The Performance of Query Control Schemes for the Zone Routing Protocol

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

***Abstract*—In this paper, we study the performance of route query control mechanisms for the Zone Routing Protocol (ZRP) for *ad hoc* networks. ZRP proactively maintains routing information for a local neighborhood (routing zone), while reactively acquiring routes to destinations beyond the routing zone. This hybrid routing approach can be more efficient than traditional routing schemes. However, without proper query control techniques, the ZRP cannot provide the expected reduction in the control traffic.**

Our proposed query control schemes exploit the structure of the routing zone to provide enhanced detection and prevention of overlapping queries. These techniques can be applied to single- or multiple-channel ad hoc networks to improve both the delay and control traffic performance of ZRP. Our query control mechanisms allow ZRP to provide routes to all accessible network nodes, with less control traffic than purely proactive link state or purely reactive route discovery, and with less delay than conventional flood searching.

***Index Terms*—*Ad hoc* network, bordercast, hybrid routing, proactive routing, query control, reactive routing, routing protocol, routing zone, zone routing protocol, ZRP.**

I. Introduction

An *AD HOC* network is a self-organizing wireless network made up of mobile nodes and requiring no fixed infrastructure. The limitations on power consumption imposed by portable wireless radios result in a node transmission range that is typically small relative to the span of the network. To provide communication throughout the entire network, nodes are designed to serve as relays if needed. The result is a distributed multi-hop network with a time-varying topology.

Because *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 providing tactical communication for military, law enforcement, and emergency response efforts. *Ad hoc* networks can also play a role in civilian forums, such as the electronic classroom, convention centers, and construction sites. With such a broad scope of applications, it is not difficult to envision *ad hoc* networks operating over a wide range of coverage areas, node densities, and node velocities.

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 topology information becomes more detailed and up-to-date. On the other hand, in an *ad hoc* network, the topology may change quite often, requiring large and frequent exchanges of data among the network nodes. This is in contradiction with the fact that all updates in the wireless communication environment travel over the air and are costly in resources.

Existing routing protocols can be classified either as *proactive* or as *reactive*. Proactive protocols attempt to 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. Early applications of proactive routing schemes for *ad hoc* networks were Distance Vector protocols based on the Distributed Bellman-Ford (DBF) algorithm [1]. Modifications to the basic DBF algorithm (i.e., [2], [4] and [12]) were proposed to address inherent problems of convergence and excessive traffic (both of which can be quite severe in *ad hoc* networks, where bandwidth is scarce and topologies often very dynamic). The convergence problem was also addressed by the application of Link State protocols to the *ad hoc* environment (a recent addition being the Optimized Link State Routing (OLSR) protocol [7]). In general, Link State protocols converge faster than Distance Vector protocols, but at the expense of significantly more control traffic. Motivation to both improve protocol convergence and to reduce control traffic has led to the development of proactive path finding algorithms, which combine features of the Distance Vector and Link State approaches. Each node constructs its minimum spanning tree based on knowledge of its neighbors' minimum spanning trees and the link costs to each neighbor. Realizations of the path finding algorithms, like the Wireless Routing Protocol (WRP) ([9] and [10]), are able to eliminate the "counting-to-infinity" problem and to reduce the occurrence of temporary loops, often with less control traffic than traditional Distance Vector schemes.

In contrast, reactive protocols invoke a route determination procedure on an *on-demand* basis. The reactive route discovery is usually based on a query–reply exchange, where the route query is flooded through the network to reach the desired destination. In the case of the Temporally Ordered Routing Algorithm (TORA) [11], the route replies are also flooded, in a controlled manner, distributing routing information in the form of directed acyclic graphs (DAGs) rooted at the destination. In contrast, the Dynamic Source Routing (DSR) [8] and *ad hoc*

Manuscript received June 3, 1999; revised February 1, 2000; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Perkins. This work was supported in part by the National Science Foundation under Grant ANI-9980521, by the Office of Naval Research under Contract N00014-00-1-0564, and by the AFRL under Contract F30602-00-C-0032.

Z. J. Haas is with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA (e-mail: haas@ece.cornell.edu).

M. R. Pearlman is with School of Electrical and Computer Engineering, Cornell University, Ithaca, NY 14853 USA and also with GE Corporate Research and Development, Niskayuna, NY 12309 USA (e-mail: pearlman@crd.ge.com).

Publisher Item Identifier S 1063-6692(01)06855-8.

On Demand Distance Vector (AODV) [13] protocols unicast the route reply back to the querying source, along a path constructed during the route query phase. In the case of DSR, the routing information is accumulated in the query packet and the complete sequence of nodes is returned to the source (to be used for source routing). AODV, on the other hand, distributes the discovered route in the form of next-hop information stored at each node in the route. The on-demand discovery of routes can result in much less traffic than standard Distance Vector or Link State schemes, especially when innovative route maintenance schemes are employed. However, the reliance on flooding may still lead to considerable control traffic in the highly versatile *ad hoc* networking environment.

The advantage of the proactive schemes is that route information is available when needed, resulting in little delay prior to data transmission. In contrast, reactive schemes may produce significant delay in order to determine a route when route information is needed, but not available.

Routing schemes, whether proactive or reactive, require some exchange of control traffic. This overhead can be quite large in *ad hoc* networks, where the topology frequently changes. Reactive protocols produce a large amount of traffic by effectively flooding the entire network with route queries. The combination of excessive control traffic and long route query response time rule out pure reactive routing protocols for real-time communication applications. Pure proactive schemes are likewise not appropriate for *ad hoc* networks, as they *continuously* use a large portion of the network capacity to keep the routing information current. Proactive protocols tend to distribute topological changes widely in the network, even though the creation/destruction of a new link at one end of the network may not be a significant piece of information at the other end of the network. Furthermore, since *ad hoc* network nodes may move quite fast, and as the changes may be more frequent than the route requests, most of this maintained routing information is never used! This results in further waste of the network capacity.

II. The ZRP—A Short Overview

The behavior of purely proactive and reactive schemes suggest that what is needed is a protocol that initiates the route-determination procedure on-demand, but at limited search cost. Our protocol, the *ZRP* ([5] and [6]), is an example of such a hybrid reactive/proactive scheme. On one hand, it limits the scope of the proactive procedure only to the node's local neighborhood. As we shall see, the local routing information is frequently referred to in the operation of ZRP, minimizing the waste associated with the purely proactive schemes. On the other hand, the search throughout the network, although global, can be performed efficiently by querying selected nodes in the network, as opposed to querying all the network nodes. We proceed with an introduction of the routing zone concept and a brief overview of ZRP architecture.

A. Routing Zones and Intrazone Routing

In ZRP, a node proactively maintains routes to destinations within a local 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, as we shall see, is that the routing zones of neighboring nodes overlap.

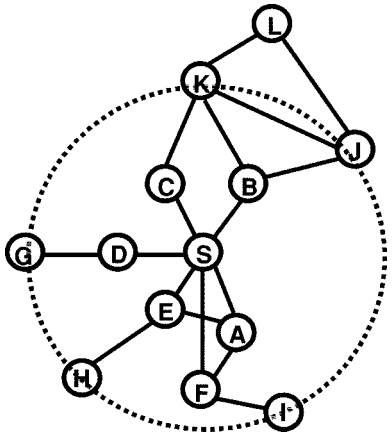

Fig. 1. A routing zone of radius 2 hops.

Fig. 1 illustrates the routing zone concept with a routing zone of radius 2 hops. This particular routing zone belongs to node S, which we refer to as the *central node* of the routing zone. Nodes A through K are members of S's routing zone. Node L, however, is three hops away from S, and is therefore 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 physical distance, but rather nodal connectivity (hops).

