The Performance of Query Control Schemes for the Zone Routing Protocol♥

Zygmunt J. Haas
School of Electrical Engineering, Cornell University
Ithaca, NY 14853
Tel: (607) 255-3454
haas@ee.cornell.edu

Marc R. Pearlman
School of Electrical Engineering, Cornell University
Ithaca, NY 14853
Tel: (607) 254-8816
pearlman@ee.cornell.edu

ABSTRACT

In this paper, we study the performance of route query control mechanisms for the Zone Routing Protocol (ZRP) for ad-hoc networks. The 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 in the generation of control traffic than traditional routing schemes. However, without proper query control techniques, the ZRP cannot provide this reduction in the control traffic.

Our proposed query control schemes exploit the structure of the routing zone to provide enhanced <u>detection</u> *(Query Detection (QD1/QD2))*, <u>termination</u> *(Loop-back Termination (LT), Early Termination (ET))* and <u>prevention</u> *(Backward Search Prevention (BSP), Selective Bordercasting (SBC))* of overlapping queries. These techniques can be applied to single-channel or multiple-channel ad-hoc networks to improve both the delay and control traffic performance of the ZRP. Our query control mechanisms allow the ZRP to provide routes to all accessible network nodes, with only a fraction of the control traffic generated by purely proactive distance vector or purely reactive flooding schemes, and in less than half the time of a route query flood.

1. 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

♥ This work is supported by the US Air Force/Rome Labs, under the contract number C-7-2544 and a grant from Motorola Corporation, the Applied Research Laboratory

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 [14]) 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 protocol (OLSR) [8]). 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) ([11] and [12]), 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) [13], 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) [9] and Ad-Hoc On Demand Distance Vector (AODV) [15] 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.

2. THE ZONE ROUTING PROTOCOL – 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 *Zone Routing Protocol (ZRP)* ([6] and [7]), 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 referred to quite often in the operation of the ZRP, minimizing the waste associated with the purely proactive schemes. On the other hand, the search throughout the network, although it is 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 the ZRP architecture.

The Notion of a Routing Zone and Intrazone Routing

In the 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.

Figure 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).

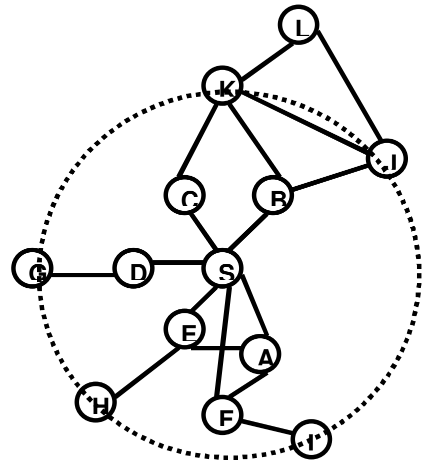


Figure 1: A Routing Zone of Radius 2 Hops

The construction of a routing zone requires a node to first know who its *neighbors* are. A neighbor is defined as a node that can communicate directly with the node in question[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 MAC-level *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.

The ZRP maintains routing zones through a proactive scheme called the IntrAzone Routing Protocol (IARP). In this paper, the IARP is implemented as a Distance Vector protocol with "split-horizon"[2]. Compared with link state alternatives, this IARP realization places a low computational load on each node, while producing less proactive routing traffic. The split horizon mechanism and the limited scope of the routing zones ease the potential convergence problems associated with distance vector protocols. Although we use the above IARP scheme, we note that , in principle, any proactive routing scheme may serve as an IARP, provided that the range of its routing messages is limited to the zone radius.

Changes in neighbor connectivity trigger the broadcast of IARP route update packets reflecting the new status to the neighbor. In the case of a neighbor discovery, the node also transmits a copy of its routing table to the new

---

[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.

[2] Route messages include route cost, next hop and next-next hop. This allows a node to identify two-hop loops, significantly improving the convergence for large routing zones.

neighbor. A node that receives an IARP route update packet will record the updated route information and determine if the update results in a change to the shortest path to that route's destination. If so, the node will broadcast an IARP route update reflecting this change. Whereas regular proactive schemes will allow updates to propagate throughout the entire network, the IARP uses the route hop count to prevent updates from propagating farther than the zone radius.

Interzone Routing

Whereas the 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. The IERP uses a query-response mechanism to discover routes on demand.

The 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 send a message to its peripheral nodes. The ZRP provides this service through a component called the Bordercast Resolution Protocol (BRP).

In its simplest form, bordercasting may be implemented through network layer unicasting or multicasting of messages to the peripheral nodes. This approach prevents non-peripheral nodes from accessing the bordercasted messages as they are relayed to the edge of the routing zone. As will be shown later, such access is central to the control of the route query process. A more suitable approach to bordercasting is to have the BRP, rather than IP, forward messages to peripheral nodes. In such an implementation, either IP unicast or IP broadcast is used as the network layer delivery mechanism.

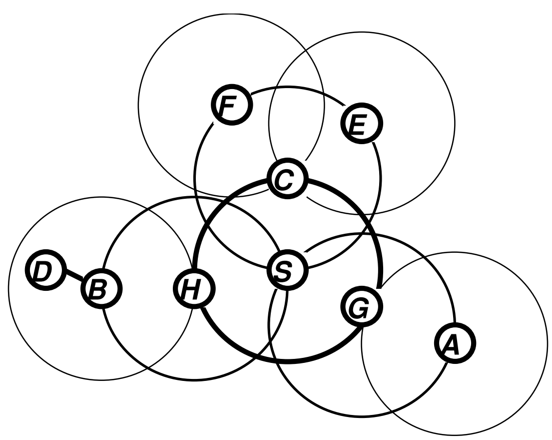


Figure 2: An Example of IERP Operation

An IERP route query is triggered by the network layer, when a data packet is to be sent to a destination that does not lie within its routing zone[3]. 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 IDs specifies an *accumulated route* from the source to the current routing zone.[4] If the

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

[4] 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.

destination does not appear in the node's routing zone, the node bordercasts the query to *its* peripheral nodes[5]. 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 Figure 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, forwards 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[6] 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).

The inter-relationship of the ZRP component protocols is illustrated in Figure 3. The proactive maintenance of the routing zone topology is performed by the IARP, through exchange of route update packets. Route updates are triggered by the MAC-level NDP, which notifies the IARP when a link to a neighbor is established or broken. The IERP reactively acquires routes to nodes beyond the routing zone using a query-reply mechanism. The IERP forwards queries to its peripheral nodes through a bordercast delivery service provided by the BRP. The BRP keeps track of the peripheral nodes through up-to-date routing zone topology information provided by the IARP. The IERP also makes use of the IARP routing zone information to determine whether a queried-for-destination belongs to its routing zone.

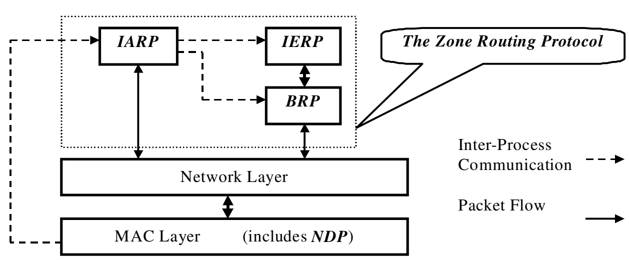


Figure 3: ZRP's Architecture

The relationship between the IARP and IERP may, at first, give the impression that the ZRP is a hierarchical routing protocol. In fact, the ZRP bears only a superficial resemblance to such protocols. Hierarchical routing relies on the strategic assignment of gateways or landmarks[16] to establish a hierarchy of subnets for the entire

---

[5] Excluding the peripheral node from which the route query was received.

