Independent Zone Routing: An Adaptive Hybrid Routing Framework for Ad Hoc Wireless Networks

Prince Samar, *Student Member, IEEE*, Marc R. Pearlman, *Member, IEEE*, and Zygmunt J. Haas, *Senior Member, IEEE*

***Abstract*—To effectively support communication in such a dynamic networking environment as the *ad hoc networks*, the routing framework has to be adaptable to the spatial and temporal changes in the characteristics of the network, such as traffic and mobility patterns. Multiscoping, as is provided through the concept of the Zone Routing Protocol (ZRP) for example, can serve as a basis for such an adaptive behavior. The Zone Routing framework implements hybrid routing by every network node proactively maintaining routing information about its local neighborhood called the routing zone, while reactively acquiring routes to destinations beyond the routing zone. In this paper, we propose the Independent Zone Routing (IZR) framework, an enhancement of the Zone Routing framework, which allows adaptive and distributed configuration for the optimal size of each node's routing zone, on the per-node basis. We demonstrate that the performance of IZR is significantly improved by its ability to automatically and dynamically tune the network routing operation, so as to flexibly and robustly support changes in the network characteristics and operational conditions. As a point of reference, through this form of adaptation, we show that the volume of routing control traffic overhead in the network can be reduced by an order of magnitude, under some set of parameter values. Furthermore, the adaptive nature of IZR enhances the scalability of these networks as well.**

***Index Terms*—Ad hoc network, adaptive routing, bordercast, hybrid routing, Independent Zone Routing, multiscope routing, proactive routing, reactive routing, routing framework, routing zone, send zone, Zone Routing Protocol.**

I. Introduction

As AD HOC networks do not rely on existing infrastructure and are self-organizing, they can be rapidly deployed to provide robust communication in a variety of hostile environments. This makes ad hoc networks very appropriate for a broad spectrum of applications ranging from providing tactical communication for the military and emergency response efforts to civilian forums such as convention centers and construction sites. With such diverse applicability, it is not difficult to envision ad hoc networks operating over a wide range of coverage areas, node densities, mobility patterns and traffic behaviors.

This potentially wide range of ad hoc network operating configurations poses a challenge for developing efficient routing protocols. On one hand, the effectiveness of a routing protocol increases as network topological information becomes more detailed and up-to-date. On the other hand, in an ad hoc network, mobility may cause frequent changes in the set of communication links of a node [27], requiring large and regular exchanges of control information among the network nodes. And if this topological information is used infrequently, the investment by the network may not pay off. Moreover, this is in contradiction with the fact that all updates in the wireless communication environment travel over the air and are, thus, costly in transmission resources.

Existing routing protocols for ad hoc networks can be classified either as proactive, reactive, or hybrid. Proactive or *table driven* protocols continuously evaluate the routes within the network, so that when a packet needs to be forwarded, the route is already known and can be immediately used. Examples of proactive protocols include DSDV [22], TBRPF [1], and WRP [14]. In contrast, reactive or *on-demand* protocols invoke a route determination procedure on an on-demand basis by flooding the network with the route query. Examples of reactive protocols include AODV [23], DSR [10], and TORA [15]. The on-demand discovery of routes can result in much less traffic than the proactive schemes, especially when innovative route maintenance schemes are employed. However, the reliance on flooding of the reactive schemes may still lead to a considerable volume of control traffic in the highly versatile ad hoc networking environment. Moreover, because this control traffic is concentrated during the periods of route discovery, the route acquisition delay can be significant. In Section II, we explore the third class of routing protocols—the hybrid protocols.

II. Protocol Hybridization

The diverse applications of ad hoc network pose a challenge for designing a single protocol that operates efficiently across a wide range of operational conditions and network configurations. Each of the purely proactive or purely reactive protocols described above performs well in a limited region of this range. For example, reactive routing protocols are well suited for networks where the "call to mobility"[1] ratio is relatively low. Proactive routing protocols, on the other hand, are well suited for networks where this ratio is relatively high. Fig. 1 shows the ad hoc network design space with node mobility and call rate as the two dimensions, and the general regions where each of these two kinds of protocols performs well. The performance of both of the protocol classes degrades when they are applied to regions of ad hoc network space between the two extremes.

Given multiple protocols, each suited for a different region of the ad hoc network design space, it makes sense to capitalize

Manuscript received August 3, 2002; revised July 10, 2003; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor N. Vaidya.

P. Samar and Z. J. Haas are with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14850 USA (e-mail: samar@ece.cornell.edu; haas@ece.cornell.edu).

M. R. Pearlman is with Kraken Ink, Clifton Park, NY 12065 USA (e-mail: pearlman@krakenink.com).

Digital Object Identifier 10.1109/TNET.2004.833153

[1]We define the "call to mobility ratio" as the rate of initial (nonrepair) route discoveries divided by the rate of changes in link status. This ratio provides a relative measure of network activity.

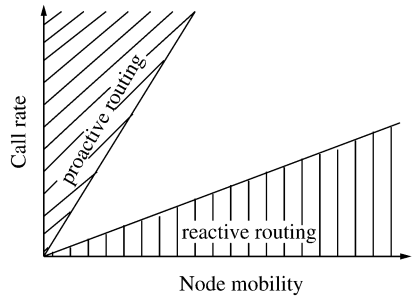

Fig. 1. Ad hoc network design space.

on each protocol's strengths by combining them into a single framework (i.e., hybridization). In the most basic hybrid framework, one of the protocols would be selected based on its suitability for the specific network's characteristics. Although not an elegant solution, such a framework has the potential to perform as well as the best suited protocol for any scenario, and may outperform either protocol over the entire ad hoc network design space. However, by not using both protocols together, this approach fails to capitalize on the potential synergy that would make the framework perform as well as *or better* than either protocol alone *for any given scenario*.

A more promising approach for protocol hybridization is to have the base protocols operate simultaneously, but with different "scopes" (i.e., hybridization through multiscoping). For the case of a two-protocol framework, protocol A would operate locally, while the operation of protocol B would be global. The key to this framework is that the local information acquired by protocol A is used by protocol B to operate more efficiently. Thus the two protocols reinforce each other's operation. This framework can be tuned to network behavior simply by adjusting the size of the protocol A's scope. In one extreme configuration, the scope of protocol A is reduced to nothing, leaving protocol B to run by itself. As the scope of protocol A is increased, more information becomes available to protocol B, thereby reducing the overhead produced by protocol B. At the other extreme, protocol A is made global, eliminating the load of protocol B altogether. So, at either extreme, the framework defaults to the operation of an individual protocol. In the wide range of intermediate configurations, the framework performs better than either protocol on its own. One example of the hybrid routing protocol class is the Zone Routing Protocol (ZRP), which will be described later.

III. Multiscope Routing

All else being equal, the value of routing information decreases with respect to the distance from the information source. For example, a node cannot compute its routes without knowing its neighbors. On the other hand, a node may make near-optimal forwarding decisions in spite of outdated or missing *distant* state information

This relationship between information value and distance is extremely valuable for routing protocol design. Simply distributing the same information at the same rate to all nodes in the network does not provide the most "bang for the buck." There is more value in providing nearby nodes with fresher and detailed information, at the expense of keeping more distant nodes less precisely informed.

To some extent, most basic protocols exhibit some degree of multiscope behavior. Many proactive routing protocols monitor the status of neighbor connectivity through neighbor broadcast HELLO beacons, which occur at a faster rate than the global link state (or distance vector) advertisements. In many reactive routing protocols, route discovery is based on querying on a global scale, whereas subsequent route repair utilizes local querying, constrained by a time-to-live (TTL) packet hop counter.

The high quality local route information provided by multiscope routing can be used to provide an assortment of new and enhanced services. By identifying overlaps in local connectivity, broadcast message can be distributed to all nodes more efficiently (e.g., OLSR's multipoint relay [2]). Moreover, local exchange of route information can be further exploited to provide a multicasting query distribution service, in which only a subset of the network's nodes need to be queried. Such a service can be applied to global route discovery, name-address translation, and general database lookups. In the case of global reactive protocols, once a route has been discovered, changes in local connectivity can be quickly identified, allowing for proactive route repair or route shortening [18]. Local multihop feedback of link layer acknowledgments can be used to discover and reliably use unidirectional links [19]. Intelligent node participation/sleep-mode algorithms can use local route information to determine if a node's absence would compromise network connectivity [16].

Perhaps the most familiar examples of multiscope routing are the various flavors of hierarchical routing (e.g., [9], [21]). In basic clustered routing, nodes are aggregated into subnets. Each node knows the topology of its subnet through a local proactive protocol. On a global scale, each subnet is represented by a clusterhead, which knows the connectivity to other subnets' clusterheads, but not the details of the other subnets' topologies. Nodes are located in the hierarchy through relative addressing that associates nodes with clusterheads. This two level example can be easily extended by grouping clusterheads into intermediate level subnets, thus creating a deeper hierarchy.

A variation on the clustered routing is landmark routing [3]. As in the previous example, nodes are organized into local subnets and assume hierarchical addresses. However, the clusterheads are replaced with globally visible landmarks, such that each node maintains a route to them using a global distance vector routing protocol. The role of the landmark is to identify the general location of the associated subnet toward which data packets can be forwarded.

Another hierarchical routing approach is based on the concept of a *core*. In this approach, local topology is proactively monitored in order to select a set of core nodes, such that every node has at least one core node neighbor. The purpose of the core nodes is to determine routes on behalf of the nodes that they cover, through global route discovery. Although route discovery occurs in the core, the core nodes apply knowledge of their local topology to construct routes that do not necessarily pass through the core. An example is the Dynamic Virtual Backbone scheme [11]. In addition to the basic core operation, CEDAR [28] introduces additional local scoping behavior by advertising higher quality links over greater distances.

