Mobile Ad Hoc Networking Working Group George Aggelou
INTERNET DRAFT University of Surrey, UK
13 September 1999 Rahim Tafazolli
University of Surrey, UK

Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) Protocol
draft-ietf-manet-rdmar-00.txt

Status of This Memo

Abstract

This document describes the Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) protocol for use in mobile ad hoc networks (MANETs). The protocol is highly adaptive, bandwidth-efficient and scaleable. A key concept in its design is that protocol reaction to link failures is typically localised to a very small region of the network near the change. This desirable behaviour is achieved through the use of a novel mechanism for route discovery, called Relative Distance Micro-discovery (RDM). The concept behind RDM is that a query flood can be localised by knowing the relative distance (RD) between two terminals. To accomplish this, every time a route search between the two terminals is triggered, an iterative algorithm calculates an estimate of their RD, given an average nodal mobility and information about the elapsed time since

they last communicated and their previous RD. Based on the newly calculated RD, the query flood is then localised to a limited region of the network centred at the source node of the route discovery and with maximum propagation radius that equals to the estimated relative distance. This ability to localise query flooding into a limited area of the network serves to increase scalability and minimise routing overhead and overall network congestion.

Contents

1. Introduction

The Relative Distance Micro-discovery Ad hoc Routing (RDMAR) protocol is designed for operation in mobile ad hoc networks [MANET]. In mobile ad hoc networks (MANETs), nodes are free to move around randomly and organise themselves arbitrarily; thus, the network's wireless topology may change rapidly and unpredictably. In many packet-radio networks the packet radios do not have direct radio links to all other packet radios in the network and thus store-and- forward routing of the packets is required.
Therefore, nodes are acting also as routers (also called Mobile Routers) and dynamically establishing routing patterns among themselves to form an infrastructure-less network.

RDMAR is a source-initiated on-demand routing protocol and allows nodes to maintain routes to destinations that are in active communication. One distinguishing feature of RDMAR is its use of an optimised route discovery mechanism, called Relative Distance Micro-discovery (RDM). According to this mechanism, the routing protocol limits the range of route searching in order to save the cost of flooding a route request message into the entire wireless area. Thus, in contrast to pure flooding mechanism where a route query would reach every node that is reachable in the wireless network, in RDMAR a query is propagated only to a limited region of the network for the successful discovery of the destination terminal.

This is achieved by estimating the relative distance between the source and destination of the route search, thus restricting the range of route discovery within an area centred at the source node of the route discovery and with maximum radius that equals to the estimated relative distance. Another feature of RDMAR is that the maintenance of active paths (i.e., paths that carry active calls) is a distributed operation that exploits the spatial relationship of nodes when a failure along an active route occurs. Depending on the relative distance of the node that reports the failure from the calling and called nodes, two heuristics are considered:
a) if its relative distance from the called node is smaller or equal to this from the calling node, then RDM is to be applied to localise the repair of the failed route on the region of the network where the failure occurs; otherwise,
b) the node proceeds and informs the calling node about the failure to deliver the call through this path.

RDMAR offers a number of potential advantages over other routing protocols for mobile ad hoc networks. First, RDMAR uses no periodic beaconing to keep routing tables updated thus significantly reducing network bandwidth overhead, conserving battery power and reducing the probability of packet collision. In addition, in RDMAR nodes do not make use of their routing caches to reply to route queries. Using other nodes' cashes results to a storm of route replies and repetitive updates in hosts' caches, yet early query quenching can not stop the propagation of all query messages which are flooded all over the network. Furthermore, RDMAR does not rely on any specific location aided technology in order to compute routing patterns and to limit the query flood to a restriction region.
Finally, in the presence of asymmetrical links traditional link-state or distance vector protocols may compute routes that do not work. Several factors such as interference, shadowing, differing radio or

antenna capabilities, may turn a link to function asymmetrically. RDMAR, however, has been designed to compute correct routes even in the presence of asymmetric links.