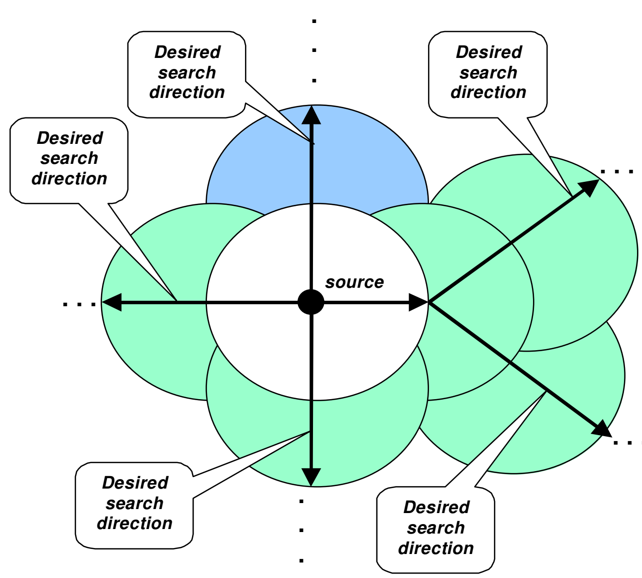
[6] Typical path metrics include hop count, delay, capacity, etc.

network.[7] 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 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 the 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 the ZRP as a flat rather than a hierarchical routing protocol.

3. QUERY CONTROL MECHANISMS

Because the routing zones heavily overlap, the route query will be forwarded to many network nodes, multiple times. In fact, it is very possible that the query will be forwarded to all the network nodes, effectively flooding the network. But a more disappointing result is that, due to fact that bordercasting involves sending the query over a path of length equal to the zone radius, the IERP will result in more traffic than the flooding itself! 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 a 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: A node's response to a route query contains information about that node's entire routing zone. From this perspective, excess route query traffic can be regarded as a result of overlapping query threads (i.e. overlapping queried routing zones). Thus, the design objective of query control mechanisms should be to reduce the amount of route query traffic by steering threads outward from the source's routing zone and away from each other (see Figure 4). This problem is addressed from two different perspectives: thread overlap detection/termination and thread overlap prevention.



[7] 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.

3.1. Query Detection and Query Termination

The standard approach to query thread termination is to discard a thread when it appears at a previously queried node. However, this does not fully exploit the structure of the routing zone. A broader approach is to discard a thread when it appears in a previously queried zone. This introduces the first challenge for the design of an effective termination mechanism: How can a previously queried zone be identified when only one node (the central node) was queried?

3.1.1. Query Detection (QD1/QD2)

A majority of thread overlapping occurs by a thread appearing in a zone that was previously queried by another thread. The ability to terminate in this situation strongly depends on the ability of nodes to detect that a routing zone that they belong to has been previously queried. Clearly, the central node in the routing zone (which processed the query) is aware that its zone has been queried. In order to notify the remaining routing zone nodes, without introducing additional control traffic, some form of "eavesdropping" needs to be implemented. Based on the ZRP architecture described earlier, it is most convenient to perform query detection at the BRP, which is responsible for query delivery. The first level of Query Detection - QD1 - allows the intermediate nodes, which relay queries to the edge of the routing zone, to detect these queries. In single-channel networks, it may be possible for queries to be detected by any node within the range of a query-transmitting node. This extended query detection capability - QD2 - can be implemented by means of promiscuous IP and MAC layer operation. Figure 5 illustrates both levels of advanced query detection. In this example, node S bordercasts to two peripheral nodes, B and D. The intermediate nodes A and C are able to detect passing threads using QD1. If QD2 is implemented, node E will be able to "eavesdrop" on A's transmissions and record the query as well.

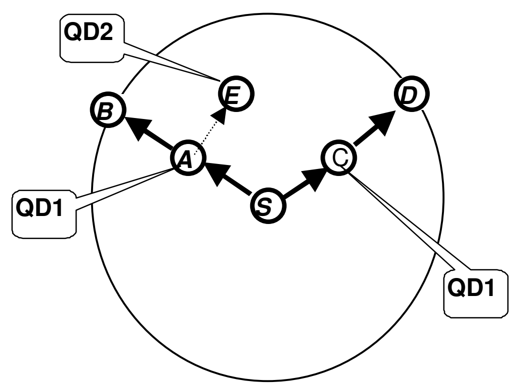


Figure 5: Advanced Query Detection (QD1/QD2)

At a minimum, the query detection scheme needs to record the query's source node 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 are the addresses of the most recently visited peripheral node and neighbor. This information provides valuable insight into the recent history of the query thread, which can be used to terminate or prevent redundant queries.

3.1.2. Early Termination (ET)

The standard flood termination criteria can be tightened by discarding a thread as it enters a previously queried *zone*. When the ability to terminate threads is limited to peripheral nodes, threads are allowed to penetrate *into* previously covered areas, generating redundant control traffic. This excess traffic can be reduced, by extending the thread termination capability to the *intermediate nodes* that relay the thread. We refer to this approach as Early Termination (ET).

In order for ET to operate correctly, the proper termination criteria must be identified. A natural choice might be to terminate all but the first detected query packet. However, this approach turns out to be too strong, causing the query to fail under certain circumstances. A less stringent, but correct, approach is to accept query packets only if the ID of the most recent peripheral node does not exceed the ID of all previously detected bordercasting nodes. Figure 6 illustrates the operation of the basic ET mechanism. Assume that node B has previously detected a query. Node S bordercasts a route query, with node C as one of the intended recipients. Intermediate node A passes along the query to B. First assume that node B has previously detected a query bordercasted by node Z. Since S < Z (alphabetically), node B will relay the query to node C. On the other hand, if node B had previously detected a query that was bordercasted by node M, then node B would terminate this query thread (because S > M).

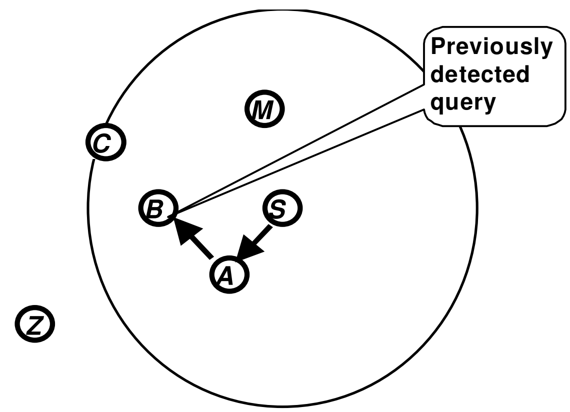


Figure 6: Early Termination (ET)

We note that ET only allows partial participation of intermediate nodes in the route query process. Intermediate nodes are restricted from issuing new queries (i.e. in the case of QD2 eavesdropping). Otherwise, the ZRP would degenerate into a flooding protocol.