The construction of a routing zone requires a node to first know who its *neighbors* are. A neighbor is defined as a node with whom *direct* communication can be established[1] (and is thus one hop away). Identification of a node's neighbors may be provided directly by the media access control (MAC) protocols, as in the case of polling-based protocols. In other cases, neighbor discovery may be implemented through a separate *Neighbor Discovery Protocol (NDP)*. Such a protocol typically operates through the 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 proactive monitoring of routing zones through the IntrAzone Routing Protocol (IARP). IARP can be derived from globally proactive Link State routing protocols that provide a complete view of network connectivity (for example, the Shortest Path First OSPF). The base protocol needs to be modified to ensure that the scope of the route updates is restricted to the radius of the

[1]The determination of a direct connection between two nodes is typically based on measurements of link quality, such as received signal power, bit error rate (BER), signal-to-interference ratio (SIR), link stability, etc. The application of the network often determines the minimal level of link quality to support a direct connection between two nodes.

node's routing zone. In this paper, IARP is based on a simple, timer-based, Link State protocol. To track the topology of ρ hop routing zones, each node periodically broadcasts its link state for a depth of ρ hops [controlled by a time-to-live (TTL) field in the update message].

B. Interzone Routing

Whereas IARP maintains routes to nodes within the routing zone, the *IntErzone Routing Protocol (IERP)* is responsible for acquiring routes to destinations that are located beyond the routing zone. IERP uses a query–response mechanism to discover routes on demand.

IERP is distinguished from standard flooding algorithms by exploiting the structure of the routing zone, through a process known as *bordercasting*. Bordercasting is a packet delivery service that allows a node to efficiently send a message to its peripheral nodes. ZRP provides this service through a component called the Bordercast Resolution Protocol (BRP). In its simplest form, bordercasting *could* be implemented through network layer multicasting of messages to peripheral nodes. This approach prevents the routing protocol from accessing route query messages until they arrive at the edge of the routing zone. As we will show later, proper control of the route query process requires that the routing protocol monitor and relay query messages on a *hop-by-hop* basis. The network layer is used to deliver query messages to a set of downstream neighbors identified by the BRP.

An IERP route query is triggered at the network layer, when a data packet is destined for a node that does not lie within its routing zone.[2] The source generates a route query packet, which is uniquely identified by a combination of the source node's ID and request number. The query is then bordercast to all the source's peripheral nodes. Upon receipt of a route query packet, a node adds its ID to the query. The sequence of recorded node ID's specifies an *accumulated route* from the source to the current routing zone.[3] If the destination does not appear in the node's routing zone, the node bordercasts the query to *its* peripheral nodes. If the destination is a member of the routing zone, a *route reply* is sent back to the source, along the path specified by reversing the accumulated route. As with standard flooding algorithms, a node will discard any replicated route query packet of a query that it has previously encountered.

An example of this *Route Discovery* procedure is demonstrated in Fig. 2. The source node S prepares to send data to the destination D. S first checks whether D is within its routing zone. If so, S already knows the route to node D. Otherwise, S sends a query to all its peripheral nodes (C, G, and H). Now, in turn, each one of these nodes, after verifying that D is not in its routing zone, forward the query to its peripheral nodes. In particular, H sends the query to B, which recognizes D as being in its routing zone and responds to the query, indicating the forwarding path: S–H–B–D.

A nice feature of this route discovery process is that a single route query can return multiple route replies. The quality of these returned routes can be determined based on hop count (or any other path metric[4] accumulated during the propagation of the query). The best route can be selected based on the relative quality of the route (i.e., choose the route with the smallest hop count, or shortest accumulated delay).

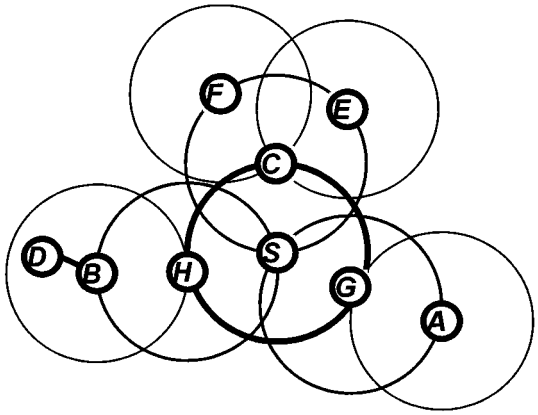

Fig. 2. An example of IERP operation.

C. Constructing the Bordercast Tree

In ZRP, efficient route discovery is based on a routing zone based packet delivery service called bordercasting. Rather than blindly *broad*casting a route query from neighbor to neighbor, *border*casting allows the query to be directed outward, via multicast, to a set of surrounding peripheral nodes. Execution of a bordercast requires construction of a bordercast tree, from which packet forwarding instructions for each tree member can be extracted. Since a complete bordercast tree extends from a root node to all its peripheral nodes, only the root has sufficient topological knowledge to construct this tree. This implies that the root constructs its bordercast tree on behalf of all tree members, appending forwarding instructions to the route query packet [Fig. 3(a)]. This *root directed* approach adds a per packet overhead that increases more than linearly with the zone radius. The increased query packet length works against the expected reduction in query packets, obscuring the benefits of a hybrid proactive/reactive routing strategy.

In order to support a *distributed* bordercast, an interior member of a bordercast tree must be able to independently reconstruct that bordercast tree. To achieve this, each node must proactively track the topology of a region that extends beyond its own routing zone, and encompasses all routing zones for which it is an interior member. Specifically, if an interior bordercast tree member (up to $\rho - 1$ hops from the tree's root node) is to construct the entire ρ hop tree, then that node needs to proactively track the topology of a $\rho + (\rho - 1) = 2\rho - 1$ hop *extended routing zone* [Fig. 3(b)]. Maintaining an extended routing zone adds extra load to the proactive IARP, but preserves the expected savings in reactive route discovery (IERP) traffic. Thus, the distributed bordercast upholds the desired overhead tradeoff between locally proactive and globally reactive routing components.

[2]Remember that a node knows the identity, distance to, and a route to all the nodes in its zone.

[3]Because each node maintains a routing zone, interzone routes can be specified as a sequence of nodes separated by a distance equal to the zone radius.

[4]Typical path metrics include hop count, delay, capacity, etc.

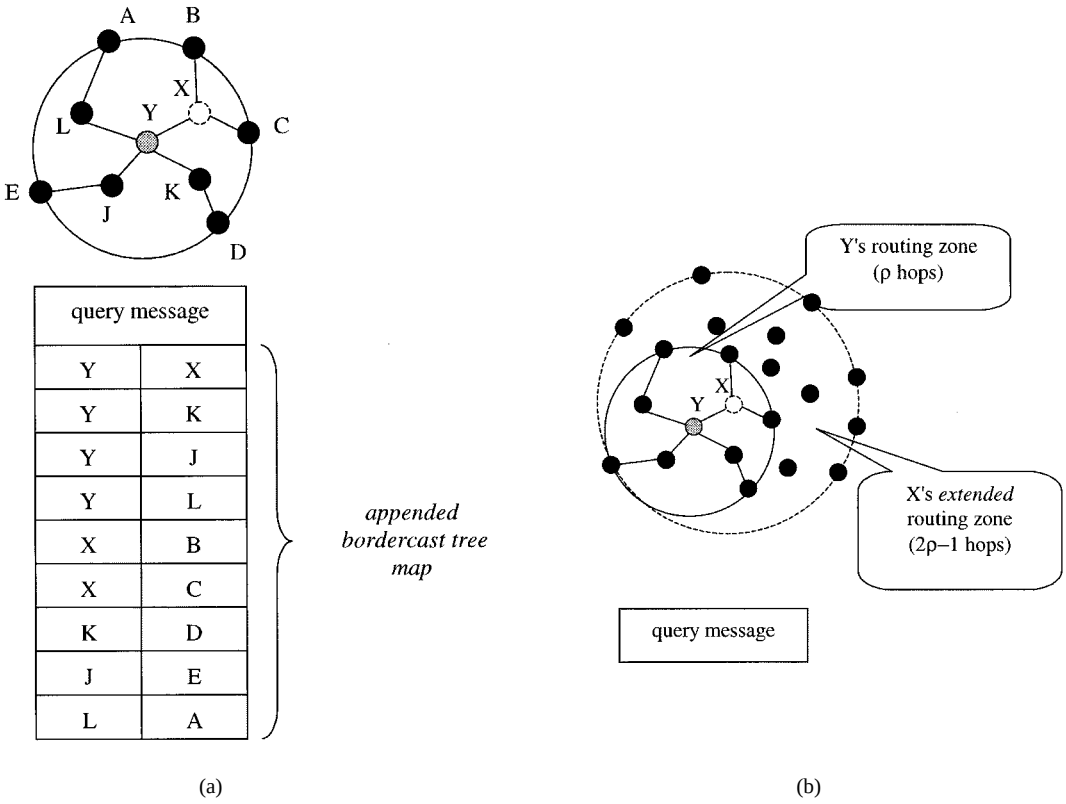

Fig. 3. (a) RDB. (b) DB.

