The Zone Routing Protocol (ZRP) for Ad Hoc Networks
<draft-zone-routing-protocol-00.txt>

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”

To view the entire list of current Internet-Drafts, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), ftp.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this memo is unlimited.

Abstract

This document describes a routing protocol for ad hoc networks. In particular, it is suitable for highly versatile networks, characterized by large range of nodal mobilities and large network diameters. The protocol is a hybrid of a proactive and a reactive schemes, allowing adjustment of its operation to the current network operational conditions.

1. Introduction

One of the major challenges in designing a routing protocol for the ad hoc networks stems from the fact that, on one hand, to determine a packet route, at least the reachability information of the source neighbors needs to be known to the source node. On the other hand, in an ad hoc network, the network topology can change quite often. Furthermore, as the number of network nodes can be large, the potential number of destinations is also large, requiring large and frequent exchange of data (e.g., routes, routes updates, or routing tables) among the network nodes. Thus, the amount of update traffic can be quite high. This is in contradiction with the fact that all updates in a wirelessly interconnected ad hoc network travel over the air and, thus, are costly in resources.
In general, the 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. The family of Distance-Vector protocols is an example of a proactive scheme. Reactive protocols, on the other hand, invoke a route determination procedure on demand only. Thus, when a route is needed, some sort of global search procedure is employed. The family of classical flooding algorithms belong to the reactive group. Some examples of reactive (also called on-demand) ad hoc network routing protocols are [Johnson] and [TORA].

The advantage of the proactive schemes is that, once a route is needed, there is little delay until the route is determined. In reactive protocols, because route information may not be available at the time a datagram is received, the delay to determine a route can be quite significant. Furthermore, the global search procedure of the reactive protocols requires significant control traffic. Because of this long delay and excessive control traffic, pure reactive routing protocols may not be applicable to real-time communication. However, pure proactive schemes are likewise not appropriate for the ad hoc networking environment, as they continuously use a large portion of the network capacity to keep the routing information current. Since nodes in an ad hoc networks move quite fast, and as the changes may be more frequent than the route requests, most of this routing information is never even used! This results again in an excessive waste of the wireless network capacity. What is needed is a protocol that, on one hand, initiates the route-determination procedure on-demand, but at limited search cost. The presented here protocol, termed the "Zone Routing Protocol (ZRP)," is an example of a hybrid reactive/proactive routing protocol.

The ZRP, on one hand, limits the scope of the proactive procedure only to the node's local neighborhood. On the other hand, the search throughout the network, although global in nature, is done by efficiently querying selected nodes in the network, as opposed to querying all the network nodes.

A related issue is that of updates in the network topology. For a routing protocol to be efficient, changes in the network topology have to have local effect only. In other words, creation of a new link at one end of the network is an important local event but, most probably, not a significant piece of information at the other end of the network. Proactive protocols tend to distribute such topological changes widely in the network, incurring large costs. The ZRP limits propagation of such information to the neighborhood of the change only, thus limiting the cost of topological updates.
2.0 The Notion of Routing Zone, Zone Radius, and Bordercasting

A routing zone is defined for each node and includes the nodes whose minimum distance in "hops" from the node in question is at most some predefined number, which is referred to as the Zone Radius. An example of a routing zone (for node A) of radius 2 is shown below.

\[
\begin{array}{c}
\text{Zone of node A} \\
of \text{radius}=2
\end{array}
\]

Note that in this example nodes B through E are within the routing zone of A. Node G is outside A’s routing zone. Also note that E can be reached by two path from A, one with length 2 hops and one with length 3 hope. Since the minimum is less or equal than 2, E is within A’s routing zone.

Peripheral nodes are nodes whose minimum distance to the node in question is equal exactly to the zone radius. Thus, in the above figure, nodes D, F, and E are A’s peripheral nodes.

Bordercasting is a process of sending an IP datagram ([RFC-0791]) by a node to all its peripheral nodes. Bordercasting can be implemented either through regular IP unicast or through IP multicast (Distance Vector Multicast Routing Protocol [RFC-1075]). Of course, the multicasting approach is preferred to reduce the amount of traffic over the air.

