Mobile Ad hoc Networks Working Group
Internet-Draft
Expires: April 26, 2006

I. Chakeres
Boeing
E. Belding-Royer
UC Santa Barbara
C. Perkins
Nokia
October 23, 2005

Dynamic MANET On-demand (DYMO) Routing
draft-ietf-manet-dymo-03

Abstract

The Dynamic MANET On-demand (DYMO) routing protocol is intended for use by mobile nodes in wireless multihop networks. It offers adaptation to changing network topology and determines unicast routes between nodes within the network.

Table of Contents

1. Overview

The Dynamic MANET On-demand (DYMO) routing protocol enables reactive, multihop routing between participating nodes that wish to communicate. The basic operations of the DYMO protocol are route discovery and management. During route discovery the originating node initiates dissemination of a Route Request (RREQ) throughout the network to find the target node. During this dissemination process, each intermediate node records a route to the originating node. When the target node receives the RREQ, it responds with a Route Reply (RREP) unicast toward the originating node. Each node that receives the RREP records a route to the target node, and then the RREP is unicast toward the originating node. When the originating node receives the RREP, routes have then been established between the originating node and the target node in both directions.

In order to react to changes in the network topology nodes maintain their routes and monitor their links. When a packet is received for a route that is no longer available the source of the packet is notified. A Route Error (RERR) is sent to the packet source to indicate the current route is broken. Once the source receives the RERR, it re-initiates route discovery if it still has packets to deliver.

In order to enable extension of the base specification, DYMO defines a generic element structure and handling of future extensions. By defining a fixed structure and default handling, future extensions are handled in a predetermined fashion.

DYMO uses sequence numbers as they have been proven to ensure loop freedom [3]. Sequence numbers enable nodes to determine the order of DYMO route discovery packets, thereby avoiding use of stale routing information.

All DYMO packets are transmitted via UDP on port TBD.

2. Terminology

IP Destination Address (IPDestinationAddress)

The destination of a packet, indicated by examining the IP header.

IP Source Address (IPSourceAddress)

The source of a packet, indicated by examining the IP header.

DYMOcast

Packet transmission to all DYMO routers. DYMOcast packets should be sent with an IPDestinationAddress of IPv4 TBD (IPv6 TBD), the DYMOcastAddress.

Routing Element (RE)

A DYMO message element that is used to distribute routing information.

Route Invalidation

Disabling the use of a route, causing it to be unavailable for forwarding data.

Route Reply (RREP)

Upon receiving a RREQ, the target node generates a Route Reply (RREP). A RREP is a RE with a unicast IPDestinationAddress, indicating that this RE is to be unicast hop-by-hop toward the TargetAddress.

Route Request (RREQ)

A node generates a Route Request (RREQ) to discover a valid route to a particular destination (TargetAddress). A RREQ is simply a RE with the DYMOcastAddress in the IPDestinationAddress field of the IP packet. Also, the A-bit is set to one (A=1) to indicate that the TargetNode must respond with a RREP.

Valid Route

A known route where the Route.ValidTimeout is greater than the current time.

3. Data Structures

3.1 Route Table Entry

The route table entry is a conceptual data structure. Implementations may use any internal representation that conforms to the semantics of a route as specified in this document.

- Route.DestAddress
- Route.DeleteTimeout
- Route.HopCnt
- Route.IsGateway
- Route.NextHopAddress
- Route.NextHopInterface
- Route.Prefix
- Route.SeqNum
- Route.ValidTimeout

These fields are defined as follows:

Route Node Address (Route.DestAddress)

The IP address of the node associated with the routing table entry.

Route Delete Timeout (Route.DeleteTimeout)

If the time current is after Route.DeleteTimeout the corresponding routing table entry MUST be deleted.

Route Hop Count (Route.HopCnt)

The number of intermediate node hops before reaching the Route.DestAddress.

Route Is Gateway (Route.IsGateway)

1-bit selector indicating whether the Route.DestAddress is a gateway.

