastructured networks, i.e., those networks with fixed and wired gateways. The bridges for these networks are known as base stations. A mobile unit within these networks connects to, and communicates with, the nearest base station that is within its communication radius. As the mobile travels out of range of one base station and into the range of another, a "handoff" occurs from the old base station to the new, and the mobile is able to continue communication seamlessly throughout the network. Typical applications of this type of network include office wireless local area networks (WLANs).

The second type of mobile wireless network is the infrastructureless mobile network, commonly known as an ad-hoc network. Infrastructureless networks have no fixed routers; all nodes are capable of movement and can be connected dynamically in an arbitrary manner. Nodes of these networks function as routers which discover and maintain routes to other nodes in the network. Example applications of ad-hoc networks are emergency search-and-rescue operations, meetings or conventions in which persons wish to quickly share information, and data acquisition operations in inhospitable terrains.

This paper examines routing protocols designed for these ad-hoc networks by first describing the operation of each of the protocols and then comparing their various characteristics. The remainder of the paper is organized as follows: Section 2 presents a discussion of two subdivisions of ad-hoc routing protocols. Subsection 2.1 discusses current table-driven protocols, while Subsection 2.2 describes

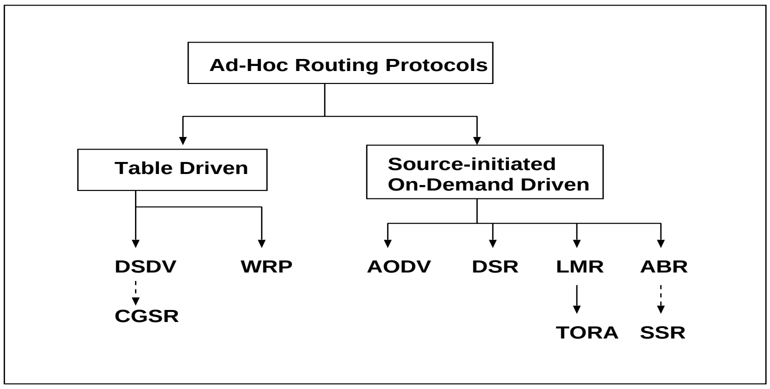


Figure 1: Categorization of Ad-Hoc Routing Protocols.

those protocols which are classified as on-demand. Section 3 presents qualitative comparisons of table-driven protocols, followed by on-demand-driven protocols, and finally a general comparison of table-driven and on-demand protocols. Applications and challenges facing ad-hoc mobile wireless networks are discussed in Section 4. Finally, Section 5 concludes the paper.

2 Existing Ad-Hoc Routing Protocols

Since the advent of DARPA packet radio networks in the early 1970s [11], numerous protocols have been developed for ad-hoc mobile networks. Such protocols must deal with the typical limitations of these networks, which include high power consumption, low bandwidth, and high error rates. As shown in Figure 1, these routing protocols may generally be categorized as: (a) table-driven and (b) source-initiated on-demand driven. Solid lines in this figure represent direct descendants while dotted lines depict logical descendants. Despite being designed for the same type of underlying network, the characteristics of each of these protocols are quite distinct. The following sections describe the protocols and categorize them according to their characteristics.

2.1 Table-Driven Routing Protocols

The table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every other node in the network. These protocols require each node to maintain one or more tables to store routing information, and they respond to changes in network topology by propagating updates throughout the network in order to maintain a consistent network view. The areas where they differ are the number of necessary routing-related tables and the methods by which changes in network structure are broadcast. The following sections discuss some of the existing table-driven ad-hoc routing protocols.

2.1.1 Destination-Sequenced Distance-Vector Routing (DSDV)

The Destination-Sequenced Distance-Vector Routing protocol (DSDV) described in [17] is a table-driven algorithm based on the classical Bellman-Ford routing mechanism [7]. The improvements made to the Bellman-Ford algorithm include freedom from loops in routing tables.

Every mobile node in the network maintains a routing table in which all of the possible destinations within the network and the number of hops to each destination are recorded. Each entry is marked with a sequence number assigned by the destination node. The sequence numbers enable the mobile nodes to distinguish stale routes from new ones, thereby avoiding the formation of routing loops. Routing table updates are periodically transmitted throughout the network in order to maintain table consistency. To help alleviate the potentially large amount of network traffic that such updates can generate, route updates can employ two possible types of packets. The first is known as a "full dump." This type of packet carries all available routing information and can require multiple network protocol data units (NPDUs). During periods of occasional movement, these packets are transmitted infrequently. Smaller "incremental" packets are used to relay only that information which has changed since the last full dump. Each of these broadcasts should fit into a standard size NPDU, thereby decreasing the amount of traffic generated. The mobile nodes maintain an additional table where they store the data sent in the incremental routing information packets.

New route broadcasts contain the address of the destination, the number of hops to reach the destination, the sequence number of the information received regarding the destination, as well as a new sequence number unique to the broadcast [17]. The route labeled with the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric is used in order to optimize (shorten) the path. Mobiles also keep track of the settling time of routes, or the weighted average time that routes to a destination will fluctuate before the route with the best metric is received (see [17]). By delaying the broadcast of a routing update by the length of the settling time, mobiles can reduce network traffic and optimize routes by eliminating those broadcasts that would occur if a better route was discovered in the very near future.

2.1.2 Clusterhead Gateway Switch Routing (CGSR)

The Clusterhead Gateway Switch Routing (CGSR) protocol differs from the previous protocol in the type of addressing and network organization scheme employed. Instead of a "flat" network, CGSR is a clustered multihop mobile wireless network with several heuristic routing schemes [4]. The authors state that by having a cluster head controlling a group of ad-hoc nodes, a framework for code separation (among clusters), channel access, routing and bandwidth allocation can be achieved. A cluster head selection algorithm is utilized to elect a node as the cluster head using a distributed algorithm within the cluster. The disadvantage of having a cluster head scheme is that frequent cluster head changes can adversely affect routing protocol performance since nodes are busy in cluster head selection rather than packet relaying. Hence, instead of invoking cluster head reselection every time the cluster membership changes, a Least Cluster Change (LCC) clustering algorithm is introduced. Using LCC, cluster heads only change when two cluster heads come into contact, or when a node moves out of contact of all other cluster heads.