<u>ET termination criteria:</u>

1. *query*→{source,id} ∈ *detected queries*
2. *query*→prev_periph > min {*detected queries*[][8]→prev_periph}

[8] Unless stated otherwise, a reference to the detected queries[ ] implies the entries corresponding to the {source, id} pair of the current query (i.e. query→{source,id})

3.1.3. Loop-back Termination (LT)

When all nodes are configured with the same zone radius, routing zone membership is commutative (i.e. if node A belongs to node B's routing zone, then node B belongs to node A's routing zone). It is relatively easy to identify a thread that returns to a routing zone that it previously queried. A node simply examines the accumulated route in the received route query packet to determine if any hop prior to the most recent bordercasting node lies *inside* of its routing zone. If the loop-back condition exists, the thread is discarded. An example of this scheme, which we refer to as Loop-back Termination (LT), is shown in Figure 5. Node S bordercasts a route query to A, which bordercasts it to B, which in turn bordercasts it to C. C terminates the thread (i.e. does not bordercast) because node S, which appears in the accumulated route, also lies within C's routing zone. LT is an ideal mechanism to handle thread loop-back, because the information provided by the accumulated route is sufficient to identify all cases of loop-back.

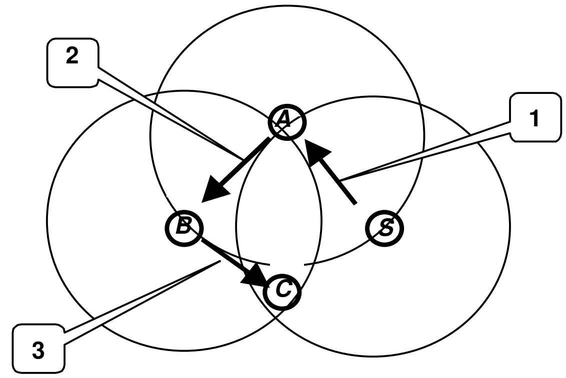


Figure 7: Loop-back Termination (LT)

LT termination criteria:

1. node $\in$ *route*
2. $i < j$, where *route*[i] = = node and *route*[j] = = prev_periph
3. node $\in$ {*routing zone*}
4. node $\notin$ {*peripheral nodes*}

3.2. Query Overlap Prevention

The mechanisms described in Section 3.1 are used to remedy an existing condition of overlapping query threads. Further querying efficiency can be achieved through techniques that use knowledge of the local network topology and/or query history to encourage outward propagation of the query. We introduce two mechanisms, Backward Search Prevention (BSP) and Selective Bordercasting (SBC), that prevent local overlap of query threads by focussing the bordercast to a properly chosen subset of peripheral nodes.

3.2.1. Backward Search Prevention (BSP)

The information acquired from the query detection (QD) mechanisms can be used by bordercasting nodes to determine which of its peripheral nodes are "upstream", i.e. in the direction of the query source. Overlap prevention then becomes a matter of excluding these "upstream" peripheral nodes from the set of bordercast recipients. In the case of traditional flood search, an upstream node is identified as the neighbor from which the query was received. Backward Search Prevention (BSP) extends this idea to the general case of bordercasting. Using detected query information, a node first determines who its upstream *neighbors* are. Having identified the upstream neighbors, the upstream *peripheral nodes* are designated as the peripheral nodes that can be reached, in minimum distance, from the upstream neighbors.

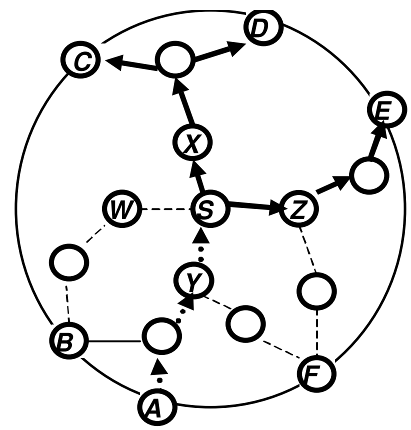


Figure 8: Backward Search Prevention (BSP)

The operation of BSP is illustrated in Figure 8. Node S receives a route query from node Y. By definition this makes Y an upstream neighbor. By examination of its IARP routing table, S determines that peripheral nodes A, B and F are upstream since their routing zone membership is satisfied through (upstream) neighbor Y (i.e. the zone radius is 3 hops and A, B and F are reachable through Y in 3 hops). Node S proceeds to bordercast the query (outward) to the remaining peripheral nodes: C, D and E.

> <u>BSP bordercast selection criteria:</u>
>
> Given the modified routing zone {routing zone'}:
>
> {*routing zone'*} = {*routing zone*}- {*detected queries*[]→prev_neighbor}
>
> Select node as bordercast recipient if:
>
> node ∈ {*peripheral node'*}

3.2.2. Selective Bordercasting (SBC)

Bordercasting is based on the notion that a node can use knowledge of the local network topology to direct queries outward, toward target peripheral nodes. These peripheral nodes represent the extent of a node's routing zone, and thus the extent of a node's ability to steer its descendent threads away from each other. Selective Bordercasting (SBC) expands the range of a bordercasting node's control, by working within the context of a larger "extended routing zone". In the case of a ρ hop "standard" routing zone, sufficient knowledge of a $k \cdot \rho$ hop "extended" routing zone can be used to prevent k successive generations of descendent threads (relayed through k layers of peripheral nodes) from overlapping. In particular, the IARP must provide information about the reachability of $(j-1)^{th}$ layer peripheral nodes to j^{th} layer peripheral nodes ($1 \le j \le k$).

For the remainder of this paper, we will restrict our attention to the simple case of SBC with k=2. For convenience, we will refer to the first layer peripheral nodes (distance ρ hops) as inner peripheral nodes, and the second layer peripheral nodes (distance 2·ρ hops) as outer peripheral nodes. Prior to bordercasting, a node first determines the subset of outer peripheral nodes covered by its assigned inner peripheral nodes. The node then bordercasts to a subset of the assigned inner peripheral nodes which form a "minimal" partitioning set of the outer peripheral nodes. Figure 9 provides an illustrative example of a SBC application. Node S's inner peripheral nodes are A, B and C. Node S's outer peripheral nodes are F, G, H, X, Y and Z. We can see from the overlapping routing zones that the two inner peripheral nodes of node B (H and X) are also inner peripheral nodes of A and C. Consequently, node S can choose to eliminate node B from its list of bordercast recipients, as node A can provide coverage to H and node C can cover X. In this way, we are able to provide full coverage over the extended zone while preventing overlapping queries.

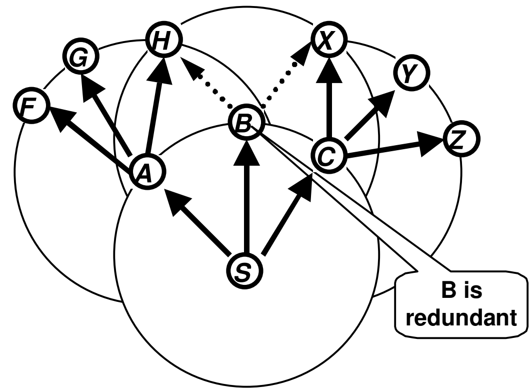


Figure 9: Selective Bordercasting (SBC)

The proposed technique for computing this minimal partitioning set is based on the "greedy" heuristic introduced in [8]. Each of the selected inner peripheral nodes is sent a list (in the route query packet) of the outer peripheral nodes that it partitions. This list becomes the recipient node's set of assigned inner peripheral nodes. The restriction imposed on the set of inner peripheral nodes helps to direct query threads outward from the source, rather than overlap or loop back on themselves. A detailed description of the SBC algorithm can be found in the appendix.

