Scalable Routing Protocols for Mobile Ad Hoc Networks

Xiaoyan Hong, Kaixin Xu, and Mario Gerla, University of California at Los Angeles

Abstract

The growing interest in mobile ad hoc network techniques has resulted in many routing protocol proposals. Scalability issues in ad hoc networks are attracting increasing attention these days. In this article we will survey the routing protocols that address scalability. The routing protocols we intend to include in the survey fall into three categories: flat routing protocols, hierarchical routing approaches, and GPS augmented geographical routing schemes. The article will compare the scalability properties and operational features of the protocols and discuss challenges in future routing protocol designs.

With the advance of wireless communication technologies, small-size and high-performance computing and communication devices are increasingly used in daily life and computing (e.g., commercial laptops and personal digital assistants equipped with radios). In this article we consider a large population of such devices wishing to communicate tetherlessly. While the infrastructured cellular system is a traditional model for a mobile wireless network, here we focus on a network that does not rely on a fixed infrastructure and works in a shared wireless media. Such a network, called a *mobile ad hoc network* (MANET) [1], is a self-organizing and self-configuring multihop wireless network, where the network structure changes dynamically due to member mobility. Ad hoc networks are very attractive for tactical communication in the military and law enforcement. They are also expected to play an important role in civilian fora such as convention centers, conferences, and electronic classrooms. Nodes in this network model share the same random access wireless channel. They cooperate in a friendly manner to engage in multihop forwarding. Each node functions not only as a host but also as a router that maintains routes to and forwards data packets for other nodes in the network that may not be within direct wireless transmission range. Routing in ad hoc networks faces extreme challenges from node mobility/dynamics, potentially very large numbers of nodes, and limited communication resources (e.g., bandwidth and energy). The routing protocols for ad hoc wireless networks have to adapt quickly to frequent and unpredictable topology changes and must be parsimonious of communications and processing resources.

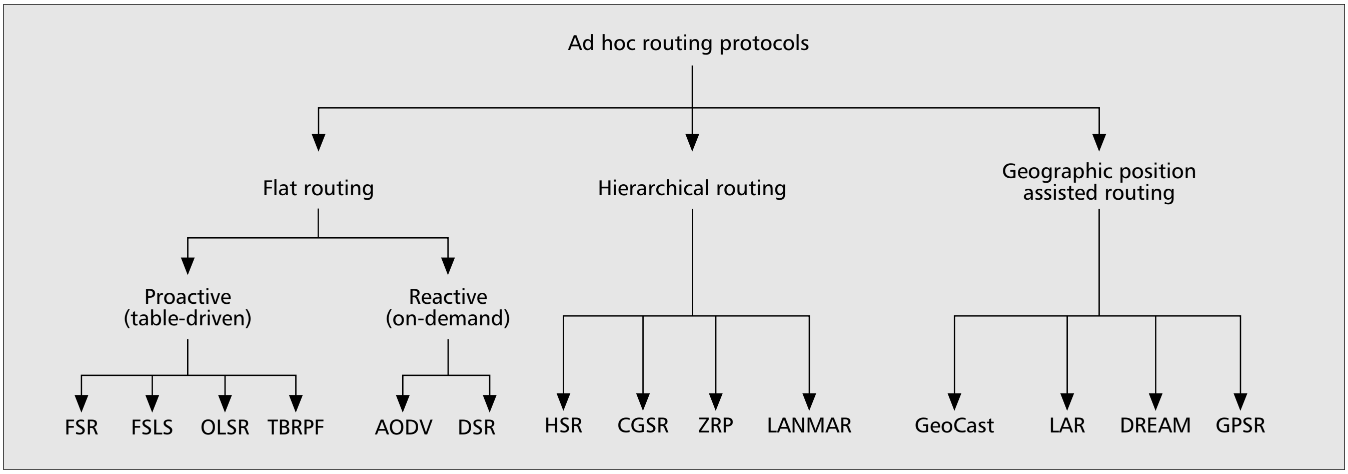
Due to the facts that bandwidth is scarce in MANET nodes and the population in a MANET is small compared to the wireline Internet, the scalability issue for wireless multihop routing protocols is mostly concerned with excessive routing message overhead caused by the increase of network population and mobility. Routing table size is also a concern in MANETs because large routing tables imply a large control packet size and hence large link overhead. Routing protocols generally use either distance-vector or link-state routing algorithms [2]. Both types find shortest paths to destinations. In distance-vector routing (DV), a vector containing the cost (e.g., hop distance) and path (next hop) to all the destinations is kept and exchanged at each node. DV protocols are generally known to suffer from slow route convergence and a tendency to create loops in mobile environments. The link-state routing (LS) algorithm overcomes the problem by maintaining global network topology information at each router through periodical flooding of link information about its neighbors. Mobility entails frequent flooding. Unfortunately, this LS advertisement scheme generates larger routing control overhead than DV. In a network with population N, LS updating generates routing overhead on the order of $O(N^2)$. In large networks, the transmission of routing information will ultimately consume most of the bandwidth and consequently block applications, rendering it unfeasible for bandwidth limited wireless ad hoc networks. Thus, reducing routing control overhead becomes a key issue in achieving routing scalability. In some application domains (e.g., digitized battlefield) scalability is realized by designing a hierarchical architecture with physically distinct layers (e.g., point-to-point wireless backbone) [3]. However, such a physical hierarchy is not cost-effective for many other applications (e.g., sensor networks). Thus, it is important to find solutions to the scalability problem of a homogeneous ad hoc network strictly using scalable routing protocols.

The scalability is more challenging in the presence of both large numbers and mobility. If nodes are stationary, the large population can be effectively handled with conventional hierarchical routing. In contrast, when nodes move, the hierarchical partitioning must be continuously updated. Mobile IP solutions work well if there is a fixed infrastructure supporting the concept of the *home agent*. When all nodes move (including the home agent), such a strategy cannot be directly applied.

A considerable body of literature has addressed research on routing and architecture of ad hoc networks. Relating to

---

This work was supported in part by ONR "MINUTEMAN" project under contract N00014-01-C-0016, in part by DARPA under contract DAAB07-97-C-D321.



■ Figure 1. *Classification of ad hoc routing protocols.*

the problem described above, we present a survey with a focus on solutions toward scalability in large populations that are able to handle mobility. Classification according to routing strategy, that is, proactive (or table-driven) and reactive (or on demand), has been used in other articles [4–8]. Here we provide a classification according to the network structure underlying routing protocols. Different structures affect the design and operation of the routing protocols; they also determine the performance with regard to scalability. Reviews and performance comparisons of ad hoc routing protocols have been presented in many earlier publications [4, 5, 9–12]. While some overlap with previous surveys is inevitable in order to preserve the integrity of our presentation, our choice of protocols includes recent examples that reveal unique features in term of scalability.

In the remainder of this article, we review key routing protocols in ad hoc networks in three broad categories (Fig. 1):

- Flat routing schemes, which are further classified into two classes: proactive and reactive, according to their design philosophy
- Hierarchical routing
- Geographic position assisted routing

Flat routing approaches adopt a flat addressing scheme. Each node participating in routing plays an equal role. In contrast, hierarchical routing usually assigns different roles to network nodes. Some protocols require a hierarchical addressing system. Routing with assistance from geographic location information requires each node to be equipped with the Global Positioning System (GPS). This requirement is quite realistic today since such devices are inexpensive and can provide reasonable precision. The article concludes with a summary of the scalable features of protocols in the three categories and future research directions.

Routing in a Flat Network Structure

The protocols we review here fall into two categories: proactive and on-demand routing. Many proactive protocols stem from conventional LS routing. On-demand routing, on the other hand, is a new emerging routing philosophy in the ad hoc area. It differs from conventional routing protocols in that no routing activities and no permanent routing information are maintained at network nodes if there is no communication, thus providing a scalable routing solution to large populations.

Proactive Routing Protocols

Proactive routing protocols share a common feature, that is, background routing information exchange regardless of communication requests. The protocols have many desirable properties, especially for applications including real-time communications and QoS guarantees, such as low-latency route access and alternate QoS path support and monitoring. Many proactive routing protocols have been proposed for efficiency and scalability.