Route Next Hop Address (Route.NextHopAddress)

The IP address of the next node on the path toward the Route.DestAddress.

Route Next Hop Interface (Route.NextHopInterface)

The interface used to send packets toward the Route.DestAddress.

Route Prefix (Route.Prefix)

6-bit field that specifies the size of the subnet reachable through the Route.DestAddress, see Section 4.7. The definition of the Prefix field is different for gateways; entries with Route.IsGateway set to one (1).

Route Sequence Number (Route.SeqNum)

The sequence number of the Route.DestAddress.

Route.ValidTimeout

The time at which a route table entry is scheduled to be invalidated. The routing table entry is no longer considered valid if the current time is after Route.ValidTimeout.

3.2 DYMO Message Elements

3.2.1 Generic DYMO Element Structure

All DYMO message elements MUST conform to the fixed data structure below.

```
 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      |           Len           |    TTL    |I|Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.          NotifyAddress (Only Types with M-bit set)            .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
. TargetAddress (for non-DYMOcastAddress IPDestinationAddresses).
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                                                               .
.                             Data                              .
.                     Type-Specific Payload                     .
.                                                               .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1

Element Type (Type)

```
 0                          0
 0 1 2 3 4 5 6 7 8          0 1 2 3 4 5 6 7 8
+-+-+-+-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+-+
|     Type      |     =    |M| H |           |
+-+-+-+-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+-+
```

Figure 2

The Type field identifies the element as well as the handling by nodes that do not implement or understand the element. The most significant bit, the M-bit, denotes whether the element requires notification via an Unsupported-element Error (UERR) when the element is not understood or handled by a particular node. The next two bits, H-bits, identify how the Type is to be handled by nodes not implementing the Type, regardless of UERR delivery. Section 4.6.3 describes the handling behavior based on the Type.

I-bit (I)

1-bit selector indicating whether the element has been ignored by some node that has relayed this element. If I=1 the element has been ignored.

Reserved (Reserved, Reservd, Res, R)

Reserved bits. These bits are set to zero (0) during element creation and ignored during processing.

Element Time to Live (TTL)

6-bit field that identifies the maximum number of times the element is to be retransmitted. The TTL field operates similar to IPTTL (MaxCount) and is decremented at each hop. When TTL reaches zero (0) the element is dropped.

Element Length (Len)

12-bit field that indicates the size of the element in bytes, including the fixed portion.

Element Notify Address (NotifyAddress)

The node to send a UERR if the Element Type is unsupported or not handled by the processing node. The NotifyAddress field is only present if the Type field has the M-bit is set to one (1).

Element Target Address (TargetAddress)

The node that is the ultimate destination of the element. This field is only required if the IPDestinationAddress is not the DYMOcastAddress. During hop-by-hop transmission of a DYMO packet the IPDestinationAddress is filled with the Route.NextHopAddress of the route table entry associated with

      the TargetAddress.

   Element Data (Data)

      Type-specific payload.

3.2.2 Routing Element (RE)

```
    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      |           Len           |     TTL     |I|A|S| Res |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                          TargetAddress                          .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           TargetSeqNum                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  THopCnt   |Res|                                                .
   +-+-+-+-+-+-+-+-+                                                 .
   .                                                                 .
   .                    Routing Block 1 (RBlock1)                    .
   .                                                                 .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   .                                                                 .
   .                    Additional Routing Blocks                    .
   .                                                                 .
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

                                Figure 3

   A-bit (A)

      1-bit selector indicating whether this RE requires a RREP by
      the TargetAddress. If A=1 a RREP is required. The
      instructions for generating a RREP are described in
      Section 4.3.2.

   S-bit (S)

1-bit selector indicating whether this RE requires a unicast message be sent to the previous hop address. This message MAY used by the previous hop to ensure that the link traversed is not unidirectional. The handling instructions for the S-bit is explained in Section 4.3.2.