Unlike the other query control techniques, SBC does not come for free. The amount of IARP traffic increases to provide information about the extended zone topology. In addition, the length of each IERP query packet increases in order to accommodate the list of assigned peripheral nodes. To be viable, this added cost must be offset by the reduction in query packet transmissions due to overlap prevention.

4. EVALUATION OF THE ZRP

The performance of the ZRP was evaluated based on simulations of 500 node ad-hoc networks, over a range of routing zone radii (ρ), from purely reactive routing (ρ = 1 hop) to purely proactive routing (ρ→∞ hops). Performance was gauged by measuring the control traffic generated by the ZRP and the average response time of the reactive route discovery process.

Measurements of control traffic are reported in units of ID fields transmitted at the network layer. This allows us to account for the variable length of the IERP control packets due to factors such as route accumulation. The use of ID fields also allows us to report results that are independent of the size (bits) of a node ID. The overall ZRP control traffic is viewed as the sum of the ID fields in the transmitted intrazone update packets and the interzone route request/reply packets. The neighbor discovery beacons are excluded from our measurement of ZRP control traffic because we assume that this service is already provided in conjunction with the MAC protocol.

The delay performance of the ZRP is reflected by the average delay of an IERP route discovery (delay between the time that a route query packet is issued and the first route response packet is received). Like our measurements of control traffic, we generalize the delay performance by expressing it in terms of the transmission delay of an ID field.

Delay is evaluated under low load and high load network conditions, for a representative scenario of mobile speed v = 20 [m/s] and route query rate R_{query} = 0.1 [query/sec]. In the low load scenario, the amount of application traffic is negligible in comparison to the ZRP control traffic. In the high load case, each node relays 0.05 [Erlangs] of randomly generated application traffic in addition to the ZRP control traffic. For our single channel system where each node has an average of six neighbors, 0.05 [Erlang] represents about 1/3 the available channel capacity. The remaining 2/3 channel capacity is sufficient to evaluate the ZRP over a wide range of sub-optimal configurations , without risking buffer overflow. 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 500 mobile nodes, whose initial positions are chosen from a uniform random distribution over an area of 4000 [m] by 4000 [m]. All nodes move at a constant speed, v, with an initial direction,[9] 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_{xmit} = 500 [m]. Within a range of d_{xmit}, the average power of the desired signal (and resulting average SIR) 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.

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

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 our model, 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} = 0.25 seconds. Neighbor connectivity is determined by the reception of the HELLO beacons. If a new beacon fails to arrive within $2 \cdot T_{beacon}$ of the most recent beacon, a link failure is 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 are given highest transmission priority and are not destroyed by collisions. This prevents the inaccurate reporting of link failures for the allowed $2 \cdot T_{beacon}$ window.

The ZRP evaluation was performed using a custom-built event driven simulator. Simulations were run on 50 randomly distributed network configurations, each for a duration of 125 seconds. No data was collected for the first 5 seconds of the simulations to avoid measurements before 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.

Parameter	Symbol	Value
Number of nodes	N	500
Network coverage area	A	4000 [m] x 4000 [m]
Transmission radius	d_{xmit}	250 [m]
Beacon period	T_{beacon}	0.2 [sec]
Transmission rate	R_{xmit}	2.0 [Mbps]

Table 1: Fixed Simulation Parameters

Parameter	Symbol	Values
Routing zone radius	ρ	1-8 [hops]
Node speed	V	10-75 [m/sec]
Mean route query rate	R_{query}	0.1-10.0 [query/s/node]

Table 2: Variable Simulation Parameters

5. PERFORMANCE RESULTS

Figure 10 illustrates the dependence of intrazone control packets on the routing zone radius, ρ, for various rates of network reconfiguration. A distinction is made between the full bordercasting and selective bordercasting schemes because the selective bordercasting requires the IARP to maintain an extended zone of radius $2 \cdot \rho$. The increase in IARP traffic resulting from the extended routing zone is shown to be quite significant. In both cases, the amount of IARP control traffic *per node* is approximately proportional to ρ^2. This behavior is to be expected, since 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. It should be noted that there is no intrazone control overhead for $\rho = 1$. All nodes within a routing zone of $\rho = 1$ are, by definition, neighbors. Consequently, the Neighbor Discovery Protocol provides all of the information needed to maintain connectivity within the routing zone.

We begin our examination of the IERP control traffic by determining the effectiveness of the 48 possible combinations of query control mechanisms. 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 ID fields and IERP packets. IERP *ID fields* 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.

Table 3 demonstrates the power of the overlap prevention schemes (SBC and BSP). Selective bordercasting is able to effectively eliminate redundant IERP query traffic (both packets and fields), even without any other control mechanisms in place. Of course, this benefit comes at the expense of additional IARP packets, IERP data and increased processing overhead. Without SBC, effective control of query packets requires a combination of mechanisms that includes the preventative BSP and either QD1 or QD2. In the case of QD1, which represents the highest level of query detection for multiple channel systems, ET is also required. When QD2 is employed (i.e. in single channel systems) ET is optional.

Because the IERP needs only to record the IDs of *peripheral* nodes in the accumulated route, the amount of *data per query packet* naturally decreases as zone radius increases (approximately proportional to $1/\rho$). This eases the query control requirements: the amount of data may decrease with the zone radius even if the number of packets remains constant or increases slightly. We find that when some form of advanced query detection is employed (QD1 or QD2) <u>either</u> ET or BSP is sufficient to effectively control the amount of IERP data.

				NO-SBC		SBC	
QD	ET	LT	BSP	Packet	Field	Packet	Field
No	NO	NO	NO			✓	✓
No	NO	NO	YES			✓	✓
No	NO	YES	NO			✓	✓
No	NO	YES	YES			✓	✓
No	YES	NO	NO			✓	✓
No	YES	NO	YES		✓	✓	✓
No	YES	YES	NO			✓	✓
No	YES	YES	YES		✓	✓	✓
QD1	NO	NO	NO			✓	✓
QD1	NO	NO	YES		✓	✓	✓
QD1	NO	YES	NO			✓	✓
QD1	NO	YES	YES		✓	✓	✓
QD1	YES	NO	NO		✓	✓	✓
QD1	YES	NO	YES	✓	✓	✓	✓
QD1	YES	YES	NO		✓	✓	✓
QD1	YES	YES	YES	✓	✓	✓	✓
QD2	NO	NO	NO			✓	✓
QD2	NO	NO	YES	✓	✓	✓	✓
QD2	NO	YES	NO			✓	✓
QD2	NO	YES	YES	✓	✓	✓	✓
QD2	YES	NO	NO		✓	✓	✓
QD2	YES	NO	YES	✓	✓	✓	✓
QD2	YES	YES	NO		✓	✓	✓
QD2	YES	YES	YES	✓	✓	✓	✓

Table 3: Effective Combinations of Query Control Mechanisms

It is interesting to note the limited role of LT in controlling the spread of query packets. Adding LT to a combination of ineffective control mechanisms does not help to make it effective. Likewise, any combination that is effective with LT is also effective without it. This is because LT only terminates query threads that loop-back on themselves. In practice, most redundancy is due to the overlap of two parallel threads, which is directly addressed by the other mechanisms.