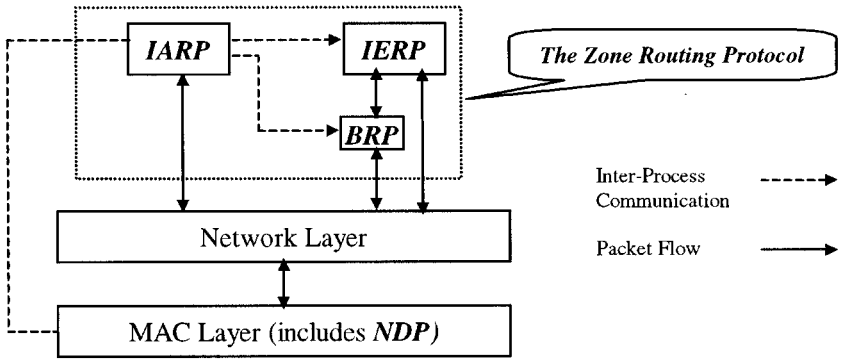

Fig. 4. ZRP architecture.

D. ZRP Architecture

The relationship between ZRP component protocols is illustrated in Fig. 4. The proactive maintenance of the routing zone topology is performed by IARP, through exchange of route update packets. Route updates are triggered by the MAC-level NDP,[5] which notifies IARP when a link to a neighbor is established or broken. IERP reactively acquires routes to nodes beyond the routing zone using a query-reply mechanism. IERP forwards queries using the BRP's bordercast packet delivery service. Bordercasting leverages IARP's up-to-date view of the local topology to efficiently guide route queries away from the query source. IERP also uses IARP's local routes to respond to incoming route queries. The route responses are then relayed back to the query source through network layer unicast.

The relationship between IARP and IERP may, at first, give the impression that ZRP is a hierarchical routing protocol. In fact, ZRP bears only a superficial resemblance to such protocols. Hierarchical routing relies on the strategic assignment of gateways or landmarks [] to establish a hierarchy of subnets for the entire network.[6] Access to a subnet is provided through that subnet's assigned gateway or landmark. As a result, two nodes that belong to different subnets must send their communication up the hierarchy to a subnet that is common to both nodes. This

[5]If a neighbor discovery service is not provided by the MAC layer, then ZRP provides its own neighbor discovery.

[6]Gateways or landmarks must be assigned in such a way that every node is able to access every level of the hierarchy. Furthermore, in order to guarantee communication between any two network nodes, there must be a "top" subnet or landmark which is accessible/visible by all network nodes.

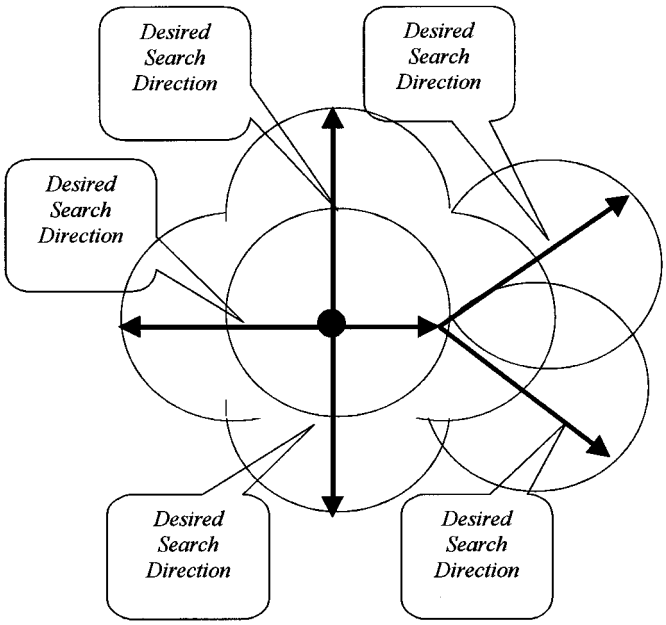

Fig. 5. Guiding the search in desirable directions.

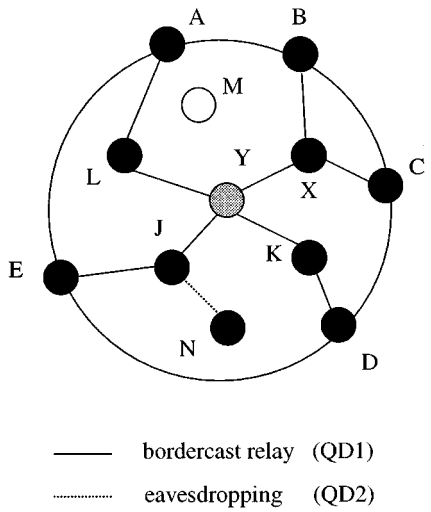

Fig. 6. Query Detection (QD1/QD2).

constraint often leads to sub-optimal routes. In contrast, access to a ZRP routing zone is provided not through a single gateway or landmark, but through the "best" of the multiple peripheral nodes that define the extent of the zone. Communication beyond a routing zone is passed across overlapping routing zones in a peer-to-peer manner, rather than up to a higher tier with broader coverage. As a result, the routing inefficiencies associated with hierarchical routing protocols are avoided in ZRP, permitting optimal routing to a destination. Furthermore, this results in an increase in the reuse of the wireless spectrum. In this sense, it is more accurate to categorize ZRP as a flat, rather than hierarchical, routing protocol.

III. Query-Control Mechanisms

ZRP is based on the idea that querying can be performed more efficiently than flooding by directing route requests to target peripheral nodes. However, because neighboring routing zones heavily overlap, each node may forward a route request multiple times, resulting in more control traffic than flooding. To prevent this from happening, the query termination and query forwarding strategies used in traditional flooding algorithms need to be properly extended for use in the routing zone architecture.

In order to understand the cause of the ZRP control traffic problem, it is important to stress one of the key features of the routing zone: when a node bordercasts a query, the node's entire routing zone is effectively covered by the query. From this perspective, excess route query traffic is the result of query messages returning to covered *zones* (as opposed to *nodes*, in the case of simple flood searching). Thus, the design objective of query control mechanisms should be to reduce route query traffic by directing query messages outward from the query source and away from covered routing zones (Fig. 5).

In this section, we introduce a collection of query control mechanisms that meet the basic design objectives. Through advanced query detection and knowledge of the local topology, each node is able to identify surrounding regions that have already been covered by the query. Nodes can steer queries away from those areas by early termination of stray messages, encouraging the search to proceed outward. In some cases, delaying the early termination processing for a random period of time provides a valuable opportunity to detect recent additions in query coverage.