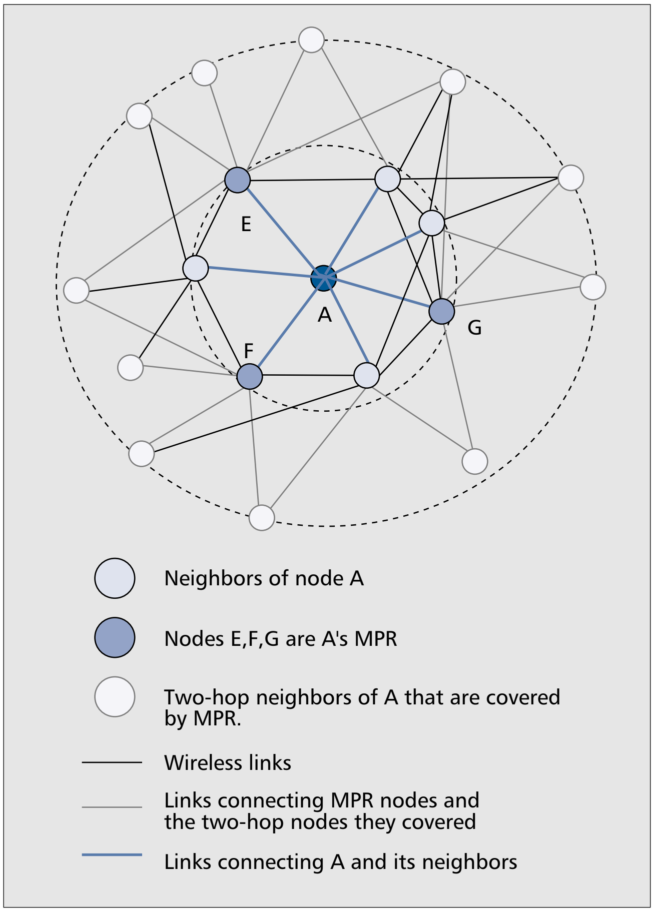
Fisheye State Routing — Fisheye State Routing (FSR) described in [13, 14] is a simple, efficient LS type routing protocol that maintains a topology map at each node and propagates link state updates. The main differences between FSR and conventional LS protocols are the ways in which routing information is disseminated. First, FSR exchanges the entire link state information only with neighbors instead of flooding it over the network. The link state table is kept up to date based on the information received from neighbors. Second, the link state exchange is periodical instead of event-triggered, which avoids frequent link state updates caused by link breaks in an environment with unreliable wireless links and mobility. Moreover, the periodical broadcasts of the link state information are conduced in different frequencies for different entries depending on their hop distances to the current node. Entries corresponding to faraway (outside a predefined *scope*) destinations are propagated with lower frequency than those corresponding to nearby destinations. As a result, a considerable fraction of entries are suppressed from link state exchange packets. FSR produces accurate distance and path information about the immediate neighborhood of a node, and imprecise knowledge of the best path to a distant destination. However, this imprecision is compensated by the fact that the route on which the packet travels becomes progressively more accurate as the packet approaches its destination. Similar work is also presented in Fuzzy Sighted Link State (FSLS) routing [6]. FSLS includes an optimal algorithm called Hazy Sighted Link State (HSLS), which sends a link state update (LSU) every $2^k * T$ to a scope of 2^k, where k is hop distance and T is the minimum LSU transmission period. Thus, both FSR and FSLS achieve potential scalability by limiting the scope of link state update dissemination in space and over time. Theoretical analysis on routing overhead and optimization for this type of “myopic” routing can be found in [6].

Optimized Link State Routing Protocol — Optimized Link State Routing Protocol (OLSR) [15] is an LS protocol. It periodically exchanges topology information with other nodes in the network. The protocol uses *multipoint relays (MPRs)* [16] to reduce the number of “superfluous” broadcast packet retransmissions and also the size of the LS update packets, leading to efficient flooding of control messages in the network.

A node, say node A, periodically broadcasts HELLO messages to all immediate neighbors to exchange neighborhood information (i.e., list of neighbors) and to compute the MPR set. From neighbor lists, node A figures out the nodes that are two hops away and computes the minimum set of one-hop relay points required to reach the two-hop neighbors. Such set is the MPR set. Figure 2 illustrates the MPR set of node A. The optimum (minimum size) MPR computation is NP-complete. Efficient heuristics are used. Each node informs its neighbors about its MPR set in the HELLO message. Upon receiving such a HELLO, each node records the nodes (called MPR *selectors*) that select it as one of their MPRs. In routing information dissemination, OLSR differs from pure LS protocols in two aspects. First, by construction, only the MPR nodes of A need to forward the link state updates issued by A. Second, the link state update of node A is reduced in size since it includes only the neighbors that select node A as one of their MPR nodes. In this way, partial topology information is propagated, that is, say, node A can be reached only from its MPR selectors. OLSR computes the shortest path to an arbitrary destination using the topology map consisting of all of its neighbors and of the MPRs of all other nodes. OLSR is particularly suited for dense networks. When the network is sparse, every neighbor of a node becomes a multipoint relay. The OLSR then reduces to a pure LS protocol.

Topology Broadcast Based on Reverse Path Forwarding — Topology Broadcast Based on Reverse Path Forwarding (TBRPF) [17, 18] is also an LS protocol. It consists of two separate modules: the neighbor discovery module (TND) and the routing module. TND is performed through periodical "differential" HELLO messages that report only the changes (up or lost) of neighbors. The TBRPF routing module operates based on partial topology information obtained through both periodic and differential topology updates. Operation in full topology is provided as an option by including additional topology information in updates.

TBRPF works as follows. Assume node S is the source of update messages. Every node i in the network chooses its next hop (say, node p) on the minimum-hop path toward S as its parent with respect to node S. Instead of flooding to the entire net, TBRPF only propagates link-state updates in the reverse direction on the spanning tree formed by the minimum-hop paths from all nodes to the source of the updates, that is, node i only accepts topology updates originated at node S from parent node p, and then forwards them to the children pertaining to S. Furthermore, only the links that will result in changes to i's source tree are included in the updates. Thus, a smaller subset of the source tree is propagated. The leaves of the broadcast tree do not forward updates. Each node can also include the entire source tree in the updates for full topology operation. The topology updates are broadcast periodically and differentially. The differential updates are issued more frequently to quickly propagate link changes (additions and deletions). Thus, TBRPF adapts to topology change faster, generates less routing overhead, and uses a smaller topology update packet size than pure LS protocols.



■ Figure 2. *OLSR: an illustration of multipoint relays.*

On-Demand Routing Protocols

On-demand routing is a popular routing category for wireless ad hoc routing. The design follows the idea that each node tries to reduce routing overhead by only sending routing packets when a communication is awaiting. Examples include Ad Hoc On Demand Distance Vector Routing (AODV) [19], Associativity-Based Routing (ABR) [20], Dynamic Source Routing (DSR) [21], Lightweight Mobile Routing (LMR) [22], and Temporally Ordered Routing Algorithms (TORA) [23]. Among the many proposed protocols, AODV and DSR have been extensively evaluated in the MANET literature and are being considered by the Internet Engineering Task Force (IETF) MANET Working Group as the leading candidates for standardization. They are described briefly here to demonstrate the on-demand routing mechanism. Interested readers are referred to [4, 9, 24] for performance evaluation.

On-demand algorithms typically have a route discovery phase. Query packets are flooded into the network by the sources in search of a path. The phase completes when a route is found or all the possible outgoing paths from the source are searched. There are different approaches for discovering routes in on-demand algorithms. In AODV, on receiving a query, the transit nodes "learn" the path to the source (called *backward learning*) and enter the route in the forwarding table. The intended destination eventually receives the query and can thus respond using the path traced by the query. This permits establishment of a full duplex path. To reduce new path search overhead, the query packet is dropped during flooding if it encounters a node which already has a route to the destination. After the path has been established, it is maintained as long as the source uses it. A link failure will be reported to the source recursively through the intermediate nodes. This in turn will trigger another query-response procedure in order to find a new route.