2.1 The Zone Routing Protocol (ZRP) ([Haas-1],[Haas-2])

The ZRP is based on two procedures: the IntrAzone Routing Protocol (IARP) and the IntErzone Routing Protocol (IERP). Through the use of the IARP, each node learns the identity of and the (minimal) distance to all the nodes in its routing zone. The actual IARP is not specified and can include any number of protocols, such as the derivatives of Distance Vector Protocol (e.g., Ad Hoc On-Demand Distance Vector [AODV], Shortest Path First (e.g., OSPF [RFC-2178]), [Murthy]). In fact, different portions of an ad hoc network may choose to operate based on different choice of the IARP protocol. Whatever the choice of IARP is, the protocol needs to be modify to ensure that the scope of this operation is restricted to the zone of the node in question.

Note that as each node needs to learn the distances to the nodes within its zone only, the nodes are updated about topological
changes only within their routing zone. Consequently, in spite of
the fact that a network can be quite large, the updates are only
locally propagated.

2.2 The Interzone Routing Protocol (IERP)

While IARP finds routes within a zone, IERP is responsible for
finding routes between nodes located at distances larger than the
zone radius. IERP relies on bordercasting. Bordercasting is possible
as any node knows the identity and the distance to all the nodes in
its routing zone by the virtue of the IARP protocol.

The IERP operates as follows: The source node first checks whether
the destination is within its routing zone. (Again, this is possible
as every node knows the content of its zone). If so, the path to the
destination is known and no further route discovery processing is
required. If, on the other hand, the destination is not within the
source's routing zone, the source bordercasts a route request
(referred to here as a "request") to all its peripheral nodes. Now,
in turn, all the peripheral nodes execute the same algorithm: check
whether the destination is within their zone. If so, a route reply
(referred to here as a "reply") is sent back to the source
indicating the route to the destination. If not, the peripheral node
forwards the query to its peripheral nodes, which, in turn, execute
the same procedure. An example of this Route Discovery procedure is
demonstrated in the figure below. As we be shown, Thus, a route
within a network is specified as a sequence of nodes, separated by
approximately the zone radius.

```
+---+    | F |
|---+    | D |
|---+    | E |
|---+    | H |
|---+    | B |
|---+    | A |
|---+    | G |
|---+    | I |
|---+    | J |
|---+    | C |
|---+    | K |
|---+    | L |
```

The node A has a datagram to node L. Assume routing zone radius of
2. Since L is not in A’s routing zone (which includes B,C,D,E,F,G),
A bordercast a routing request to its peripheral nodes: D,F,E, and
G. Each one of these peripheral nodes check whether L exists in their
routing zones. Since L is not found in any routing zones of these
nodes, the nodes bordercast the request to their peripheral nodes. In particular, G bordercasts to K, which realizes that L is in its routing zone and returns the requested route (L-K-G-A) to the query source, namely A.

2.3 Route Accumulation procedure

The process by which the node receiving a query knows the path back to the source of the query is the Route Accumulation procedure. In particular, the Route Accumulation procedure is used to return the route to the source of the query by the "last peripheral node" in which routing zone the destination resides. The Route Accumulation procedure is based on each node that forwards a query writing into the query packet its identification. The sequence of these identifications represents a path from the source node to the current node, and, by reversing the order, a path from the current node to the source node.

2.4 Terminating the IERP flood

It is desirable for route requests to propagate away from the source and not to loop-back into a previously queried routing zone. To encourage this directed spread of route requests, IERP employs two loop-back prevention mechanisms. The first mechanism terminates threads which loop-back on themselves. Such threads are terminated when the accumulated route (excluding the previous hop) contains a host which lies within its routing zone. A separate mechanism, based on packet eavesdropping, is employed to reduce the overlapping of parallel threads. When a host receives a route request, the IERP records the request ID in its Processed Request List.* If a node "officially" receives a request that has been previously recorded, the new copy is discarded. Without these measures, the bordercast transmission can actually generate more overhead than flooding.