A. Query Detection (QD1/QD2)

Redundant querying occurs when a query message reappears in the routing zone of a node that has already bordercast the query. Preventing query overlap depends on the ability of nodes to detect local query relaying activity. Clearly, a bordercasting node is aware that its own zone has been queried. If the query message were relayed from a bordercasting node to its peripheral nodes via IP, the query would travel through the routing zone, undetected by ZRP. By using BRP to direct the query, hop by hop along bordercast trees, all relaying nodes in the tree are able to detect the query (QD1). In single-channel networks, it is possible for queries to be detected by any node within the transmission range of a relaying node. This extended query detection capability (QD2) can be implemented by means of IP and MAC layer broadcasts.

Fig. 6 illustrates both levels of advanced query detection. In this example, node Y bordercasts to peripheral nodes A–E. The intermediate relaying nodes (J, K, L and X) are able to detect the query through QD1. Using QD2 "eavesdropping," node N is able to detect J's query transmission, even though N does not belong to Y's bordercast tree. QD2 offers a high level of query detection, but does not guarantee that the entire routing zone will be informed. In this example, node M does not overhear the query message and is thus unaware that node M's routing zone is covered by the query.

At a minimum, the query detection scheme needs to record the query source node's address and query id in a *Query Detection Table*. This {source, id} pair is sufficient to uniquely identify all queries in the network. Other query control mechanisms may require QD to record additional information contained in

the route query packet. Of particular importance is the ID of the node that most recently bordercast the query. As we will see in the next section, this information provides valuable insight into the local coverage of the query, which can be used to terminate or prevent redundant queries.

B. Early Termination (ET)

When a node bordercasts a query, all nodes within its routing zone are effectively covered by the query. Any further query messages directed into this region are redundant and represent a potential inefficiency of bordercasting. In general, it is not possible to guide the query perfectly outward into uncovered regions of the network. Fortunately, information obtained through advanced query detection (QD1/QD2), combined with knowledge of the local topology, can support Early Termination (ET) of many query messages that otherwise would stray inward.

When a node relays a query along a bordercast tree, it can safely prune any downstream branches leading to peripheral nodes *inside* covered regions of the network (i.e., interior routing zone members of nodes that already have bordercast the query). The relaying node can use the known topology of its extended routing zone (or standard routing zone plus cached bordercast trees, in the case of root directed bordercast) *interior* routing zone members of each previously bordercast node in the Detected Queries Table. Furthermore, the relaying node can prune a peripheral node if it has already relayed the query downstream to that peripheral node. Relaying the same query message to a peripheral node for a second time would not add to the overall query coverage.[7]

The ET mechanism is demonstrated in Fig. 7. Node X first receives a query message to relay for bordercasting node Y. X takes advantage of its extended routing zone and QD to identify all of Y's interior routing zone nodes as being covered. X then reconstructs Y's bordercast tree (again, based on the extended routing zone) and relays the query message to two downstream peripheral nodes. These downstream peripheral nodes are also considered by X to be covered. Later, X receives a second copy of the query to relay on behalf of bordercasting node Z. As before, X identifies the interior nodes of Z's routing zone and reconstructs Z's bordercast tree. According to Z's bordercast tree, X should relay the query to two of Z's peripheral nodes. However, X recognizes that both peripheral nodes have already been covered (one is an interior member of Y's routing zone, and the other is a peripheral node of Y's routing zone that was already relayed to by X). Based on the ET criteria, node X can prune both peripheral nodes from the bordercast tree and safely discard the query.

C. Random Query Processing Delay (RQPD)

When a node initiates a bordercast to its peripheral nodes, the node's routing zone is instantly covered by the query. However, it takes some finite amount of time for the query to make its way along the bordercast tree, and be detected through the QD mechanisms. During this bordercast propagation window, the routing zone is vulnerable to query overlap from nearby bordercasts. Although this window of vulnerability is not very large, it can be a real problem when nearby nodes initiate bordercasts at roughly the same time. This is common, especially in single-channel networks, when neighboring peripheral nodes receive a query message and simultaneously re-bordercast the message farther out into the network.

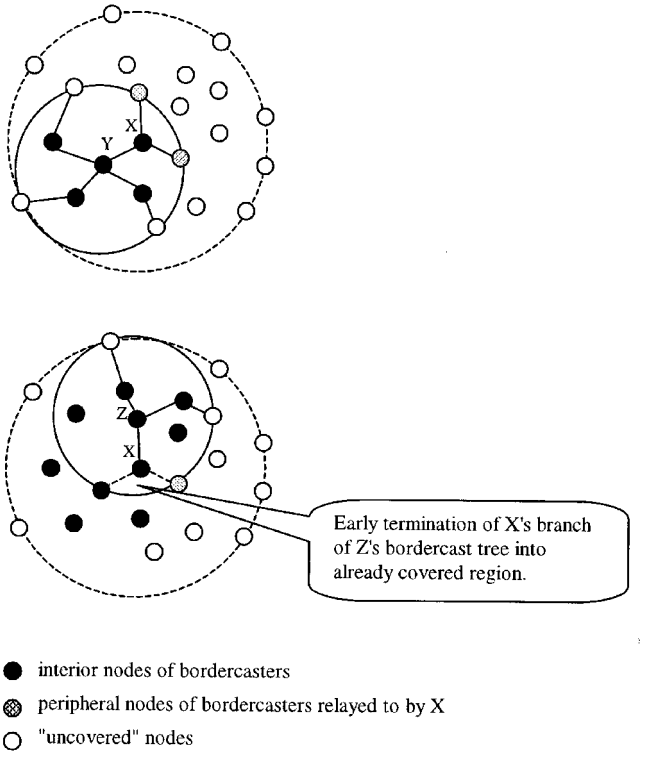

Fig. 7. ET.

This problem of "simultaneous" bordercasts can be addressed by spreading out the bordercasts with a Random Query Processing Delay (RQPD). Specifically, each bordercasting node schedules a random delay prior to bordercast tree construction and ET. During this time, the waiting node benefits from the opportunity to detect the added query coverage from earlier bordercasting nodes. This, in turn, promotes a more thorough pruning of the bordercast tree (through ET) when it is time for the waiting node to bordercast. Increasing the average RQPD can significantly improve performance, up to a point. Once the bordercast times are sufficiently spread out, further increases in delay have a negligible impact on query efficiency.

The benefit of RQPD is shown in Fig. 8. Nodes X and Y are peripheral nodes that share a common upstream neighbor in the zone's bordercast tree. Assuming neither node is pruned from the bordercast tree through ET, X and Y will receive the route query at approximately the same time. Without RQPD, X and Y will both proceed to bordercast the query to their peripheral nodes. Only later will both nodes determine (via QD) that their bordercasts were redundant. However, when RQPD is applied, X and Y each "back off" a random period of time with sufficiently large mean. In this case, X schedules its bordercast far enough in advance of Y, allowing Y to detect X's bordercast

[7]A proof of ZRP correctness with these early termination criteria can be found in [6].

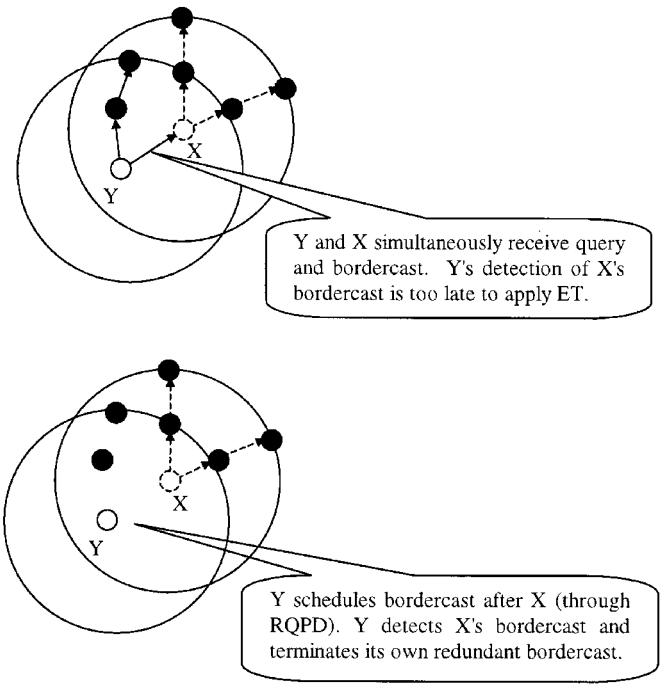

Fig. 8. RQPD.

before launching its own. X then applies this detected query information to prune its remaining downstream peripheral nodes (since they all lie inside of Y's routing zone).