Specialized node roles and regional node addressing help hierarchical routing protocols to scale with network size, especially when there is structure in the underlying network connectivity (for example, group mobility) to be exploited [12]. However, as network behavior becomes more dynamic and less coordinated, the overhead of the hierarchy maintenance (e.g., clusterhead election, node re-addressing) becomes a limiting factor for scaling. In addition, hierarchical routing suffers from single points of failure and may introduce uneven resource utilization, traffic hotspots and, in some cases, sub-optimal routing.

It is also possible to provide multiscope routing without the limitations and the overhead of hierarchy management. For example, in FSR [20] the distance (scope) of each link state update is a function of time, with longer distance updates occurring at lower frequencies. This provides each node with a fresh view of the surrounding topology, but a more dated view of farther network regions. The less accurate distant views effectively serve as landmarks, getting data packets forwarded in the right general direction. As the data packets approach the destination, the path becomes more accurate and the forwarding more refined.

Additional benefits of multiscope routing can be realized when larger scope protocols are able to exploit the information provided by a smaller scope. Two protocols that exhibit this kind of scope integration are OLSR and Zone Routing Protocol. In OLSR [2], an extended neighbor discovery provides each node with the topology of its surrounding two hops. This local information is used to provide an efficient global link state broadcast, based on multipoint relay. Multipoint relay identifies a "minimal" subset of neighbors needed to relay a message, such that all nodes two hops away will receive the message. In the case of Zone Routing [8], a proactive routing protocol is used to provide each node with a view of its surrounding "routing zone" topology. This local information enables an efficient route query distribution (*bordercasting*), which is used by a global reactive route discovery protocol. The global protocol efficiency increases with the size of the local "zone." The cost of local versus global scope can be traded off, and ultimately optimized, through the adjustment of a single parameter—the zone radius.

IV. Framework Tuning

The motivation behind hybrid routing and multiscope routing is to provide a framework that can be configured to match the network's operational condition and characteristics. Therefore, an integral component of the framework is a tuning mechanism. In particular, the three basic ingredients for tuning are: a means for measuring relevant network characteristics, a mapping of these measurements onto the framework's configuration, and a scheme to update the configuration of affected nodes.

The most basic approach to tuning is to determine the network characteristics and proper configuration offline, prior to the network deployment. Typically, the configurations are determined through network simulation and subsequent parameter optimizations. The nodes are loaded with the proper configuration and then activated. When it is not possible to pre-configure all nodes individually, a small number of nodes may be configured, and this configuration can be shared with other nodes as part of an automatic bootstrapping configuration procedure.

The main advantages of pre-configuration are that it requires limited network intelligence, low real-time processing overhead, and ensures stable and consistent configuration. However, for many applications, pre-configuration is not an option. Pre-configuration requires a central configuration authority, which may not exist for distributed applications such as ad hoc networks. In addition, the network characteristics may not be known apriori, or may vastly vary over time, preventing the offline analysis and reducing the effectiveness of the static configuration.

Ad hoc networks naturally lend themselves to dynamic reconfiguration. Through the course of normal operation, nodes directly measure (or infer) local network characteristics. Each node may use its own local measurements for independent self-configuration. Alternatively, the measurements could be relayed to a central configuration node or shared with surrounding nodes for a distributed configuration approach.

At first glance, centralized dynamic reconfiguration may appear to prevent inconsistent configuration, as is the case for centralized static configuration. However, the multihop nature of ad hoc networks makes it impossible to reliably perform tightly synchronized configuration updates for all nodes. This means that, for some period of time, the network could be in an inconsistent state. As this also affects distributed and independent reconfiguration, it is necessary that a dynamically tunable routing framework be able to deal with, support, and possibly exploit, nonuniform node configurations. The way in which a routing framework supports nonuniform configuration depends on its particular design. In Sections V–IX, we will explore how nonuniform configuration is supported in the Zone Routing framework.

With support for nonuniform configuration, reconfiguration decisions, associated measurements and control traffic can be localized, thereby providing for scalable framework tuning. Furthermore, the framework can be fine-tuned to adapt to changes in regional and even nodal behavior, rather than broadly tracking average network behavior. This can lead to significant performance improvements, especially in the case of networks where node behavior has regional dependencies.

V. Zone Routing Framework

The concepts of protocol hybridization and multiscope operation form the basis of the Zone Routing framework. At the local level, a proactive routing protocol provides a detailed and fresh view of each node's surrounding topology. The knowledge of the local topology is used to support services such as proactive route maintenance, unidirectional link discovery, and guided message forwarding and distribution. One particular message distribution service, called *bordercasting*, directs queries throughout the network. Bordercasting is used in place of traditional broadcasting to improve the efficiency of a global reactive routing protocol.

The benefits provided by the available topology information about the node's neighborhood, weighed against the overhead of proactively tracking this information, determine the optimal framework configuration. In this section, we describe the operation of the Zone Routing framework.

A. Local Proactive (Intrazone) Routing

In Zone Routing, the Intrazone Routing Protocol (IARP) proactively maintains routes to destinations within a local

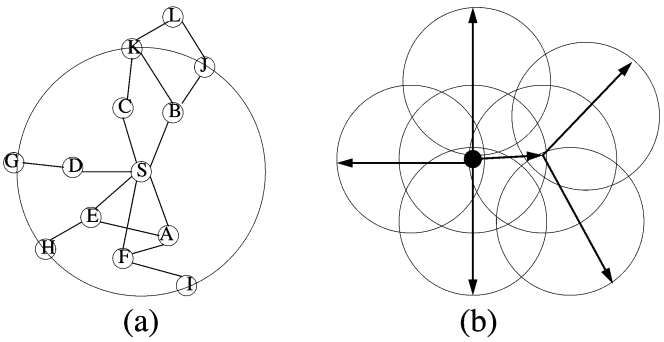

Fig. 2. (a) A routing zone of radius 2 hops. (b) Guiding the search in desirable directions (indicated by arrows).

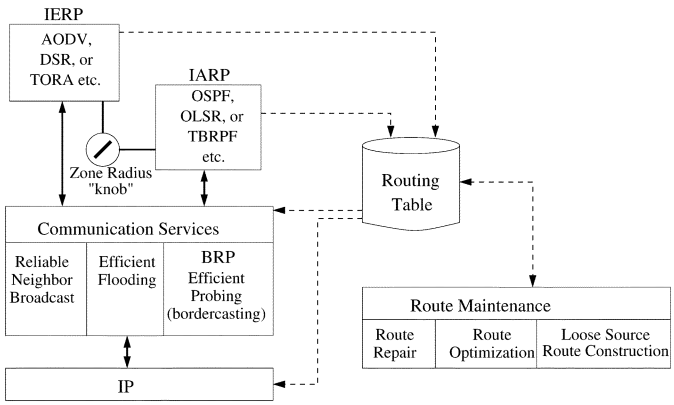

Fig. 3. Architecture of the Zone Routing framework.

neighborhood, which we refer to as a *routing zone*. More precisely, a node's routing zone is defined as a collection of nodes whose minimum distance in hops from the node in question is no greater than a parameter referred to as the *zone radius*. Note that each node maintains its own routing zone. An important consequence is that the routing zones of neighboring nodes overlap.

Fig. 2(a) illustrates the routing zone concept with a routing zone of radius 2 hops. This particular routing zone belongs to node S, with nodes A–K as its routing zone members. Node L, which is three hops away from S, is outside of S's routing zone. An important subset of the routing zone nodes is the collection of nodes whose minimum distance to the central node is exactly equal to the zone radius. These nodes are aptly named *peripheral nodes*. In our example, nodes G–K are peripheral nodes of node S. We typically illustrate a routing zone as a circle centered around the central node. However, one should keep in mind that the zone is not a description of a physical distance, but rather of nodal connectivity (hops).

The construction of a routing zone requires a node to first know who its neighbors are. A neighbor is defined as one with whom the node shares a direct communication link and is, thus, one hop away.[2] Identification of a node's neighbors may be provided directly by the media access control (MAC) protocol. Alternatively, neighbor discovery can be facilitated by periodic broadcasting of HELLO beacons. The reception (or quality of reception) of a HELLO beacon can be used to indicate the status of a connection to the beaconing neighbor.

Neighbor discovery information is used as a basis for IARP. IARP can be derived from globally proactive link state routing protocols that provide a complete view of the network connectivity (for example, OSPF [13], OLSR [2], or TBRPF [1], as shown in Fig. 3). The base protocol needs to be modified to ensure that the scope of the route updates is restricted to the radius of the node's routing zone [7]. In this paper, IARP is based on a simple, timer-based, link state protocol. To track the topology of R-hop routing zones, each node periodically broadcasts its link state for a depth of R hops (controlled by a time-to-live (TTL) field in the update message).

Note that routing zones are designed to be "circular" or "regular" in shape. This is because regular shaped routing zones are relatively easy to maintain. Moreover, ZRP fully exploits the regular shape of routing zones to effect efficient bordercasting as well as query control, as described next.[3]

B. Global Reactive (Interzone) Routing

Route discovery in the Zone Routing framework is distinguished from standard broadcast-based route discovery through a message distribution service known as *bordercasting* [5]. Rather than blindly broadcasting a route query from a neighbor to a neighbor, bordercasting allows the query to be directed outward, toward regions of the network (specifically, toward peripheral nodes) that have not yet been "covered" by the query; where a covered node is defined as one that belongs to the routing zone of a node that has received a route query. The query control mechanisms [4] reduce route query traffic by directing query messages outward from the query source and away from covered routing zones, as illustrated in Fig. 2(b).

A node can determine local query coverage by observing the addresses of neighbors that have forwarded the query. In the case of multiple channel networks, a node can typically detect only those query packets that have been directly forwarded to it. For single channel networks, the shared media enables a node to detect query packets forwarded within radio range. When a node identifies a query-forwarding neighbor, all known members of that neighbor's routing zone (i.e., members common to both the node's and the neighbor's routing zones) are marked as covered.