* ICMP provides a service similar to IERP’s route failure notification. However, the IERP service provides additional diagnostic information, allowing the source to respond to the route failure more effectively.

2.5 Route failures notifications

The IERP also provides a mechanism to reactively respond to route failures. A route failure is detected by the IP when the next hop in a source route is determined to be unreachable (i.e., does not appear in the Intrazonal Routing Table). Upon detection of a route failure, the IERP is alerted, and a route failure packet is generated.** The route failure packet propagates back to the route’s source in the same manner as a route reply. When the route’s source receives notification of the route failure, the expired route is removed from its Interzone Routing Table. The IERP may also be configured to locally repair the damaged Interzone route by...
initiating a route discovery to the unreachable next hop.

** Eavesdropping nodes are only permitted to listen to route requests (and record them in their Processed Request List); they are prohibited from processing the request any further.

### 2.6 Bordercast Resolution Protocol (BRP)

The Bordercast Resolution Protocol (BRP) is included with the IERP in order to provide bordercasting services which do not exist in IP. In the current version of the BRP, the content of a bordercast message is considered to be accessible to any host which receives the message.*** Future versions of the BRP may allow for semi-private bordercasting, where bordercast messages are only delivered to the higher layers of the bordercast destinations (peripheral nodes).

*** When the BRP delivers the message to the next higher layer, a flag is set to indicate whether the packet was overheard or received at its intended destination. Note that this does not restrict access to the message; it only serves to provide the access control status of the message.

### 3.0 The ZRP Architecture

```
+---------+    :    +---------+         +---------+    :    +---------+
|   NDM   |~~~~:~~~>|  IARP   |~~~~~~~~>|  IERP   |<~~~:~~~~|  ICMP   |
+---------+    :    +---------+         +---------+    :    +---------+
```

Legend:

- **A<----B** exchange of packets between protocols A & B
- **A->->B** information passed from protocol A to protocol B

**Existing Protocols**

```
---
IP     Internet Protocol
ICMP   Internet Control Message Protocol
---
```
ZRP Entities
------------
IARP        IntrAzone Routing Protocol
IERP        IntErzone Routing Protocol
BRP         Bordercast Resolution Protocol

Additional Protocols
---------------------
NDM          Neighbor Discovery/Maintenance Protocol

Note, it is assumed that the neighbor discovery operation can be implemented by the MAC/link-layer protocols. Thus the NDM protocol remains unspecified here.

4.0 Implementation Details

4.1 The IntrAzone Routing Protocol (IARP)

Although the IARP can be implemented through various proactive protocols, we present here an example of an implementation based on a modified version of the Distance Vector algorithm that restricts the of the algorithm’s operation to the range of the routing zone radius.

A terminal may receive new route information either from a received IARP packet or from an interrupt generated by its Neighbor Discovery/Maintenance (NDM) process\$. In the special case when a host has discovered a neighbor, the IARP is responsible for sending the new neighbor the shortest route to each host which is common to both of their routing zones. The terminal then records the new route information in its Intrazone Routing Table. If the shortest path to the host has changed, the terminal broadcasts an IARP packet reflecting the change.

\$ IARP relies on the services of a separate protocol (referred to here as the Neighbor Discovery/Maintenance Protocol) to provide current information about a host’s neighbors. At the least, this information must include the IP addresses of all the neighbors. The IARP can be readily configured to support supplemental neighbor information, such as link cost.

A. Packet Format

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Destination Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Next Hop Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

  1                   2                   3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Hop Cnt |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Haas, Pearlman           Expires May 1998

[Page 7]
Field Description:
- **Destination Address** (32 bits)
  IP address of the destination host
- **Next Hop Address** (32 bits)
  IP address of the "next hop" host to the destination host
- **Hop Count** (4 bits)
  Length of the route to the destination host, measured in hops

B. Structures

B.1 Intrazone Routing Table

<table>
<thead>
<tr>
<th>Host</th>
<th>0</th>
<th>1</th>
<th>==&gt;</th>
<th>M-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Next : Hop</td>
<td>Next : Hop</td>
<td>==&gt;</td>
<td>Next : Hop</td>
<td></td>
</tr>
<tr>
<td>Hop : Count</td>
<td>Hop : Count</td>
<td>Hop : Count</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