An alternate scheme for tracing on-demand paths is DSR. DSR uses *source routing*, that is, a source indicates in a data packet's header the sequence of intermediate nodes on the routing path. In DSR, the query packet copies in its header the IDs of the intermediate nodes it has traversed. The desti-

	FSR	OLSR	TBRPF	AODV	DSR
Routing philosophy	Proactive	Proactive	Proactive	On-demand	On-demand
Routing metric	Shortest path	Shortest path	Shortest path	Shortest path	Shortest path
Frequency of updates	Periodically	Periodically	Periodically, as needed (link changes)	As needed (data traffic)	As needed (data traffic)
Use sequence numbers	Yes	Yes	Yes (HELLO)	Yes	No
Loop-free	Yes	Yes	Yes	Yes	Yes
Worst case exists	No	Yes (pure LS)	No	Yes (full flooding)	Yes (full flooding)
Multiple paths	Yes	No	No	No	Yes
Storage complexity	$O(N)$	$O(N)$	$O(N)$	$O(e)$	$O(e)$
Comm. complexity	$O(N)$	$O(N)$	$O(N)$	$O(2N)$	$O(2N)$

■ Table 1. *Characteristics of flat routing protocols.*

nation then retrieves the entire path from the query packet, and uses it (via source routing) to respond to the source, providing the source with the path at the same time. Data packets carry the source route in the packet headers. A DSR node aggressively caches the routes it has leaned so far to minimize the cost incurred by the route discovery. Source routing enables DSR nodes to keep multiple routes to a destination. When link breakage is detected (through *passive acknowledgments*), route reconstruction can be delayed if the source can use another valid route directly. If no such alternate routes exist, a new search for a route must be reinvoked. The path included in the packet header makes the detection of loops very easy.

To reduce the route search overhead, both protocols provide optimizations by taking advantage of existing route information at intermediate nodes. *Promiscuous listening* (overhearing neighbor propagation) used by DSR helps nodes to learn as many route updates as they can without actually participating in routing. *Expanding ring search* (controlled by the *time-to-live* field of route request packets) used by AODV limits the search area for a previous discovered destination using the prior hop distance.

Comparisons of Flat Routing Protocols

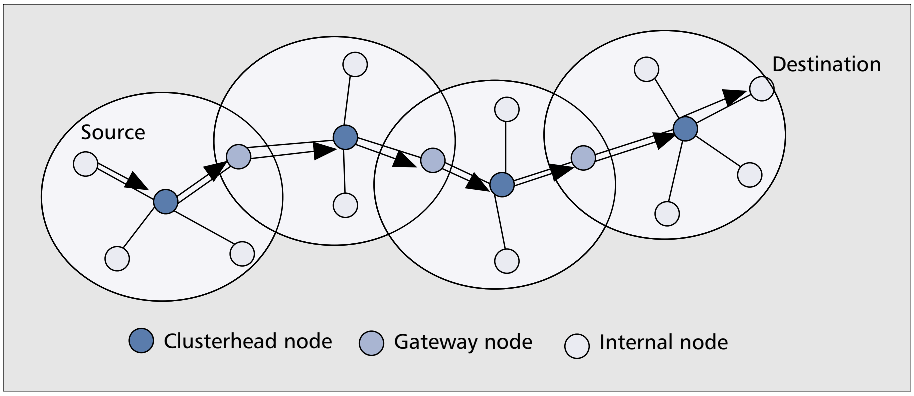
Key characteristics of the protocols are summarized in Table 1. In the table, N denotes the number of nodes in the network and e the number of communication pairs. Storage complexity measures the order of the table size used by the protocols. Communication complexity gives the number of messages needed to perform an operation when an update occurs.

The proactive protocols adopt different ways toward scalability. FSR introduces the notion of multilevel fisheye scope to reduce routing update overhead through reducing the routing packet sizes and update frequency. FSLS/HSLS further drives this limited dissemination approach to an optimal point. OLSR produces less control overhead than FSR because it forces the propagation of link state updates only at MPR nodes, leading to fewer nodes participating in link state update forwarding. Similarly, TBRPF reduces the LS updates forwarding at leaf nodes of each source tree and disseminates differential updates. It also generates smaller HELLO messages than OLSR. Both OLSR and TBRPF achieve more efficiency than classic LS algorithms when networks are dense, that is, OLSR obtains a larger compression ratio of number of MPRs over number of neighbors, and TBRPF trims more leaf nodes from propagation. The multilevel scope reduction from FSR and FSLS, however, will not reduce propagation frequency when a network grows dense. In contrast, the scope reduction works well when a network grows in diameter (in terms of hop distance). Multiple scopes can effectively reduce the update frequency for nodes many hops away. However, all four protocols require nodes to maintain routing tables containing entries for all the nodes in the network (storage complexity $O(N)$). This is acceptable if the user population is small. As the number of mobile hosts increases, so does the overhead. This affects the scalability of the protocols in large networks.

Operations of on-demand routings react only to communication needs. The routing overhead thus relates to the discovery and maintenance of the routes in use. With light traffic (directed to a few destinations) and low mobility, on-demand protocols scale well to large populations (low bandwidth and storage overhead). However, for heavy traffic with a large number of destinations, more sources will search for destinations. Also, as mobility increases, the prediscovered route may break down, requiring repeated route discoveries on the way to the destination. Route caching becomes ineffective with high mobility. Since flooding is used for query dissemination and route maintenance, routing control overhead tends to grow very high [24] in this case. Longer delays are also expected in large mobile networks. In addition, DSR generates larger routing and data packets due to the stored path information. In large networks where longer paths prevail, source routing packets cause larger overhead.

In terms of scattered traffic pattern and high mobility, proactive protocols produce higher routing efficiency than on-demand protocols. The routes to all the destinations are known in advance. Fresh route information is maintained periodically. No additional routing overhead needs to be generated for finding a new destination or route. The cost of these features is that proactive protocols constantly consume bandwidth and energy due to the periodic updates. This property makes proactive schemes undesirable for some resource critical applications (e.g., sensor networks).

For AODV and DSR, since a route has to be entirely discovered prior to the actual data packet transmission, the initial search latency may degrade the performance of interactive applications (e.g., distributed database queries). In contrast, FSR, OLSR, and TBRPF avoid the extra work of "finding" the destination by retaining a routing entry for each destination all the time, thus providing low single-packet transmission latency. Proactive schemes such as FSR, OLSR, and TBRPF can easily extend to QoS monitoring by including bandwidth and channel quality information in link state entries. Thus, the quality of the path (e.g., bandwidth, delay) is known prior to call setup. For AODV and DSR, the quality of the path is not known a priori.



■ **Figure 3.** *CGSR routing: showing a data path from source to destination.*

It can be discovered only while setting up the path and must be monitored by all intermediate nodes during the session, at the cost of additional latency and overhead penalty.

Hierarchical Routing Protocols

Typically, when wireless network size increase (beyond certain thresholds), current "flat" routing schemes become infeasible because of link and processing overhead. One way to solve this problem, and to produce scalable and efficient solutions is hierarchical routing. An example of hierarchical routing is the Internet hierarchy, which has been practiced in the wired network for a long time. Wireless hierarchical routing is based on the idea of organizing nodes in groups and then assigning nodes different functionalities inside and outside a group. Both routing table size and update packet size are reduced by including in them only part of the network (instead of the whole); thus, control overhead is reduced. The most popular way of building hierarchy is to group nodes geographically close to each other into explicit clusters. Each cluster has a leading node (*clusterhead*) to communicate to other nodes on behalf of the cluster. An alternate way is to have implicit hierarchy. In this way, each node has a local scope. Different routing strategies are used inside and outside the scope. Communications pass across overlapping scopes. More efficient overall routing performance can be achieved through this flexibility. Since mobile nodes have only a single omnidirectional radio for wireless communications, this type of hierarchical organization will be referred to as *logical hierarchy* to distinguish it from the physically hierarchical network structure.