When a node has to relay a bordercast message, it again uses its routing zone topology to construct a bordercast tree that is rooted at itself and spans its uncovered peripheral nodes. The message is then forwarded to those neighbors in the bordercast tree. By virtue of the fact that this node has forwarded the query, all of its routing zone members are marked as covered. Therefore, a bordercasting node will not forward a query more than once.

Query detection can be enhanced by introducing a random delay prior to construction of the bordercast tree. During this time, the waiting node benefits from the opportunity to detect additional query coverage from other bordercasting neighbors. This, in turn, promotes a more thorough pruning of the bordercast tree, significantly decreasing the routing overhead. The use

[2]Note that we define the term *neighbor* to have a distinct meaning than the term *neighborhood*. A neighborhood refers to a set of topologically nearby nodes, potentially spanning multiple hops.

[3]As will be seen later, the Independent Zone Routing (IZR) framework also borrows this philosophy to define regular shaped routing zones.

of *short* random delays does not necessarily result in extra route discovery delay. Many route discovery protocols use random pre-transmission jitter to dilute the "instantaneous" channel load of neighboring query retransmissions. This forwarding jitter may be scheduled any time between query packet reception and query packet retransmission, including just prior to the bordercast tree construction.

Given the implementation of an underlying bordercast service, the operation of Zone Routing's reactive Interzone Routing Protocol (IERP) is quite similar to the standard route discovery protocols. An IERP route discovery is initiated when no route is locally available (i.e., through IARP) to the destination of an outgoing data packet. The source generates a route query packet, which is uniquely identified by a combination of the source node's address and request number. The query is then relayed to a subset of neighbors as determined by the bordercast algorithm. Upon receipt of a route query packet, a node records its ID in the packet. The sequence of recorded node IDs specifies an accumulated route from the source to the current node. If a valid route for the destination is not known (i.e., the destination is not in the node's routing zone and an active route does not appear in the node's route cache) then the node re-bordercasts the query. This process continues until the query reaches the destination or a node with a valid route to the destination. At that time, a route reply is sent back to the source, along the path specified by reversing the accumulated route. The operation of IERP is sufficiently general, so that many existing reactive protocols can be used as IERP with minimal modifications [6]. In particular, DSR [10] or AODV [23] can be incorporated into the Zone Routing framework as its reactive component (IERP), as shown in Fig. 3.

VI. Motivation for Independent Zones

According to the description of Zone Routing in Section V, every node participating in network routing needs to be tuned to the same zone radius value. This implies that before the network becomes operational, all the nodes in the network need to reach a consensus on the optimal value of the zone radius by some extraneous means. Also, any node that subsequently joins the network or undergoes a reboot needs to learn the common zone radius value for the network.

Most applications of ad hoc networks require that the network be formed and be operational quickly, and nodes be free to join and leave the system at their own will, without the need for any external configuration. In such networks, the constraint of having uniform zone radius for all the nodes may not be desirable. Having independently sized routing zones capability within the Zone Routing framework would allow nodes to dynamically and automatically configure their optimal zone radii in a distributed fashion, thus making the framework truly flexible.

Intuition, confirmed by simulation results of Zone Routing, dictates that high mobility and/or low call rates favor smaller zone radius. And vice versa, low mobility and/or high call rates favor larger zone radius. Now consider a network whose different parts have different mobility and call-rate patterns. Due to these differences, it may turn out that the nodes in different parts have different *optimal* zone radii. This motivates the development of the Zone Routing framework with independent zones capability, such that nodes in the network can possibly be assigned different zone radii. Such a framework should perform better than the single zone-size case, as it can be fine-tuned to the local conditions of the network. Furthermore, if the network characteristics change over time, such a framework can easily and quickly adapt to the changes, leading to optimal performance.

All these points motivate the development of Zone Routing with the capability to have independently sized routing zones. Such a framework would help determine the balance of proactive and reactive contributions which is appropriate for the specific characteristics and operational conditions of the network. The balance of proactive and reactive contributions can easily be adjusted over time and location by changing a single parameter—the zone radius of each node. Such a framework would not only reduce the routing overhead, but would be responsive to the needs of the network traffic as well.

VII. IZR Introduction

Protocol hybridization, multiscope operation, and dynamic reconfiguration, the key features of adaptable and scalable routing [26], form the basis of the Independent Zone Routing (IZR) framework. In the IZR framework, different nodes may have differently sized "routing zones." What does it mean for the nodes to have independent routing zones and how does such a routing protocol operate? Before exploring these issues, we begin by re-defining some terms in the IZR context.

- *Routing Zone or Receive Zone:* The neighborhood around each node about which a node proactively maintains routing information is called its routing zone. A node maintains this information by receiving proactive updates from these nodes in the neighborhood, hence this zone is also called its receive zone. This neighborhood consists of the set of all nodes, whose minimum distance, in hops, from the node is not more than the *zone radius*, R.
- *Send Zone*: All the nodes which require proactive updates from the node in question in order to maintain their intra-zone routing information belong to the node's send zone. A node is expected to broadcast proactive updates to the members of its send zone.
- *Peripheral Nodes:* The farthest members of a node's routing zone, whose minimum distance in hops from the node is R, are called its peripheral nodes.

We have seen that in the case of equally sized routing zones, a node broadcasts proactive routing information to all the members of its zone and also receives the same from each one of them. Thus, the send zone of a node is the same as its receive zone, when all nodes have equal zone radius. However, when the nodes in the network are allowed to have independent routing zones, this may not be the case.

In IZR, the routing zone or the receive zone is also *regular* in shape—that is, it can be represented by a circle of radius proportional to the zone radius of the node. All nodes with lesser number of hops from the node lie inside this circle and the peripheral nodes lie on the circle. In contrast, the send zones may not have such a regular shape. The members of the send zone of a particular node S consist of all nodes of which S is a routing zone member (thus those send zone nodes expect to receive routing updates from S). Because nodes in S's send zone may

have different receive zone radii, S's resulting send zone may be irregularly shaped. It is to be noted that the send zone may not even be a connected (contiguous) area.

Fig. 4 shows the routing or receive zones of nodes S, C, E, L, M, G, and H, which are regular in shape. As S is a routing zone member of C, E, L, M, G, and H, they belong to its send zone, which is irregular in shape.

VIII. IZR Details

The basic operation of IZR is similar to Zone Routing as discussed before. If a source node has a packet to send to a destination node which is not a member of its routing zone, it bordercasts a route query packet. However, due to the presence of unequal routing zones in the network, a somewhat different bordercasting scheme is used. As unequal routing zones imply that the send zone of a node may be irregular in shape, the Intrazone Routing Protocol (IARP) has to be modified in order to distribute the proactive updates in such a send zone. Below, we discuss the IARP and the BRP (Bordercast Resolution Protocol) for the IZR framework. Note that as the receive zones are still of regular shape, the operation of IERP remains the same as in regular ZRP.

A. Intrazone Routing Protocol (IARP)

Each node maintains proactive routing information about the members of its routing zone. For this to happen, each node needs to broadcast its proactive updates to the members of its send zone. As the send zone may be irregular in terms of the distance in hops to the "boundary" nodes, a node first needs to infer its send-zone's size and shape.

Consider a scenario where each node broadcasts "zone building packets" to all the members of its *routing zone*. As the routing zones are regular in shape, this can easily be done by setting the time-to-live (TTL) field of the packet equal to the zone radius R. The value in the TTL field is decremented by one each time the packet travels one hop. If the TTL value reaches zero, the packet is dropped, else it is rebroadcasted. In Fig. 4, nodes S, C, E, L, M, G, and H broadcast their zone building packets to their routing zone members (marked by a circle around each of them). Thus, each node will receive a zone building packet from all those nodes to whose routing zone it belongs. In particular, S would receive zone building packets from C, E, L, M, G, and H, as it is a member of each of their routing zones. Note that reception of a node's zone building packet means that the node is a send zone member. Thus, in Fig. 4, C, E, L, M, G, and H belong to S's send zone.

Based on the above, in the general case of independent zone radius, a node can find out the size and extent of its send zone. Thus, a node can determine the distance in hops to the farthest member of its send zone. The following scheme is used by the nodes to distribute the proactive updates in their send zones. Along with the zone radius field and the dynamic time-to-live TTL field, an update packet also has a field which contains the initial value of TTL at the source, the TTL_0 field. The source node sets the values of the TTL and the TTL_0 fields equal to the distance in hops to the farthest member of its send zone.

Initializing the TTL field as described above makes the updates reachable to all the members of a node's send zone. For example, in Fig. 4, S sets the TTL (and TTL_0) field equal to the distance in hops to one of the farthest members of its send zone (node C, E, or G). However, sending updates to *all* the nodes within this distance may lead to extraneous overhead. That is, updates would be broadcasted in the dashed area in the figure (which lies outside S's send zone). In order to reduce this overhead, each of the peripheral nodes of S maintains information about the members of S's send zone which lie further away from S than itself. That is, A maintains information about C, D about E, and F about G. S's other peripheral nodes, H, J, and K do not have any such nodes. Hence, when A, D, and F receive S's proactive updates, they send it toward C, E, and G respectively. If H, J, and K receive S's updates, they do not forward the update packets, thus reducing the extra overhead.

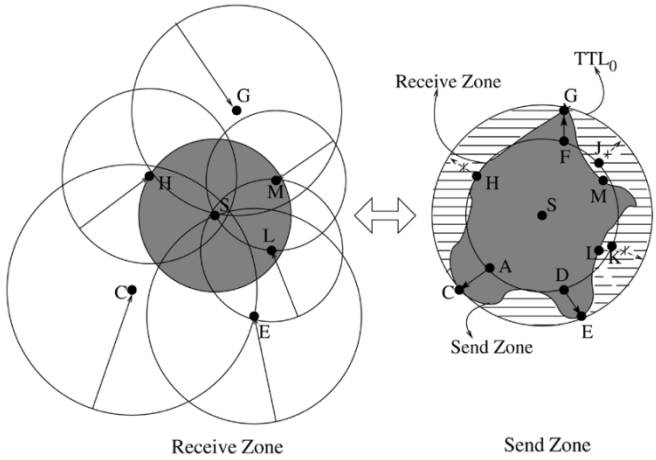

Fig. 4. The receive zone is regular in shape but the send zone may not be.