Element Target Address (TargetAddress)

The node that is the ultimate destination of the Routing Element.

Target Sequence Number (TargetSeqNum)

The sequence number of the ultimate destination of this Routing Element. If the Sequence Number is unknown for this particular Route.DestAddress then TargetSeqNum is set to zero (0).

Target Hop Count (THopCnt)

6-bit field that identifies the number of intermediate nodes through which a packet traversed on the route to this particular TargetAddress the last time a route was available. The THopCnt is the Route.HopCnt of the TargetAddress, stored in the routing table of the RREQ originator. If the hop count information is not available at the originating node then the THopCnt is set to zero (0).

Routing Block (RBlock)

Data structure that describes routing information related to a particular IP address, RBNodeAddress.

Routing Block (RBlock)

```
 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
                                +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                                |G|  RBPrefix   |Res| RBHopCnt  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                          RBNodeAddress                        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          RBNodeSeqNum                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4

G-bit (G)

1-bit selector to indicate whether the RBNodeAddress is a gateway. If G=1 RBNodeAddress is a gateway. For more information on gateway operation see Section 4.8.

Prefix Size (Prefix)

7-bit field that specifies the size of the subnet reachable through the associated node, see Section 4.7. The definition of Prefix is different for gateways.

Routing Block Hop Count (RBHopCnt)

6-bit field that identifies the number of intermediate nodes through which the associated RBlock has passed.

Routing Block Node Address (RBNodeAddress)

The IP address of the node associated with this RBlock.

Routing Block Node Sequence Number (RBNodeSeqNum)

The sequence number of the node associated with this RBlock.

3.2.3 Route Error (RERR)

```
 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      |           Len           |    TTL    |I|Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                          UNodeAddress1                          .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          UNodeSeqNum1                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.               Additional UNodeAddressN (if needed)              .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Additional UNodeSeqNumN (if needed)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5

Unreachable Node Address (UNodeAddress)

The IP address of the unreachable node.

Unreachable Node Sequence Number (UNodeSeqNum)

The sequence number of the unreachable node, if known; otherwise, zero (0). RERR generation is described in Section 4.5.3.

3.2.4 Unsupported-element Error (UERR)

```
 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      |          Len            |    TTL    |I|Reserved |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                          TargetAddress                        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                       UElemTargetAddress                      .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
.                        UERRNodeAddress                        .
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   UElemType   |
+-+-+-+-+-+-+-+-+
```

Figure 6

Element Target Address (TargetAddress)

The node that is the ultimate destination of the element, NotifyAddress.

Unsupported-element Target Address (UElemTargetAddress)

Address of the destination of the element that caused generation of this UERR; TargetAddress from the offending fixed DYMO element.

Unsupported-element Node Address (UERRNodeAddress)

The IP address of the node that created the UERR.

Unsupported-element Type (UElemType)

The Type that required generation of the UERR.

4. Detailed Operation

4.1 Sequence Numbers

4.1.1 Maintaining a Sequence Number

DYMO requires each node in the network to maintain its own sequence number (OwnSeqNum). The circumstances for a node to change its OwnSeqNum are described in Section 4.3.1.

4.1.2 Incrementing a Sequence Number

When a node increments its OwnSeqNum (as described in Section 4.3.1 and Section 4.3.2) it MUST do so by treating the sequence number value as if it was an unsigned number. The sequence number zero (0) is reserved and is used in several DYMO data structures to represent an unknown sequence number.

4.1.3 Sequence Number Rollover

If the sequence number has been assigned to be the largest possible number representable as a 32-bit unsigned integer (i.e., 4294967295), then the sequence number MUST be set to one (1) when incremented.

4.1.4 Actions After Sequence Number Loss

If a node's OwnSeqNum is lost, it must take certain actions to avoid creating routing loops. To prevent this possibility after sequence number loss a node MUST wait for at least ROUTE_DELETE_PERIOD before transmitting any DYMO packet other than RERR generated by this node. If a DYMO control packet is received during this period, the node SHOULD process it normally but MUST not retransmit any DYMO control packets. If a data packet is received during this waiting period the node MUST send a RERR message to the IPSourceAddress with the UNodeSeqNum set to zero (0) and restart its waiting period before transmitting any DYMO control packets except RERR generated by this node.