Clusterhead-Gateway Switch Routing

Clusterhead-Gateway Switch Routing (CGSR) [25] is typical of cluster-based hierarchical routing. A stable clustering algorithm, *Least Clusterhead Change* (LCC), is used to partition the whole network into clusters, and a *clusterhead* is elected in each cluster. A mobile node that belongs to two or more clusters is a *gateway* connecting the clusters. Data packets are routed through paths having a format of "Clusterhead–Gateway Clusterhead–Gateway…" between any source and destination pairs.

CGSR is a distance vector routing algorithm. Two tables, a cluster member table and a DV routing table, are maintained at each mobile node. The cluster member table records the clusterhead for each node and is broadcast periodically. A node will update its member table on receiving such a packet. The routing table only maintains one entry for each cluster recording the path to its clusterhead, no matter how many members it has. To route a data packet, the current node first looks up the clusterhead of the destination node from the cluster member table. Then it consults its routing table to find the next hop to that destination cluster and routes the packet toward the destination clusterhead. The destination clusterhead will finally route the packet to the destination node, which is a member of it and can be directly reached. This procedure is demonstrated in Fig. 3.

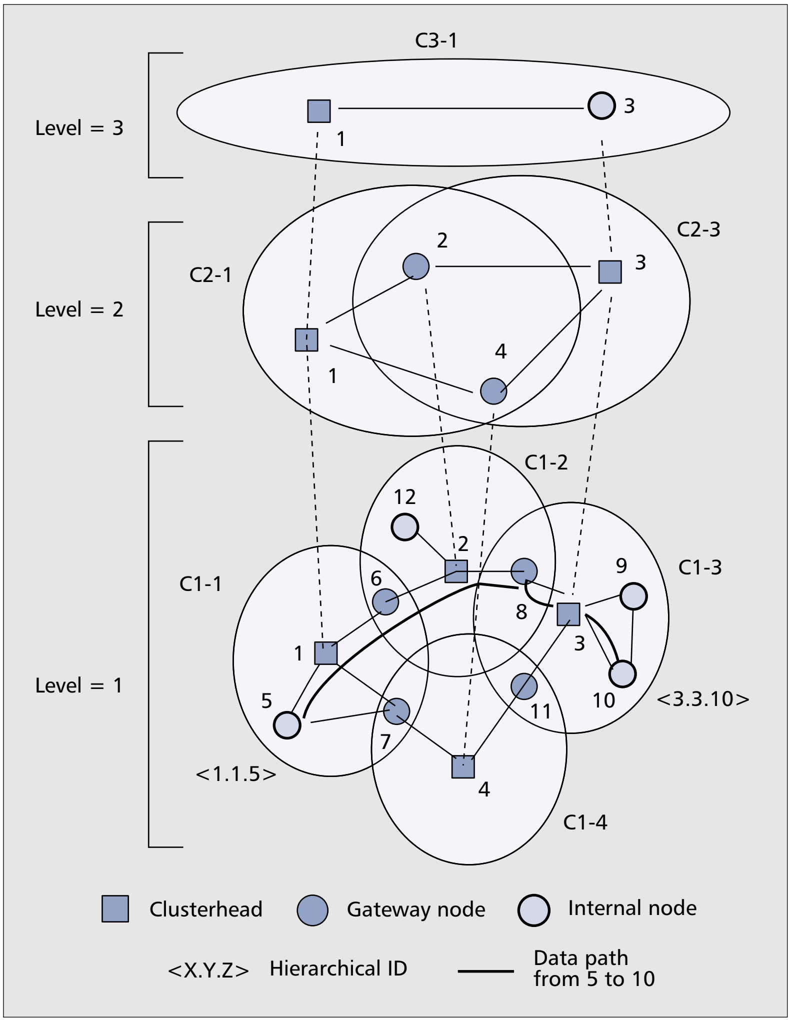
The major advantage of CGSR is that it can greatly reduce the routing table size compared to DV protocols. Only one entry is needed for all nodes in the same cluster. Thus, the broadcast packet size of routing table is reduced. These features make a DV routing scale to large network size. Although an additional cluster member table is required at each node, its size only decided by the number of clusters in the network. The drawback of CGSR is the difficulty of maintaining the cluster structure in a mobile environment. The LCC clustering algorithm introduces additional overhead and complexity in the formation and maintenance of clusters.

Hierarchical State Routing

Hierarchical State Routing (HSR) [26] is a multilevel clustering-based LS routing protocol. It maintains a logical hierarchical topology by using the clustering scheme recursively. Nodes at the same logical level are grouped into clusters. The elected clusterheads at the lower level become members of the next higher level. These new members in turn organize themselves in clusters, and so on. The goal of clustering is to reduce routing overhead (i.e., routing table storage, processing, and transmission) at each level. An example of a three-level hierarchical structure is demonstrated in Fig. 4. Generally, there are three kinds of nodes in a cluster: clusterheads (e.g., nodes 1, 2, 3, and 4), gateways (e.g., nodes 6, 7, 8, and 11), and internal nodes (e.g., nodes 5, 9, and 10). A clusterhead acts as a local coordinator for transmissions within the cluster.

HSR is based on LS routing. At the first level of clustering (also the physical level), each node monitors the state of the link to each neighbor (i.e., link up/down and possibly QoS parameters, e.g., bandwidth) and broadcasts it within the cluster. The clusterhead summarizes link state information within its cluster and propagates it to the neighbor cluster heads (via the gateways). The knowledge of connectivity between neighbor clusterheads leads to the formation of level 2 clusters. For example, as shown in Fig. 4, neighbor clusterheads 1 and 2 become members of the level 2 cluster C2. Link state entries at level 2 nodes contain the "virtual" links in C2. A "virtual" link between neighbor nodes 1 and 2 consists of the level 1 path from clusterhead 1 to clusterhead 2 through gateway 6. The virtual link can be viewed as a "tunnel" implemented through lower level nodes. Applying the aforementioned clustering procedure recursively, new cluster heads are elected at each level, and become members of the higher-level cluster. If QoS parameters are required, the clusterheads will summarize the information from the level they belong to and carry it into the higher level. After obtaining the link state information at one level, each virtual node floods it down to nodes of the lower-level clusters. As a result, each physical node has "hierarchical" topology information through the hierarchical address of each node (described below), as opposed to a full topology view as in flat LS schemes.

The hierarchy so developed requires a new address for each node, the hierarchical address. The node IDs shown in Fig. 4 (at level = 1) are physical (e.g., MAC layer) addresses. They are hardwired and unique to each node. In HSR, the *hierarchical ID* (HID) of a node is defined as the sequence of MAC addresses of the nodes on the path from the top hierar-



■ Figure 4. *HSR: an example of multilevel clustering.*

chy to the node itself. For example, in Fig. 4 the hierarchical address of node 5, HID(5), is <1,1,5>. The advantage of this hierarchical address scheme is that each node can dynamically and locally update its own HID on receiving the routing updates from the nodes higher up in the hierarchy. The hierarchical address is sufficient to deliver a packet to its destination from anywhere in the network using HSR tables. Gateway nodes can communicate with multiple clusterheads and thus can be reached from the top hierarchy via multiple paths. Consequently, a gateway has multiple hierarchical addresses, similar to a router in the wired Internet, equipped with multiple subnet addresses. These benefits come at the cost of longer (hierarchical) addresses and frequent updates of the cluster hierarchy and the hierarchical addresses as nodes move. In principle, a continuously changing hierarchical address makes it difficult to locate and keep track of nodes.

Zone Routing Protocol

The Zone Routing Protocol (ZRP) [7] is a hybrid routing protocol that combines both proactive and on-demand routing strategies and benefits from advantages of both types. The basic idea is that each node has a predefined *zone* centered at itself in terms of number of hops. For nodes within the zone, it uses proactive routing protocols to maintain routing information. For those nodes outside of its zone, it does not maintain routing information in a permanent base. Instead, on-demand routing strategy is adopted when interzone connections are required.