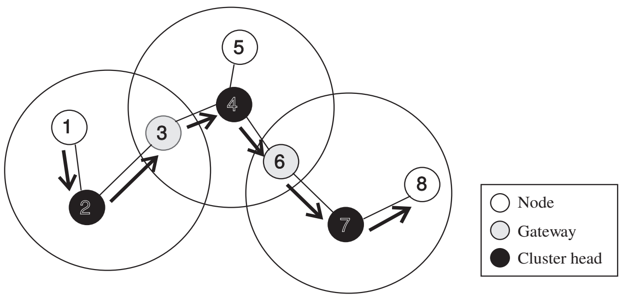


Figure 2: CGSR: Routing from Node 1 to Node 8.

CGSR uses DSDV as the underlying routing scheme, and hence has much of the same overhead as DSDV. However, it modifies DSDV by using a hierarchical cluster head-to-gateway routing approach to route traffic from source to destination. Gateway nodes are nodes that are within communication range of two or more cluster heads. A packet sent by a node is first routed to its cluster head, and then the packet is routed from the cluster head to a gateway to another cluster head, and so on until the cluster head of the destination node is reached. The packet is then transmitted to the destination. Figure 2 illustrates an example of this routing scheme. Using this method, each node must keep a "cluster member table" where it stores the destination cluster head for each mobile node in the network. These cluster member tables are broadcast by each node periodically using the DSDV algorithm. Nodes update their cluster member tables on the reception of such a table from a neighbor.

In addition to the cluster member table, each node must also maintain a routing table, which is used to determine the next hop in order to reach the destination. On receiving a packet, a node will consult its cluster member table and routing table to determine the nearest cluster head along the route to the destination. Next the node will check its routing table to determine the node in order to reach the selected cluster head. It then transmits the packet to this node.

2.1.3 The Wireless Routing Protocol (WRP)

The Wireless Routing Protocol (WRP) described in [14] is a table-based protocol with the goal of maintaining routing information among all nodes in the network. Each node in the network is responsible for maintaining four tables: (a) distance table, (b) routing table, (c) link-cost table, and (d) message retransmission list (MRL) table. Each entry of the MRL contains the sequence number of the update message, a retransmission counter, an acknowledgment-required flag vector with one entry per neighbor, and a list of updates sent in the update message. The MRL records which updates in an update message need to be retransmitted and which neighbors should acknowledge the retransmission [14].

Mobiles inform each other of link changes through the use of update messages. An update message is sent only between neighboring nodes and contains a list of updates (the destination, the distance to the destination, and the predecessor of the destination), as well as a list of responses indicating which mobiles should acknowledge (ACK) the update. Mobiles send update messages after processing updates from neighbors or detecting a change in a link to a neighbor. In the event of the loss of a link between two nodes, the nodes send update messages to their neighbors. The neighbors then update their distance table entries and check for new possible paths through other nodes. Any new paths are relayed back to the original nodes so that they can update their tables accordingly.

Nodes learn of the existence of their neighbors from the receipt of acknowledgments and other messages. If a node is not sending messages, it must send a *hello* message within a specified time period to ensure connectivity. Otherwise, the lack of messages from the node indicates the failure of that link; this may cause a false alarm. When a mobile receives a *hello* message from a new node, that new node is added to the mobile's routing table, and the mobile sends the new node a copy of its routing table information.

Part of the novelty of WRP stems from the way in which it achieves loop freedom. In WRP, routing nodes communicate the distance and second-to-last hop information for each destination in the wireless networks. WRP belongs to the class of path finding algorithms with an important exception. It avoids the "count-to-infinity" problem [21] by forcing each node to perform consistency checks of predecessor information reported by all its neighbors. This ultimately (though not instantaneously) eliminates looping situations and provides faster route convergence when a link failure event occurs.

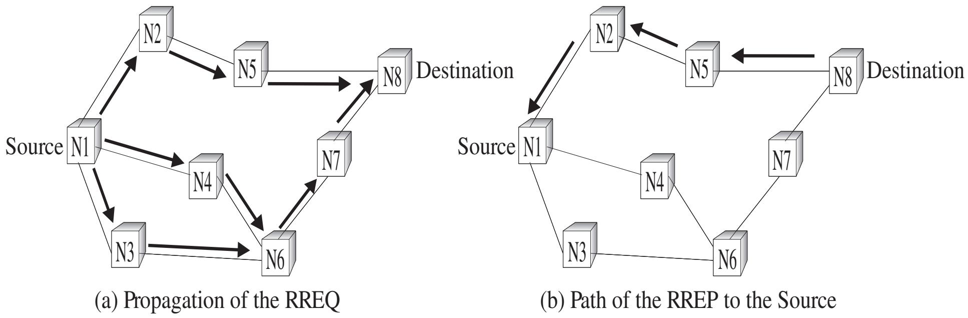


Figure 3: AODV Route Discovery.

2.2 Source-Initiated On-Demand Routing

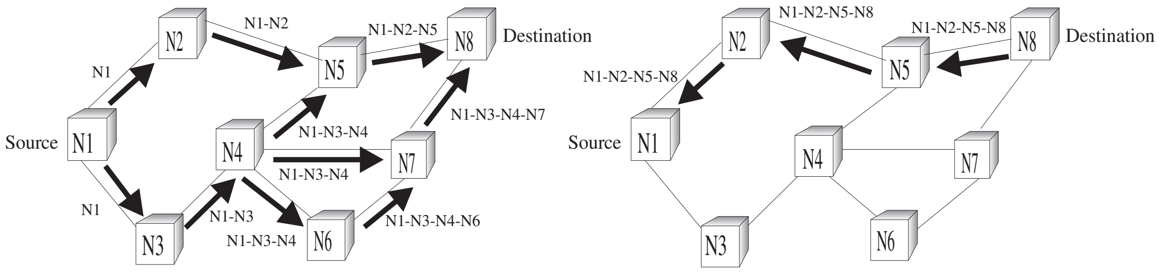
A different approach from table-driven routing is source-initiated on-demand routing. This type of routing creates routes only when desired by the source node. When a node requires a route to a destination, it initiates a route discovery process within the network. This process is completed once a route is found or all possible route permutations have been examined. Once a route has been established, it is maintained by some form of route maintenance procedure until either the destination becomes inaccessible along every path from the source or until the route is no longer desired.