4.2 DYMO Routing Table Operations

4.2.1 Creating or Updating a Route Table Entry from a Routing Block

While processing a RE, as described in Section 4.3.2, a node checks its routing table for an entry to the RBNodeAddress using longest-prefix matching. In the event that no matching entry is found, an entry is created.

If a matching entry is found, the routing information about

RBNodeAddress contained in this RBlock is NOT stale if the result of subtracting the Route.SeqNum from RBNodeSeqNum is greater than zero (0) using signed 32-bit arithmetic.

If a matching entry is found, the routing information about RBNodeAddress contained in this RBlock is NOT stale if the result of subtracting the Route.SeqNum from RBNodeSeqNum is equal to zero (0) using signed 32-bit arithmetic but it SHOULD be disregarded if:

- o the Route.ValidTimeout has not passed and RBHopCnt is greater than or equal to Route.HopCnt, OR

- o the Route.ValidTimeout has passed and RBHopCnt is greater than Route.HopCnt plus one (1).

If the information in this RBlock is stale or disregarded and this RBlock is the first RBlock in a RREQ this DYMO packet MUST be dropped. For other RBlocks containing stale or disregarded routing information, the RBlock is simply removed from this RE and the RELen adjusted. Removing stale and disregarded RBlocks ensures that unused information is not propagated further.

If the route information for RBNodeAddress is not stale, disregarded or a disregarded RREP, then the following actions occur to the route table entry for RBNodeAddress:

1. the Route.HopCnt is set to the RBHopCnt,

2. the Route.IsGateway is set to the G-bit,

3. the Route.NextHopAddress is set to the node that transmitted this DYMO packet (IPSourceAddress),

4. the Route.NextHopInterface is set to the interface that this DYMO packet was received on,

5. the Route.Prefix is set to RBPrefix,

6. the Route.SeqNum is set to the RBNodeSeqNum,

7. and the Route.ValidTimeout is set to the current time + ROUTE_TIMEOUT.

If a valid route exists to RBNodeAddress, the route can be used to send any queued data packets and to fulfill any outstanding route requests.

4.2.2 Route Table Entry Timeouts

If the current time is later than a routing entry's Route.ValidTimeout, the route is stale and it is not be used to route packets. The information in invalid entries is still used for generating RREQ messages.

If the current time is after Route.DeleteTimeout the corresponding routing table entry MUST be deleted.

4.3 Routing Element

4.3.1 Routing Element Creation

When a node creates a RREQ it SHOULD increment its OwnSeqNum by one according to the rules specified in Section 4.1.2. When a node creates a RREP, then it increments its OwnSeqNum under the following conditions:

- o TargetSeqNum is greater than OwnSeqNum OR
- o TargetSeqNum is equal to OwnSeqNum AND THopCnt is less than to RBHopCnt.

In either case (for RREQ or RREP), the node MUST create the first RBlock. It sets the RBNodeAddress to its own address. The RBNodeSeqNum is the node's OwnSeqNum. The node may advertise a prefix using the Prefix field, as described in Section 4.7. Otherwise, the Prefix field is set to zero (0). The node may advertise it is a gateway by setting the G-bit if it is a gateway, as described in Section 4.8. Otherwise, the G-bit is set to zero (0). The TTL SHOULD be set to NET_DIAMETER, but MAY be set smaller. For the case of RREQ, the TTL MAY be set in accordance with an expanding ring search as described in [2].

4.3.2 Routing Element Processing

After general DYMO element pre-processing (Section 4.6.2), the RBHopCnt for the first RBlock is incremented by one (1). A route to the first RBlock is then created or updated, as described in Section 4.2.1. If this RBlock does not result in a valid route the packet MUST be dropped.