The ZRP protocol consists of three components. Within the zone, proactive *Intrazone Routing Protocol (IARP)* is used to maintain routing information. IARP can be any LS routing or distance vector routing depending on the implementation. For nodes outside the zone, reactive *Interzone Routing Protocol (IERP)* is performed. IERP uses the *route query* (RREQ)/*route reply* (RREP) packets to discover a route in a way similar to typical on-demand routing protocols. IARP always provides a route to nodes within a node's zone. When the intended destination is not known at a node (i.e., not in its IARP routing table), that node must be outside of its zone. Thus, a RREQ packet is broadcast via the nodes on the border of the zone. Such a RREQ broadcast is called *Bordercast Resolution Protocol* (BRP). Route queries are only broadcast from one node's border nodes to other border nodes until one node knows the exact path to the destination node (i.e., the destination is within its zone). The hybrid proactive/reactive scheme limits the proactive overhead to only the size of the zone, and the reactive search overhead to only selected border nodes. However, potential inefficiency may occur when flooding of the RREQ packets goes through the entire network.

Landmark Ad Hoc Routing Protocol

Landmark Ad Hoc Routing Protocol (LANMAR) [8, 27] is designed for an ad hoc network that exhibits group mobility. Namely, one can identify logical subnets in which the members have a commonality of interests and are likely to move as a group (e.g., a brigade or tank battalion in the battlefield). LANMAR uses an IP-like address consisting of a group ID (or subnet ID) and a host ID: < *GroupID, HostID* >. LANMAR uses the notion of *landmarks* to keep track of such logical groups. Each logical group has one dynamically elected node serving as a *landmark*. A global distance vector mechanism (e.g., DSDV [28]) propagates the routing information about all the landmarks in the entire network. Furthermore, LANMAR works in symbiosis with a local scope routing scheme. The local routing scheme can use the flat proactive protocols mentioned previously (e.g., FSR). FSR maintains detailed routing information for nodes within a given scope *D* (i.e., FSR updates propagate only up to hop distance *D*). As a result, each node has detailed topology information about nodes within its local scope and has a distance and routing vector to all landmarks. When a node needs to relay a packet to a destination within its scope, it uses the FSR routing tables directly. Otherwise, the packet will be routed toward the landmark corresponding to the destination's logical subnet, which is read from the logical address carried in the packet header. When the packet arrives within the scope of the destination, it is routed using local tables (that contain the destination), possibly without going through the landmark.

LANMAR reduces both routing table size and control overhead effectively through the truncated local routing table and "summarized" routing information for remote groups of nodes. In general, by adopting different local routing schemes [12], LANMAR provides a flexible routing framework for scalable routing while still preserving the benefits introduced by the associated local scope routing scheme.

	CGSR	HSR	ZRP	LANMAR
Hierarchy	Explicit two levels	Explicit multiple levels	Implicit two levels	Implicit two levels
Routing philosophy	Proactive, distance vector	Proactive, link state	Hybrid, DV and LS	Proactive, DV and LS
Loop-free	Yes	Yes	Yes	Yes
Routing metric	Via critical nodes	Via critical nodes	Local shortest path	Local shortest path
Critical nodes	Yes (clusterhead)	Yes (clusterhead)	No	Yes (landmark)
Storage complexity	$O(N/M)$	$O(M*H)$	$O(L) + O(e)$	$O(L) + O(G)$
Comm. complexity	$O(N)$	$O(M*H)$	$O(N)$	$O(N)$

■ Table 2. *Characteristics of hierarchical routing protocols.*

Comparisons of Hierarchical Routing Protocols

Table 2 summarizes the features of the four hierarchical routing protocols. Some symbols used in the table are N, the total number of mobile nodes in the network; M, the average number of nodes in a cluster; L, the average number of nodes in a node's local scope, which is used by both ZRP and LANMAR and is given here an identical scope size (r hops). The difference between M and L is that M usually only includes one-hop nodes, while L includes nodes up to r hops. The relation between M and L is $L = r^2 * M$. Also in the table, H is the number of hierarchical levels of HSR. G is the number of logical groups in LANMAR. The number of communication pairs is denoted as e. The storage and communication complexity have the same definitions as given in an earlier section.

The explicit hierarchical protocols CGSR and HSR force a path to go through some critical nodes like clusterheads and gateways, leading to possibly suboptimal paths. The two implicitly hierarchical protocols ZRP and LANMAR use a shortest path algorithm at each node. However, LANMAR guarantees shortest paths only when destinations are within the scope. For remote nodes, though data packets are first routed towards remote landmarks through shortest paths, extra hops may be traveled before a destination is hit. Similarly, ZRP does not provide an overall optimized shortest path if the destination has to be found through IERP.

CGSR maintains two tables at each node, a cluster member table and a routing table. The routing table contains one route to each cluster (actually clusterhead). Its storage complexity is $O(N/M)$. For the cluster member table, again only one entry is needed for each cluster. Thus, the storage complexity of CGSR is $O(N/M)$. In HSR, nodes at different levels have different storage requirements. The worst case occurs at the top level. The top-level nodes have to maintain a routing table of its clusters at each level. Thus, its storage complexity is $O(M*H)$. ZRP has separate tables for IARP and IERP. IARP is proactive and its storage complexity is $O(L)$. IERP is on-demand routing; thus, the table size depends on traffic pattern, leading to storage on the order of $O(L) + O(e)$. In LANMAR routing, each node also keeps two routing tables. One is a local routing table keeping track of all nodes in the scope. The other is a distance vector routing table maintaining paths to all landmarks. Thus, its storage complexity is $O(L) + O(G)$. Usually, the number of groups (G) is small (comparing to network size N). For an example of a simple network with equal partitions, when group size is 25 nodes, a 100-nodes network has four groups. A 1000-node network generates 40 groups.

The communication complexity of CGSR is $O(N)$ since the routing table and cluster member table have to be propagated throughout the whole network. Link updates in HSR are propagated along the hierarchical tree. In the worst case, if the top-level clusterhead is changed, corresponding worst-case communication complexity is $O(M*H)$. The worst case in ZRP occurs when a link change requires rediscovery of a new route over the entire network; thus, communication complexity is $O(N)$. In LANMAR, though the local proactive protocol has communication complexity in the order of $O(L)$, the total complexity is still $O(N)$ as the landmark distance vectors have to be propagated throughout the whole network.

The comparisons of the storage and communication complexities show that hierarchical routing protocols maintain smaller routing tables compared to flat proactive routing protocols. Even though the basic protocols have equivalent communication complexity as in flat routing, routing overhead is greatly reduced because smaller message size is used. For example in HSR, the storage $O(M*H)$ can be expressed as $O(M*\log N/\log M)$ (because the total number of nodes N can be expressed as $O(M^H)$) and the routing overhead is $O((M*\log N/\log M)^2)$, and in LANMAR, routing overhead is $O((L + G)*N)$. Both are smaller than $O(N^2)$ in flat LS routing. Reduction in overhead greatly improves hierarchical routing protocol scalability to large network sizes.

However, in the face of mobility, explicit cluster-based hierarchical protocols will induce additional overhead in order to maintain the hierarchical structure. HSR further requires complex management for HID registrations and translations [29]. This will not be the case for the "implicitly hierarchical" ZRP and LANMAR.

Both ZRP and LANMAR use proactive routing for local operations. However, they differ in outside scope routing. ZRP adopts an on-demand scheme and LANMAR uses a proactive scheme. Thus, when network size increases, so destinations are more likely to be outside the local scope, ZRP's behavior becomes similar to on-demand routing with unpredictable large overhead, while LANMAR has the advantage that the landmark distance vector is small and grows slowly. LANMAR greatly improves routing scalability to large MANETs. The main limitation of LANMAR is the assumption of group mobility.

Geographic Position Information Assisted Routing

The advances in the development of GPS nowadays make it possible to provide location information with a precision within a few meters. It also provides universal timing. While location information can be used for directional routing in distributed ad hoc systems, the universal clock can provide global synchronizing among GPS equipped nodes. Research has shown that geographical location information can improve routing performance in ad hoc networks. Additional care must be taken into account in a mobile environment, because locations may not be accurate by the time the information is used. All the protocols surveyed below assume that the nodes know their positions.