The use of RQPD 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. In particular, a forwarding jitter scheduled after packet reception and before bordercast tree construction/ET effectively serves as RQPD.

IV. Evaluation of ZRP

The performance of ZRP was evaluated based on simulations of mobile *ad hoc* networks, over a range of routing zone radii (ρ), from purely reactive routing ($\rho = 1$ hop) to purely proactive routing ($\rho \rightarrow \infty$ hops). Performance was gauged by measurements of control traffic generated by ZRP and the average response time of the reactive route discovery process.

Measurements of control traffic are reported in terms of control traffic packets. The overall ZRP control traffic is viewed as the sum of the transmitted NDP neighbor discovery beacons, IARP route update packets and IERP request/reply packets. The delay performance of ZRP is reflected by the initial route acquisition delay for each destination node. Delay is evaluated under low-load network conditions, for a representative scenario of mobile speed $v = 25$ [m/s] and route query rate $R_{\text{query}} = 1.0$ [queries/s]. In the low-load scenario, the amount of application traffic is negligible in comparison to ZRP control traffic. Application traffic is sent in packets of 1000 [bits] and is given low transmission priority relative to our routing control traffic.

Our simulated network consists of 200 mobile nodes, whose initial positions are chosen from a uniform random distribution over an area of 1000 [m] by 1000 [m]. All nodes move at a constant speed, v, with an initial direction[8] θ, which is uniformly distributed between 0 and 2π. When a node reaches the edge of the 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 the velocity is not altered.

In the absence of a packet collision, we assume that background channel interference and receiver noise limit the transmission range of packets and busy tones to a physical radius of $d_{\text{xmit}} = 100$ [m]. Within a range of d_{xmit}, the average power (and resulting average SIR) of the desired signal rapidly increase to support reliable packet transmission. As significant improvements can be realized through the addition of error control coding, we approximate the rapid increase of packet reliability by a simple threshold packet delivery model: *Once access to the channel has been established*, a packet can be delivered (error-free) to any receiver within d_{xmit} from the transmitting node. Receivers farther than d_{xmit} from the transmitting node will not receive the packet.

In our single-channel networks, nodes contend for the channel based on the Dual Busy Tone Multiple Access (DBTMA) protocol [3]. Prior to transmitting a data packet, a node secures access to the channel through an RTS/CTS handshake (performed on a separate control channel). After completing the RTS/CTS handshake, the transmitter sends the data packet, while simultaneously activating a *transmit busy tone*. The intended receiver, in turn, activates a separate *receive busy tone* as soon as this data transmission is detected. The dual busy tones are used to block attempts by neighboring nodes to access a channel already in use. In particular, the transmit busy tone prevents neighbors of the transmitter from accepting incoming RTS requests. Likewise, the receive busy tone prevents the receiver's neighbors from initiating the RTS/CTS handshake. This effectively prevents the "hidden terminal problem" associated with wireless channel access. In addition, DBTMA inherently avoids the "exposed terminal problem" by permitting neighboring nodes to transmit data simultaneously to different (and available) receivers.

In contrast to the single-channel networks, we assume that channel access in our multiple-channel networks is contention free. The underlying media access control is responsible for assigning each incoming/outgoing link a locally unique channel (frequency, time slot, code) to avoid channel contention. Although there are no packet collisions, retransmissions are still possible, as a receiving node may be busy receiving or transmitting another packet.

Neighbor discovery is based on the reception of HELLO beacons that are unreliably broadcast at the MAC layer. These short beacons (containing only source address) are transmitted at random intervals of mean T_{beacon}. T_{beacon} is inversely proportional to the relative node speed ($T_{\text{beacon}} = (d_{\text{xmit}}/20)/v$), so networks with different mobility experience the same acceptable level of neighbor connectivity (subject to available bandwidth). Neighbor connectivity is determined by the reception of the HELLO beacons. If a new beacon fails to arrive within $2 \cdot T_{\text{beacon}}$ of the most recent beacon, a link failure is

[8]Direction is measured as an angle relative to the positive x axis.

TABLE I
FIXED SIMULATION PARAMETERS

Parameter	Symbol	Value
Number of nodes	N	200
Network coverage area	A	1000 [m] x 1000 [m]
Transmission radius	d_{xmit}	100 [m]
Transmission rate	R_{xmit}	10.0 [Mbps]

TABLE II
VARIABLE SIMULATION PARAMETERS

Parameter	Symbol	Values
Routing zone radius	ρ	1-10 [hops]
Node speed	v	10-75 [m/sec]
Beacon period	T_{beacon}	$\frac{d_{xmit}/20}{v}$
IARP update period	T_{IARP}	$3 \cdot T_{beacon}$
Mean route query rate	R_{query}	0.1-10.0 [query/s/node]

reported. Because the links are bi-directional, the need for a more complex HELLO→I-HEAR-YOU packet exchange is eliminated. Furthermore, we assume that neighbor discovery beacons are given 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 ZRP evaluation was performed using a custom-built event driven simulator. For each network configuration (see Tables I and II), simulations were run on 50 randomly distributed node layouts, each for a duration of 125 s. No data was collected for the first 5 s of the simulations while the initial intrazone route discovery process stabilized. IARP traffic is generated based on changes in link status detected by the neighbor discovery protocol. IERP route queries are triggered at a rate of R_{query} for destinations selected from a uniform random distribution of all nodes outside of the querying node's routing zone. These route queries represent both the initial query performed at the beginning of a session and subsequent queries due to reported route failures.

V. PERFORMANCE RESULTS

Fig. 9 illustrates the dependence of proactive intrazone control packets on the routing zone radius ρ for various rates of network reconfiguration. A distinction is made between the root-directed bordercasting (RDB) and distributed bordercasting (DB) because DB requires IARP to maintain an extended zone of radius $2\rho. - 1$ hops. The increase in IARP traffic resulting from the extended routing zone is shown to be quite significant. For unbounded networks with a uniform distribution of nodes, we expect the amount of intrazone control traffic per node to be $O(\rho^2)$. The amount of proactive routing traffic per node is proportional to the number of nodes that are being "tracked" in the routing zone, and the number of zone nodes is proportional to the "area" (ρ^2) of the zone. However, because our network is of finite size, the resulting boundary effect makes the dependence less than ρ^2.

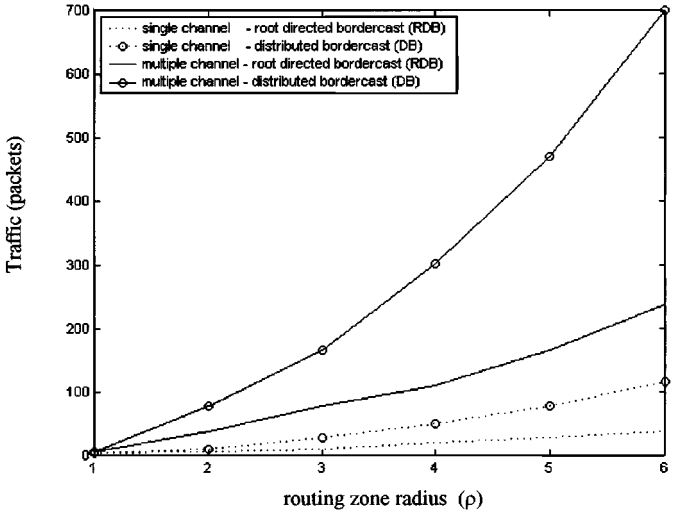

Fig. 9. IARP + NDP traffic per route update per node.