Because the LT, ET, QD1/QD2 and BSP techniques can be implemented with no additional traffic and negligible computational overhead, the full array of *valid* query control techniques can be applied to provide the best IERP traffic performance. Figure 11 clearly demonstrates the extent to which the proposed query control mechanisms suppress redundant query traffic. As stated earlier, in the absence of an effective query control strategy, the amount of reactive traffic will increase with the size of the routing zone. When none of the proposed query control schemes are employed, we observe that the amount of query traffic increases linearly with the routing zone radius. For $\rho = 3$, for example, the IERP without query control generates about twice as much traffic as flooding ($\rho = 1$), and about 10 – 50 times as much traffic as the most effective query control mechanisms.

Even without the powerful overlap prevention capabilities of SBC, the ZRP produces much less query traffic than flooding. In multiple channel networks, increasing the routing zone to just $\rho = 2$ hops results in 40% less query traffic than flooding. Even better performance can be achieved in single channel networks, which can implement QD2. The added benefit of eavesdropping can lead to a *further* savings of 60% less query traffic. The impact of these mechanisms becomes more significant as the routing zone radius increases.

All else being equal, selective bordercasting provides a substantial reduction in IERP traffic compared with full bordercasting. With the SBC, the ZRP generates less than 20% of the query traffic of a traditional flood search. We note that in the case of SBC, there is not a significant performance improvement between QD1 and QD2. Because SBC is so successful at preventing local query overlap, the relative impact of these termination mechanisms is diminished.

Having analyzed the behavior of the individual IARP and IERP components, we now focus our attention on the total ZRP control traffic. Figures 12 a-i show how the ZRP can be optimized for different conditions of node mobility and call activity, through the adjustment of the routing zone radius. Keeping the route query rate fixed, we see that the optimal routing zone radius decreases as nodal velocity increases. Increased nodal velocity causes the network to reconfigure more rapidly, resulting in an increased of IARP route update traffic. Likewise, we find that, for a constant node velocity, the optimal routing zone radius increases with the route query rate. Increased call activity results in the generation of additional IERP route query traffic, but has no effect on the reconfiguration rate of the network (i.e. no effect on the IARP traffic). We summarize these trends as follows: increased CMR[10] favors a more proactive ZRP configuration (larger routing zones). Likewise, decreased CMR favors a more reactive ZRP configuration (smaller routing zones).

Comparing selective bordercasting with full bordercasting, we find that the selective bordercasting implementation favors a more reactive ZRP configuration. This is to be expected, since selective bordercasting was shown to improve the efficiency of the reactive IERP, while adding significant cost to the proactive IARP. In single channel networks, the best full bordercasting solution appears comparable to the best selective bordercasting approach. In multi-channel networks, where QD2 may not be employed, selective bordercasting may result in about half as much traffic as a full bordercasting approach.

The route discovery delay is closely related to the amount of ZRP control traffic. Excessive amounts of control traffic increase channel contention, thereby increasing queueing delays. Configuring the ZRP for minimum control traffic relieves much of the channel contention burden, but does *not necessarily* minimize the route discovery delay. While the proactive IARP traffic may occur at any time, the IERP traffic is concentrated during the query periods. As a result, for low network loads, the amount of IERP traffic tends to exert more influence over route discovery delay than the IARP traffic. Comparing figure 13a with 12d, we see that, under low load operation, improved query delay performance may be achieved by reducing the IERP traffic at the expense of additional IARP (through an increase in zone radius). Under high load operation (fig 13b), the large amount of application traffic increases the delay of the proactive *IARP* traffic. As a result, more IARP traffic is present in the system at any given time. Under these conditions, the influence of the IARP traffic may match (or even exceed) the IERP traffic's influence on route discovery delay.

It is interesting to note that the route discovery delays for QD1 and QD2 are similar, even though the eavesdropping QD2 produced half the query traffic. Recall that the QD schemes reduce the amount of query traffic by terminating query packets that return to previously queried regions. Most of these redundant packets appear *after* the original "successful" packet has left the local region. As a result, the channel's availability for the "successful" query packet does not strongly depend on the amount of *that query's* redundant query traffic. On the other hand, in situations where the average route query rate (R_{query}) is high, the impact of the redundant traffic of *other queries* becomes more pronounced. Such circumstances lead to a greater disparity between the QD1 and QD2 delays.

[10] Call-to-mobility ratio. Increased CMR corresponds to increased route query rate or decreased node mobility. Likewise, decreased CMR corresponds to either decreased route query rate or increased node mobility.

The optimal route discovery delay does not appear to depend heavily on the use of selective bordercasting or eavesdropping. Under high load conditions, the route discovery delay is approximately half the delay of traditional flood-search queries. For low network loads, the ZRP can respond to queries three times faster than the flood-search. If the zone radius is configured to minimize control traffic *rather than delay*, the resulting route discovery delay is within about 15% of the optimal value, only slightly diminishing the delay performance relative to flooding.

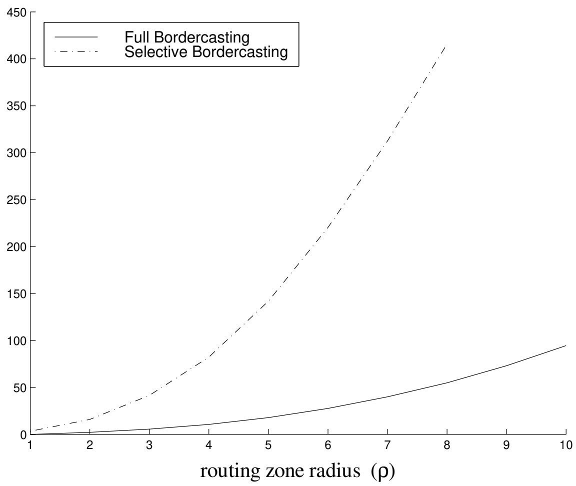


Figure 10: IARP Traffic Per m/s

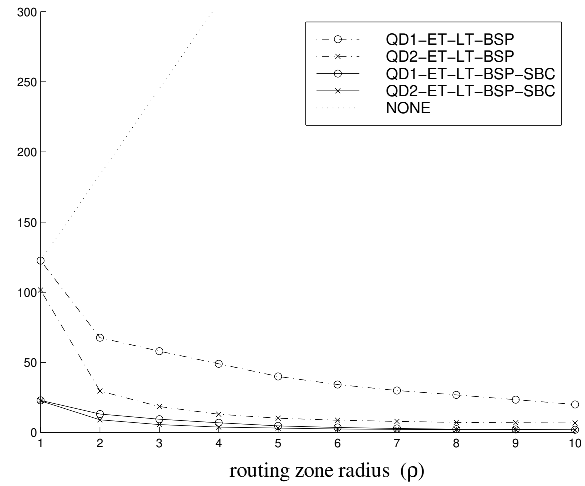


Figure 11: IERP Traffic per Route Discovery per Node

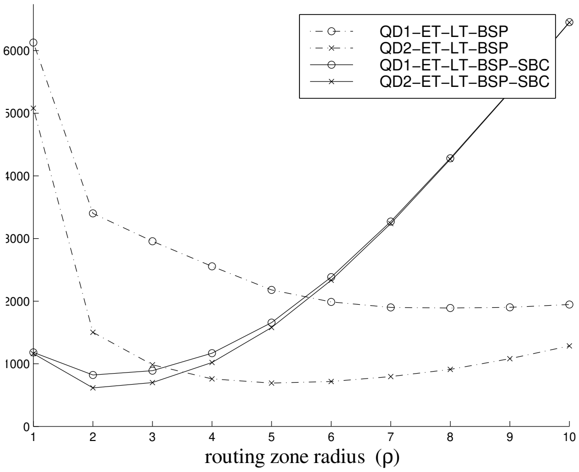


Figure 12a: Total ZRP Traffic per Node

v = 10 [m/s], R_{query} = 0.1 [query/s] (CMR = 10 [query/km])