Geographic Addressing and Routing

Geographic Addressing and Routing (GeoCast) [30] allows messages to be sent to all nodes in a specific geographical area using geographic information instead of logical node addresses. A geographic destination address is expressed in three ways: point, circle (with center point and radius), and polygon (a list of points, e.g., $P(1)$, $P(2)$, …, $P(n-1)$, $P(n)$, $P(1)$). A point is represented by geographic coordinates (latitude and longitude). When the destination of a message is a polygon or circle, every node within the geographic region of the polygon/circle will receive the message. A geographic router (*GeoRouter*) calculates its *service area* (geographic area it serves) as the union of the geographic areas covered by the networks attached to it (Fig. 5). This service area is approximated by a single closed polygon. GeoRouters exchange service area polygons to build routing tables. This approach builds hierarchical structure (possibly wireless) consisting of GeoRouters. The end users can move freely about the network.

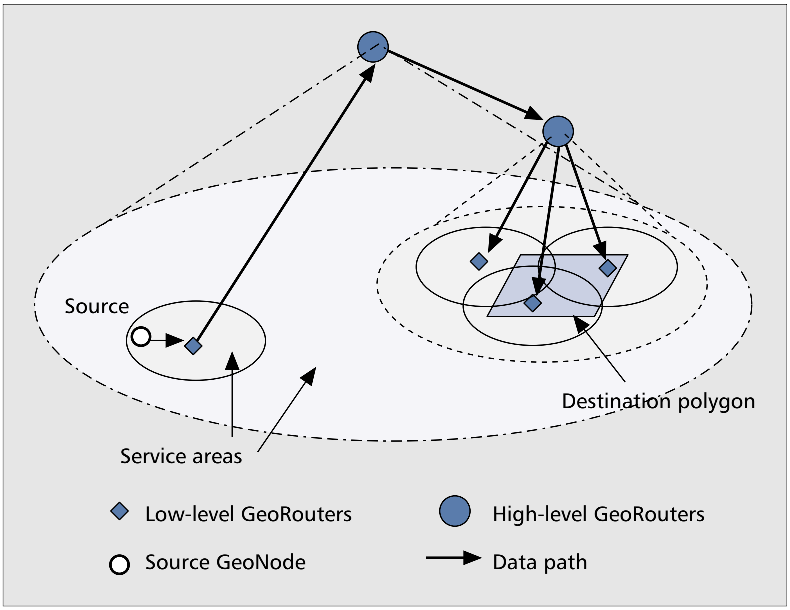


Figure 5. *An example of GeoCast.*

Data communication starts from a computer host capable of receiving and sending geographic messages (*GeoHost*). Data packets are then sent to the local *GeoNode* (residing in each subnet), which is responsible for forwarding the packets to the local GeoRouter. A GeoRouter first checks whether its service area intersects the destination polygon. As long as a part of the destination area is not covered, the GeoRouter sends a copy of the packet to its parent router for further routing beyond its own service area. Then it checks the service area of its child routers for possible intersection. All the child routers intersecting the target area are sent a copy of the packet. When a router's service area falls within the target area, the router picks up the packet and forwards it to the GeoNodes attached to it. Figure 5 illustrates the procedure of routing over GeoRouters.

As GeoCast is designed for group reception, multicast groups for receiving geographic messages are maintained at the GeoNodes. The incoming geographic messages are stored for a lifetime (determined by the sender) and during the time, they are multicast periodically through assigned multicast address. Clients at GeoHosts tune into the appropriate multicast address to receive the messages.

Location-Aided Routing

The Location-Aided Routing (LAR) protocol presented in [31] is an on-demand protocol based on source routing. The protocol utilizes location information to limit the area for discovering a new route to a smaller *request zone*. As a consequence, the number of route request messages is reduced.

The operation of LAR is similar to DSR [21]. Using location information, LAR performs the route discovery through *limited flooding* (i.e., floods the requests to a request zone). Only nodes in the request zone will forward route requests. LAR provides two schemes to determine the request zone.

Scheme 1: The source estimates a circular area (*expected zone*) in which the destination is expected to be found at the current time. The position and size of the circle is calculated based on the knowledge of the previous destination location, the time instant associated with the previous location record, and the average moving speed of the destination. The smallest rectangular region that includes the expected zone and the source is the request zone (Fig. 6a). The coordinates of the four corners of the zone are attached to a route request by the source. During the route request flood, only nodes inside the request zone forward the request message.

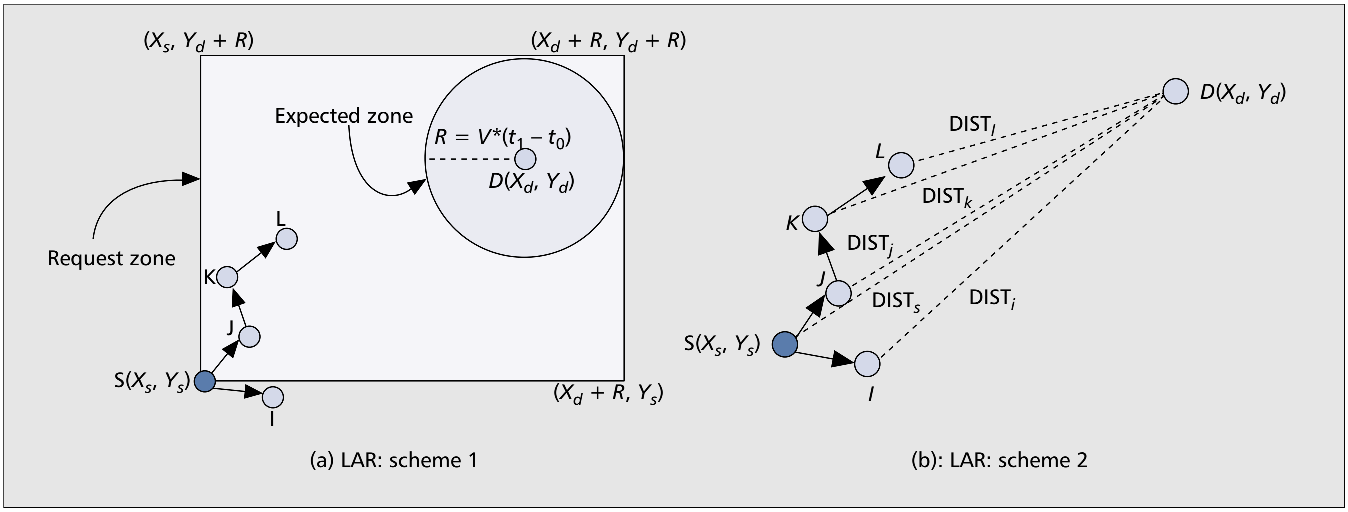
Scheme 2: The source calculates the distance to the destination based on the destination location known to it. This distance, along with the destination location, is included in a route request message and sent to neighbors. When a node receives the request, it calculates its distance to the destination. A node will relay a request message only if its distance to the destination is less than or equal to the distance included in the request message. For example, in Fig. 6b, nodes I and J will forward the requests from S. Before a node relays the request, it updates the distance field in the message with its own distance to the destination.

Distance Routing Effect Algorithm for Mobility

Distance Routing Effect Algorithm for Mobility (DREAM) [32] is a proactive routing protocol using location information. It provides distributed, loop-free, multipath routing and is able to adapt to mobility. It minimizes the routing overhead by using two new principles for the routing update frequency and message lifetime. The principles are *distance effect* and *mobility rate*. With the distance effect, the greater the distance separating two nodes, the slower they appear to be moving with respect to each other. With the mobility rate, the faster a node moves, the more frequently it needs to advertise its new location. Using the location information obtained from GPS, each node can realize the two principles in routing.

In DREAM, each node maintains a *location table (LT)*. The table records locations of all the nodes. Each node periodically broadcasts control packets to inform all other nodes of its location. The distance effect is realized by sending more frequently to nodes that are more closely positioned. In addition, the frequency of sending a control packet is adjusted based on its moving speed.