Each additional RBlock SHOULD be processed. For each RBlock the RBHopCnt is incremented by one (1), then a route is created or updated as defined in Section 4.2.1. Each RBlock resulting in a valid route entry may alleviate a future route discovery. Any RBlocks that do not result in a valid route update or that are not

processed MUST be removed from the RE.

If this node is the TargetAddress AND the A-bit is set (A=1), this node MUST respond with a RREP. The target node creates a new RE as described in Section 4.3.1. The TargetAddress in the new RE is set to the RBNodeAddress1 from the RE currently being processed. The THopCnt is the hop count for the TargetAddress. The A-bit is set to (A=0). The IPDestinationAddress is set to the Route.NextHopAddress for the TargetAddress. The TargetSeqNum is set to Route.SeqNum for the TargetAddress. Then the new RE undergoes post-processing, according to Section 4.6.4.

After processing a RE, a node MAY append its routing information to the RE, according to the process described in Section 4.3.3. The additional routing information will reduce route discoveries to this node.

If this node is not the TargetAddress, the current RE SHOULD be handled according to Section 4.6.4.

If this node is the TargetAddress, the current packet and any additional elements are processed, but this packet is not retransmitted.

If the S-bit is set to one (1) in the RE, then a unicast message SHOULD be sent or have been sent to the previous hop within UNICAST_MESSAGE_SENT_TIMEOUT. Any unicast packet will serve this purpose, but it MAY be an ICMP REPLY message. If a message is not sent, then the previous hop may assume that the link is unidirectional and may blacklist this node.

4.3.3 Appending Additional Routing Information to an Existing Routing Element

Appending routing information will alleviate route discovery attempts to this node from other nodes that process the resultant RE. Nodes MAY append a RBlock to RE processed if the believes that this additional routing information will alleviate future RREQ.

Prior to appending a RBlock to a RE, a node MUST increment its OwnSeqNum as defined in Section 4.1.2. Then it appends its IP address, OwnSeqNum, Prefix and G-bit to the RE in a RBlock. The RBHopCnt is set to zero (0). The RE Len is also adjusted according to the number of RBlocks in the RE.

4.4 Route Discovery

A node generates a Route Request (RREQ) to discover a valid route to

a particular destination (TargetAddress). A RREQ is a RE with the A-bit is set to one (A=1) to indicate that the TargetNode must respond with a RREP. If a sequence number is known for the TargetAddress it is placed in the TargetSeqNum field. Otherwise, TargetSeqNum is set to zero (0). A TargetSeqNum of zero MAY be set to indicate that only the destination may respond to this RREQ. If a hop count is known for the TargetAddress it is placed in the THopCnt field. Otherwise, the THopCnt is set to zero (0). The IPDestinationAddress is set to the DYMOcastAddress. Then the RE is then transmitted according to the procedure defined in Section 4.6.5.

After issuing a RREQ, the originating node waits for a route to be created to the TargetNode. If a route is not received within RREQ_WAIT_TIME milliseconds, this node MAY again try to discover a route by issuing another RREQ.

To reduce congestion in a network, repeated attempts at route discovery for a particular TargetNode SHOULD utilize a binary exponential backoff. The first time a node issues a RREQ, it waits RREQ_WAIT_TIME milliseconds for a route to the TargetNode. If a route is not found within that time, the node MAY send another RREQ. If a route is not found within two (2) times the current waiting time, another RREQ may be sent, up to a total of RREQ_TRIES. For each additional attempt, the waiting time for the previous RREQ is multiplied by two (2) so that the waiting time conforms to a binary exponential backoff.

Data packets awaiting for a route SHOULD be buffered.

If a route discovery has been attempted RREQ_TRIES times without receiving a route to the TargetNode, all data packets destined for the corresponding TargetNode SHOULD be dropped from the buffer and a Destination Unreachable ICMP message SHOULD be delivered to the application.