C. Interfaces

C.1 Higher Layer (N/A)
C.2 Lower Layer (IP)
  C.2.1 Send() (specified in IP standard)
  C.2.2 Deliver() (specified in IP standard)
C.3 NDM
  C.3.1 Neighbor_Lost(host)
    Used by the NDM to notify the IARP that direct contact has been lost with "host".
  C.3.2 Neighbor_Found(host)
    Used by the NDM to notify the IARP that direct contact has been confirmed with "host".
C.4 IERP
  C.4.1 Host_Lost(host)
    Used by IARP to notify the IERP that a host no longer exists within the routing zone.

D. State Machine

The IARP protocol consists of only one state (IDLE). Therefore, no state transitions need to be specified. The IARP immediately acts upon an event and then returns back to the IDLE state.
Notes:  1) X is used as a label for the host running this state machine.
       2) INF is a reserved field value corresponding to "infinity".

D.1  Event: An IARP packet is received containing route information to a destination D. The hop count associated with the received route is LESS THAN the routing zone radius.

       Action: The received route is recorded in the Intrazone Routing Table. If the received route is shorter than the previous shortest route to D, then a new IARP packet containing route information to D through X is broadcasted.

D.2  Event: An IARP packet is received containing route information to a destination D. The hop count is EQUAL TO the routing zone radius.

       Action: The received route is recorded in the Intrazone Routing Table.

D.3  Event: An IARP packet is received containing route information to a destination D. The hop count is equal to INF.

       Action: The route to D is removed from the Intrazone Routing Table.
               1) If the Intrazone Routing Table still contains a route to D and the length of the shortest route has increased due to the route removal, then an IARP packet containing the shortest route to D through X is broadcasted.
               2) If the Intrazone Routing Table contains no more routes to D, then an IARP packet containing a route to D through X with hop count of INF is broadcast. A "Host Lost" interrupt is generated to alert the IERP that D has moved beyond the routing zone.

D.4  Event: A "Neighbor Found" interrupt is received, indicating the discovery of a neighbor host N.

       Action: For each destination in X's Intrazone Routing Table, an IARP packet is sent to N containing the best route to that destination. An IARP packet is then broadcasted containing the 1 hop route to N through X.
Event: A "Neighbor Lost" interrupt is received, indicating that host N is no longer a neighbor of X

Action: The one hop route to N is removed from the Intrazone Routing Table.
1) If the Intrazone Routing Table still contains a route to N and the length of the shortest route has increased due to the route removal, then the an IARP packet containing the shortest route to N through X is broadcasted.
2) If the Intrazone Routing Table contains no more routes to N, then an IARP packet containing a route to D through X with hop count of INF is broadcast. A "Host Lost" interrupt is generated to alert the IERP that D has moved beyond the routing zone.

E. Pseudocode Implementation

E.1 Update Intrazone Routing Table

if (packet arrived)
  (host, route->next_hop,route->hop_count) <-- packet
else
  {host} <-- intrpt
  route->next_hop=host
  if (type(intrpt) == "Neighbor Found")
  {
    for recorded_host = each host in Intrazone_Routing_Table
    {
      best_route = Intrazone_Routing_Table[recorded_host,0]
      if (best_route->hop_count < ROUTING_ZONE_RADIUS)
      {
        packet <-- {recorded_host,my_id,hop_count+1}
        send(packet,host)
      }
      route->hop_count=1
    }
  }
  else
  {
    route->hop_count=INF
  }
former_best_route = Intrazone_Routing_Table[host,0]
if (route->hop_count < INF)
    add(Intrazone_Routing_Table[host], route)
else
    remove(Intrazone_Routing_Table[host], route)
best_route = Intrazone_Routing_Table[host,0]
if (best_route != NULL)
    if (best_route->hop_count != former_best_route->hop_count
        && best_route->hop_count < ROUTING_ZONE_RADIUS)
        {  
            packet <-- {host, my_id, best_route->hop_count+1}
            broadcast(packet)
        }
    else
        {
            force_intrpt("IERP","Host Lost", {host})
            packet <-- {host, my_id, INF}
            broadcast(packet)
        }

4.2 IntErzone Routing Protocol (IERP)

The Interzone Routing Protocol (IERP) is responsible for discovering routes to hosts which are beyond a terminal’s routing zone. The IERP collects routing information reactively, through bordercasted queries which contain the accumulated routes from the source.