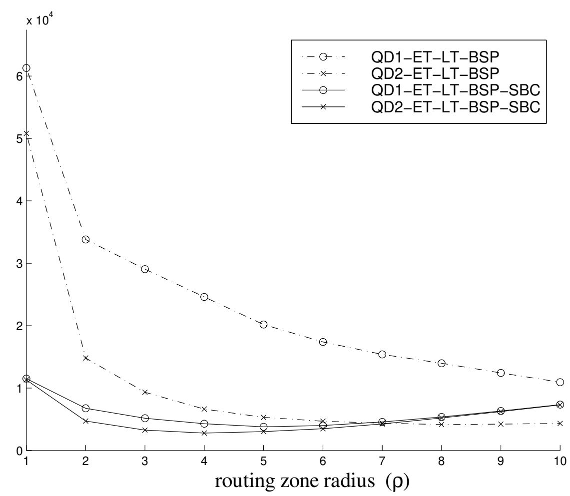


Figure 12b: Total ZRP Traffic per Node

v = 10 [m/s], R_{query} = 1.0 [query/s] (CMR = 100 [query/km])

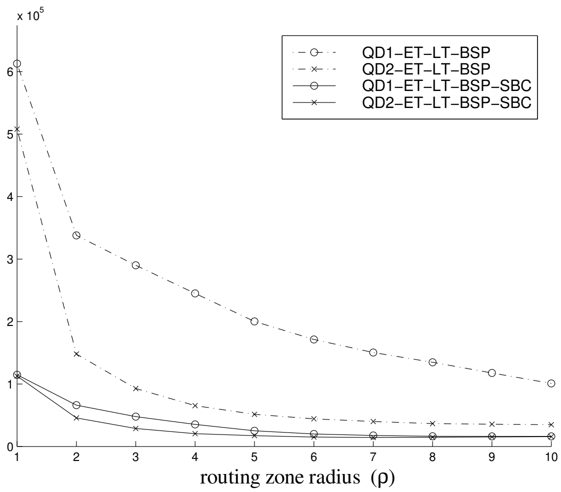


Figure 12c: Total ZRP Traffic per Node
v = 10 [m/s], R_{query} = 10.0 [query/s] (CMR = 1000

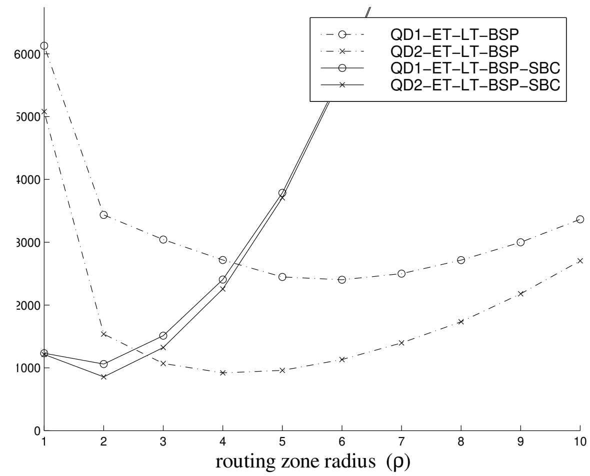


Figure 12d: Total ZRP Traffic per Node
v = 25 [m/s], R_{query} = 0.1 [query/s] (CMR = 4 [query/km])

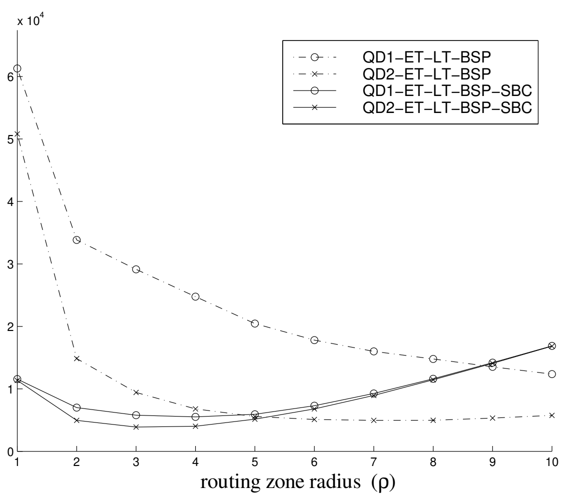


Figure 12e: Total ZRP Traffic per Node
v = 25 [m/s], R_{query} = 1.0 [query/s] (CMR = 40 [query/km])

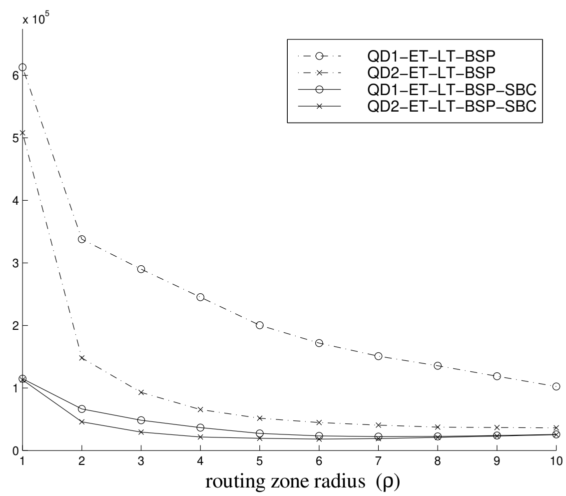


Figure 12f: Total ZRP Traffic per Node
v = 25 [m/s], R_{query} = 10.0 [query/s] (CMR = 400 [query/km])

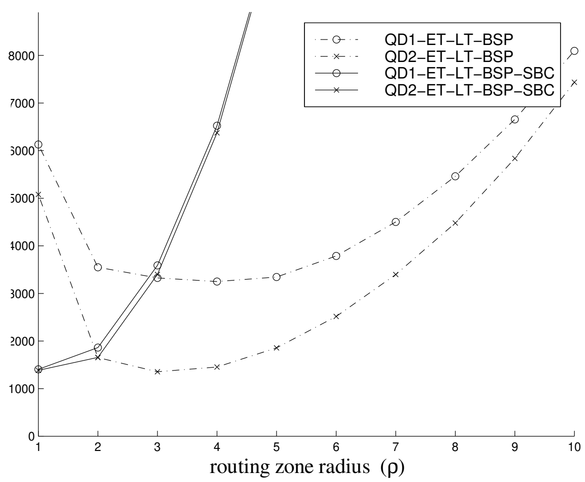


Figure 12g: Total ZRP Traffic per Node
v = 75 [m/s], R_{query} = 0.1 [query/s] (CMR = 1.3 [query/km])

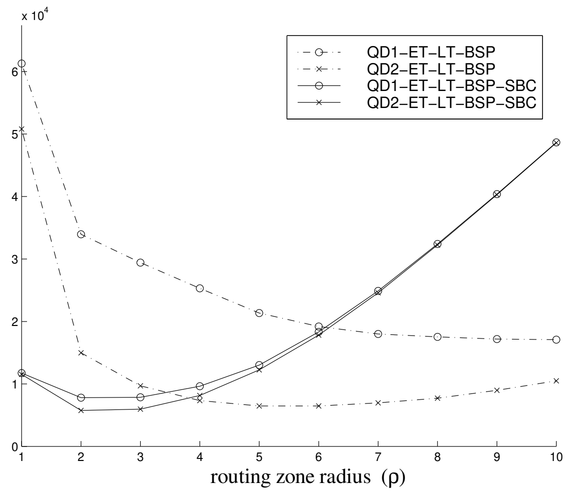


Figure 12h: Total ZRP Traffic per Node
v = 75 [m/s], R_{query} = 1.0 [query/s] (CMR = 13 [query/km])