This information to reduce the overhead is maintained as follows. A node A maintains a list of all nodes for whom it serves as a peripheral node. For each node S in this list, A maintains another list called the *expecting_nodes_list*, which consists of all nodes C, whose zone building packets are received by A such that the current value of TTL in the packet is not less than the zone radius of S. (This implies that S lies in C's routing zone, or equivalently, C lies in S's send zone.)

Now a peripheral node H of a node S does not forward a proactive update packet originated at S, if H has no nodes in the *expecting_nodes_list* for node S. This reduces unnecessary traffic going beyond the peripheral nodes, if there are no nodes in that region which have S in its routing zone. Note that all these conditions can be checked by using the TTL, TTL_0, and the zone radius values that are available in the zone building or update packets.

Using the above scheme, each node in the network broadcasts proactive update packets by initializing the values of TTL and TTL_0 as above. Propagation of unnecessary update packets will be terminated by the peripheral nodes, if no nodes beyond them are *expecting* these packets, as determined by examining the maintained lists.

The "zone building packets" may, in practice, be combined with the proactive update packets to reduce the overhead. The broadcasting of the proactive updates by IARP can be based on one of the strategies proposed in [25] for more efficient performance.

B. Bordercast Resolution Protocol (BRP) and Query Control

With independently sized routing zones in the network, it is possible that some of the nodes in the bordercast tree of the source node have a routing zone which is small, so that it lies completely within the source node's routing zone. Such nodes' routing zones will not cover any newer, unexplored regions of the network, and these nodes will not be able to correctly judge who to forward the query packets to. In order to deal with situations like these, a different bordercasting mechanism is used.

The source node constructs a bordercast tree to its *uncovered* peripheral nodes and identifies the following two kinds of nodes.

- *Rebordercasting Node:* The node closest to the source node on the bordercast path[4] from the source node to a peripheral node, such that its routing zone extends beyond the source node's routing zone, is called a rebordercasting node of the source node corresponding to that peripheral node. For example, in Fig. 5, H is a rebordercasting node corresponding to O and N, J is a rebordercasting node corresponding to P, etc. However, B, with zone radius of 1, does not qualify as a rebordercasting node as its routing zone does not extend past its downstream peripheral nodes N, O, and P. Mathematically, the following condition holds true for a source node s and its rebordercasting node b:

$$\mathcal{H}(s,b) + \mathcal{R}(b) > \mathcal{R}(s) \tag{1}$$

  where $\mathcal{H}(s,b)$ is the minimum distance in hops between the source node s and the rebordercasting node b, and $\mathcal{R}(b)$ is the routing zone radius of b.
- *Forwarding Node:* Nodes lying on the bordercast path between the source node and a rebordercasting node belong to the set of forwarding nodes corresponding to that rebordercasting node. For example, B is a forwarding node corresponding to J and H, while A does not have any forwarding node. Note that if a node's routing zone is no larger than the routing zones of all its bordercast tree neighbors, the set of forwarding nodes is empty.

The following bordercasting mechanism is used by the nodes in order to guide a route query "outward," toward unexplored regions of the network [as in Fig. 2(b)].

1) Source node S constructs the bordercast tree to *uncovered* peripheral nodes.
2) S chooses rebordercasting nodes corresponding to each of its *uncovered* peripheral nodes.
3) S then sends the query packet to each of these rebordercasting nodes via the forwarding nodes, if any.
4) The rebordercasting nodes, on receiving the query packet, become bordercasting nodes and go back to step 1.

Fig. 5 shows the bordercast tree of the source node S, which has a zone radius of 3. Nodes A, B, C, and D are the bordercast tree neighbors of S. The zone radius of A is 3 and its zone extends beyond S's routing zone. However, as the zone radii of B, C, and D are small (1, 2, and 2, respectively), their routing zones lie completely inside S's routing zone. So, S examines the nodes which are two hops from it in the bordercast tree downstream from B, C, and D. It finds that the routing zones of E, F, G, H, and J include newer regions outside its own routing zone, and so they get selected as rebordercasting nodes. S then sends the route query packet to A, E, F, G, H, and J (via hop-by-hop relaying through forwarding nodes, if needed), which could, in turn, query unexplored regions of the network.

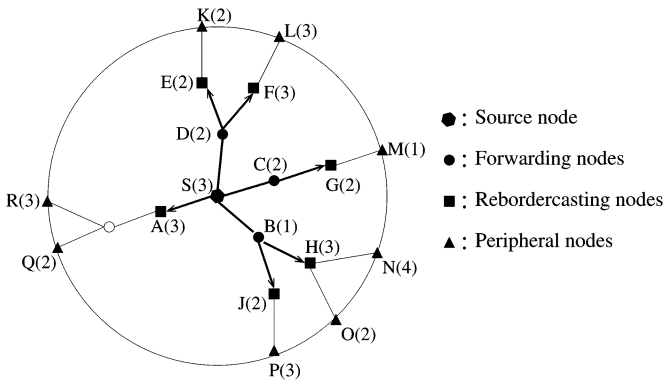

Fig. 5. The bordercast tree of the source node S. The zone radii are indicated in parentheses next to the node labels.

For query control, a node marks certain members of its zone as *covered* and tries to steer the query away from such nodes. Note that a node bordercasting a route query is in a position to make routing decisions on behalf of all the nodes in its routing zone. Therefore, a node receiving a route query marks the known members of the *last* bordercasting node's routing zone as covered. The following rules are used by a node to identify such covered nodes.

- A *rebordercasting node* marks
  - — the nodes lying in the intersection of its zone with the zone of the bordercasting node as covered, if the bordercasting node is a member of its zone.[5]
  - — the nodes lying in the intersection of its zone with the zone of the last forwarding node as covered, if the bordercasting node does not lie in its zone.[6]
- A *forwarding node* marks all the members of its zone as covered.

This mechanism ensures that the query always gets bordercasted by nodes whose routing zones cover newer, unexplored regions of the network.

The correctness of the IZR framework is proved in the Appendix.

C. Zone Radius Determination Algorithm

Another integral component of the IZR framework is the zone radius determination algorithm, which tunes the framework to the network characteristics and operating conditions. The following are some desirable features that one would like to incorporate in the design of this algorithm. The algorithm should be

[4]Bordercast path is a path on the bordercast tree.

[5]As the bordercasting node is a member of the rebordercasting node's routing zone, the rebordercasting node knows the bordercasting node's position relative to the other members of its routing zone and can, thus, infer the intersection of their routing zones.

[6]The last forwarding node will be a member of the rebordercasting node's routing zone and, thus, the rebordercasting node has the required information to mark the intersection of their routing zones as covered.

able to determine the optimal zone radius of each node in the network. Also, it should be quick in adapting the node configuration to any changes in the network characteristics. The algorithm should lead to minimal (if any) amount of extra overhead in the network. Further, there should not be any dependence on nodes with special roles (e.g., arbiter nodes determining the zone radius for the region), as this may give rise to single points of failure, excess power drainage and possible congestion around such nodes.

These features can best be achieved by having nodes independently configure their own zone radius based on local measurements. As discussed in Section IV, centralized configuration schemes are not desirable in ad hoc networks, particularly because they depend on special nodes for their operation. Distributed coordination efforts among multiple nodes (to either determine a common zone radius or individual zone radii for each member of the group) would certainly lead to an increase in the control traffic overhead due to the need for the nodes to communicate and coordinate among themselves. Such coordination efforts would also be slow in adapting to any changes in the network characteristics. The zone radius configuration scheme described below independently determines the optimal zone radius of each node by monitoring the control traffic passing through the node, and can fine-tune the framework to adapt to regional, and even nodal, behavior rather than broadly tracking average network behavior.

A hybrid of *Min Searching* and *Adaptive Traffic Estimation* schemes [17] is used to dynamically configure the optimal zone radius of each node in a distributed fashion.

The *Min Searching* scheme involves iteratively searching for the minima of the routing control traffic curve by incrementally increasing or decreasing the routing zone radius of a node by one hop. During each estimation interval, the amount of routing control traffic passing through the node is measured. If the amount of routing traffic in the current estimation interval is less than that in the previous interval, the zone radius is further incremented/decremented in the same direction. Otherwise, the direction of the zone radius change is reversed. The process continues until a zone radius R is detected that leads to minimum control traffic based on the following condition: $Z(R) < Z(R-1)$ and $Z(R) < Z(R+1)$, where $Z(R)$ represents the total control traffic corresponding to the zone radius of R hops. Fig. 6(a) illustrates the operation of Min Searching, where the algorithm starts at $t = 0$ and converges to the optimal value at $t = 4$.

The Min Searching scheme converges to a local minima, provided that the network characteristics do not change substantially during the search and that the estimation interval is long enough to provide a relatively stable measurement of the routing control traffic. Results in [17] show that the proactive traffic for a node is a nondecreasing function of the zone radius. Similarly, the reactive traffic is a nonincreasing function of the zone radius. Hence, the total control traffic, which is a sum of these two components, is a convex function. Thus, the local minima found by the Min Searching algorithm is, in fact, also the global minima. Note that it is possible that this minima may lie at one of the extremes, i.e., at a zone radius of one or at a zone radius equal to the network diameter.

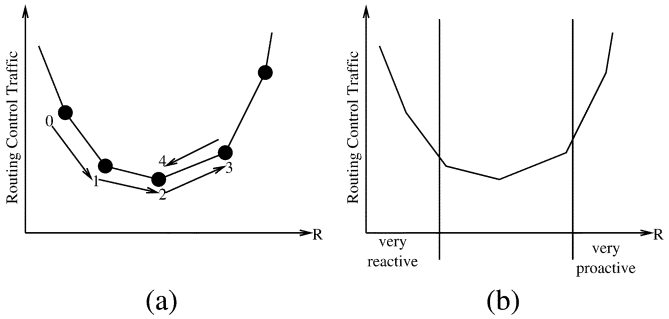

Fig. 6. (a) Min Searching. (b) Adaptive Traffic Estimation.