2.2.1 Ad-hoc On-Demand Distance Vector Routing (AODV)

The Ad-hoc On-Demand Distance Vector (AODV) routing protocol described in [19] builds on the DSDV algorithm previously described. AODV is an improvement on DSDV because it typically minimizes the number of required broadcasts by creating routes on an on-demand basis, as opposed to maintaining a complete list of routes as in the DSDV algorithm. The authors of AODV classify it as a *pure on-demand route acquisition* system, as nodes that are not on a selected path do not maintain routing information or participate in routing table exchanges [19].

When a source node desires to send a message to some destination node and does not already have a valid route to that destination, it initiates a *Path Discovery* process to locate the other node. It broadcasts a route request (RREQ) packet to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a "fresh enough" route to the destination is located. Figure 3a illustrates the propagation of the broadcast RREQs across the network. AODV utilizes destination sequence numbers to ensure all routes are loop-free and contain the most recent route information. Each node maintains its own sequence number, as well as a broadcast ID. The broadcast ID is incremented for every RREQ the node initiates, and together with the node's IP address, uniquely identifies a RREQ. Along with its own sequence number and the broadcast ID, the source node includes in the RREQ the most recent sequence number it has for the destination. Intermediate nodes can reply to the RREQ only if they have a route to the destination whose corresponding destination sequence number is greater than or equal to that contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record in their route tables the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. If additional copies of the same RREQ are later received, these packets are discarded. Once the RREQ reaches the destination or an intermediate node with a fresh enough route, the destination/intermediate node responds by unicasting a route reply (RREP) packet back to the neighbor from which it first received the RREQ (Figure 3b). As the RREP is routed back



(a) Building of the Route Record during Route Discovery

(b) Propagation of the Route Reply with the Route Record

Figure 4: Creation of the *Route Record* in DSR.

along the reverse path, nodes along this path set up forward route entries in their route tables which point to the node from which the RREP came. These forward route entries indicate the active forward route. Associated with each route entry is a route timer which will cause the deletion of the entry if it is not used within the specified lifetime. Because the RREP is forwarded along the path established by the RREQ, AODV only supports the use of symmetric links.

Routes are maintained as follows. If a source node moves, it is able to reinitiate the route discovery protocol to find a new route to the destination. If a node along the route moves, its upstream neighbor notices the move and propagates a *link failure notification* message (a RREP with infinite metric) to each of its active upstream neighbors to inform them of the erasure of that part of the route [19]. These nodes in turn propagate the *link failure notification* to their upstream neighbors, and so on until the source node is reached. The source node may then choose to re-initiate route discovery for that destination if a route is still desired.

An additional aspect of the protocol is the use of *hello* messages, periodic local broadcasts by a node to inform each mobile node of other nodes in its neighborhood. Hello messages can be used to maintain the local connectivity of a node. However the use of hello messages is not required. Nodes listen for retransmissions of data packets to ensure the next hop is still within reach. If such a retransmission is not heard, the node may use any one of a number of techniques, including the reception of hello messages, to determine whether the next hop is within communication range. The *hello* messages may list the other nodes from which a mobile has heard, thereby yielding a greater knowledge of the network connectivity.

2.2.2 Dynamic Source Routing (DSR)

The Dynamic Source Routing (DSR) protocol presented in [10] is an on-demand routing protocol that is based on the concept of source routing. Mobile nodes are required to maintain route caches that contain the source routes of which the mobile is aware. Entries in the route cache are continually updated as new routes are learned.

The protocol consists of two major phases: route discovery and route maintenance. When a mobile node has a packet to send to some destination, it first consults its route cache to determine whether it already has a route to the destination. If it has an unexpired route to the destination, it will use this route to send the packet. On the other hand, if the node does not have such a route, it initiates route discovery by broadcasting a *route request* packet. This *route request* contains the address of the destination, along with the source node's address and a unique identification number. Each node receiving the packet checks whether it knows of a route to the destination. If it does not, it adds its own address to the *route record* of the packet and then forwards the packet along its outgoing links. To limit the number of *route requests* propagated on the outgoing links of a node, a

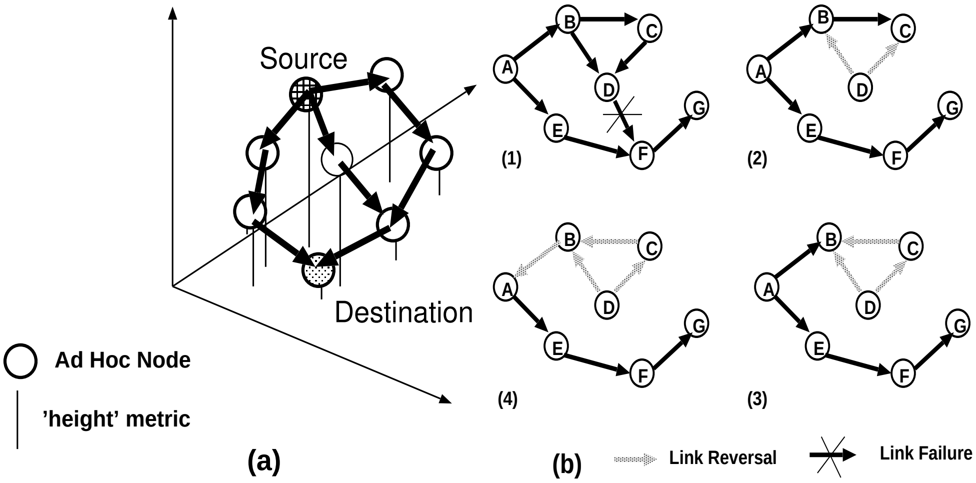


Figure 5: (a) Route creation (showing link direction assignment), and(b) Route Maintenace (showing link reversal phenonemon) in TORA.

mobile only forwards the *route request* if the request has not yet been seen by the mobile and if the mobile's address does not already appear in the *route record*.

A *route reply* is generated when either the *route request* reaches the destination itself, or when it reaches an intermediate node which contains in its route cache an unexpired route to the destination [2]. By the time the packet reaches either the destination or such an intermediate node, it contains a *route record* yielding the sequence of hops taken. Figure 4a illustrates the formation of the *route record* as the *route request* propagates through the network. If the node generating the *route reply* is the destination, it places the *route record* contained in the *route request* into the *route reply*. If the responding node is an intermediate node, it will append its cached route to the *route record* and then generate the *route reply*. To return the *route reply*, the responding node must have a route to the initiator. If it has a route to the initiator in its route cache, it may use that route. Otherwise, if symmetric links are supported, the node may reverse the route in the *route record*. If symmetric links are not supported, the node may initiate its own route discovery and piggyback the *route reply* on the new *route request*. Figure 4b shows the transmission of the *route reply* with its associated *route record* back to the source node.