With the location information stored at routing tables, data packets are partially flooded to nodes in the direction of the destination. The source first calculates the direction toward the destination, then it selects a set of one-hop neighbors that are located in the direction. If this set is empty, the data is flooded to the entire network. Otherwise, the set is enclosed in the data header and transmitted with the data. Only nodes specified in the header are qualified to receive and process the data packet. They repeat the same procedure by selecting their own set of one-hop neighbors, updating the data header, and sending the packet out. If the selected set is empty, the data packet is dropped. When the destination receives the



(a) LAR: scheme 1

(b): LAR: scheme 2

■ Figure 6. *LAR: limited flooding of route request: a) scheme 1: expected zone; b) scheme 2: closer distances.*

data, it responses with an ACK to the source in a similar way. However, the destination will not issue an ACK if the data is received via flooding. The source, if it does not receive an ACK for data sent through a designated set of nodes, retransmits the data again by pure flooding.

Greedy Perimeter Stateless Routing

Greedy Perimeter Stateless Routing (GPSR) [33] is a routing protocol that uses only neighbor location information in forwarding data packets. It requires only a small amount of per-node routing state, has low routing message complexity, and works best for dense wireless networks. In GPSR, beacon messages are periodically broadcast at each node to inform its neighbors of its position, which results in minimized one-hop-only topology information at each node. To further reduce the beacon overhead, the position information is piggybacked in all the data packets a node sends. GPSR assumes that sources can determine through separate means the location of destinations and include such locations in the data packet header. A node makes forwarding decisions based on the relative position of destination and neighbors.

GPSR uses two data forwarding schemes: *greedy forwarding* and *perimeter forwarding*. The former is the primary forwarding strategy, while the latter is used in regions where the primary one fails. Greedy forwarding works this way: when a node receives a packet with the destination's location, it chooses from its neighbors the node that is geographically closest to the destination and then forwards the data packet to it. This local optimal choice repeats at each intermediate node until the destination is reached. When a packet reaches a dead end (i.e., a node whose neighbors are all farther away from the destination than itself), perimeter forwarding is performed.

Before performing the perimeter forwarding, the forwarding node needs to calculate a *relative neighborhood graph (RNG)*, that is, for all the neighbor nodes, the following inequality holds:

$$\forall w \neq u, v : d(u, v) \leq \max[d(u, w), d(v, w)], \qquad (1)$$

where, u, v and w are nodes, and $d(u, v)$ is the distance of edge (u, v). A distributed algorithm of removing edges violating Inequality 1 from the original neighbor list yields a network without crossing links and retaining connectivity.

Perimeter forwarding traverses the RNG using the right-hand rule hop by hop along the perimeter of the region. During perimeter forwarding, if the packet reaches a location that is closer to the destination than the position where the previous greedy forwarding of the packet failed, the greedy process is resumed. Possible loops during perimeter forwarding occur when the destination is not reachable. These will be detected and packets dropped. In the worst case, GPSR will possibly generate a very long path before a loop is detected.

Comparisons of Geographic Position Assisted Routing

With the knowledge of node locations, routing can be more effective and scalable in the realm of routing philosophy at the cost of the overhead incurred by exchanging coordinates. Key characteristics and properties of the protocols are summarized in Table 3. The same notations used in previous tables are used here.

GeoCast integrates the physical location into routing and addressing in the network design, and provides effective group communication to a geographic region. The hierarchical arrangement of GeoRouters based on the nested service areas reduces the size of the routing tables. LAR inherits the bandwidth saving of on-demand routing when there is no data to send. Moreover, it reduces DSR overhead by restricting the propagation of route request packets. However, when no path is available within the limited request zone or when location information is obsolete, LAR reverts to DSR's full area flooding. Geographic information is used only in flood reduction during route discovery. DREAM adopts a pure proactive approach for location updates at each node. It makes data forwarding decisions based on the geographic information carried by the data packet. Partial flooding of the data packet toward the direction of the destination results in multipath forwarding of copies of the original packets to the destination. This multiple delivery increases the probability of reception and protects DREAM from mobility. Both LAR and DREAM involve network-wise flooding to obtain location information. Thus, the control overhead increases when the network grows.

GPSR decouples geographic forwarding from location services. The routing overhead is limited to only periodic beacon massages and a small table for neighbor locations (compared to GeoCast and DREAM, where tables contain all the nodes in the network). Thus, GPSR achieves its scalability by being insensitive to the number of nodes in the network. However, additional overhead for location services (including location registration and location database lookup) must be considered when GPSR is used. Overhead is usually restricted because only destinations need to register to the location database and only sources need to query the database. Also, lookup is performed only once at the time communication starts. Ongoing connections will exchange location updates through the data packet headers. A scalable location lookup scheme can be found in [34].

	GeoCast	LAR	DREAM	GPSR
Support location propagation	Yes	Yes	Yes	No
Data forwarding by location	Yes	No	Yes	Yes
Routing philosophy	Proactive	On-demand	Proactive	Proactive (beacons only)
Sensitive to mobility	No	Yes	No	No
Routing metric	Shortest path	Shortest path	Shortest path	Closest distance
Loop-free	Yes	Yes	Yes	No
Worst case exists	No	Yes (full flooding)	No	Yes (loops and longer paths)
Multiple receivers	Yes	No	No	No
Storage complexity	$O(N)$	$O(N)$	$O(N)$	$O(M)$
Comm. complexity	$O(N)$	$O(e)$	$O(N)$	$O(M)$

Table 3. *Characteristics of GPS assisted routing.*

Conclusions

Protocols described in this article reveal the influence of underlying network structure on the routing protocols. They also show how the routing strategy differs in various design considerations. Flat proactive routing schemes with great advantages of immediate route availability and strong quality of service support have been studied using FSR, FSLS, OLSR, and TBRPF as examples. In these protocols, routing overhead is efficiently limited. FSR and FSLS achieve routing traffic reduction by selectively adjusting routing update frequencies. OLSR reduces both the size of routing packets and the number of nodes forwarding such packets. TBRPF limits the propagation of routing updates at leaf nodes and reports only differential information on source trees. Both OLSR and TBRPF work more efficiently in dense networks, while FSR and FSLS are more suitable for large diameter networks. The drawbacks of proactive schemes are constant bandwidth consumption due to periodic routing updates. On-demand routing schemes overcome this problem by searching for available routes to destinations only when needed, thus keeping bandwidth usage and routing table storage low. Two popular on-demand schemes, AODV and DSR, scale well for large networks when the communication pattern is sparse and mobility is low.

However, flat routing schemes only scale up to a certain degree: on one hand, routing table sizes in proactive schemes grow more than linearly when network size increases, resulting in overly congested channels and blocked data traffic; on the other hand, on-demand schemes incur huge amounts of flooding packets in large networks in search of destinations. The major advantage of hierarchical routing is the drastic reduction of routing table storage and processing overhead. CGSR, HSR, ZRP, and LANMAR only store routing entries about nearby nodes. CGSR and HSR organize the routing information dissemination and data forwarding in an explicit hierarchical approach through clusterheads. HSR can achieve multilevel hierarchy through a hierarchical address scheme at the cost of complex bookkeeping for logical addresses. LANMAR overcomes the limitations of address remapping by exploiting group mobility. The protocol reduces the routing table size greatly by keeping only a landmark for each remote group. LANMAR is suitable for large networks presenting the grouped motion feature.

With the help from GPS, directional data forwarding can reduce routing information propagation as shown in LAR and GPSR, and can improve data reception, for example, in GeoCast and DREAM. However, extra overhead is induced if mapping from addresses to locations is required. For example, DREAM generates larger overhead than GPSR due to node coordination dissemination. A possible solution is to use a scalable location lookup service. Moreover, location assisted routing protocols have great advantages in geographic related applications (e.g., group communications associated with a particular region, as seen in GeoCast).

We have reviewed a broad range of routing protocols designed for ad hoc networks. All protocols address the challenges of scalability. Since ad hoc networks will be used in various applications ranging from military to commercial, diversity in routing protocol designs is inevitable. In this article we provide descriptions of the protocols and discuss the differences among them, highlighting particular important features impacting scalability. No protocol emerges as the winner for all scenarios. All the previously mentioned schemes offer different, competitive, and complementary advantages, and are thus appropriate for different applications. Routing protocols capable of adapting to various application domains are desirable in future designs. With the recent rapid growth of ad hoc networks, future research will face even more challenges in the attempt to find the best match between scalable routing and media access control, security, and service management.