The *Adaptive Traffic Estimation* scheme tunes the framework by exploiting some special properties of the IZR routing control traffic components. When the zone radius is larger than the optimal zone radius and the corresponding amount of control traffic is significantly more than the optimal, the control traffic is dominated by the proactive IARP updates (i.e., reactive traffic/proactive traffic $\ll 1$). This is because increasing the zone radius increases the proactive traffic and decreases the reactive traffic. Similarly, when the zone radius is less than the optimal zone radius and the corresponding amount of control traffic is significantly more than the optimal, the control traffic is dominated by the reactive queries (i.e., reactive traffic/proactive traffic $\gg 1$). These regions are depicted in Fig. 6(b). The optimal zone radius lies between these two regions of proactive and reactive traffic component dominance, where the ratio of the two components is within the two extremes. The Adaptive Traffic Estimation scheme tries to track the optimal zone radius by iteratively increasing or decreasing the zone radius in order to reduce the reactive or the proactive traffic component dominance in the total routing overhead.

Let $\Gamma(R)$ be the ratio of the reactive (IERP) traffic to the proactive (IARP) traffic at zone radius R during a certain estimation interval, as measured at one network node. Adjustments to the zone radius are made by comparing this ratio with a predetermined threshold, Γ_{thres}. Intuitively, Γ_{thres} represents the ratio of the reactive traffic to the proactive traffic at the optimal zone radius. If $\Gamma(R) > \Gamma_{\text{thres}}$, the zone radius is increased by one hop to decrease the reactive traffic dominance. If $\Gamma(R) < \Gamma_{\text{thres}}$, the zone radius is decreased by one hop to decrease the proactive traffic dominance. However, changing the zone radius after each estimation interval could lead to too frequent adaptation of the zone radius, possibly resulting in network instability. Hence, a triggering mechanism is introduced by a multiplicative hysteresis term, so that if $\Gamma(R) > \Gamma_{\text{thres}} \cdot H$, the zone radius is increased by one hop; if $\Gamma(R) < \Gamma_{\text{thres}}/H$, the zone radius is decreased by one hop.

Note that in order to determine $\Gamma(R)$, a node measures the reactive and the proactive components of all the routing control traffic passing through itself during an estimation interval. As a node has to bordercast queries originated at other nodes, in addition to its own route queries, the zone radius it chooses affects the overhead associated with route queries originated at other nodes as well. The aim of the zone radius configuration algorithm is to minimize the overall routing overhead in the network.

A node can bordercast more efficiently with a larger zone radius, causing a decrease in the reactive control traffic. However, a larger zone radius implies maintaining topology information about more nodes, leading to an increase in the proactive control traffic. The chosen zone radius of a node should be able to efficiently bordercast the queries received (or originated) at a node, while at the same time, keeping the cost of maintaining the routing zone topology low. The Adaptive Traffic Estimation scheme tries to track this zone radius by adjusting the ratio of reactive traffic in a node's vicinity to its proactive traffic in order to bring it close to the threshold. This minimizes the total routing control traffic in the network.

A combination of Min Searching and Adaptive Traffic Estimation schemes is used for IZR zone radius determination. Initially Min Searching is applied, which starts from a zone radius of one and searches for the zone radius which would cause the least overhead. Once the minimum of the control traffic curve is found, a node sets Γ_{thres} equal to the ratio of the reactive component to the proactive component at the optimal zone radius. Our experience suggests that this ratio is within the vicinity of 1 for most networks. Min Searching is then replaced with Adaptive Traffic Estimation which adaptively tracks the changing characteristics of the network to adjust the node's zone radius. The Adaptive Traffic Estimation algorithm uses the Γ_{thres} that was determined during the Min Searching phase.

The two extreme cases of optimal zone radius equal to the network diameter and optimal zone radius of one need some extra attention. For example, consider a static network with a high call rate. According to the above description of the Adaptive Traffic Estimation scheme, the scheme needs to choose between the following two zone radii: zone radius equal to the network diameter (in which case only the proactive component exists) and zone radius equal to one less than the network diameter (in which case the reactive component dominates). The algorithm restricts its adjustment to these two radii, as they represent the boundaries of reactive and proactive dominance. Because the zone radius is integer valued, there are no other possible radii that lie between these boundaries. Similarly, consider a network with high mobility (relative to the call rate) so that the optimal zone radius is one. In this case, the two choices for the Adaptive Traffic Estimation algorithm are: zone radius of one (totally reactive) and zone radius of two (highly proactive due to the high mobility). For both these special boundary cases, the algorithm needs to switch to the Min Searching mode in order to decide which one of the two choices leads to lower routing control traffic.

While the algorithm is operating in the Min Searching mode, the estimate of Γ_{thres} is updated. When the minimum is identified by Min Searching, the value of the traffic ratio for that interval is passed to the Adaptive Traffic Estimation scheme. Due to the useful information provided by the Min Searching scheme, it may be beneficial to occasionally switch to Min Searching.[7] This can be done when a large change is detected in the magnitude of $\Gamma(R)$ (indicating a significant change in the network characteristics), or simply periodically.

[7]Note that Min Searching may be more expensive as it may need to evaluate the routing overhead at the neighboring zone radii in order to ascertain the minimum.

It can be shown that the zone radius determination scheme is stable in the sense that for a certain zone radius R_0 (which depends on the values of Γ_{thres}, H, and the network characteristics), the probability of a zone radius $R > R_0$ being chosen decreases and becomes vanishingly small for large R. Thus, we state the following theorem.

Therorem 1: The zone radius determination scheme for IZR is nondivergent.

The reader is referred to [24] for a proof of Theorem 1.

IX. Performance Evaluation

The OPNET$^{\text{TM}}$ simulation environment was used to simulate the IZR framework. Link-state-based IARP, described in [7], was used as the basis for the proactive component, and a source-route-based IERP, described in [6], was used as the reactive component. Neighbor discovery is based on the reception of $HELLO$ beacons transmitted at random intervals of mean T_{beacon}. If a new beacon fails to arrive within $2 \cdot T_{\text{beacon}}$ of the most recent beacon, a link failure is reported. We assume that neighbor discovery beacons are given the highest transmission priority and are not destroyed by collisions. This prevents the inaccurate reporting of link failures for the allowed $2 \cdot T_{\text{beacon}}$ window.

The network consists of 100 nodes spread randomly in an area of 1300×1300 meter^2 (unless noted otherwise). A node moves at a constant speed v and is assigned an initial direction θ, which is uniformly distributed between 0 and 2π. When a node reaches an edge of the square simulation region, it is reflected back into the coverage area by setting its direction to $-\theta$ (horizontal edges) or $\pi - \theta$ (vertical edges). The magnitude of its velocity is not altered. In the absence of packet collisions, we assume that background channel interference and receiver noise limit the transmission range of packets to a physical radius of 225 meters.

A node's session with a randomly chosen destination consists of sending a certain number of data packets; the number of data packets per session is Poisson distributed with an average of 10 packets. The interarrival time between sessions are exponentially distributed. The source of a particular session generates data packets at the constant rate of 16 packets per second, where the size of each packet is 1000 bits. Measurements of the routing control traffic are reported in terms of the number of the control traffic packets. The total routing overhead is viewed as the sum of the IARP and the IERP components. IARP traffic is generated based on changes in link status detected by a node [25]. The IERP component's traffic is constituted by the initial route query performed at the beginning of a session and any subsequent queries due to reported route failures.

Our primary performance evaluation metric is the routing control traffic in the network. Minimizing control traffic is an important goal for a routing protocol as it affects the performance of an ad hoc network in a number of ways. Smaller control traffic translates to lower power consumption, less congestion, smaller delays, reduced memory and processing requirements, and faster access to the communication channel. Performance of IZR is also presented in terms of other metrics like fraction of data packets delivered and route discovery delay.

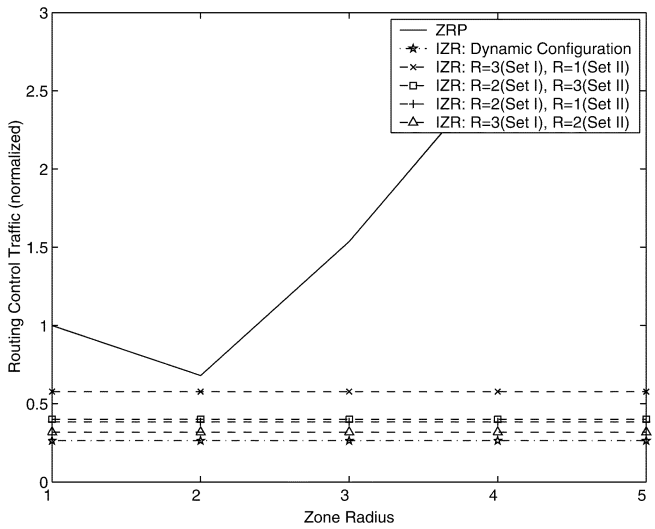

Fig. 7. Total routing overhead for IZR as compared to the different zone radii settings of ZRP.

As different simulation runs were performed for different zone radius settings and parameter values, the network behavior was made to remain exactly the same; i.e., the nodes move in exactly the same path and start sessions with the same nodes at the very same instants. No data was collected for the first 5 s of the simulations, while the initial intrazone route discovery process stabilized.

Fig. 7 shows the amount of the routing control traffic generated during a simulation duration of 180 s. The scenario consists of half of the nodes (Set I) moving at a constant speed (v) of 1 m/s and have a mean session interarrival delay (MSID) of 5 s. The other half (Set II) move at a speed of 10 m/s and have a mean session interarrival delay of 25 s. The vertical axis has been normalized by the total control traffic produced at the zone radius of one for the regular Zone Routing Protocol (which corresponds to simple flooding-based route discovery). From the plot, it can be seen that IZR with dynamic radius configuration leads to more than 60% reduction in routing control traffic as compared to the optimal setting of regular Zone Routing. The plot also shows the amount of routing control traffic for IZR with fixed but different zone radius (R) assignments for the two sets of nodes. The reduction in the control traffic for this case reinforces our intuition that different zone radii may be preferable for nodes with different characteristics.