Route maintenance is accomplished through the use of *route error* packets and acknowledgments. *Route error* packets are generated at a node when the data link layer encounters a fatal transmission problem. When a *route error* packet is received, the hop in error is removed from the node's route cache and all routes containing the hop are truncated at that point. In addition to *route error* messages, acknowledgments are used to verify the correct operation of the route links. Such acknowledgments include passive acknowledgments, where a mobile is able to hear the next hop forwarding the packet along the route.

2.2.3 Temporally-Ordered Routing Algorithm (TORA)

TORA (Temporally-Ordered Routing Algorithm) is a highly adaptive, loop-free, distributed routing algorithm based on the concept of link reversal [16]. TORA is proposed to operate in a highly dynamic mobile networking environment. It is source-initiated and provides multiple routes for any desired source/destination pair. The key design concept of TORA is the localization of control messages to a very small set of nodes near the occurrence of a toplogical change. To accomplish this, nodes need to maintain routing information about adjacent (1-hop) nodes. The protocol performs three basic functions: (a) route creation, (b) route maintainence, and (c) route erasure.

During the route creation and maintenance phases, nodes use a "height" metric to establish a directed acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or downstream) based on the relative height metric of neighboring nodes, as shown in Figure 5a. This process of establishing a DAG is similar to the query/reply process proposed in LMR (Lightweight Mobile Routing) [5]. In times of node mobility, the DAG route is broken and route maintenace is necessary to re-establish a DAG rooted at the same destination. As shown in Figure 5b, upon failure of the last downstream link, a node generates a new reference level which results in the propagation of that reference level by neighboring nodes, effectively coordinating a structured reaction to the failure. Links are reversed to reflect the change in adapting to the new reference level. This has the same effect as reversing the direction of one or more links when a node has no downstream links.

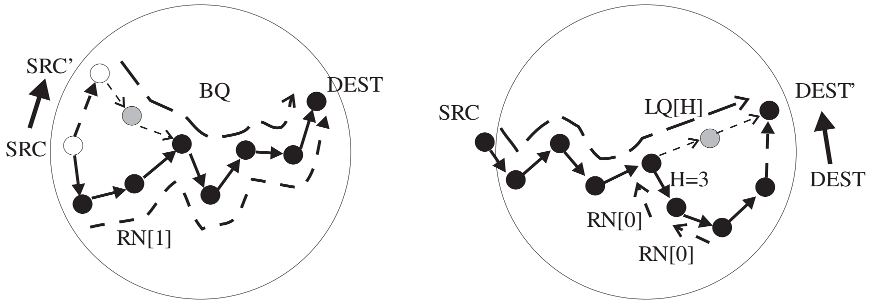
Timing is an important factor for TORA because the "height" metric is dependent on the logical time of a link failure; TORA assumes all nodes have synchronized clocks (accomplished via an external time source such as Global Positioning System). TORA's metric is a quintuple comprised of five elements, namely: (a) logical time of a link failure, (b) the unique ID of the node that defined the new reference level, (c) a reflection indicator bit, (d) a propagation ordering parameter, and (e) the unique ID of the node. The first three elements collectively represent the reference level. A new reference level is defined each time a nodes loses its last downstream link due to a link failure. TORA's route erasure phase essentially involves flooding a broadcast "clear packet" (CLR) throughout the network to erase invalid routes.

In TORA, there is a potential for oscillations to occur, especially when multiple sets of coordinating nodes are concurrently detecting partitions, erasing routes, and building new routes based on each other. Because TORA uses internodal coordination, its instability problem is similar to the "count-to-infinity" problem in distance-vector routing protocols, except that such oscillations are temporary and route convergence will ultimately occur.

2.2.4 Associativity-Based Routing (ABR)

A totally different approach in mobile routing is proposed in [22]. The Associativity-Based Routing (ABR) protocol is free from loops, deadlock, and packet duplicates, and defines a new routing metric for ad-hoc mobile networks. This metric is known as the *degree of association stability*. In ABR, a route is selected based on the degree of association stability of mobile nodes. Each node periodically generates a beacon to signify its existence. When received by neighboring nodes, this beaconing causes their associativity tables to be updated. For each beacon received, the associativity tick of the current node with respect to the beaconing node is incremented. Association stability is defined by connection stability of one node with respect to another node over time and space. A high degree of association stability may indicate a low state of node mobility, while a low degree may indicate a high state of node mobility. Associativity ticks are reset when the neighbors of a node or the node itself moves out of proximity. A fundamental objective of ABR is to derive longer-lived routes for ad-hoc mobile networks.

The three phases of ABR are: (a) route discovery, (b) route re-construction (RRC), and (c) route deletion. The route discovery phase is accomplished by a broadcast query and await-reply (BQ-REPLY) cycle. A node desiring a route broadcasts a BQ message in search of mobiles that have a route to the destination. All nodes receiving the query (that are not the destination) append their addresses and their associativity ticks with their neighbors along with QoS information to the



(a) Route Maintenance for a Source Move (b) Route Maintenance for a Destination Move

Figure 6: Route Maintenance for Source and Destination Movement in ABR.

query packet. A successor node erases its upstream node neighbors' associativity tick entries and retains only the entry concerned with itself and its upstream node. In this way, each resultant packet arriving at the destination will contain the associativity ticks of the nodes along the route to the destination. The destination is then able to select the best route by examining the associativity ticks along each of the paths. In the case where multiple paths have the same overall degree of association stability, the route with the minimum number of hops is selected. The destination then sends a REPLY packet back to the source along this path. Nodes propagating the REPLY mark their routes as valid. All other routes remain inactive and the possibility of duplicate packets arriving at the destination is avoided.

Route re-construction may consist of partial route discovery, invalid route erasure, valid route updates, and new route discovery, depending on which node(s) along the route move. Movement by the source results in a new BQ-REPLY process, as shown in Figure 6a. The RN[1] message is a route notification that is used to erase the route entries associated with downstream nodes. When the destination node moves, the immediate upstream node erases its route and determines if the node is still reachable by a localized query (LQ[H]) process, where H refers to the hop count from the upstream node to the destination (Figure 6b). If the destination receives the LQ packet, it REPLYs with the best partial route; otherwise, the initiating node times out and the process backtracks to the next upstream node. Here an RN[0] message is sent to the next upstream node to erase the invalid routes and inform this node it should invoke the LQ[H] process. If this process results in backtracking more than halfway to the source, the LQ process is discontinued and a new BQ process is initiated at the source.