We begin our examination of the IERP control traffic by determining the effectiveness of the 24 possible combinations of query control mechanisms (see Table III). To be considered effective, we require that the amount of IERP traffic per route discovery be a decreasing function of the routing zone radius. For this evaluation, we consider traffic both in terms of IERP *bits* and IERP packets. IERP bits reflect the amount of channel capacity consumed by the *transmission* of all IERP control traffic, while the number of IERP *packets* provides a better indication of the capacity required for *pre-transmission* channel contention. The relative importance of these two criteria depends on the overall network load: channel contention becomes a stronger factor under higher network loads.

Both RDB and DB can be effective in reducing the route discovery *packet* load. However, RDB does not reduce the route discovery's overall *bit* load. The query packet increases in length with respect to the zone radius due to the appended bordercast tree map. The savings in transmitted packets is not sufficient to offset the growing packet length.

Effective zone-based route discovery requires some form of advanced query detection and early termination. Single-channel networks also need RQPD to compensate for excess re-bordercasting of query packets by neighboring nodes. Multiple-channel networks can achieve efficient route discovery with only QD1 and ET. The looser query control requirements are due to the fact that, in a multiple-channel network, a node can focus its query forwarding to a subset of its neighbors, rather than the "all-or-nothing" alternative in single-channel networks. Of course, more efficient route discovery is possible with the addition of RQPD. Since "eavesdropping" is not possible in multiple-channel networks, QD2 is not applicable.

TABLE III
EFFECTIVE COMBINATIONS OF QUERY–CONTROL MECHANISMS

				Single Channel		Multiple Channel	
Advanced Query Detection	ET	RQPD	Bordercasting	Packets	Bits	Packets	Bits
NO	NO	NO	Distributed				
NO	NO	NO	Root Directed				
NO	NO	YES	Distributed				
NO	NO	YES	Root Directed				
NO	YES	NO	Distributed				
NO	YES	NO	Root Directed				
NO	YES	YES	Distributed				
NO	YES	YES	Root Directed				
QD1	NO	NO	Distributed				
QD1	NO	NO	Root Directed				
QD1	NO	YES	Distributed				
QD1	NO	YES	Root Directed				
QD1	YES	NO	Distributed			✓	✓
QD1	YES	NO	Root Directed			✓	
QD1	YES	YES	Distributed	✓	✓	✓	✓
QD1	YES	YES	Root Directed	✓		✓	
QD2	NO	NO	Distributed			N/A	N/A
QD2	NO	NO	Root Directed			N/A	N/A
QD2	NO	YES	Distributed			N/A	N/A
QD2	NO	YES	Root Directed			N/A	N/A
QD2	YES	NO	Distributed			N/A	N/A
QD2	YES	NO	Root Directed			N/A	N/A
QD2	YES	YES	Distributed	✓	✓	N/A	N/A
QD2	YES	YES	Root Directed	✓		N/A	N/A

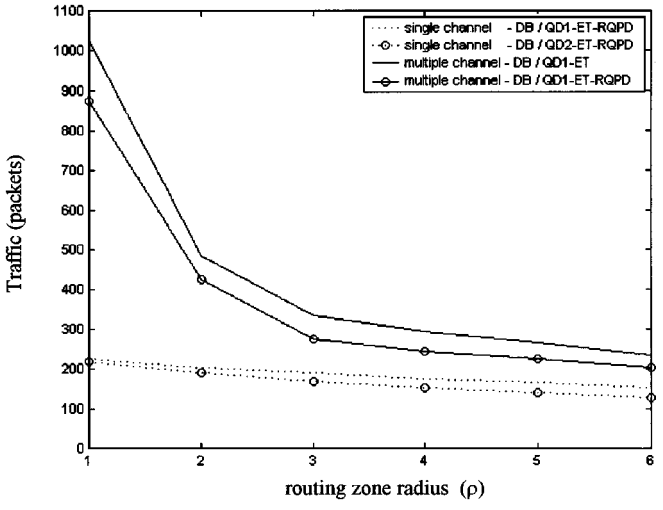

Fig. 10. IERP traffic per route discovery per node.

Fig. 10 demonstrates the extent to which the proposed query control mechanisms suppress redundant query traffic. In multiple-channel networks, setting the routing zone to $\rho = 2$ hops results in about 50% less query traffic than flooding ($\rho = 1$, no RQPD). The extended query detection opportunities provided by RQPD translate to an extra 10% savings in route discovery traffic. As explained earlier, ZRP's impact on single-channel route discovery is not as dramatic as multiple-channel route discovery. Increasing the zone radius from $\rho = 1$ to $\rho = 2$ reduces route discovery overhead by 15%.[9] These effective query control combinations provide further savings as the routing zone radius increases.

Having analyzed the behavior of the individual IARP and IERP components, we now focus our attention on the total ZRP control traffic [Fig. 11(a)–(i)]. The amount of control traffic depends on both node mobility and route query rate. However, we can characterize the *relative* amounts of traffic based on the call to mobility ratio (CMR) (measured as the ratio of route query rate to node speed). As the CMR increases, the ZRP configuration favors larger routing zone radii. For large CMR scenarios, node mobility is relatively low and the cost of maintaining larger routing zones is justified by the resulting reduction in route discovery traffic. In contrast, lower CMR corresponds to relatively

[9]We note for single-channel networks, $\rho = 1$ (with or without RQPD) is slightly more efficient than conventional flood searching. If a node receives a query from all of its known neighbors before making its forwarding decision, it will drop the packet. Flood searching in single-channel networks is equivalent to $\rho = 0$ (no neighbor discovery).