For IZR in Fig. 7, a default value of $H = 12$ has been used. From our simulations with different values of H in the range 4 to 28, we found that the total routing control traffic only differed by about 6%. Further, H between 8 and 16 led to least control traffic. This is explained as follows. A low value of hysteresis (e.g., $H = 4$) implies a smaller range around the threshold Γ_{thres} over which variations can be tolerated without a change in the zone radius. Thus a low value of H causes relatively frequent changes in the zone radius, leading to increased control traffic. At the same time, a high value of hysteresis (e.g., $H = 28$) is not desirable either, as it makes the zone radius configuration algorithm slow in adapting to changes in the network characteristics, leading to sub-optimal performance.

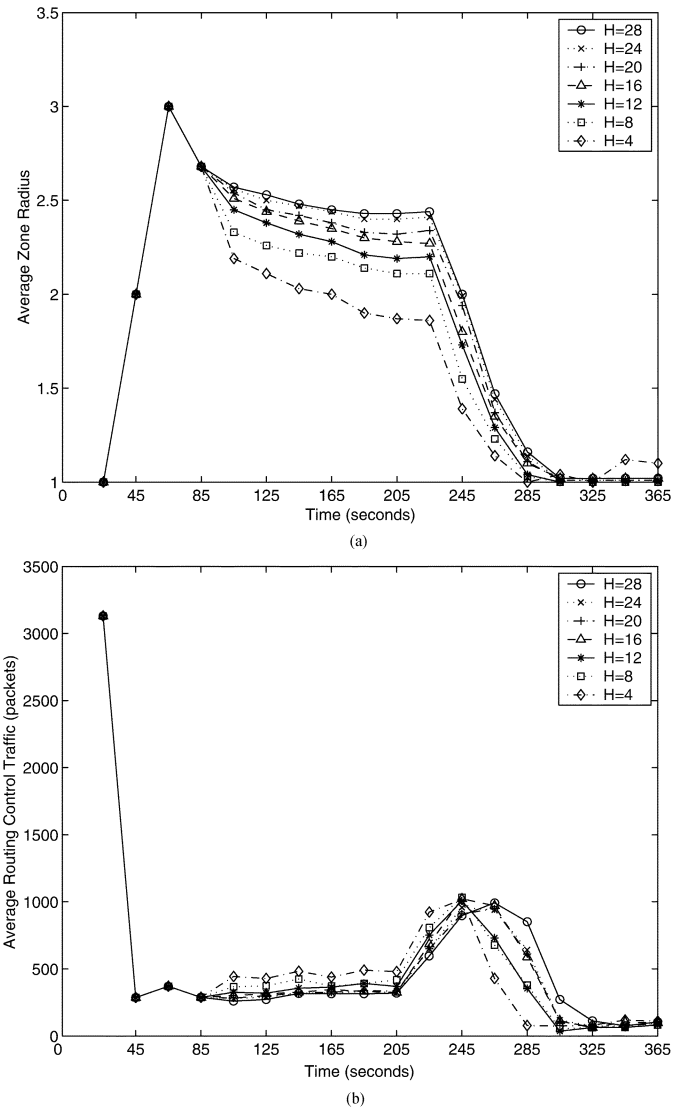

Fig. 8. The variation in (a) the average zone radius of the nodes and (b) routing control traffic, for different H values.

In order to study the adaptivity of the IZR framework to changes in network characteristics and its dependence on the parameter H, the following experiment was performed. Nodes in the network initially move with the velocity of 0.5 m/s and have MSID of 3 s. After 205 s of simulation time, the characteristics change to the velocity of 15 m/s and MSID of 200 s. These changes in the network characteristics correspond to a change in the optimal zone radius of the nodes from 2 or 3 (function of initial characteristics) to 1 (function of latter characteristics). The following results demonstrate the effectiveness of IZR in adaptively tracking these changes.

Fig. 8(a) shows how the average zone radius of the nodes changes as a function of time for the experiment described above. The points correspond to the average zone radius of the nodes during the last estimation interval of 20 s. Fig. 8(b) plots the corresponding average routing control traffic of the nodes during the interval. Initially, the Min Searching algorithm increases the zone radius until the optimal zone radius is found. Once the zone radius producing the minimum control traffic is

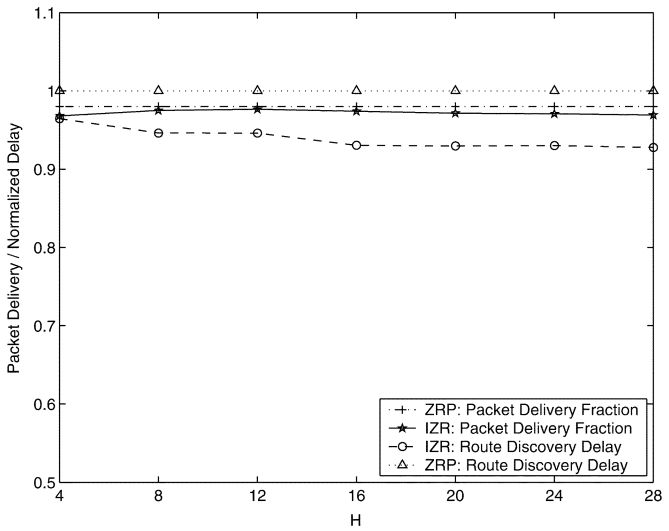

Fig. 9. Data Packet delivery fraction and route discovery delay (normalized) for IZR as compared to ZRP.

found, the Adaptive Traffic Estimation scheme assumes control and the average zone radius approximately lies between 2 and 3 depending upon the value of the hysteresis (H). When the network characteristics change after 205 s, the Adaptive Traffic Estimation scheme soon brings the zone radius of the nodes close to the new optimal value of 1.

In Fig. 8(b), initially the routing overhead decreases rapidly as the optimal zone radius is found by the Min Searching scheme[8]. It is to be noted that although the average zone radius varies for different values of the hysteresis (H), the actual routing overhead remains low until 205 s for all values of H. When the network characteristics change after 205 s and the current zone radius is no longer optimal, the routing control traffic increases. However, the Adaptive Traffic Estimation algorithm soon finds the new optimal zone radius and the routing control traffic decreases to a low level again. Comparing the curves corresponding to $H = 28$ and $H = 4$ in Fig. 8(a) and (b), one can see that a high value of H can be slow in adapting to changes in network characteristics, while a low value of H can lead to slightly higher routing overhead (as compared to other H values), for reasons explained before.

Next we look at the performance of IZR in terms of data packet delivery fraction and route discovery delay. Data Packet delivery fraction is defined as the ratio of the total number of data packets delivered at the destinations to the total number of data packets generated by the sources. Fig. 9 shows the packet delivery fraction of IZR as a function of H for the experiment described above. As can be seen from the figure, the packet delivery fraction remains high for the values of H considered, and is quite close to that offered by the regular Zone Routing Protocol at its optimal configuration of zone radius equal to 2 for all the nodes. Fig. 9 also compares the mean delay in discovering a route for IZR and the regular Zone Routing Protocol. The values in the plot are normalized to the mean route discovery delay for

[8]For fair comparison, the routing control traffic values for the case of IZR with dynamic zone radius configuration in Fig. 7 does not include this initial overhead of the scheme during its stabilization.

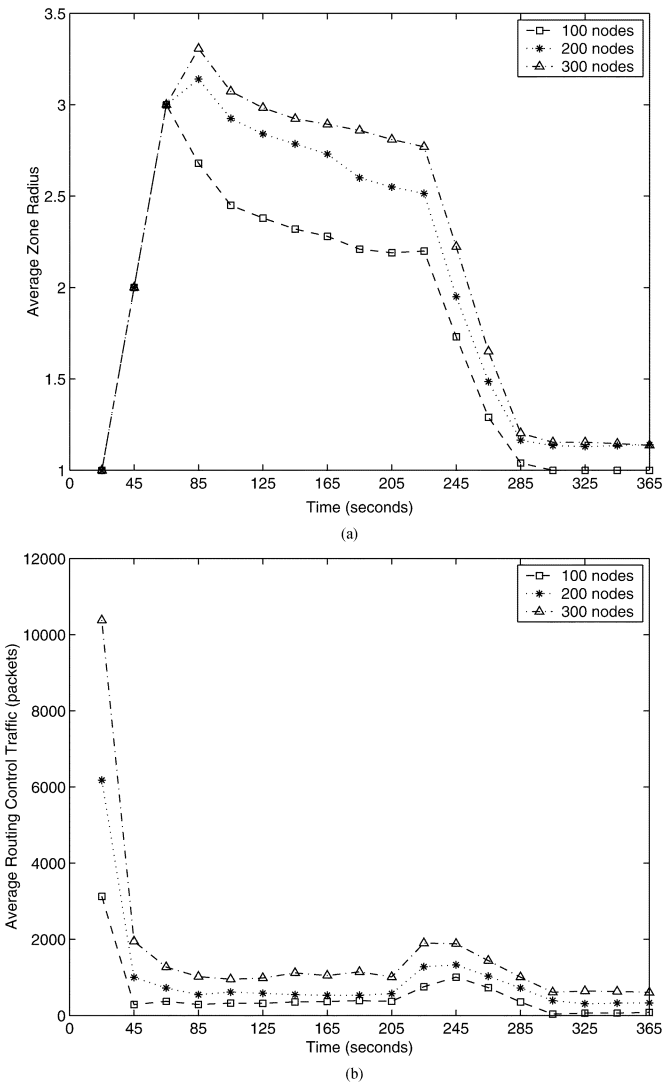

Fig. 10. The variation in (a) the average zone radius of the nodes and (b) routing control traffic, as the network size is increased.

the Zone Routing Protocol at its optimal configuration in order to illustrate the improvement offered by IZR. The plot shows that IZR can decrease the route discovery delay by about 4% to 7%.