4.5 Route Maintenance

4.5.1 Active Link Monitoring

Before a route can be used for forwarding a packet, it MUST be checked to make sure that the route is still valid. If the Route.ValidTimeout is earlier than the current time, the packet cannot be forwarded, and a RERR message MUST be generated (see section Section 4.5.3). In this case, the Route.DeleteTimeout is set to Route.ValidTimeout + ROUTE_DELETE_TIMEOUT.

If the current time is after Route.DeleteTimeout, then the route SHOULD be deleted, though a route MAY be deleted at any time.

Nodes MUST monitor links on active routes. This may be accomplished by one or several mechanisms. Including:

o Link layer feedback

o Hello messages

o Neighbor discovery

o Route timeout

Upon detecting a link break the detecting node MUST set the Route.ValidTimeout to the current time for all routes active routes utilizing the broken link.

A RERR MUST be issued if a data packet is received and it cannot be delivered to the next hop. RERR generation is described in Section 4.5.3. A RERR SHOULD be issued after detecting a broken link of an active route to quickly notify nodes that a link break occurred and a route or routes are no longer available. If a route has not been used, a RERR SHOULD NOT be generated.

4.5.2 Updating Route Lifetimes

To avoid route timeouts for active routes, a node MUST update the Route.ValidTimeout to the IPSourceAddress to be the current time + ROUTE_TIMEOUT upon receiving a data packet.

To avoid route timeouts for active routes, a node SHOULD update the Route.ValidTimeout to the IPDestinationAddress to be the current time + ROUTE_TIMEOUT upon successfully transmitting a packet to the next hop.

4.5.3 Route Error Generation

When a data packet is received for a destination without a valid routing table entry, a Route Error (RERR) MUST be generated by this node. A RERR informs the source that the current route is no longer available.

In the RERR, the UNodeAddress1 field is the address of the unreachable node (IPDestinationAddress) from the data packet. If the UNodeSeqNum is known, it is placed in the RERR; otherwise, zero (0) is placed in the UNodeSeqNum field of the RERR. The TTL SHOULD be set to NET_DIAMETER, but may be set smaller. The IPDestinationAddress is set to the DYMOcastAddress.

Additional unreachable nodes that required the same unavailable link

(routes with the same Route.NextHopAddress and Route.NextHopInterface) as the UNodeAddress1 SHOULD be appended to the RERR. For each unreachable node the UNodeAddress and UNodeSeqNum are appended. The Len is set accordingly.

The RERR is then processed as described in Section 4.6.5.

4.5.4 Route Error Processing

When a node processes a RERR after generic element pre-processing (Section 4.6.2), it SHOULD set the Route.ValidTimeout to the current time for each route to a UNodeAddress that meets all of the following conditions:

1. The Route.NextHopAddress is the same as the RERR IPSourceAddress.

2. The Route.NextHopInterface is the same as the interface on which the RERR was received.

3. The UNodeSeqNum is zero (0) OR the result of subtracting Route.SeqNum from UNodeSeqNum is less than or equal to zero using signed 32-bit arithmetic.

Each UNodeAddress that did not result in a change to Route.ValidTimeout SHOULD be removed from the RERR.

Prior to generic post processing a node MAY remove any UNodeAddressN, UNodeSeqNumN pairs except UNodeAddress1 to decrease the element size.

If at least one UNodeAddress remains and at least one route remains in the RERR it SHOULD be handled as described in Section 4.6.4 to continue notification of nodes effected by the broken link. Otherwise, the RERR is dropped.

4.6 General DYMO Processing

4.6.1 DYMO Control Packet Processing

A DYMO packet may consist of multiple DYMO elements. Each element is processed individually and in sequence, from first to last. An incoming DYMO packet MUST be completely processed prior to any DYMO packet transmissions.

The length of IP addresses (32-bits for IPv4 and 128-bits for IPv6) inside DYMO elements is dependent on the IP packet header. For example, if the IP header is IPv6 then all DYMO elements contained in the payload use IPv6 addresses.