When a discovered route is no longer desired, the source node initiates a route delete (RD) broadcast so that all nodes along the route update their routing tables. The RD message is propagated by a full broadcast, as opposed to a directed broadcast, because the source node may not be aware of any route node changes that occurred during route re-constructions.

2.2.5 Signal Stability Routing (SSR)

Another on-demand protocol is the Signal Stability based Adaptive Routing protocol (SSR) presented in [6]. Unlike the algorithms described so far, SSR selects routes based on the signal strength between nodes and on a node's location stability. This route selection criteria has the effect of choosing routes that have "stronger" connectivities. SSR can be divided into two cooperative protocols: the Dynamic Routing Protocol (DRP) and the Static Routing Protocol (SRP).

The DRP is responsible for the maintenance of the Signal Stability Table (SST) and the Routing Table (RT). The SST records the signal strength of neighboring nodes, which is obtained by periodic beacons from the link layer of each neighboring node. The signal strength may be recorded as either a strong or weak channel. All transmissions are received by, and processed in, the DRP. After updating all appropriate table entries, the DRP passes a received packet to the SRP.

The SRP processes packets by passing the packet up the stack if it is the intended receiver or looking up the destination in the RT and then forwarding the packet if it is not. If no entry is found in the RT for the destination, a *route-search* process is initiated to find a route. Route requests are propagated throughout the network but are only forwarded to the next hop if they are received over strong channels and have not been previously processed (to prevent looping). The destination chooses the first arriving *route-search* packet to send back because it is most probable that the packet arrived over the shortest and/or least congested path. The DRP then reverses the selected route and sends a *route-reply* message back to the initiator. The DRP of the nodes along the path update their RTs accordingly.

Route-search packets arriving at the destination have necessarily chosen the path of strongest signal stability, as the packets are dropped at a node if they have arrived over a weak channel. If there is no *route-reply* message received at the source within a specific timeout period, the source changes the *PREF* field in the header to indicate that weak channels are acceptable, as these may be the only links over which the packet can be propagated.

When a failed link is detected within the network, the intermediate nodes send an error message to the source indicating which channel has failed. The source then initiates another *route-search* process to find a new path to the destination. The source also sends an erase message to notify all nodes of the broken link.

3 Comparisons

The following sections provide comparisons of the previously described routing algorithms. Section 3.1 compares table-driven protocols, and Section 3.2 compares on-demand protocols. Section 3.3 presents a discussion of the two classes of algorithms. In Tables 1 and 2, Time Complexity is defined as the number of steps needed to perform a protocol operation, and Communication Complexity is the number of messages needed to perform a protocol operation [5], [23]. Also, the values for these metrics represent *worst case* behavior.

3.1 Table-Driven Protocols

Our discussion here will be based on Table 1. As stated earlier, DSDV routing is essentially a modification of the basic Bellman-Ford routing algorithm. The modifications include the guarantee of loop-free routes and a simple route update protocol. While only providing one path to any given destination, DSDV selects the shortest path based on the number of hops to the destination. DSDV provides two types of update messages, one of which is significantly smaller than the other. The smaller update message can be used for incremental updates so that the entire routing table need not be transmitted for every change in network topology. However, DSDV is inefficient because of the requirement of periodic update transmissions, regardless of the number of changes in the network topology. This effectively limits the number of nodes that can connect to the network since the overhead grows as $O(n^2)$.

In CGSR, DSDV is used as the underlying routing protocol. Routing in CGSR occurs over cluster heads and gateways. A cluster head table is necessary in addition to the routing table. One advantage of CGSR is that several heuristic methods can be employed to improve the protocol's performance. These methods include priority token scheduling, gateway code scheduling, and path reservation [4].

Parameters	DSDV	CGSR	WRP
Time Complexity (link addition / failure)	O(d)	O(d)	O(h)
Communication Complexity (link addition / failure)	O(x=N)	O(x=N)	O(x=N)
Routing Philosophy	Flat	Hierarchical	Flat[1]
Loop Free	Yes	Yes	Yes, but not instantaneous
Multicast Capability	No	No[2]	No
Number of Required Tables	Two	Two	Four
Frequency of Update Transmissions	Periodically & as needed	Periodically	Periodically & as needed
Updates Transmitted to	Neighbors	Neighbors & cluster head	Neighbors
Utilizes Sequence Numbers	Yes	Yes	Yes
Utilizes "Hello" Messages	Yes	No	Yes
Critical Nodes	No	Yes (cluster head)	No
Routing Metric	Shortest Path	Shortest Path	Shortest Path

Table 1: Comparisons of the Characteristics of Table-Driven Routing Protocols.

Abbreviations:

$N = Number\ of\ nodes\ in\ the\ network$
$d = Network\ diameter$
$h = Height\ of\ routing\ tree$
$x = Number\ of\ nodes\ affected\ by\ a\ topological\ change$

The WRP protocol differs from the other protocols in several ways. WRP requires each node to maintain four routing tables. This can lead to substantial memory requirements, especially when the number of nodes in the network is large. Furthermore, the WRP protocol requires the use of *hello* packets whenever there are no recent packet transmissions from a given node. The *hello* packets consume bandwidth and disallow a node to enter sleep mode. However, though it belongs to the class of path finding algorithms, WRP has an advantage over other path finding algorithms because it avoids the problem of creating temporary routing loops that these algorithms have through the verification of predecessor information, as described in Section 2.1.3.

Having discussed the operation and characteristics of each of the existing table-driven based routing protocols, it is important to highlight the differences. During link failures, WRP has lower time complexity than DSDV since it only informs neigboring nodes about link status changes. During link additions, *hello* messages are used as a presence indicator such that the routing table entry can be updated. Again, this only affects neighboring nodes. In CGSR, because routing performance is dependent on the status of specific nodes (cluster head, gateway or normal nodes), time complexity of a link failure associated with a cluster head is higher than DSDV, given the additional time needed to perform cluster head reselection. Similarly, this applies to the case of link additions associated with the cluster head. There is no gateway selection in CGSR since each node declares it is a gateway node to its neighbors if it is responding to multiple radio codes. If a gateway node moves out of range, the routing protocol is responsible for routing the packet to another gateway.