2. RDMAR Terminology

Node

A device in the ad hoc network willing to participate in the routing protocol.

link

A communication facility or medium over which nodes can communicate at the link layer, such as an Ethernet (simple or bridged). A link is the layer immediately below IP.

packet

An IP header plus payload.

active route

A routing table entry with an unexpired Lifetime and a finite metric. A routing table may contain entries that are not active. Only active entries can be used to forward data packets.

Route Discovery

The mechanism in RDMAR where a node S discovers a route to some node D when one is needed.

Route Maintenance

The mechanism in RDMAR whereby a node is able to detect, while using a route, if the network topology has changed such that it can no longer use this route.

2.1 Specification Language

This protocol specification uses conventional capitalised keywords such as "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" which are to be interpreted as described in RFC 2119 [RFC].

3. Protocol Overview

In RDMAR, calls are routed between the stations of the network by using routing tables which are stored at each station of the network; each node is treated as a host as well as a store-and-forward node. Each routing table lists all available destinations, and the number of hops to each. Therefore, the routing table of each node is a column vector of maximum (N - 1) row entries, where N is the set of participating nodes in the network. Apart from the available destination addresses, additional information is maintained for each destination address D. This includes: the "Default Router" field that indicates the next hop node through which the current node can reach D, the

"RD" field which shows an estimate of the relative distance (RD) (in hops) between the node and D, the "Time_Last_Update" (TLU) field that indicates the time elapsed since the node last received routing information for D, a "RT_Timeout" field which records the remaining amount of time before the route is considered invalid, and a "Route Flag" field which declares whether the route to D is active.

Each mobile node maintains two data structures, in addition to the routing table, wherein all routing and management information, learned during protocol operation, is kept; namely these are: the Data Re-transmission Table and the Route Request Table.

When a source node S wishes to send a data packet to some destination D, it first examines if it has a route to this destination. If so, it keeps a copy of the packet in its Data Re-transmission table and then proceeds and transmits the packet over its network interface to the next hop identified in its routing table. The Data Re-transmission Buffer, therefore, is a queue of data packets that are awaiting the receipt of an explicit acknowledgement from their destination. Each intermediate node (IN) upon reception of a data packet, first acknowledges (link-level acknowledgement) its correct reception to the previous hop, and forwards it to the next hop, if a path to destination is available. If an IN is unable to forward the packet, it starts the Route Maintenance Phase, as described in section 3.3

On the other hand, if the source, S, of the data packet does not have a route to destination (either because S did not have previous information for the destination node, D, or because S had a valid route for D but the lifetime associated with this route expired and hence erased from its Routing Table), the node buffers the packet and attempts to discover one using the Route Discovery procedure, as described in section 3.2. All information related to the most recent Route Discovery, is stored in the Route Request Table. Finally, the node transmits the original packet once the route is learned from route discovery.

3.1 Packet Formats

Route Requests (RREQs), Route Replies (RREPs), and FN (Failure Notification) are the three message types defined by RDMAR. These messages carry all the control information needed for the correct operation of RDMAR.

3.1.1 Route Request (RREQ) Message Format

This packet is used during the route discovery phase. It is a fixed length packet.   The various fields contained in this packet are:

```
    0                   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 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type      |E|         Reserved             |      RD      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          RREQ_ID                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Destination address                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source IP address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Source Sequence Number                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Route Request message is illustrated above, and contains the following fields:

Type   Indicates which control packet is sent.

E   Emergency flag; set to '1' when a microdiscovery procedure (RDM) is triggered as a result of a failure of an intermediate node to forward a call, according to the rules described in Section 3.3. Otherwise, the flag is set to '0'.

Reserved   Sent as 0; ignored on reception.

Relative Distance (RD)

The number of hops from the Source IP Address to the node handling the request.

RREQ_ID

A sequence number uniquely identifying the particular RREQ when taken in conjunction with the source and destination node's IP address.

Destination Address

The address of the destination for which a route is desired.

Source IP Address

The IP address of the node which originated the Route Request.

Source Sequence Number

The current sequence number to be used for route entries pointing to (and generated by) the source of the route request.

3.1.2 Route Reply (RREP) Message Format

This packet is used during the route discovery phase and is sent by the destination node of a RREQ in response to the RREQ control packet. It is a fixed length packet. The various fields contained in this packet are:

```
    0                   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 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     Type        |E|             Reserved        |     RD      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          RREP_ID                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Destination address                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Source IP address                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Lifetime                             |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Route Reply message is illustrated above, and contains the following fields:

Type Indicates which control packet is sent.

E Emergency flag; set when the RREP is a response of a RREQ that had the 'E' bit set.

Reserved Sent as 0; ignored on reception.

Relative Distance (RD)

The number of hops from the receiver of the packet to the Source IP Address.

RREP_ID

A sequence number uniquely identifying the particular RREP when taken in conjunction with the source and destination node's IP address.

Destination IP Address

The IP address of the node that requested the a route.

Source IP Address

The IP address of the node for which a route is desired.

Lifetime

The time for which nodes receiving the RREP consider the route to be valid.

3.1.3 Failure Notification (FN) Message

This Failure Notification (FN) packet is invoked during the Route Maintenance phase and is sent by an immediate neighboring node that has detected a link or nodal failure along an active path. Its packet length is fixed and it has the following format:

```
     0                   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 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      Type      |             Reserved                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Source  Address                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Destination  Address                        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Dependent_List.Node                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The format of the Failure Notification message is illustrated above, and contains the following fields:

Type   Indicates which control packet is sent.

Reserved   Sent as 0; ignored on reception.

Source Address

                The IP address of the destination of the data packet.

Destination Address

                The IP address of the source of the data packet.

Dependent_List.Node

                The IP address of the node (neighbor) listed in the Dependent_List (see section 3.3).

3.2 Route Discovery

3.2.1 Generating and Forwarding Route Requests

When an incoming call arrives at node S for destination node D and there is no route available to route the packet to D, S initiates a route discovery phase. Here, S has two options; either to flood the network with a route query in which case the route query packets are broadcast into the whole network, or instead, to limit the discovery in a smaller region of the network, if some kind of location prediction model for D can be established.

The former case is straightforward. In the latter case, the source of the route discovery, S, refers to its routing table in order to retrieve information on its previous relative distance with D and the time elapsed since S last received routing information for D. Let us designate this time as Tm. Based on this information and assuming a moderate velocity, Micro_Velocity, node S is then able to estimate its new relative distance to destination node D in terms of actual number of hops.

To accomplish this, a stochastic model, called relative distance estimation (RDE) algorithm, has been developed [RDMAR_MODEL] to estimate the minimum relative distance for the successful location of the destination terminal based on the previous relative distance of the terminals and the elapsed time since they last communicated (i.e., Tm). In this model, the total cost is defined as the total network resources spent during the route discovery procedure and RDE is then used to determine the optimal relative distance (in terms of hops) that results in the minimum cost.

Thereafter, node S limits the distribution for route queries by inserting the normalized hop-wise value of their RD in the Time-to-Live (TTL) [IP] field in the header of the route request (RREQ) packet. As TTL decreases at every node that forwards the packet, the range of broadcast is eventually restricted in a small area of the network with maximum radius from the source of the broadcast equal to RD hops. In the worst case, the newly calculated RD is equal to the radii of the network; that is, pure flooding. This procedure is called Relative-Distance Microdiscovery (RDM).