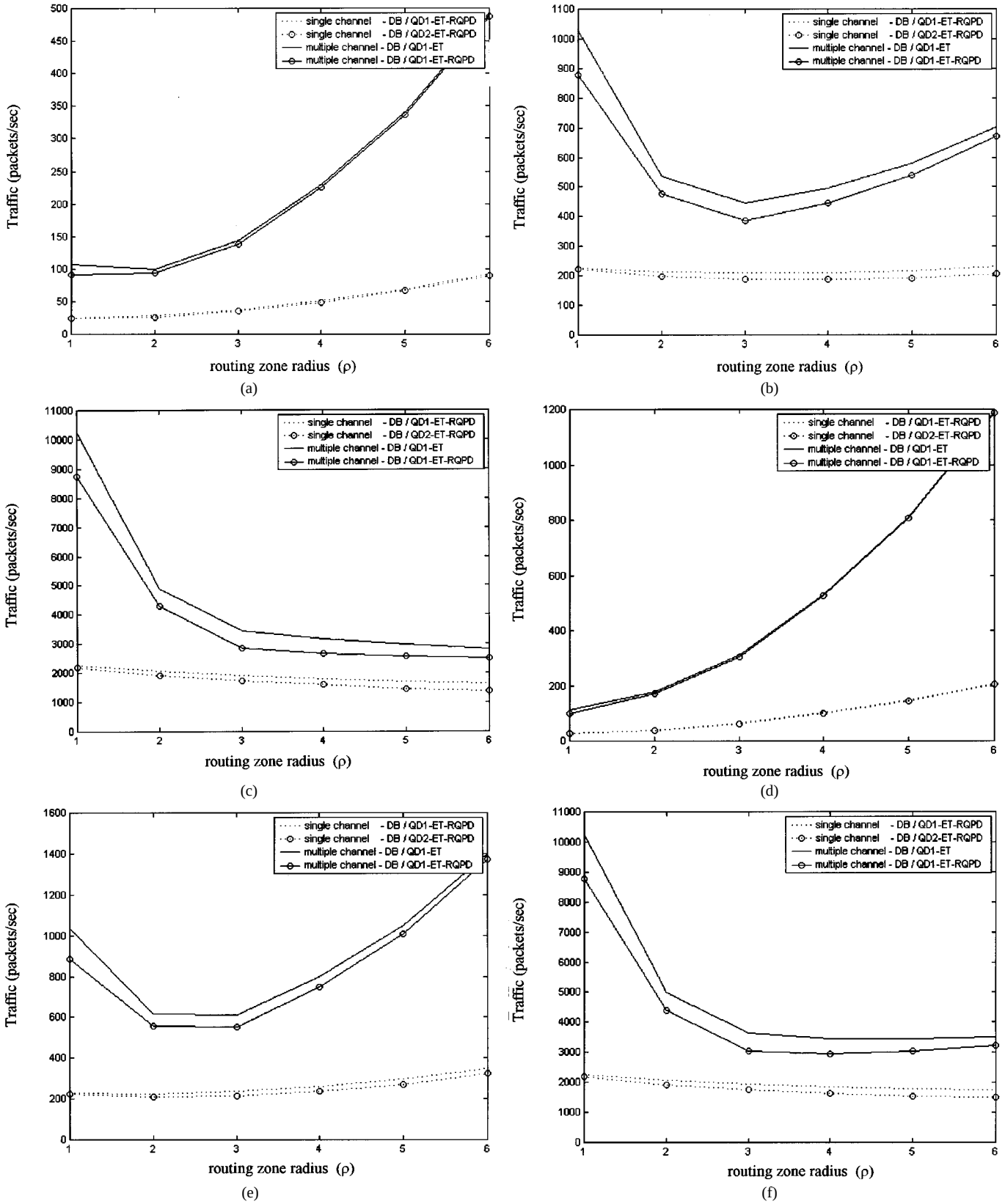

Fig. 11. (a) Total ZRP traffic per node, $v = 10$ [m/s], $R_{\text{query}} = 0.1$ [query/s] (CMR = 10 [query/km]). (b) Total ZRP traffic per node, $v = 10$ [m/s], $R_{\text{query}} = 1.0$ [query/s] (CMR = 100 [query/km]). (c) Total ZRP traffic per node, $v = 10$ [m/s], $R_{\text{query}} = 10.0$ [query/s] (CMR = 1000 [query/km]). (d) Total ZRP traffic per node, $v = 25$ [m/s], $R_{\text{query}} = 0.1$ [query/s] (CMR = 4 [query/km]). (e) Total ZRP traffic per node, $v = 25$ [m/s], $R_{\text{query}} = 1.0$ [query/s] (CMR = 40 [query/km]). (f) Total ZRP traffic per node, $v = 25$ [m/s], $R_{\text{query}} = 10.0$ [query/s] (CMR = 400 [query/km]).

higher mobility scenarios, where routing zone maintenance becomes more costly.

In the context of multiple-channel networks, a zone radius of $\rho = 1$ hop (equivalent to flood searching) is appropriate for

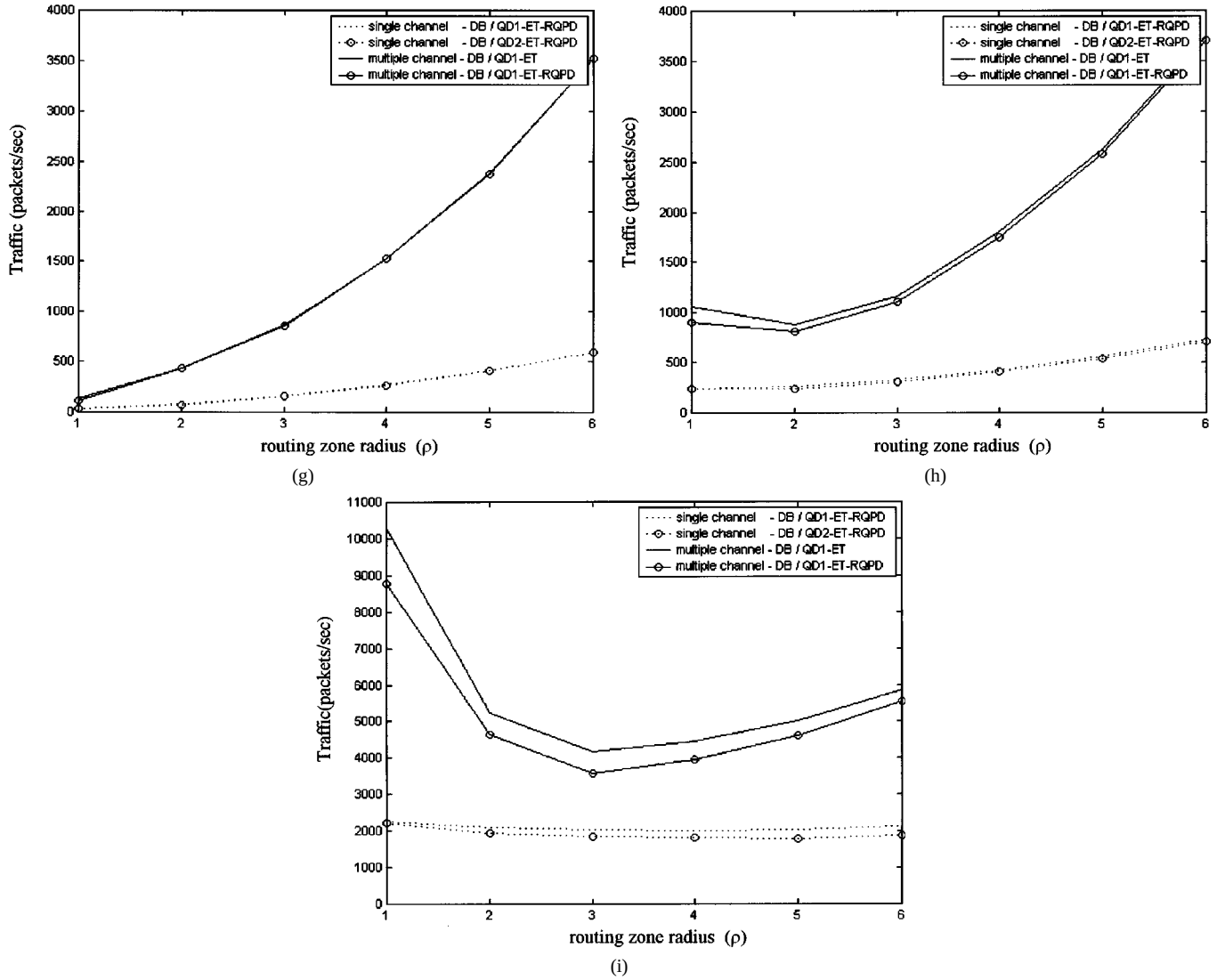

Fig. 11. *(Continued.)* (g) Total ZRP traffic per node, $v = 75$ [m/s], $R_{\text{query}} = 0.1$ [query/s] (CMR = 1.3 [query/km]). (h) Total ZRP traffic per node, $v = 75$ [m/s], $R_{\text{query}} = 1.0$ [query/s] (CMR = 13 [query/km]). (i) Total ZRP traffic per node, $v = 75$ [m/s], $R_{\text{query}} = 10.0$ [query/s] (CMR = 130 [query/km]).

CMR below 15 [query/km]. For larger CMR, routing zones provide an improvement in the overall amount of routing traffic. As an example, a network with CMR = 100 [query/km] [Fig. 11(b)] has an optimal zone radius of $\rho = 3$ hops, generating half the routing traffic of flood searching. Because bordercasting has less impact for single channels, a larger CMR (30 [query/km]) is required to outperform flood searching. Again referring to the example of CMR = 100 [query/km], an optimal ZRP configuration of $\rho = 3$ outperforms purely reactive route discovery by 10%.