When IP receives a data packet intended for an unknown destination (i.e., destination is not recorded in either the Intrazone or Interzone Routing Tables), the IERP is interrupted. The IERP responds by initiating a route discovery and bordercasting a route query. Each query in the network is uniquely identified by the IP address of the source and a request ID (local to the source).

Upon receipt of a route request packet, a host searches its Intrazone Routing Table to determine if the requested destination is within its routing zone. If so, the terminal responds with a route reply which is returned to the source along the (reversed) accumulated route. If the destination is not within the terminal’s routing zone, the terminal adds its terminal ID to the accumulated route and bordercasts the route request.

A. Packet Format
Field Description:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>(4 bits) Identifies the type of IERP packet. The current version of IERP contains three packet types: REQUEST: request an Interzone source route to the destination host specified in Destination Address. REPLY: response to a request packet; includes the source route to the destination host specified in Destination Address. FAILURE: control message sent to a host indicating that the received data packet contains an invalid source route.</td>
</tr>
<tr>
<td>Request ID</td>
<td>(10 bits) Sequence number which, along with the Source Address (see below) uniquely identifies any route query in the network. (used only for REQUEST packets)</td>
</tr>
<tr>
<td>Last Hop</td>
<td>(6 bits) Indicates the previous recipient of the IERP packet (referenced as an index into the Source Route (see below))</td>
</tr>
<tr>
<td>Hop Marker</td>
<td>(6 bits) Currently used to indicate the hop where a route failure was detected (referenced as an index into the Source Route (see below)) (used only for FAILURE packets)</td>
</tr>
</tbody>
</table>
* Max Hops (6 bits)
   Maximum number of Interzone hops which a route query may
   propagate. This field allows nodes to adaptively
   configure the depth of a route search in order to
   control the amount of IERP traffic generated. (used only
   for REQUEST packets)

* Destination Address (32 bits)
   IP address of the destination host

* Source Route (N*32 bits)
   Variable length field which contains the IP addresses of
   an N hop source route. The first subfield of the Source
   Route contains the IP address of the route’s source.
   In general, the n-th subfield contains the IP address of
   the n-th hop in the route.
   For REQUEST packets, the Source Route represents the
   incomplete accumulated route to the destination host
   (as indicated by the Destination Address)
   For REPLY and FAILURE packets, the Source Route contains
   the complete route from the source host to the
   destination host.

B. Structures

B.1 Intrazone Routing Table (See IARP Description)

B.2 Interzone Routing Table

```
+-------+-----------+-----------+-----+-----------+-------+
|       | Routes     |           |     |           |       |
| + Dest| +-----------+-----------+-----|-----------| + Dest |
|       |           |           |     |           |       |
| 0     |           |           |     |           | 1     |
|       |           |           |     |           |       |
| +-----|-----------|-----------|-----|-----------| +-----|
|       |           |           |     |           |       |
| ==>   |           |           |     |           | ==>   |
|       |           |           |     |           |       |
|       |           |           |     |           |       |
+-------+-----------+-----------+-----+-----------+-------+
```

source

source

first

(N-1)

host

hop

hop

Haas, Pearlman
Expires May 1998

[Page 13]
B.3 Processed Request List

+-------------------------+
| Source | Request ID |
+-------------------------+

C. Interfaces

C.1 Higher Layer (N/A)

C.2 Lower Layer (BRP)

C.2.1 Send()   (same interface as IP send())
   Used by the IERP to request transmission of a data packet.