After broadcasting a RREQ, a node waits for a RREP. If the RREP is not received within RREP_WAIT_TIME milliseconds, the node MAY rebroadcast the RREQ, up to a maximum of RREQ_RETRIES times. Each rebroadcast MUST increment the RREQ_ID field.

Data packets waiting for a route (i.e., waiting for a RREP after RREQ has been sent) SHOULD be buffered at the source node. The buffering SHOULD be FIFO. If a RREQ has been rebroadcast RREQ_RETRIES times without receiving any RREP, all data packets destined for the corresponding destination SHOULD be dropped from the buffer.

When a node (source of RREQ or any intermediate node) receives a RREP and the node has a valid (in the sense not expired) pending

RREQ, the node keeps a record in its routing table of the correct value of its RD to the source of the RREP control packet (i.e., destination of RREQ).

A Route Discovery for a destination SHOULD NOT be initiated unless the initiating node has a packet in the Data Re-transmission Buffer requiring delivery to that destination. A Route Discovery for a given target node MUST NOT be initiated unless permitted by the rate-limiting information contained in the Route Request Table.

A node that receives a RREQ packet which is not destined for it, the node SHOULD NOT retransmit the packet, SHOULD NOT request explicit RDMAR acknowledgement from the next hop, SHOULD NOT expect a passive acknowledgement, and SHOULD NOT place the packet in the RREQ table for retransmission.

3.2.2 Generating Route Replies

As the RREQ is propagated hop-by-hop outward, each intermediate node that receives the RREQ packet creates a reverse route to the source of the RREQ in its routing table with next hop set to the address of the previous hop included in the RREQ packet.

When a node receives a RREQ or RREP packet it checks the source IP address, the RREQ_ID/RREP_ ID value and the destination IP address of the packet. The node first checks whether it is the intended destination of the packet. If so, the node proceeds and sends a Route Reply back to the source of the RREQ, according to the rules described in section 3.2.2.
If, however, the node is not the destination of the RREQ, it checks to see whether the broadcast packet has already been received in the past, and thus whether it has already transmitted the broadcast packet. If there is no existing list entry in its RREQ Table containing the same IP source address, RREQ_ID/RREP_ID value and the destination IP address of the packet the node retransmits the broadcast packet. If there is such a list entry with matching source IP address , RREQ_ID/RREP_ID value and the destination IP address of the packet the node MUST not propagate any copies of it after the first, avoiding the overhead of forwarding additional copies that reach this node along different paths.

A Route Reply (RREP) packet MUST be generated in response to a RREQ only by the destination node for which the RREQ is sent for. The route is made available by unicasting a RREP back to the source of the RREQ packet.

If a new route for some destination D is offered to a mobile station, then if the mobile station has already a route to D it compares the hop count (i.e., the "RD" field in the RREP message) of the new route to the RD for the route that already exists in its table. If the new route carries a smaller RD than the RD of the pending route, then the new route will be selected. If, however, the node does not have a route for the destination, it "blindly" adds the newly coming route. This is because in RDMAR a RREP is always generated from the destination node of a RREQ. Hence, it is highly impossible for nodes to receive stale routing information.

Finally, an intermediate node that receives a new RREP to forward, it SHOULD send replies not only to the destination node listed in this RREP, but also to all nodes that have previously sent RREQ to the source of the RREP and no reply has been forwarded to them yet.

In conclusion, in RDMAR route decision MUST be performed at the destination node and only the best selected route will be valid while all other possible routes remain passive; therefore avoid stale routes and packet duplicates.

3.3 Packet Forwarding and Route Maintenance

An intermediate node K, upon reception of a data packet, first processes the routing header and then forwards the packet to the next hop. In addition to that, node K SHOULD send an explicit message to the previous node on an attempt to examine whether bi-directional link can be established with the node where the packet is received from.

RDMAR, therefore, does not assume bi-directional links but in contrast nodes SHOULD exercise the possibility of having bi-directional links. In this way, nodes that forward a data packet will always have routing information to send the future acknowledgement back to the source.