The ZRP delay performance is heavily influenced by the use of RQPD. Without RQPD, the route discovery time depends solely on the instantaneous channel load. Compared with the average traffic load, the instantaneous traffic contains a higher percentage of reactive IERP traffic (in particular, the IERP traffic for the current route query). Thus, the zone radius that minimizes route discovery time is at least as large as the zone radius that minimizes control traffic. Comparing Figs. 12 with 11(e), we find that both control traffic and delay are minimized by a zone radius of $\rho = 3$. In this example, the ZRP responds to the route query 60% faster than purely reactive routing.

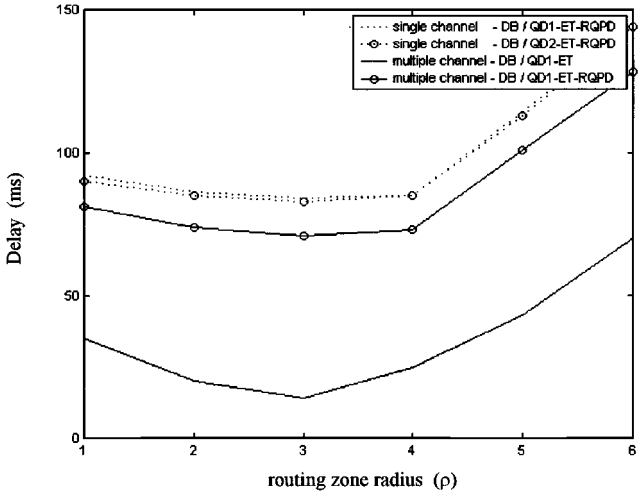

Fig. 12. Route discovery delay: low load, $v = 25$ [m/s], $R_{\text{query}} = 1.0$ [query/s].

When RQPD is applied, the route discovery time mainly consists of the scheduled query-processing delay. As this

scheduled delay is generally several times larger than the MAC layer queueing delay, the route discovery time increases and the zone radius that minimizes the route discovery time also increases. For multiple-channel networks, the 10% traffic savings may not be worth the increase in delay. For example (Fig. 12), an RQPD of 10 ms increases the route discovery time by as much as 300%. For single-channel networks, RQPD and its resulting latency is not optional. However, RQPD can also serve as packet-forwarding jitter (used to ease reduce channel contention and packet loss) used by many reactive routing protocols. As a result, the single-channel ZRP with RQPD is capable of responding faster to a route query than a purely reactive routing protocol with the same forwarding jitter.

VI. Conclusion

The ZRP provides a flexible solution to the challenge of discovering and maintaining routes in a wide variety of *ad hoc* network environments. ZRP combines two radically different methods of routing into one protocol. Intrazone routing uses a proactive protocol to maintain up-to-date routing information to all nodes within its routing zone. In contrast, interzone routing is based on a reactive global route discovery. The amount of intrazone control traffic required to maintain a routing zone increases with the size of the routing zone. However, the structure of the routing zone can be exploited to significantly reduce the amount of reactive interzone control traffic. Using a multicast-based probing service that we refer to as bordercasting, queries may be efficiently directed toward the edge of the queried routing zone, rather than being blindly relayed to all neighbors. Because routing zones heavily overlap, an uncontrolled bordercast can actually produce much more traffic than flood searching. In response to this problem, we have introduced a suite of advanced query control mechanisms (QD1/QD2, ET, RQPD) that effectively combat redundant querying.

For networks characterized by highly mobile nodes and very unstable routes, the hybrid proactive-reactive routing scheme produces less average total ZRP control traffic than purely reactive ($\rho = 1$) or purely proactive ($\rho \to \infty$) routing. Increasingly reactive ZRP configurations (smaller routing zones) appear to be more suitable for networks that exhibit low call to mobility ratios. On the other hand, networks characterized by slower moving, highly active nodes (frequent route requests), lend themselves to a more proactive configuration (larger routing zones).

A ZRP configuration that minimizes control traffic generally provides near-optimal route discovery delay. ZRP has been shown to respond twice as fast as traditional flood-search queries in multiple-channel networks. In the single-channel environment, ZRP's response time is comparable to that of flood searching, but with less routing control traffic. The improvements in route response time are even greater when we consider that a node can immediately provide routes for all of its routing zone nodes.

Acknowledgment

The authors thank the following organizations and individuals for their continuous support of this work: Dr. R. Peterson (Motorola), Dr. M. Kotzin (Motorola), and S. Tabrizi (Rome Laboratories).

References

[1] D. Bertsekas and R. Gallager, *Data Networks*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[2] C. Cheng, R. Reley, S. P. R. Kumar, and J. J. Garcia-Luna-Aceves, "A loop-free extended Bellman-Ford routing protocol without bouncing effect," *ACM Comput. Commun. Rev.*, vol. 19, no. 4, pp. 224–236, 1989.
[3] J. Deng and Z. J. Haas, "Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks," in *Proc. IEEE ICUPC'98*, Florence, Italy, Oct. 1998.
[4] J. J. Garcia-Luna-Aceves, "Loop-free routing using diffusing computations," *IEEE/ACM Trans. Networking*, vol. 1, pp. 130–141, Feb. 1993.
[5] Z. J. Haas, "A routing protocol for the reconfigurable wireless networks," in *Proc. IEEE ICUPC'97*, San Diego, CA, Oct. 1997.
[6] Z. J. Haas and M. R. Pearlman, "The zone routing protocol: A hybrid framework for routing in *ad hoc* networks," in *Ad Hoc Networks*, C. E. Perkins, Ed. Reading, MA: Addison-Wesley, 2000.
[7] P. Jacquet, P. Muhlethaler, and A. Qayyum, "Optimized link state routing protocol," in *IETF MANET*: Internet Draft, Nov. 1998.
[8] D. B. Johnson and D. A. Maltz, "Dynamic source routing in *ad hoc* wireless networking," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Norwell, MA: Kluwer, 1996.
[9] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proc. ACM Mobile Computing and Networking Conf., MOBICOM'95*, Nov. 1995.
[10] 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.
[11] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. IEEE INFOCOM '97*, Kobe, Japan, 1997.
[12] 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.
[13] C. E. Perkins and E. M. Royer, "*Ad hoc* on-demand distance vector routing," in *Proc. IEEE WMCSA'99*, New Orleans, LA, Feb. 1999.
[14] P. F. Tsuchiya, "The landmark hierarchy: A new hierarchy for routing in very large networks," *ACM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 35–42, 1988.

Zygmunt J. Haas (S'84–M'88–SM'90) received the B.Sc. degree in 1979 and the M.Sc. degree in 1985, both in electrical engineering, and the Ph.D. from Stanford University, Stanford, CA, in 1998.

He then joined AT&T Bell Laboratories, Holmdel, NJ, in the Network Research Department, where he pursued research on wireless communications, mobility management, fast protocols, optical networks, and optical switching. From September 1994 until July 1995, he was with the AT&T Wireless Center of Excellence, Whippany, NJ, where he investigated various aspects of wireless and mobile networking, concentrating on TCP/IP networks. As of August 1995, he has been with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY. He is an author of numerous technical papers and holds 14 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 tutorials at major IEEE and ACM conferences, and serves as Editor of several journals and magazines, including the IEEE Transactions on Networking. He has been a Guest Editor of three IEEE Journal on Selected Areas in Communication issues (Gigabit Networks, Mobile Computing Networks, and *Ad Hoc* Networks). He is a voting member of ACM, and a Chair of the IEEE Technical Committee on Personal Communications.

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 currently working toward the Ph.D. degree in the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY.

His research interests include routing protocol design and wireless channel modeling for *ad hoc* networks.