The above experiment was performed again for larger sized networks. Fig. 10(a) and (b) shows the results as a function of time for networks with 100, 200, and 300 nodes. As the number of nodes was increased, the network area was also increased such that the per unit area density of the nodes remained constant (at $100/1300^2$ $\mathrm{nodes/meter}^2$). $H = 12$ has been used for the plots. The points in Fig. 10(a) and (b) correspond to the average zone radius and the average routing control traffic, respectively, for a window of 20 s. The performance remains similar with an increase in the network size, as IZR is able to quickly configure the nodes to their optimal zone radii at the start and when the network characteristics change, keeping the level of routing control traffic low. Note that the figures show an increase in the average routing control traffic and the average zone radius as the number of nodes in the network is increased. This is because the mean session interarrival delay for each node is

kept constant while increasing the number of nodes in the network. This causes an increase in the reactive control traffic experienced by the network (and thus each node), triggering a small increase in the average zone radius of the nodes which tries to balance the increase in the reactive component.

The results related to Figs. 7–9 demonstrate that IZR is not very sensitive to the only parameter of the framework, H. Simulation studies for a variety of scenarios show that values of H in the range between 8 and 16 lead to good performance, in general.

For the simple scenario considered in Fig. 7, the plots show that IZR can lead to about 60% reduction in the routing control traffic, as compared to the optimal setting of the regular Zone Routing Protocol. However, the real benefit of IZR is seen in networks where different regions have different characteristics or where the network characteristics change with time. In such networks, the performance of the regular Zone Routing Protocol's global zone radius assignment is not nearly as good as the performance provided by IZR, which is able to fine tune and adapt to the network conditions. For example, in Fig. 8(a) and (b), when the network characteristics change, IZR is able to adapt the zone radii of the nodes so that the routing control traffic again falls to a low level. Here, the regular Zone Routing Protocol would continue to operate at the (now) suboptimal zone radius, resulting in about 10 times more control traffic overhead!

These results have demonstrated that the IZR framework enhances Zone Routing by enabling each node in the network to independently and adaptively configure its optimal zone radius. Further, it can lead to a significant reduction in the routing control traffic as well, as observed from the simulation results. IZR enables setting the zone radius of each of the nodes to its near optimal value over time and space. Moreover, the ability of IZR to adaptively change the zone radii of the nodes makes it robust to changes in network characteristics.

X. Conclusion

Hybridization, multiscope operation, and dynamic reconfiguration form the basis for scalable, adaptable routing, as demonstrated by the IZR framework. IZR provides a flexible solution to the challenge of discovering and maintaining routes in a wide variety of ad hoc networking environments, by adapting the balance of proactive and reactive routing. The independent zone sizing capability allows the IZR framework to be fine tuned to the local network characteristics. Each of the nodes in the network can dynamically and automatically configure its zone radius to the temporally and spatially optimal value in a distributed fashion. This configuration is done at each node by analyzing the local route control traffic only, making the tuning mechanism itself scalable. All these factors lead to significant performance improvements and increase the scalability and robustness of the routing protocol.

Possible future directions consist of extending the hybrid routing framework into additional dimensions, for example balancing the tradeoffs between bandwidth efficiency and local processing/storage requirements. The performance of the IZR framework may further be enhanced by incorporating multiscoped proactive (IARP) and reactive (IERP) components. Another potential area that needs to be explored further is the application of hybridization to hierarchical routing. The impact of multiscope routing on providing quality of service (QoS) in the network may also be investigated. Intelligent power saving algorithms can be developed that can prolong the life of the network by utilizing information about residual energy of the nodes in the routing zone [16]. Integration of security into the IZR framework is another area on which work is in progress.

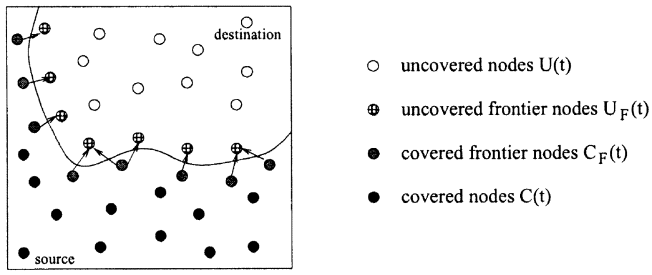

Fig. 11. Covered and uncovered frontier nodes as a query propagates in the network.

While enabling functionalities like QoS, security and power savings, the hybrid nature of the routing framework gives rise to new research issues. Mutual interaction between the constituent protocols of the framework may provide useful additional information which can be exploited for better performance. At the same time, it may introduce specific problems that do not exist in the individual protocols. Existing research on providing these functionalities in purely proactive or reactive routing protocols is insufficient to satisfactorily handle these issues. Hence, more research is warranted in this area.

Appendix
IZR Correctness

Correct operation of IZR means that a route discovery issued for a reachable destination will discover at least one route to that destination. A necessary and sufficient condition for correct IZR operation is that the route discovery's query distribution provides full coverage—that is, a route discovery for a nonreachable destination results in every reachable node being covered by the query in finite time.

The following proofs demonstrate the correctness of the IZR framework. For analysis of the route query mechanism, we assume that the network topology remains static during query propagation, and that IARP has already converged to prior topological changes.

Let the set of all nodes which belong to the routing zones of the nodes that have already received the query packet at time t be called the set of *covered nodes*, $C(t)$. The nodes in the network which do not belong to $C(t)$ form the set of *uncovered nodes*, $U(t)$ ($U(t) = C^c(t)$). Let $C_F(t)$ be a subset of $C(t)$ such that each node in $C_F(t)$ has at least one neighbor in $U(t)$. $C_F(t)$ is referred to as the set of *covered frontier nodes*. Similarly, we define the set of *uncovered frontier nodes*, $U_F(t)$, which is a subset of $U(t)$ such that each node in $U_F(t)$ has at least one neighbor that belongs to $C(t)$. Members of $C_F(t)$ and $U_F(t)$ form a boundary between covered and uncovered regions of the network, as demonstrated in Fig. 11. The set of nodes which form their own bordercast trees to bordercast the query at time t belong to the set of bordercast or *relaying nodes*, $R(t)$.

TABLE I
SETS USED IN THE IZR CORRECTNESS PROOF

Covered nodes	$C(t)$
Uncovered nodes	$U(t)$
Covered frontier nodes	$C_F(t)$
Uncovered frontier nodes	$U_F(t)$
Relaying nodes	$R(t)$
Neighbors of x	$N(x)$
Peripheral nodes of x	$P(x)$
Rebordercasting node of x, upstream of peripheral node p	$B(x,p)$
Forwarding nodes of x, to reach rebordercasting node b	$F(x,b)$
Peripheral nodes of x, downstream of rebordercasting node b	$P(x,b)$
Frontier peripheral nodes of x	$P_F(x,t)$
Frontier peripheral nodes of x, downstream of rebordercasting node b	$P_F(x,b,t)$

As explained in Section VIII-B, let $\mathcal{H}(s,b)$ be the minimum distance in hops between a source node s and its rebordercasting node b, and let $\mathcal{R}(s)$ denote the routing zone radius of s. Then the following condition is satisfied:

$$\mathcal{H}(s,b) + \mathcal{R}(b) > \mathcal{R}(s). \tag{2}$$

Table I gives a summary of the sets used in the correctness proof of the IZR framework.

Lemma 1: Each uncovered frontier node is a neighbor of a covered frontier node.

Proof: By definition, an uncovered frontier node, $u \in U_F(t)$ has at least one covered neighbor $c \in C_F(t)$. Since c has at least one uncovered neighbor, c is a covered frontier node.

Lemma 2: Given a node r, with a peripheral node p and a rebordercasting node $b \in B(r,p)$ that lies on a minimum hop path between r and p:

a) node p lies inside b's routing zone;
b) all nodes not more than $k = \mathcal{R}(b) + \mathcal{H}(r,b) - \mathcal{R}(r)$ hops from p belong to b's routing zone. In particular, all of p's neighbors belong to b's routing zone (i.e., $k \geq 1$).

Proof:

a) As p is a peripheral node of r, the distance in hops between r and p is $\mathcal{R}(r)$. Hence the distance in hops from b to p is $\mathcal{R}(r) - \mathcal{H}(r,b)$. As b is a rebordercasting node of r corresponding to p, from (2), $\mathcal{R}(r) - \mathcal{H}(r,b) < \mathcal{R}(b)$. This implies that the hop-distance between b and p is less than the zone radius of b. Therefore, p lies inside b's routing zone.
b) From above, the distance in hops from b to p is $\mathcal{R}(r) - \mathcal{H}(r,b)$. Hence, the nodes k hops from p are not more than $k + \mathcal{R}(r) - \mathcal{H}(r,b) = \mathcal{R}(b)$ hops from b. Thus all nodes k hops from p *belong* to b's routing zone. From (2), $\mathcal{R}(b) + \mathcal{H}(r,b) - \mathcal{R}(r) \geq 1$. This implies $k \geq 1$. Consequently, all of p's neighbors, which are exactly 1 hop away from p, *belong* to b's routing zone.

Lemma 3: When a relaying node $r \in R(t)$ relays the query to a rebordercasting node $b \in B(r,p)$ (via forwarding nodes, if any), all frontier nodes covered by r and downstream of b ($P_F(r,b,t)$) are removed from the frontier, and all nodes not more than $k = \mathcal{R}(b) + \mathcal{H}(r,b) - \mathcal{R}(r)$ hops downstream from the nodes in $P_F(r,b,t)$ are covered. In particular, all uncovered frontier neighbors of the nodes in $P_F(r,b,t)$ become covered ($k \geq 1$).