If node K is unable to forward the packet because there is no route available or a forwarding error occurs along the data path as a result of a link or node failure, K MAY attempt a number of additional re-transmissions of the same data packet, up to a maximum number of retries MAX_DATA_RETRIES. The reason for multiple attempts is that this failure could have been caused by temporary factors such as noise bursts, moving node was in a radio shadow area etc. If, however, the failure persists, node K initiates the Route Maintenance Phase.

During the Route Maintenance procedure, node K exploits the spatial relation between itself and the source and destination nodes of the data packet. Depending on its relative position from the source and destination nodes of the packet, K may optionally initiate an RDM procedure or notify instead the source of the active call. This results from the relative distance of K from the source and destination nodes of a data packet. If K is close to the destination of the packet at the time of the failure (referring to its routing table), it SHOULD proceed and initiate an RDM procedure (to distinguish this RDM phase from the one that is triggered from a data source node, let us designate this as RM_RDM); otherwise, if K is close to the source of the data packet it SHOULD proceed and notify the source about the failure to deliver the packet through this path.
The latter case is called Failure Notification (FN) phase.

The RREQ traffic generated during the RM_RDM phase MUST have the 'E' flag set to '1', so as the recipients of this traffic to prioritize the traffic. This can be achieved by assigning queue priorities (e.g., by emulation of Priority Queue Discipline [QoS]) so that this traffic is always transmitted ahead of other types of traffic.

The reasons for prioritizing this control traffic are twofold. On one hand, an intermediate node that triggers the RM_RDM should receive a RREP fast so that a fast re-routing is performed in a manner seamless to data source; seamless in terms of the application's performance perspective and the graceful degradation of real-time traffic. On the other hand, data packets destined to the node for which a route discovery is in progress, keep accummulating on the input buffer of the node while the discovery is in progress (i.e., a RREP has not been received yet). Therefore, unless a route is received fast, the resources (buffers) of the source node of the RM_RDM will saturate and local congestion problems may arise at this node. Congestion in turn leads to the dropping of packets which causes big delays and intermittent application disruptions.

After broadcasting a RREQ, a node waits for a RREP. If the RREP is not received within RREP_WAIT_TIME_MICRO milliseconds, the node MAY rebroadcast the RREQ, up to a maximum of RREQ_RETRIES times. Each rebroadcast MUST increment the RREQ_ID field.

During the FN phase, each node K, that receives a packet for forwarding (i.e., the node is not the intended destination of the packet), keeps a list of all the neighbours that use this node as their default router to reach a destination D.
We call this list of nodes as "Dependent List" (similar to the concept of Dependent Downstream Routers, described in [DVMRP]). The importance of

this list is that the need of broadcasting is eliminated
by sending FN messages only to the nodes that actually use
the failed path. In this way, nodes that currently need a
route to the destination are notified to search for a
new path, if one still needed. The FN message propagates
upwards towards the source of the data packet.

Each node, upon receipt of a FN message, MUST remove from
its routing table the route associated with the destination
of the data packet, if the next hop to reach this destination
is the one through which the FN message is
received. Additionally, to secure the propagation of error
messages, nodes that receive a FN message but have an empty
"Dependent List" (thus being unable to forward the error
message), MUST use their routing caches to forward the message
to its destination. In this way, we ensure that
the affected data sources from the path failure are always
notified.

Nodes MUST NOT ptrigger the Route Maintenance phase for control
packets (i.e., RREQ, RREP and FN) or for Acknowledgement packets.

A node that receives a FN packet SHOULD NOT retransmit the
packet, SHOULD NOT request explicit RDMAR acknowledgement from the
next hop and SHOULD NOT expect a passive acknowledgement.

4. Quality of Service

RDMAR currently provides some minimal controls to
enable mobile nodes in an ad hoc network to specify, as part
of a Route Discovery, certain Quality of Service parameters that
a route to a destination must satisfy. In particular, a RREQ MAY
include a diverse of QoS Metrics such as Delay or Bandwidth bounds.