References

[1] S. Corson and J. Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," RFC 2501, Jan. 1999.

[2] S. Keshav, *An Engineering Approach to Computer Networking: ATM networks, the Internet, and the Telephone Network*, Ch. 11, Addison Wesley.

[3] K. Xu, X. Hong, and M. Gerla, "An Ad Hoc Network with Mobile Backbones," *Proc. IEEE ICC 2002*, New York, NY, Apr. 2002.

[4] S. R. Das, R. Castaneda, and J. Yan, "Simulation Based Performance Evaluation of Mobile, Ad Hoc Network Routing Protocols," *ACM/Baltzer MONET J.*, July 2000, pp. 179–89.

[5] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," *IEEE Pers. Commun.*, Apr. 1999, pp. 46–55.

[6] C. Santivanez, R. Ramanathan, and I. Stavrakakis, "Making Link-State Routing Scale for Ad Hoc Networks," *Proc. 2001 ACM Int'l. Symp. Mobile Ad Hoc Net. Comp.*, Long Beach, CA, Oct. 2001.

[7] Z. J. Haas and M. R. Pearlman, "The Performance of Query Control Schemes for the Zone Routing Protocol," *ACM/IEEE Trans. Net.*, vol. 9, no. 4, Aug. 2001, pp. 427–38.

[8] G. Pei, M. Gerla and X. Hong, "LANMAR: Landmark Routing for Large Scale Wireless Ad Hoc Networks with Group Mobility," *Proc. IEEE/ACM MobiHOC 2000*, Boston, MA, Aug. 2000, pp. 11–18.

[9] J. Broch *et al.*, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols," *Proc. ACM/IEEE MOBICOM '98*, Dallas, TX, Oct. 1998, pp. 85–97.

[10] S.-J. Lee, C.-K. Toh, and M. Gerla, "Performance Evaluation of Table-Driven and On-Demand Ad Hoc Routing Protocols," *Proc. IEEE PIMRC '99*, Osaka, Japan, Sept. 1999, pp. 297–301.

[11] C. E. Perkins, Ed., *Ad Hoc Networking*, Addison Wesley, 2001.

[12] X. Hong *et al.*, "Scalable Ad Hoc Routing in Large, Dense Wireless Networks Using Clustering and Landmarks," *Proc. ICC 2002*, New York, NY, Apr. 2002.

[13] A. Iwata *et al.*, "Scalable Routing Strategies for Ad-hoc Wireless Networks," IEEE JSAC, Aug. 1999, pp. 1369–79.

[14] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks," *Proc. ICC 2000*, New Orleans, LA, June 2000.

[15] P. Jacquet *et al.*, "Optimized Link State Routing Protocol," draft-ietf-manet-olsr-05.txt, Internet Draft, IETF MANET Working Group, Nov. 2000.

[16] A. Qayyum, L. Viennot, and A. Laouiti. "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks," INRIA res. rep. RR-3898, 2000
[17] B. Bellur and R. G. Ogier, "A Reliable, Efficient Topology Broadcast Protocol for Dynamic Networks," *Proc. IEEE INFOCOM '99*, New York, NY, Mar. 1999.
[18] R. G. Ogier *et al.*, "Topology Broadcast based on Reverse-Path Forwarding (TBRPF)," draft-ietf-manet-tbrpf-05.txt, INTERNET-DRAFT, MANET Working Group, Mar. 2002.
[19] C. E. Perkins and E. M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," *Proc. IEEE WMCSA '99*, New Orleans, LA, Feb. 1999, pp. 90–100.
[20] C.-K. Toh, "Associativity-Based Routing For Ad Hoc Mobile Networks," *WL Pers. Commun. J.*, Special Issue on *Mobile Networking and Computing Systems*, Kluwer, vol. 4, no. 2, Mar. 1997, pp. 103–39.
[21] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing*, T. Imielinski and H. Korth, Eds., Ch. 5, Kluwer, 1996, pp. 153–81.
[22] M. S. Corson, A. Ephremides, "A Distributed Routing Algorithm for Mobile Wireless Networks," *ACM/Baltzer WL Net. J.*, vol. 1, no. 1, Feb. 1995.
[23] V. D. Park and M.S. Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *Proc. IEEE INFOCOM '97*, Kobe, Japan, Apr. 1997, pp. 1405–13.
[24] S. R. Das, C. E. Perkins and E. M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks," *Proc. IEEE INFOCOM 2000*, Tel Aviv, Israel, Mar. 2000.
[25] C. -C. Chiang and M. Gerla, "Routing and Multicast in Multihop, Mobile Wireless Networks," *Proc. IEEE ICUPC '97*, San Diego, CA, Oct. 1997.
[26] G. Pei *et al.*, "A Wireless Hierarchical Routing Protocol with Group Mobility," *Proc. IEEE WCNC '99*, New Orleans, LA, Sept. 1999.
[27] M. Gerla, X. Hong, and G. Pei, "Landmark Routing for Large Ad Hoc Wireless Networks," *Proc. IEEE GLOBECOM 2000*, San Francisco, CA, Nov. 2000.
[28] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *Proc. ACM SIGCOMM '94*, London, U.K., Sept. 1994, pp. 234–44.
[29] G. Pei and M. Gerla, "Mobility Management in Hierarchical Multi-hop Mobile Wireless Networks," *Proc. IEEE ICCCN '99*, Boston, MA, Oct. 1999.
[30] J. C. Navas and T. Imielinski, "Geographic Addressing and Routing," *Proc. 3rd ACM/IEEE Intn'l. Conf. Mobile Comp. Net.*, Budapest, Hungary, Sept. 26–30, 1997.
[31] Y.-B. Ko and N. H. Vaidya, "Location-aided Routing(LAR) in Mobile Ad Hoc Networks," *ACM/IEEE Int'l. Conf. Mobile Comp. Net.*, 1998, pp. 66–75.
[32] S. Basagni *et al.*, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *ACM/IEEE Int'l. Conf. Mobile Comp. Net.*,1998, pp. 76–84.
[33] B. Karp and H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. 6th Annual Int'l. Conf. Mobile Computing and Networking (MobiCom 2000)*, Boston, MA, USA, 2000, pp. 243–54.
[34] J. Li *et al.*, "A Scalable Location Service for Geographic Ad Hoc Routing," *ACM Mobicom 2000*, Boston, MA.

Biographies

XIAOYAN HONG (hxy@cs.ucla.edu) is a Ph.D. candidate in the Computer Science Department of the University of California at Los Angeles (UCLA). She has been in the Wireless Adaptive Mobility Lab since 1997. Her research is in the area of mobile computing and wireless networking, particularly ad hoc networks and energy concerned sensor networks. Her research interests include mobility issues, MAC protocols, unicast and multicast routing protocols, scalability, wireless network QoS support, and adaptability.

KAIXIN XU (xkx@cs.ucla.edu) is a Ph.D. student in the Computer Science Department at UCLA. He joined the Network Research Lab of UCLA in 2000. His research focuses on the ad hoc wireless networking especially protocols at the MAC, network, and transport layers. His recently work includes enhancing TCP performance in multihop ad hoc networks, TCP fairness in IEEE 802.11 MAC-based ad hoc networks, as well as MAC protocols for utilizing directional antennas and mobility track.

MARIO GERLA (gerla@cs.ucla.edu) received a graduate degree in engineering from the Politecnico di Milano, in 1966, and his M.S. and Ph.D. degrees in engineering from UCLA in 1970 and 1973, respectively. He joined the faculty of the UCLA Computer Science Department in 1977. His research interests cover the performance evaluation, design, and control of distributed computer communication systems; high-speed computer networks; wireless LANs (Bluetooth); ad hoc wireless networks. He has been involved in the design, implementation, and testing of wireless ad hoc network protocols (channel access, clustering, routing and transport) within the DARPA WAMIS and GloMo projects, and most recently the ONR MINUTEMAN project. He has also carried out design and implementation of QoS routing, multicasting protocols, and TCP transport for the next-generation Internet. He is currently an associate editor of *IEEE Transactions on Networking.*