Unless specific element processing requires dropping the DYMO packet, it is retransmitted after processing, according to the method described in Section 4.6.5.

4.6.2 Generic Element Pre-processing

Each element in a DYMO packet undergoes pre-processing before the element specific processing occurs. During pre-processing, the TTL is decremented by one (1).

4.6.3 Processing Unsupported DYMO Element Types

This section describes the processing for unsupported DYMO element Types. The Type field identifies the handling by nodes that do not implement, support or understand a particular Element Type. The most significant bit (M-bit) indicates whether an Unsupported-element Error (UERR) SHOULD be sent to the NotifyAddress. The next two bits (H-bits) identify how the element should be handled.

```
 0                          0
 0 1 2 3 4 5 6 7 8          0 1 2 3 4 5 6 7 8
+-+-+-+-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+-+
|      Type       |   =    |M| H |           |
+-+-+-+-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+-+
```

If the M-bit is set in a DYMO element being processed by a node that does not support this Element Type a UERR SHOULD be sent to the NotifyAddress. This is accomplished by following the instructions in Section 4.6.3.1.

Regardless of whether or not a UERR is sent in response to this unsupported Element Type, the processing node MUST also examine the H-bits to determine how this unsupported element is handled. The unsupported element Type MUST be handled as follows:

- If H == 00 skip the element and continue as if the packet did not contain this element.

- If H == 01 remove the unsupported element (using the Len field) from the packet and continue as if the packet did not include this element.

- If H == 10 set the ignored bit (I-bit) skip this element and continue, as if the packet did not contain this element.

- If H == 11 drop the packet without processing any other DYMO elements.

4.6.3.1 Generating an Unsupported-element Error

When an unsupported element type is received with the M-bit set, the processing node SHOULD generate an Unsupported-element Error (UERR). The TargetAddress is set to the NotifyAddress. The IPDestinationAddress is set to the Route.NextHopAddress toward the NotifyAddress. The UElemTargetAddress is set to the TargetAddress from the unsupported element. The UERRNodeAddress is set to the node address generating this UERR. The UElemType is the Type from the unsupported element. The TTL SHOULD be set to NET_DIAMETER, but MAY be set smaller. The Len is set to the total number of bytes in this UERR. The element is then processed as described in Section 4.6.4.

4.6.4 Generic Element Post-processing

If the first element TTL is zero (0) the DYMO packet is dropped after processing of all elements. If the TTL of the first element is greater than zero the DYMO packet is re-transmitted after processing of all elements. If the TTL of any element is zero (0) after processing it MUST be removed from the DYMO packet prior to transmission.

4.6.5 DYMO Control Packet Transmission

DYMO packet transmission and re-transmission is controlled by the IPDestinationAddress. If the IPDestinationAddress is a unicast address, the packet IPDestinationAddress is replaced by the Route.NextHopAddress from a route table lookup for the TargetAddress. If a route for the TargetAddress is unknown or invalid the packet is dropped and a RERR SHOULD be generated.

For all currently defined DYMO packets the IPTTL (IPMaxCount) SHOULD be set to 1 (IPTTL=1), since all DYMO packet communications are between direct neighbors.

4.7 Routing Prefix

Any node can advertise connectivity to a subset of other nodes within its address space by using the prefix field in RE. The nodes within the advertised prefix SHOULD NOT participate in the MANET and MUST be reachable by forwarding packets to the node advertising connectivity. For example, 192.168.1.1 with a prefix of 16 indicates all nodes with the prefix 192.168.X.X are reachable through 192.168.1.1.

The meaning of the prefix field is altered for routes to the gateway; Route.IsGateway is one (1). If the G-bit is set the prefix in association with the IP address indicates that all nodes outside the subnet are reachable via the gateway node. For example, a route to a

gateway with IP address 192.168.1.1 and a prefix of 16 indicates that all nodes with an IP address NOT matching 192.168.X.X are reachable via this route.