If, after establishment of such a route, any node along the
path detects that the requested Quality of Service parameters
can no longer be maintained, that node MUST trigger Route
Maintenance phase and re-discover a QoS route.

Although the RDMAR protocol is not designed to support QoS
guaranteed, the 'QoS Metrics' in RDMAR route discovery and
selection procedure can be customized in accordance to application
QoS requirements. The RDMAR route selection algorithm presented

earlier considers routes with the highest degree of stability and overall network resource consumption as the more important QoS metrics, followed by minimum-hop routes and minimum aggregate delay.

In RDMAR, there exists flexibility in terms of route selection based on which QoS parameter is viewed to be more important than others. Path stability is chosen to be the most important factor to be considered first since it determines how long the other QoS paramters can be maintained before the route is invalidated by mobile hosts' movements.

5. Optimizations

A number of optimizations can be added to the basic operation of Route discovery and Route Maintenance as described in Sections 3.2 and 3.3, respectively, that can improve load balancing, efficient bandwidth utilisation and enhance the quality of the routes used.

Optimisations on the basic RREP mechanism are also proposed:

* Coping with transient network connectivity - Urgent_RREP (U_RREP)

A crucial effect on the performance of a dynamic routing protocol is the motion pattern of mobile stations. For example no one of the existing routing protocols works within a network where all participating nodes are moving always outside of the hearing range of each other. However, we are interested in the case where occasionally a node becomes partitioned (in the sense, the node is completely disconnected) from the rest of the networking topology.

Let us study the case where a query flood is in transit while the destination of this request is effectively partitioned. In this situation, all RREQ packets that are "in-flight" are completely unproductive. In this case, upon timeout of the recently sent RREQ packet, the source of the RREQ will continue to generate RREQ packets until it receives a RREP or the number of retransmission exceeds the maximum allowed retransmission threshold. For the latter case we employ an exponential backoff delay to limit the rate at which the new route request targeted to the same host may be initiated, after the maximum retransmission threshold has been reached. However, as the partitioned node (say DST) is approaching the ad hoc vicinity and a MANET terminal realises that its new neighbour is the one that a request is in progress (i.e., DST), this terminal immediately proceeds and sends a RREP to all nodes that need a route to DST. This technique, called Urgent Route Reply (U_RREP), therefore serves to increase protocol adaptivity in the phase of transient connectivity such as the one described here.

* Extensions for Load Balancing Support

In RDMAR, load balancing could be effectively realised during the route discovery phase where the source of the route discovery receives a number of paths from destination and picks one according to the traffic conditions at the time where the reply arrives. That is, for a certain path, if the IP layer is responsible for the monitor of the traffic load for each port interface of the nodes along this path, the objective of the route discovery is to search for the least-congested routes (instead of the shortest-path routes), and to report to the source of the RREQ an aggregate of the traffic conditions along the newly discovered paths. Furthermore, it is evident for all the MANET routing schemes proposed so far, that an active path usually remains unchanged during the call lifetime if no failure occurs along the path.

Hence, if a session runs for sufficiently long period, the batteries of the intermediate hosts may be drained out, since the battery power consumed is essentially proportional to the size of the data packets sent. To deal with the situation, we propose that the source, the destination and the intermediate nodes of a route should keep track how long the route has been used for the same call without route reconstruction. When this time exceeds some predefined limit the route must be rediscovered. This approach leads to the conclusion that one important parameter that impacts the relative behaviour of every MANET routing protocol, is the number of connection requests an intermediate node is allowed to accept for call forwarding. Due to resource constraints, a node can accept a reasonable amount of connection requests for forwarding other nodes' packets. None of the currently proposed protocols, however, examines this concern.