In terms of communication complexity, since DSDV, CGSR and WRP use distance vector shortest-path routing as the underlying routing protocol, they all have the same degree of complexity during link failures and additions.

[1]While WRP itself uses flat addressing, it can be used hierarchically [15].

[2]The protocol itself currently does not support multicast; however, there is a separate protocol described in [3], which runs on top of CGSR and provides multicast capabilty.

Performance Parameters	AODV	DSR	TORA	ABR	SSR
Time Complexity (initialization)	O(2d)	O(2d)	O(2d)	O(d+z)	O(d+z)
Time Complexity (postfailure)	O(2d)	O(2d) or 0 (cache hit)	O(2d)	O(l+z)	O(l+z)
Communication Complexity (initialization)	O(2N)	O(2N)	O(2N)	O(N+y)	O(N+y)
Communication Complexity (postfailure)	O(2N)	O(2N)	O(2x)	O(x+y)	O(x+y)
Routing Philosophy	Flat	Flat	Flat	Flat	Flat
Loop Free	Yes	Yes	Yes	Yes	Yes
Multicast Capability	Yes	No	No[3]	No	No
Beaconing Requirements	No	No	No	Yes	Yes
Multiple Route Possibilities	No	Yes	Yes	No	No
Routes Maintained in	route table	route cache	route table	route table	route table
Utilizes Route Cache/Table Expiration Timers	Yes	No	No	No	No
Route Reconfiguration Methodology	Erase Route; Notify Source	Erase Route; Notify Source	Link Reversal; Route Repair	Localized Broadcast Query	Erase Route; Notify Source
Routing Metric	Freshest & Shortest Path	Shortest Path	Shortest Path	Associativity & Shortest Path & others[4]	Associativity & Stability

Table 2: Comparisons of the Characteristics of Source-Initiated On-Demand Ad-Hoc Routing Protocols.

Abbreviations:

$l = Diameter\ of\ the\ affected\ network\ segment$
$y = Total\ number\ of\ nodes\ forming\ the\ directed\ path\ where\ the\ REPLY\ packet\ transits$
$z = Diameter\ of\ the\ directed\ path\ where\ the\ REPLY\ packet\ transits$

3.2 Source-Initiated On-Demand Routing Protocols

Table 2 presents a comparison of AODV, DSR, TORA, ABR and SSR. The AODV protocol employs a route discovery procedure similar to DSR; however, there are a couple important distinctions. The most notable of these is that the overhead of DSR is potentially larger than that of AODV since each DSR packet must carry full routing information, whereas in AODV packets need only contain the destination address. Similarly, the route replies in DSR are larger because they contain the address of every node along the route, whereas in AODV route replies need only carry the destination IP address and sequence number. Also, the memory overhead may be slightly greater in DSR because of the need to remember full routes, as opposed to only next hop information in AODV. A further advantage of AODV is its support for multicast [18]. None of the other algorithms considered in this paper currently incorporate multicast communication. On the downside, AODV requires symmetric links between nodes, and hence cannot utilize routes with assymetric links. In this aspect, DSR is superior as it does not require the use of such links, and can utilize assymetric links when symmetric links are not available.

The DSR algorithm is intended for networks in which the mobiles move at a moderate speed with respect to packet transmission latency [10]. Assumptions that the algorithm makes for operation are that the network diameter is relatively small and that the mobile nodes can enable a promiscuous receive mode, whereby every received packet is delivered to the network driver software without filtering by destination address. An advantage of DSR over some of the other on-demand protocols

[3] Like CGSR, TORA also does not support multicast; however, there is a separate protocol, LAM [9], which runs on top of TORA and provides multicast capabilty.

[4] ABR also uses the Route Relaying Load and Cumulative Forwarding Delay as routing metrics.

is that DSR does not make use of periodic routing advertisements, thereby saving bandwidth and reducing power consumption. Hence the protocol does not incur any overhead when there are no changes in network topology. Additionally, DSR allows nodes to keep multiple routes to a destination in their cache. Hence, when a link on a route is broken, the source node can check its cache for another valid route. If such a route is found, route reconstruction does not need to be reinvoked. In this case, route recovery is faster than in many of the other on-demand protocols. However, if there are no additional routes to the destination in the source node's cache, route discovery must be reinitiated, as in AODV, if the route is still required. On the other hand, because of the small diameter assumption and because of the source routing requirement, DSR is not scalable to large networks. Furthermore, as previously stated, the need to place the entire route in both *route replies* and data packets causes greater control overhead than in AODV.

TORA is a "link reversal" algorithm that is best-suited for networks with large, dense populations of nodes [16]. Part of the novelty of TORA stems from its creation of DAGs to aid route establishment. One of the advantages of TORA is its support for multiple routes. TORA and DSR are the only on-demand protocols considered here which retain multiples route possibilities for a single source/destination pair. Route reconstruction is not necessary until all known routes to a destination are deemed invalid, and hence bandwidth can potentially be conserved because of the necessity for fewer route rebuildings. Another advantage of TORA is its support for multicast. Although, unlike AODV, TORA does not incorporate multicast into its basic operation, it functions as the underlying protocol for the Lightweight Adaptive Multicast Algorithm (LAM), and together the two protocols provide multicast capability [9]. TORA's reliance on synchronized clocks, while a novel idea, inherently limits its applicability. If a node does not have a GPS positioning system or some other external time source, it cannot use the algorithm. Additionally, if the external time source fails, the algorithm will cease to operate. Further, route rebuilding in TORA may not occur as quickly as in the other algorithms due to the potential for oscillations during this period. This can lead to potentially lengthy delays while waiting for the new routes to be determined.

ABR is a compromise between broadcast and point-to-point routing and uses the connection-oriented packet forwarding approach. Route selection is primarily based on the aggregated associativity ticks of nodes along the path. Hence, although the resulting path does not necessarily result in the smallest possible number of hops, the path tends to be longer-lived than other routes. A long-lived route requires fewer route reconstructions and therefore yields higher throughput. Another benefit of ABR is that, like the other protocols, it is guaranteed to be free from packet duplicates. The reason is that only the best route is marked valid while all other possible routes remain passive. ABR, however, relies on the fact that each node is beaconing periodically. The beaconing interval must be short enough so as to accurately reflect the spatial, temporal, and connectivity state of the mobile hosts. This beaconing requirement may result in additional power consumption. However, experimental results obtained in [24] reveal that the inclusion of periodic beaconing has a minute influence on the overall battery power consumption. Unlike DSR, ABR does not utilize route caches.