C.2.2 Deliver() (same interface as IP deliver())
   Used by the BRP to alert the IERP of the arrival of a data packet.

C.3 IARP

C.3.1 Neighbor_Lost(host)
   Used by the NDM to notify the IARP that direct contact has been lost with "host".

C.3.2 Neighbor_Found(host)
   Used by the NDM to notify the IARP that direct contact has been confirmed with "host".

C.4 ICMP

C.4.1 Host_Unreachable()   (specified in ICMP standard)

C.4.2 Source_Route_Failed() (specified in ICMP standard)

D. State Machine

The IERP protocol consists of only one state (IDLE). Therefore, no state transitions need to be specified. The IERP immediately acts upon an event and then returns back to the IDLE state.

Note: 1) X is used as a label for the host running this state machine.

D.1

Event: A "Host Lost" interrupt is received, indicating that the host H has moved beyond the routing zone.

Action: Remove every route from the Interzone Routing Table which contains H. A limited depth route discovery to H is initiated. An IERP request packet is created with the destination address set to H and the source route initialized with the IP address of X. The request packet is then bordercasted.
D.2  
Event:   A "Host Unreachable" interrupt is received from the ICMP, indicating an unknown destination host D.

Action:  A full depth route discovery to the lost host is initiated. An IERP request packet is created with the destination address set to H and the source route initialized with the IP address of X. The request packet is then bordercasted.

D.3  
Event:   An "Source Route Failed" interrupt is received from the ICMP, indicating that the next hop specified in an IP source route does not appear within the routing zone.

Action:  A route failure packet containing the invalid route is sent to the route’s source with the hop marker indicating X as the host where a route failure was detected.

D.4  
Event:   An IERP request packet is received with a destination that appears within X’s routing zone.

Action:  The request is recorded in the Processed Request List. In order to be processed further, four conditions must be satisfied. First, the received packet must not be flagged as overheard. Second, the number of hops in the route must not exceed the maximum hop count. Third, the request must not have been previously recorded. Finally, no hops in the route (except the last_hop) may lie within X’s routing zone. X appends its IP address to the route and sends an IERP reply packet to the preceding hop in the route.

D.5  
Event:   An IERP request packet is received with a destination that DOES NOT appear within X’s routing zone.

Action:  The request is recorded in the Processed Request List. In order to be processed further, four conditions must be satisfied. First, the received packet must not be flagged as overheard. Second, the number of hops in the route must not exceed the maximum hop count. Third, the request must not have been previously recorded. Finally, no hops in the route (except the last_hop) may lie within X’s routing zone. X appends its IP address to the route and is bordercasts an IERP request packet containing the updated route.
D.6
Event: An IERP reply packet is received containing X as the source host.
Action: The received source route is recorded in Interzone Routing Table.

D.7
Event: An IERP reply packet is received containing a node other than X as the source host.
Action: The route reply packet is forwarded to the preceding hop in the route

D.8
Event: An IERP failure packet is received containing X as the source host.
Action: X removes all routes from its Interzone Routing table which contain the broken link specified by the bad hop field.

D.9
Event: An IERP failure packet is received containing a node other than X as the source host.
Action: X forwards the route reply packet to the preceding hop in the route

E. Pseudocode Implementation

E.1 Intrazone Node Lost

{lost_host} <-- intrpt
for host = each host in Interzone Routing Table
{
  m=0
  while (Interzone_Routing_Table[host,m] != NULL)
  {
    route=Interzone_Routing_Table[host,m]
    if (lost_host (EXISTS IN) route)
      remove(Interzone_Routing_Table[host,m])
    else
      m++
  }
}
force_intrpt("IERP","repair",{lost_host})
E.2 Initiate Route Discovery

{dest} <-- intrpt
req_id++
route(0)=my_id
last_hop=0
if (type(intrpt) == "repair")
    max_hops=MAX_REPAIR_HOPS
else
    max_hops=MAX_REQUEST_HOPS
packet <-- {REQUEST, req_id, last_hop, NULL, max_hops, dest, route}
bordercast(packet)
add (Processed_Request_List, {my_id, req_id})