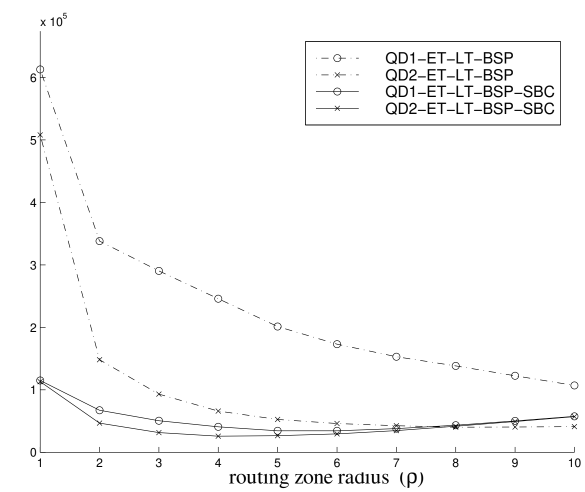


Figure 12i: Total ZRP Traffic per Node
v = 75 [m/s], R_{query} = 10.0 [query/s] (CMR = 130 [query/km])

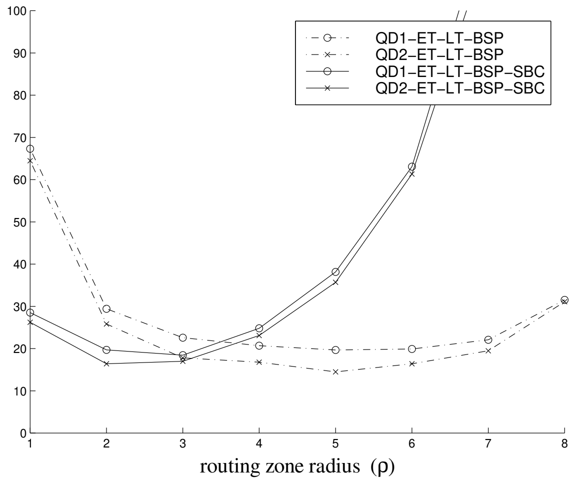


Figure 13a: Route Discovery Delay: Low Load
v = 25 [m/s], R_{query} = 0.1 [query/sec]

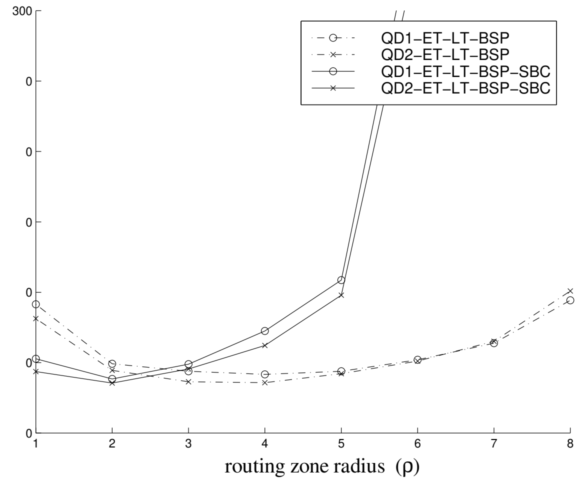


Figure 13b: Route Discovery Delay: High Load
v = 25 [m/s], R_{query} = 0.1 [query/sec]

6. SUMMARY, FUTURE WORK AND CONCLUDING REMARKS

The Zone Routing Protocol (ZRP) provides a flexible solution to the challenge of discovering and maintaining routes in a wide variety of ad-hoc network environments. The 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. By contrast, interzone route discovery is based on a reactive route request/route reply scheme.

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 mechanism that we refer to as bordercasting, queries may be passed directly to the periphery of the queried routing zone, without incurring any queueing delays at intermediate nodes. An undesirable side effect of bordercasting is the overlapping of query threads. We have introduced a suite of advanced query control mechanisms (QD1/QD2, LT, ET, BSP) that effectively combat the redundant querying, while generating no additional control traffic and requiring negligible computational overhead. Further reduction of the interzone control traffic can be achieved by preventing thread overlap locally through *selective* bordercasting (SBC). Unlike the other query control mechanisms, selective bordercasting requires additional overhead, primarily through the proactive maintenance of an extended routing zone and the non-trivial computation of peripheral node assignments.

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 (ρ=1) or purely proactive ($\rho \rightarrow \infty$) 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).

Selective bordercasting favors a more reactive ZRP configuration than full bordercasting. For single channel networks, the amount of routing traffic produced through selective bordercasting is comparable to the traffic produced through full bordercasting. For multiple channel networks, however, selective bordercasting produces about half the control traffic as full bordercasting.

A ZRP configuration that minimizes control traffic generally provides near-optimal route discovery delay. The ZRP has been shown to respond twice as fast as traditional flood-search queries under high network load, and three times faster under low network load. The improvements in route response time are even greater when we consider that a node can <u>immediately</u> provide routes for all of its routing zone nodes.

In single channel networks, selective bordercasting does not provide sufficient improvement in performance to justify its computational cost. However, selective bordercasting can substantially reduce the amount of control traffic in multiple channel networks, and may be appropriate for cases when the conservation of bandwidth justifies the processing overhead of the SBC greedy algorithm.

7. ACKNOWLEDGEMENTS

The authors thank the following organizations and individuals for their continuous support of this work:

Dr. Roger Peterson (Motorola), Dr. Michael Kotzin (Motorola), and Mr. Siamak Tabrizi (U.S. Air Force).

8. REFERENCES

[1] Bertsekas, D., and Gallager, R., *Data Networks,* Second Edition, Prentice Hall, Inc., 1992.

[2] Cheng, C., Reley, R., Kumar, S.P.R., and Garcia-Luna-Aceves, J.J., "A Loop-Free Extended Bellman-Ford Routing Protocol without Bouncing Effect," *ACM Computer Communications Review,* vol. 19, no. 4, 1989, pp. 224-236.

[3] Deng, J. and Haas, Z.J., "Dual Busy Tone Multiple Access (DBTMA): A New Medium Access Control for Packet Radio Networks," IEEE ICUPC'98, Florence, Italy, Oct. 5-9, 1998.

[4] Garcia-Luna-Aceves, J.J., "Loop-Free Routing Using Diffusing Computations," *IEEE/ACM Transactions on Networking,* vol. 1, no. 1, February 1993, pp. 130-141.

[5] Grossman, T. and Wool, A., "Computational Experience with Approximation Algorithms for the Set Covering Problem," *European J. of Operational Research*, vol. 101, pp.81-92.

[6] Haas, Z.J., "A Routing Protocol for the Reconfigurable Wireless Networks," *IEEE ICUPC'97,* San Diego, CA, October 12-16, 1997.

[7] Haas, Z.J. and Pearlman, M.R., "The Zone Routing Protocol (ZRP) for Ad-Hoc Networks," *IETF MANET*, Internet Draft, Sept. 1998.

[8] Jacquet, P., Muhlethaler, P, and Qayyum A., "Optimized Link State Routing Protocol," *IETF MANET*, Internet Draft, Nov. 1998.