The SSR algorithm is a logical descendant of ABR. It utilizes a new technique of selecting routes based on the signal strength and location stability of nodes along the path. As in ABR, while the paths selected by this algorithm are not necessarily shortest in hop count, they do tend to be more stable and longer-lived, resulting in fewer route reconstructions. One of the major drawbacks of the SSR procotol is that, unlike in AODV and DSR, intermediate nodes cannot reply to route requests sent towards a destination; this results in potentially long delays before a route can be discovered. Additionally, when a link failure occurs along a path, the route discovery algorithm must be re-invoked from the source to find a new path to the destination. No attempt is made to use partial route recovery (unlike ABR) - i.e. to allow intermediate nodes to attempt to rebuild the route themselves. AODV and DSR also do not specify intermediate node rebuilding. While this may

Parameters	On-Demand	Table-Driven
Availability of Routing Information	Available when needed	Always available regardless of need
Routing Philosophy	Flat	Mostly flat except for CSGR
Periodic route updates	Not required	Yes
Coping with mobility	Using localized route discovery as in ABRand SSR	Inform other nodes to achieve consistent routing table
Signaling traffic generated	Grows with increasing mobility of active routes (as in ABR)	Greater than that of on-demand routing
Quality of Service Support	Few can support QoS	Mainly Shortest Path as QoS metric

Table 3: Overall Comparisons of On-Demand versus Table-Driven Based Routing Protocols.

lead to longer route reconstruction times since link failures cannot be resolved locally without the intervention of the source node, the attempt and failure of an intermediate node to rebuild a route will cause a longer delay then if the source node had attempted the rebuilding as soon as the broken link was noticed. Thus it remains to be seen whether intermediate node route rebuilding is more optimal than source node route rebuilding.

3.3 Table-Driven vs On-Demand Routing

As discussed earlier, the table-driven ad-hoc routing approach is similar to the connectionless approach of forwarding packets, with no regard to when and how frequent such routes are desired. It relies on an underlying routing table update mechanism that involves the constant propagation of routing information. This is, however, not the case for on-demand routing protocols. When a node using an on-demand protocol desires a route to a new destination, it will have to wait until such a route can be discovered. On the other hand, because routing information is constantly propagated and maintained in table-driven routing protocols, a route to every other node in the ad-hoc network is always available, regardless of whether or not it is needed. This feature, although useful for datagram traffic, incurs substantial signaling traffic and power consumption. Since both bandwidth and battery power are scarce resources in mobile computers, this becomes a serious limitation. Table 3 lists some of the basic differences between the two classes of algorithms.

Another consideration is whether a flat or hierarchical addressing scheme should be used. All of the protocols considered here, except for CGSR, use a flat addressing scheme. In [1], a discussion of the two addressing schemes is presented. While flat addressing may be less complicated and easier to use, there are doubts as to its scalability.

4 Applications and Challenges

Akin to packet radio networks, ad-hoc wireless networks have an important role to play in military applications. Soldiers equipped with multi-mode mobile communicators can now communicate in an ad-hoc manner, without the need for fixed wireless base stations. In addition, small vehicular devices equipped with audio sensors and cameras can be deployed at targetted regions to collect important location and environmental information which will be communciated back to a processing node via ad-hoc mobile communications. Ship-to-ship ad-hoc mobile communication is also desirable since it provides alternate communication paths without reliance on ground- or space-based communication infrastructures.

Commerical scenarios for ad-hoc wireless networks include: (a) conferences/meetings/lectures [8], (b) emergency services, and (c) law enforcement. People today attend meetings and conferences with their laptops, palmtops and notebooks. It is therefore attractive to have instant network formation, in addition to file and information sharing without the presence of fixed base stations and systems administrators. A presenter can multicast slides and audio to intended receipents. Attendees can ask questions and interact on a commonly-shared white board. Ad-hoc mobile communication is particularly useful in relaying information (status, situation awareness, etc.) via data, video and/or voice from one rescue team member to another over a small handheld or wearable wireless device. Again, this applies to law enforcement personnel as well.

Current challenges for ad-hoc wireless networks include: (a) multicast, (b) QoS support, (c) power-aware routing [20], and (d) location-aided routing [12]. As mentioned above, multicast is desirable to support multi-party wireless communications. Since the multicast tree is no longer static (i.e., its topology is subject to change over time), the multicast routing protocol must be able to cope with mobility, including multicast membership dynamics (such as *leave* and *join*). In terms of QoS, it is inadequate to consider QoS merely at the network level without considering the underlying media access control layer [13]. Again, given the problems associated with the dynamics of nodes, hidden terminals, and fluctuating link characteristics, supporting end-to-end QoS is a non-trivial issue that requires in-depth investigation. Currently, there is a trend towards an adaptive QoS approach instead of the "plain" resource reservation method with hard QoS guarantees. Another important factor is the limited power supply in handheld devices which can seriously prohibit packet forwarding in an ad-hoc mobile environment. Hence, routing traffic based on nodes' power metric is one way to distinguish routes that are more long-lived than others. Finally, instead of using beaconing or broadcast search, location-aided routing uses positioning information to define associated regions so that the routing is spatially-oriented and limited. This is analogous to associativity-oriented and restricted broadcast in ABR.

Current ad-hoc routing approaches have introduced several new paradigms, such as exploiting user's demand, the use of location, power, and association parameters. Adaptivity and self-configuration are key features of these approaches. However, flexibility is also important. A *flexible* ad-hoc routing protocol could responsively invoke table-driven approaches and/or on-demand approaches based on situations and communication requirements. The "toggle" between these two approaches may not be trivial since concerned nodes must be "in-sync" with the toggling. Co-existence of both approaches may also exist in spatially clustered ad-hoc groups, with intra-cluster employing the table-driven approach and inter-cluster employing the demand-driven approach or vice versa. Further work is necessary to investigate the feasibility and performance of hybrid ad-hoc routing approaches. Lastly, in addition to the above, further research in the areas of media access control, security, service discovery, and internet protocol operability is required before the potential of ad-hoc mobile networking can be realized.

5 Conclusion

In this paper we have provided descriptions of several routing schemes proposed for ad-hoc mobile networks. We have also provided a classification of these schemes according to the routing strategy, i.e., table-driven and on-demand. We have presented a comparison of these two categories of routing protocols, highlighting their features, differences and characteristics. Finally, we have identified possible applications and challenges facing ad-hoc mobile wireless networks. While it is not clear that any particular algorithm or class of algorithm is the best for all scenarios, each protocol has definite advantages and disadvantages and has certain situations for which it is well-suited. The field of ad-hoc mobile networks is rapidly growing and changing, and while there are still many challenges that need to be met, it is likely that such networks will see wide-spread use within the next few years.