E.3 Report Route Failure

{route,dest} <-- intrpt
last_hop=0
while (route(last_hop) != my_id)
    last_hop++
packet <-- {FAILURE, NULL, last_hop, last_hop, NULL, dest, route}
send(packet, route(last_hop-1))

E.4 Receive IERP Packet

{pk_type, req_id, last_hop, bad_hop, max_hops, route} <-- packet
{overheard} <-- intrpt
switch(pk_type)
{
case: REQUEST
    add({Processed_Request_List, source, req_id})
    LSP1_terminate = FALSE
    n=0
    while (!LSP1_terminate && n < last_hop)
        {if (Intrazone_Routing_Table[route(n)]!=NULL)
            LSP1_terminate = TRUE
            n++
        }
    LSP2_terminate = (Processed_Request_List[source,req_id]
        != NULL)
    if (!overheard && !LSP1_terminate && !LSP2_terminate &&
        last_hop < max_hops)
        {last_hop++
            route(last_hop)=my_id
            if (dest (EXISTS IN) Intrazone_Routing_Table)
                {packet<--{REPLY,req_id,last Hop,bad_hop,max_hops, dest,route}
send(packet, route(last_hop-1))
}
else
{
    packet <-- {REQUEST, req_id, last_hop, bad_hop, max_hops,
                dest, route}
    bordercast(packet)
}
break

case:   REPLY
case:   FAILURE
if (route(0) == my_id)
{
    if (pk_type == ROUTE_REPLY)
        add(Interzone_Routing_Table, route)
    else
    {
        link(0) = route(bad_hop)
        link(1) = route(bad_hop+1)
        remove(Interzone_Routing_Table, link)
    }
}
else
{
    last_hop --
    packet <-- {pk_type, req_id, last_hop, bad_hop, max_hops,
                dest, route}
    send(packet, route(last_hop-1))
}
}

4.3 Bordercast Resolution Protocol (BRP)

The higher layer interface of the BRP is designed to be compatible with any IP based application. However, it is assumed that the routing zone hierarchy is visible only to the ZRP entities, making bordercasting services only of use to the IERP.

Upon receipt of a (IERP) packet to be bordercasted, the BRP resolves the bordercast address into the individual IP addresses of the peripheral nodes. The received packet is then encapsulated into a BRP packet and sent to each peripheral node (via IP broadcast transmission).

When a BRP packet is delivered from IP, the (IERP) data is decapsulated and passed on to the higher layer. If the BRP packet has not reached its destination, the BRP is responsible for forwarding the packet to the next hop toward its destination.
A. Packet Format

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Destination Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Next Hop Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              Data                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Field Description:

* Destination Address (32 bits)
  IP address of the destination host

* Next Hop Address (32 bits)
  IP address of the "next hop" host to the destination host

* Data (variable length)
  Encapsulated data

B. Structures

B.1 Intrazone Routing Table (see IARP description)

C. Interfaces

C.1 Higher Layer (ie. IERP)
  C.2.1 Send() (same interface as IP Send() primitive)
    Used by higher layer protocol to request transmission of a data packet
  C.2.2 Deliver() (same interface as IP Deliver() primitive)
    Used by the BRP to alert the higher layer protocol of the arrival of a data packet

C.2 Lower Layer (IP)
  C.2.1 Send() (specified in IP standard)
  C.2.2 Deliver() (specified in IP standard)

D. State Machine

The BRP protocol consists of only one state (IDLE). Therefore, no state transitions need to be specified. The BRP immediately acts upon an event and then returns back to the IDLE state.

Notes: 1) X is used as a label for the host running this state machine.
2) NULL is a reserved field value, corresponding to an invalid IP address.
D.1
Event: A packet is received from the IERP, destined for the BORDERCAST address.

Action: The IERP packet is encapsulated in a BRP packet. A copy of the BRP packet is made for each peripheral host, P. (i.e., each host in X’s Intrazone Routing Table whose shortest route is ROUTING_ZONE_RADIUS hops away from X). The packet’s destination address is set to the IP address of P and the next hop address is set to the IP address of the next hop to P (from X). Each packet is then broadcasted.