Proof: From Lemma 2, it follows that the members of $P_F(r,b,t)$ lie inside b's routing zone and all nodes not more than $k = \mathcal{R}(b) + \mathcal{H}(r,b) - \mathcal{R}(r)$ hops from $P_F(r,b,t)$ belong to b's routing zone. Therefore, when r relays the query to b, all nodes not more than k hops downstream from $P_F(r,b,t)$ become covered and the members of $P_F(r,b,t)$ are removed from the frontier. As $k \geq 1$, at least the uncovered frontier neighbors of $P_F(r,b,t)$ are covered.

Lemma 4: Each covered frontier node is covered by at least one relaying node.

Proof: Assume that each covered frontier node is covered by at least one relaying node at time t. Let r be the first member of $R(t)$ to relay the query after time t. After r relays the query and the query reaches its set of rebordercasting nodes (via the forwarding nodes, if any) at time $t+\tau$, it ceases to be a relaying node, and each of r's rebordercasting nodes, $b \in B(r,p)$, becomes a relaying node ($b \in R(t+\tau)$). From Lemma 3, the frontier nodes covered by r and downstream of b ($P_F(r,b,t)$) are removed from the frontier and all nodes not more than $k = \mathcal{R}(b) + \mathcal{H}(r,b) - \mathcal{R}(r)$ hops downstream of $P_F(r,b,t)$ are covered. Thus, the new covered frontier nodes $C_F(t+\tau)$ are a subset of the nodes exactly k hops from $P_F(r,b,t)$ and are covered by $b \in R(t+\tau)$. It follows that each covered frontier node is still covered by at least one relaying node at time $t+\tau$.

The base case is proven for $t = 0$, when all covered frontier nodes are covered by the relaying query source.

Lemma 5: All currently uncovered frontier nodes will be covered in finite time.

Proof: For each covered frontier node $p \in C_F(t)$, there exists a relaying node $r \in R(t)$ that covers p (Lemma 4) and a rebordercasting node b, $b \in B(r,p)$, that is upstream from p in r's bordercast tree. When r relays the query to b (possibly via some forwarding nodes), all of p's neighbors become covered, as they belong to b's routing zone (Lemma 3). After some time $t+\tau$, all relaying nodes $R(t)$ will have forwarded the query to their rebordercasting nodes. As the set of relaying nodes $R(t)$ covers all covered frontier nodes $P(t)$, all neighbors of $C_F(t)$ will be covered. In particular, all uncovered frontier nodes $U_F(t)$ will be covered ($U_F(t) \subset C(t+\tau)$) (Lemma 1).

Lemma 6: If no uncovered frontier node exists, then no uncovered node exists.

Proof: Because all network nodes are reachable, a path exists between any uncovered node and any covered node. This path must cross the frontier at least once, with a link connecting an uncovered frontier node to a covered frontier node. Consequently, if an uncovered frontier node does not exist, then no uncovered node exists.

Based on the above lemmas, we can now state the following theorem to conclude the IZR correctness proof.

Theorem 2: IZR provides full coverage.

Proof: Based on Lemma 5, there exists a finite increasing sequence of N time instances such that all uncovered frontier

nodes at time t_{n-1} will be covered at time t_n, with no uncovered frontier nodes remaining at time t_N. Since no uncovered frontier nodes remain at time t_N, it follows from Lemma 6 that no uncovered nodes remain at time t_N. Therefore, IZR provides full coverage.

REFERENCES

[1] B. Bellur and R. G. Ogier, "A reliable, efficient topology broadcast protocol for dynamic networks," presented at the IEEE INFOCOM, Mar. 1999.

[2] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," IETF, RFC 3626, Oct. 2003.

[3] M. Gerla, X. Hong, and G. Pei, "Landmark routing for large ad hoc wireless networks," presented at the IEEE GLOBECOM, San Francisco, CA, Nov. 2000.

[4] Z. J. Haas and M. R. Pearlman, "The performance of query control schemes for the zone routing protocol," *IEEE/ACM Trans. Networking*, vol. 9, pp. 427–438, Aug. 2001.

[5] Z. J. Haas, M. R. Pearlman, and P. Samar, "The bordercast resolution protocol (BRP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.

[6] ——, "The interzone routing protocol (IERP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.

[7] ——, "The intrazone routing protocol (IARP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.

[8] ——, "The zone routing protocol (ZRP) for ad hoc networks," IETF, MANET Internet Draft, July 2002.

[9] A. Iwata, C.-C. Chiang, G. Pei, M. Gerla, and T.-W. Chen, "Scalable routing strategies for ad hoc wireless networks," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1369–1379, Aug. 1999.

[10] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networking," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Boston, MA: Kluwer, 1996.

[11] B. Liang and Z. J. Haas, "Hybrid routing in ad hoc networks with a dynamic virtual backbone," *IEEE Trans. Wireless Commun.*, to be published.

[12] A. B. McDonald and T. Znati, "Predicting node proximity in Ad-Hoc networks: A least overhead adaptive model for electing stable routes," presented at the MobiHoc 2000, Boston, MA, Aug. 2000.

[13] J. Moy, "OSPF version 2," IETF, RFC 2178, Mar. 1997.

[14] S. Murthy and J. J. Garcia-Luna-Aceves, "An efficient routing protocol for wireless networks," *MONET*, vol. 1, no. 2, pp. 183–197, Oct. 1996.

[15] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," presented at the IEEE INFOCOM, Kobe, Japan, Apr. 1997.

[16] M. R. Pearlman, J. Deng, B. Liang, and Z. J. Haas, "Elective participation in ad hoc networks based on energy consumption," presented at the IEEE GLOBECOM, Nov. 2002.

[17] M. R. Pearlman and Z. J. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1395–1414, Aug. 1999.

[18] M. R. Pearlman, Z. J. Haas, and S. I. Mir, "Using routing zones to support route maintenance in ad hoc networks," presented at the IEEE WCNC 2000, Chicago, IL, Sept. 2000.

[19] M. R. Pearlman, Z. J. Haas, and B. P. Manvell, "Using multi-hop acknowledgment to discover and reliably communicate over unidirectional links in ad hoc networks," presented at the IEEE WCNC 2000, Chicago, IL, Sept. 2000.

[20] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing: A routing scheme for ad hoc wireless networks," presented at the IEEE ICC 2000, New Orleans, LA, June 2000.

[21] G. Pei, M. Gerla, X. Hong, and C.-C. Chiang, "A wireless hierarchical routing protocol with group mobility," presented at the IEEE WCNC'99, New Orleans, LA, Sept. 1999.

[22] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," *Proc. ACM SIGCOMM*, vol. 24, no. 4, pp. 234–244, Oct. 1994.

[23] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," presented at the IEEE WMCSA, New Orleans, LA, Feb. 1999.

[24] P. Samar, "On efficient communication protocol design for dynamic multi-hop networks," Ph.D. dissertation, Cornell Univ., Ithaca, NY, 2004.

[25] P. Samar and Z. J. Haas, "Strategies for broadcasting updates by proactive routing protocols in mobile ad hoc networks," presented at the IEEE MILCOM 2002, Anaheim, CA, Oct. 2002.

[26] P. Samar, M. R. Pearlman, and Z. J. Haas, "Hybrid routing: The pursuit of an adaptable and scalable routing framework for ad hoc networks," in *Handbook of Ad Hoc Wireless Networks*, M. Ilyas, Ed. Boca Raton, FL: CRC Press, 2002.

[27] P. Samar and S. B. Wicker, "On the behavior of communication links in a multi-hop mobile environment," in *Frontiers in Distributed Sensor Networks*, S. S. Iyengar and R. R. Brooks, Eds. Boca Raton, FL: CRC Press, 2004.

[28] P. Sivakumar, R. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed routing algorithm," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1454–1465, Aug. 1999.

Prince Samar (S'04) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Bombay, India, in 1999 and the M.S. degree in electrical and computer engineering from Cornell University, Ithaca, NY, in 2003. He will receive the Ph.D. degree in electrical and computer engineering at Cornell University in August, 2004.

His interests are in the areas of networking and wireless mobile systems. His research has focused on the design of routing protocols, mobility modeling, and its effects on link characteristics for performance optimization in ad hoc networks.

Marc R. Pearlman (M'00) received the B.S.E.E. degree (with highest honors) from Rutgers University, New Brunswick, NJ, in 1996. He is working toward the Ph.D. degree in the School of Electrical and Computer Engineering at Cornell University, Ithaca, NY.

His research interests include scalable ad hoc network routing, cross layer protocol design, collaborative systems, and emergent behavior. In October 2003, he co-founded Kraken Ink, a company advancing research and commercial development of collaborative robotic and sensing systems. Prior to the formation of Kraken Ink, he was with GE's Global Research Center, where his research and development in ad hoc networks and industrial control networks led to over 12 filed patent applications.

Zygmunt J. Haas (S'84–M'88–SM'90) received the B.Sc. degree in electrical engineering in 1979 and the M.Sc. degree in electrical engineering in 1985. In 1988, he received the Ph.D. degree from Stanford University, Stanford, CA.

He joined AT&T Bell Laboratories Network Research Department in 1988, where he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 to July 1995, he worked for the AT&T Wireless Center of Excellence, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. In August 1995, he joined the faculty of the School of Electrical and Computer Engineering at Cornell University, Ithaca, NY. He is an author of numerous technical papers and holds fifteen patents in the fields of high-speed networking, wireless networks, and optical switching. His research interests include mobile and wireless communication and networks, personal communication service, and high-speed communication and protocols.

Dr. Haas has organized several workshops, delivered numerous tutorials at major IEEE and ACM conferences, and serves as editor of several journals and magazines, including the IEEE TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE *Communications Magazine*, and the ACM/Kluwer *Wireless Networks* journal. He has been a guest editor of IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS issues on Gigabit Networks, Mobile Computing Networks, and Ad-Hoc Networks. He is a voting member of ACM, and the Chair of the IEEE Technical Committee on Personal Communications.