6 Acknowledgment

We would like to thank C.-C. Chiang, J. J. Garcia-Luna-Aceves, David Maltz, Vincent Park, and Charles Perkins for their assistance in assuring accurate descriptions of the operation of their protocols.

References

[1] D. Baker, M. S. Corson, P. Sass, and S. Ramanatham, "Flat vs. Hierarchical Network Control Architecture," ARO/DARPA Workshop on Mobile Ad-Hoc Networking, http://www.isr.umd.edu/Courses/Workshops/MANET/program.html, March 1997.

[2] J. Broch, D. B. Johnson, D. A. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," *IETF Internet Draft* draft-ietf-manet-dsr-01.txt, December 1998 (Work in Progress).

[3] C.-C. Chiang, M. Gerla, and L. Zhang, "Adaptive Shared Tree Multicast in Mobile Wireless Networks," *Proceedings of GLOBECOM '98*, pp. 1817–1822, November 1998.

[4] C.-C. Chiang, H.K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of IEEE SICON'97*, pp. 197–211, April 1997.

[5] M. S. Corson and A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM/Baltzer Wireless Networks Jouornal*, Vol. 1, No. 1, pp. 61–81, February 1995.

[6] R. Dube, C. D. Rais, K.-Y. Wang, and S.K. Tripathi, "Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks," *IEEE Personal Communications*, pp. 36–45, February 1997.

[7] L. R. Ford Jr. and D. R. Fulkerson, *Flows in Networks.* Princeton University Press, Princeton N.J., 1962.

[8] M. Gerla, C.-C. Chiang, and L. Zhang, "Tree Multicast Strategies in Mobile, Multihop Wireless Networks," To appear in *ACM/Baltzer Mobile Networks and Applications Journal*, 1998.

[9] L. Ji and M. S. Corson, "A Lightweight Adaptive Multicast Algorithm," *Proceedings of GLOBECOM '98*, pp. 1036–1042, November 1998.

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

[11] J. Jubin and J. Tornow, "The DARPA Packet Radio Network Protocols," *Proceedings of the IEEE*, Vol. 75, No. 1, pp. 21–32, 1987.

[12] Y. B. Ko and N. H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *Proceedings of ACM/IEEE MOBICOM '98*, October 1998.

[13] C. R. Lin and M. Gerla, "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network," *Proceedings of IEEE INFOCOM '97*, March 1997.

[14] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks," *ACM Mobile Networks and Applications Journal*, Special Issue on Routing in Mobile Communication Networks, pp. 183–197, October 1996.

[15] S. Murthy and J. J. Garcia-Luna-Aceves, "Loop-Free Internet Routing Using Hierarchical Routing Trees," *Proceedings of INFOCOM '97*, April 7-11, 1997.

[16] V. D. Park and M. S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proceedings of INFOCOM '97*, April 1997.

[17] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Computer Communications Review*, pp. 234–244, October 1994.

[18] C. E. Perkins and E. M. Royer, "Ad Hoc On Demand Distance Vector (AODV) Routing," *IETF Internet Draft*, draft-ietf-manet-aodv-02.txt, November 1998 (Work in Progress).

[19] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," *Proceedings of 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999.

[20] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks," *Proceedings of ACM/IEEE MOBICOM '98*, October 1998.

[21] A. S. Tanenbaum, *Computer Networks, Third Edition.* Prentice Hall, Englewood Cliffs, Chapter 5, pp. 357–358, 1996.

[22] C-K. Toh, "A Novel Distributed Routing Protocol To Support Ad-Hoc Mobile Computing," *Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communication*, pp. 480–486, March 1996.

[23] C-K. Toh, "Associativity-Based Routing for Ad-Hoc Mobile Networks," *Wireless Personal Communications*, Vol. 4, No. 2, pp. 1–36, March 1997.

[24] C-K. Toh and George Lin, "Implementing Associativity-Based Routing for Ad Hoc Mobile Wireless Networks," Unpublished Article, March 1998.

Authors' Biographies

ELIZABETH M. ROYER (eroyer@alpha.ece.ucsb.edu) received her M.S. degree in Electrical and Computer Engineering from the University of California, Santa Barbara in December 1997 and her B.S. degrees in Computer Science and Applied Mathematics from Florida State University in April 1996. She is currently a PhD student in the Electrical and Computer Engineering Department at the University of California, Santa Barbara. At UCSB she is a member of the Computer Networks and Distributed Systems Laboratory. Her research interests include routing and multicast for ad-hoc mobile networks. Elizabeth is the recipient of a National Science Foundation Graduate Fellowship and a University of California Doctoral Scholar Fellowship.

CHAI-KEONG TOH (C-K.Toh@acm.org) was born in Singapore in 1965. He received his Diploma in Electronics and Communication Engineering with a Certificate of Merit award from the Singapore Polytechnic in 1986, his BEng degree in Electronics Engineering with first class honors from the University of Manchester Institute of Science and Technology (UMIST) in 1991 and his Ph.D. degree in Computer Science from the Computer Laboratory, University of Cambridge, Cambridge, England in 1996. Dr. Toh founded and chaired the Mobile Special Interest Group (Mobile SIG) from 1994-1996. Before joining Cambridge University, C-K was a network specialist, R&D engineer and technical staff member. At Cambridge, C-K was an Honorary Cambridge Commonwealth Trust Scholar and a King's College Cambridge Research Scholar. He authored the book on 'Wireless ATM and Ad Hoc Networks: Protocols and Architectures', which was published by Kluwer Academic Press in 1996. Dr. Toh serves on the editorial board for ACM Mobile Computing and Communications Review (MC2R), IEEE Network and Personal Technologies Journal (Springer Verlag). He is a member of IEE, IEEE, USENIX, ACM, Sigma Xi Honor Society, New York Academy of Science, American Association for the Advancement of Science (AAAS) and is a Fellow of the Cambridge Philosophical Society and Cambridge Commonwealth Society. He is currently an assistant professor with the School of Electrical and Computer Engineering at the Georgia Institute of Technology, Atlanta, Georgia, directing the Mobile Multimedia and High Speed Networking Laboratory.