[9] Johnson, D.B. , and Maltz, D.A. , "Dynamic Source Routing in Ad-Hoc Wireless Networking," in *Mobile Computing,* T. Imielinski and H. Korth, editors, Kluwer Academic Publishing, 1996.

[10] Johnson, D.J., "Approximation Algorithms for Combinatorial Problems," *J. of Computer and System Sciences*, vol. 9, pp. 256-278.

[11] Murthy, S. , and Garcia-Luna-Aceves, J.J., "A Routing Protocol for Packet Radio Networks," Proc. of ACM Mobile Computing and Networking Conference, *MOBICOM'95,* Nov. 14-15, 1995.

[12] Murthy, S., and Garcia-Luna-Aceves, J.J., "An Efficient Routing Protocol for Wireless Networks," *MONET,* vol.1, no.2, pp.183-197, October 1996.

[13] Park, V.D., and Corson, M.S. "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks," *IEEE INFOCOM '97*, Kobe, Japan, 1997.

[14] Perkins, C.E., and Bhagwat, P., "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," *ACM SIGCOMM,* vol.24, no.4, Oct. 1994, pp.234-244.

[15] Perkins, C.E. and Royer, E.M., "Ad Hoc On-Demand Distance Vector Routing," *IEEE WMCSA'99,* New Orleans, LA, Feb. 1999.

[16] Tsuchiya, P.F., "The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks," *ACM Computer Communications Review*, vol. 18, no. 4, pp.35-42,1988.

Appendix: A Greedy Algorithm for Selective Bordercasting

The problem of selecting the smallest set of inner peripheral nodes that "covers" a given set of outer peripheral nodes is an application of the NP-complete "Set Covering/Partitioning Problem". Optimal solutions to this problem can be found through techniques such as the tree searching algorithms. In the case of our route discovery problem, we are willing to settle for an approximate solution that can be calculated in polynomial time. Among the standard heuristic algorithms surveyed, the "greedy" algorithm appears to provide the best balance of processing complexity and optimality [4].

A bordercasting node begins with an assigned set of inner peripheral nodes. If the bordercasting node initiated the query, then its assigned set of inner peripheral nodes is simply the full set of inner peripheral nodes. Otherwise, the assigned set of inner peripheral nodes is provided by the previous hop of the query thread.

Each inner peripheral node, j, of node k, covers a subset of the outer peripheral nodes as indicated:

$$\text{covered\_outer\_nodes}(j) = \left\{ i \in \text{nodes} \mid \text{dist}(j,i) == \rho \wedge \text{dist}(k,i) == 2 \cdot \rho \right\}$$

where ρ is the radius of the routing zone.

The union of these sets forms the total set of outer peripheral nodes that require coverage:

$$\text{outer\_nodes} = \bigcup \text{covered\_outer\_nodes}(j)$$

We begin the greedy algorithm with an empty set of bordercast nodes. During each iteration, the "best" inner peripheral node is added to the set of bordercast nodes. The best node is defined as the inner peripheral node that covers the *most* outer peripheral nodes. After the new bordercast node is identified, it is removed from the set of inner peripheral nodes and its covered nodes are removed from the set of outer peripheral nodes.[11] What remains is a simplified set covering problem to be faced by the next iteration.[12] The process continues until all outer peripheral nodes have been covered.

---

[11] These partitioned outer peripheral nodes become the new bordercast node's set of assigned inner peripheral nodes.

[12] Since each iteration further reduces the set covering problem, the greedy algorithm also lends itself to a recursive implementation.

```
bordercast_nodes=∅
while(outer_nodes ≠∅)
{
     choose  selected_inner_node  such  that :
          |outer_covered_nodes(selected_inner_node)| == max_{i∈inner_nodes} |outer_covered_nodes(i)|
     bordercast_nodes = bordercast_nodes ∪ {selected_inner_node}
     inner_nodes = inner_nodes − {selected_inner_node}
     for j ∈inner_nodes
          covered_outer_nodes(j) = covered_outer_nodes(j) − covered_outer_nodes(selected_inner_node)
     outer_nodes = outer_nodes − covered_outer_nodes(selected_inner_node)
}
```

It can be shown [10] that the solution size of the greedy set covering algorithm is within a factor of $\ln(|outer\_nodes|) + 1$ of the optimal set covering solution.

At the beginning of each iteration, it is possible to employ techniques that may reduce the set covering problem. The first simplification is to find all outer peripheral nodes that are covered by only one inner peripheral node. In these cases, the single inner peripheral node is *essential* to provide full coverage, and is automatically selected as a bordercast node.

```
covering_inner_nodes(k) = {i ∈ inner_nodes | k∈outer_nodes ∧ dist(i, k) == ρ}
for  k ∈ outer_nodes
{
     if  |covering_inner_nodes(k)|==1
     {
          bordercast_nodes = bordercast_nodes∪ covering_inner_nodes(k)
          for j∈ inner_nodes
               covered_outer_nodes(j)=covered_outer_nodes(j) − outer_nodes(covering_inner_nodes(k))
          outer_nodes = outer_nodes − outer_nodes(covering_inner_nodes(k))
          inner_nodes = inner_nodes − covering_inner_nodes(k)
     }
}
```

It should be noted that essential nodes only appear prior to the first iteration of the greedy algorithm. After all essential nodes are selected (and their corresponding outer peripheral nodes removed), all remaining outer peripheral nodes will be covered by at least two other nodes. After each iteration, <u>one</u> inner peripheral node is selected for removal, and matched with <u>all</u> of the outer peripheral node that are covered by this inner peripheral node. Each *remaining* outer peripheral node is necessarily covered by at least two inner peripheral nodes (since it was covered by more than two inner peripheral nodes in the previous iteration, all of which are present in this iteration.) Consequently, the check for essential nodes need only be performed <u>once</u>.

Each set covering problem may also be reduced, by identifying each inner peripheral node whose coverage is a subset of another inner peripheral node's coverage. In this case, we say that the inner peripheral node with the smaller coverage is dominated by the other node. Because of the greedy nature by which each bordercast node is selected, the presence of a dominated node cannot result in a better solution.

$$
\begin{array}{l}
\text{for } \mathrm{i} \in \text{inner\_nodes} \\
\quad \text{for } \mathrm{j} \in \text{inner\_nodes}, \mathrm{j} \neq \mathrm{i} \\
\quad\quad \text{if(covered\_outer\_nodes(i)} \subset \text{covered\_outer\_nodes(j} \\
\quad\quad\quad \text{inner\_nodes} = \text{inner\_nodes} - \{\mathrm{i}\}
\end{array}
$$

While these techniques simplify the set covering problem, they also add complexity to the overall algorithm. The *essential node selection* is $O(\rho^2)$. Since it is only executed once, it does not degrade the computational complexity of the $O(\rho^2)$ greedy algorithm. The *dominated node removal*, like the *essential node selection,* is also $O(\rho^2)$. However, it is executed on every iteration of the greedy algorithm, increasing the overall computational complexity to $O(\rho^3)$. For asymptotically large covering sets, application of the *dominated node removal* would not be recommended. In terms of our selective bordercasting problem, where the covering sets are typically no greater than a few tens of nodes, these complexity measures do not necessarily provide the appropriate insight into the performance of these schemes. In practice, the best combination of these schemes can be chosen, either a priori or on-line, by directly measuring the processing time associated with the various combinations of these schemes.