* Optimised Handling of Route Errors - Failure Crankback

In our attempt to optimise the handling of error messages, we use a technique similar to one proposed in PNNI specification [PNNI], called crankback. We refer to the modified version of crankback used in RDMAR, as Failure Crankback technique. According to this, a node K that receives a data packet to forward, it first keeps a temporary copy of this packet before forwarding. If K receives an error message for this packet from some subsequent node as the packet traverses the path towards its destination, K may proceed and retransmit the packet through an alternate path, if one exists, instead of forwarding the error message to its destination (i.e., source of data packet).

The impact of this technique on the overall performance, is that when a node receives a FN message and meanwhile the node has received a new route to send the packet (i.e., next hop of the route is not the hop from where the FN came from), it proceeds and sends a new copy of the data packet through the new route. In this way, protocol makes full use of data caching and distributed routing, yet there is no need to let unnecessary error messages propagate up to the source of packet, especially in case where the distance from the failure to the senders is relatively high.

* Increasing Protocol Robustness

We are making experiments on the case where a node K that receives a data packet for destination S and the next hop J to reach D is unreachable, to send FN messages to all data source nodes that currently have active calls routed through J for every destination node that turns to be also unreachable due to the failure of link K-J, and not necessarily the source nodes of the active calls for D.

6. Multicasting

RDMAR has so far addressed only unicast ad hoc routing with a strong emphasis on network resource consumption and discovery of stable shortest-path routing patterns. However, a lot of multimedia collaborative applications today involve multiparty sessions and thus it is important for a routing protocol to support multicasting. We recently began experimentation with ways to customise the reactive route query process in order to support route querying for multiple destinations and multicast groups.

7. Configuration Parameters

This section gives default values for some important values associated with RDMAR protocol operations.

Parameter Name	Value
ACTIVE_ROUTE_TIMEOUT	3 sec
RREP_WAIT_TIME	2*ESTIMATED_TTL+DELAY_OFFSET
RREP_WAIT_TIME_MICRO	2*ESTIMATED_TTL+DELAY_OFFSET_MICRO
RREQ_RETRIES	3

ESTIMATED_TTL is the hop-wise distance estimated from the RDE algorithm. DELAY_OFFSET and DELAY_OFFSET_MICRO is an estimate of the average additional latency that should be added onto the ESTIMATED_TTL to allow for queuing delays, interrupt processing times and transfer times, during the RDM and RM_RDM phase, respectively.

8. Security Considerations

Currently, RDMAR does not address any security concerns. It is assumed, however, that security issues are adequately address by IPSec authentication headers with the necessary key management to distribute keys to the members of the ad hoc network using RDMAR.

References

[DVMRP]   S. Deering. Multicast Routing in Internetworks and Extended LANs, Stanford University, Department of Computer Science Technical Report: STAN-CS-88-1214, July 1988

[IP]   J.Postel. Internet Protocol, RFC 791, Septembet 1981

[MANET]   Scott Corson and Joseph Macker. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. RFC 2501, January 1999.

[PNNI] The ATM Forum Technical committee. Private Network-to -Network Interface Specification ver. 1.0, March 1996

[QoS]   P. Ferguson, G. Huston. Delifering QoS on the Internet and in Corporate Networks, Wiley Computer Publishing, 1998

[RDMAR_MODEL]   G.Aggelou, R. Tafazolli. A Model for the Route Discovery Mechanism of the RDMAR protocol, Submitted for Publication

[RFC]   S. Bradner. Key Words for Use in RFCs to Indicate Requirement Levels, RFC 2119, March 1997

Chair's Address

The Working Group can be contacted via its current chairs:

M. Scott Corson
Institute for Systems Research
University of Maryland
College Park, MD 20742
USA

Phone: +1 301 405-6630
Email: corson@isr.umd.edu

Joseph Macker
Information Technology Division
Naval Research Laboratory
Washington, DC 20375
USA

Phone: +1 202 767-2001
Email: macker@itd.nrl.navy.mil

Author's Address

Questions about this memo can be directed to:

George Aggelou
University of Surrey
Centre for Communications Systems Research
Surrey, GU2 5XH
UK
+44 (0) 1483 300800 ext. 3468/2292
+44 (0) 1483 259504 (fax)
G.Aggelou@ee.surrey.ac.uk

Rahim Tafazolli
University of Surrey
Centre for Communications Systems Research
Surrey, GU2 5XH
UK
+44 (0) 1483 259834
+44 (0) 1483 259504 (fax)
R.Tafazolli@ee.surrey.ac.uk