D.2
Event: A packet is received from the IERP, destined for a non-BORDERCAST address.

Action: The IERP packet is encapsulated in a BRP packet. The BRP packet's destination address and next hop address fields are cleared and the BRP packet is sent to the specified destination.

D.3
Event: A BRP packet is received from the IP layer. The BRP packet contains a next hop address EQUAL TO NULL.

Action: The data is decapsulated from the BRP packet and delivered to the IERP, with the overhead flag set to FALSE.

D.4
Event: A BRP packet is received from the IP layer. The BRP packet contains a valid next hop address and a destination address EQUAL TO X.

Action: The data is decapsulated from the BRP packet and delivered to the IERP, with the overhead flag set to FALSE.

D.5
Event: A BRP packet is received from the IP layer. The BRP packet contains a valid next hop address EQUAL TO X and a destination address NOT EQUAL TO X.

Action: The data is decapsulated from the BRP packet and delivered to the IERP, with the overhead flag set to TRUE. The received BRP packet is copied, the next hop address field is updated by querying the Intrazone Routing Table, and the new packet is broadcasted.
D.6

Event: A BRP packet is received from the IP layer. The BRP packet contains a valid next hop address NOT EQUAL TO X and a destination address NOT EQUAL TO X.

Action: The data is decapsulated from the BRP packet and delivered to the IERP, with the overhead flag set to TRUE.

E. Pseudocode Implementation

E.1 Receive Data Packet from IERP

{dest} <-- intrpt
if (dest == BORDERCAST)
{
    for (host = each host in Intrazone_Routing_Table)
    {
        if (Intrazone_Routing_Table[host,0]->hop_count == ROUTING_ZONE_RADIUS)
        {
            next_hop=Intrazone_Routing_Table[host,0]->next_hop
            packet<--(host,next_hop,data_packet)
            broadcast(packet)
        }
    }
} else
{
    packet<--(dest,NULL,data_packet)
    send(packet,dest)
}

E.2 Receive Packet from IP

{dest,next_hop,data}<--packet
overheard=FALSE
if (next_hop != NULL)
{
    if (next_hop == my_id)
    {
        if(dest != my_id)
        {
            overheard=TRUE
            next_hop = Intrazone_Routing_Table[dest,0]->next_hop
            packet<--(dest,next_hop,data)
            broadcast(packet)
        }
    }
} else
    overheard=TRUE
5.0 Other Considerations

5.1 Sizing the Zone Radius

The value of the zone radius determines the performance of the ZRP protocol. In general, highly mobile networks should set the zone radius to a smaller value, while in more stationary networks the zone radius should be larger. Similarly, in very active networks (frequent query requests), the zone radius should be larger, and in low-activity networks, the zone radius should be smaller.

We believe that setting the size of the zone radius should be performed by the administration of the network, if one exists, or by the manufacturer, as a default value. Although some guidance can be given as to "optimal" value of the zone radius [Haas-3], additional constrains and operational conditions may affect this choice.

6.0 References


Postel, J., Editor, "Internet Protocol", STD 5, RFC 791, September 1981


Authors’ Information

Zygmunt J. Haas
Wireless Networks Laboratory
323 Frank Rhodes Hall
School of Electrical Engineering
Cornell University
Ithaca, NY 14853
United States of America
tel: (607) 255-3454, fax: (607) 255-9072
Email: haas@acm.org or haas@ee.cornell.edu

Marc R. Pearlman
319 Frank Rhodes Hall
School of Electrical Engineering
Cornell University
Ithaca, NY 14853
United States of America
Email: pearlman@ee.cornell.edu

The MANET Working Group contacted through its chairs:

M. Scott Corson
Institute for Systems Research
University of Maryland
College Park, MD 20742
(301) 405-6630
corson@isr.umd.edu

Joseph Macker
Information Technology Division
Naval Research Laboratory
Washington, DC 20375
(202) 767-2001
macker@itd.nrl.navy.mil