4.8 Internet Attachment

Internet attachment consists of a network of MANET nodes connected to the Internet via a gateway node. The gateway is responsible for responding to RREQs for TargetNodes outside its configured MANET subnet, as well as delivering packets to destinations outside the MANET subnet.

MANET nodes wishing to be reachable from nodes in the Internet MUST have IP addresses within the gateway's configured MANET subnet. Given a node with a globally routeable address or care-of address handled by the gateway, the gateway is responsible for routing and forwarding packets received from the Internet destined for nodes inside its MANET subnet.

Since many nodes may commonly wish to communicate with the gateway, the gateway SHOULD indicate to nodes that it is a gateway by setting the gateway bit (G-bit) in any RE created or processed. The G-bit flag indicates to nodes in the MANET that the RBNodeAddress is attached to the Internet and is capable of routing data packets to all nodes outside of the configured MANET subnet, described by the RBNodeAddress and RBPrefix fields.

4.9 Multiple Interfaces

It is likely that DYMO will be used with multiple wireless interfaces; therefore, the particular interface over which packets arrive must be known whenever a packet is received. Whenever a new route is created, the interface through which the Route.DestAddress can be reached is also recorded into the route table entry.

When multiple interfaces are available, a node transmitting a DYMOcast packet SHOULD send the packet on all interfaces that have been configured for operation in the MANET.

4.10 Packet Generation Limits

To avoid congestion, a node SHOULD NOT transmit more than RATE_LIMIT control messages per second. RREQ packets SHOULD be discarded before RREP or RERR packets.

5. Configuration Parameters

Here are some default parameter values for DYMO:

Parameter Name	Suggested Value
NET_DIAMETER	10
RATE_LIMIT	10
ROUTE_TIMEOUT	3000 milliseconds
ROUTE_DELETE_TIMEOUT	5*ROUTE_TIMEOUT
RREQ_WAIT_TIME	1000 milliseconds
RREQ_TRIES	3

For large networks or networks with frequent topology changes the default DYMO parameters should be adjusted using either experimentally determined values or dynamic adaptation. For example, in networks with infrequent topology changes ROUTE_TIMEOUT may be set to a much larger value.

It is assumed that all nodes in the network share the same parameter settings. Different parameter values for ROUTE_TIMEOUT or ROUTE_DELETE_TIMEOUT in addition to arbitrary packet delays may result in frequent route breaks or routing loops.

6. IANA Considerations

DYMO defines a Type field for each element within a packet sent to port TBD. A new registry will be created for the values for this Type field, and the following values will be assigned:

Type	Value
Routing Element (RE)	1
Route Error (RERR)	2
Unsupported-element Error (UERR)	3

Future values of the Type will be allocated using standard actions as described in [1]. For future Types with the M-bit set NotifyAddress MUST be included. Similarly for future Types that are unicast hop-by-hop (packets not sent to the DYMOcastAddress), these Types MUST include the TargetAddress field.

7. Security Considerations

Currently, DYMO does not specify any special security measures. Routing protocols, however, are prime targets for impersonation attacks. In networks where the node membership is not known, it is difficult to determine the occurrence of impersonation attacks, and security prevention techniques are difficult at best. However, when the network membership is known and there is a danger of such attacks, DYMO elements must be protected by the use of authentication techniques, such as those involving generation of unforgeable and cryptographically strong message digests or digital signatures. While DYMO does not place restrictions on the authentication mechanism used for this purpose, IPsec Authentication Element (AH) is an appropriate choice for cases where the nodes share an appropriate security association that enables the use of AH.

In particular, RE messages SHOULD be authenticated to avoid creation of spurious routes to a destination. Otherwise, an attacker could masquerade as that destination and maliciously deny service to the destination and/or maliciously inspect and consume traffic intended for delivery to the destination. RERR messages, while slightly less dangerous, SHOULD be authenticated in order to prevent malicious nodes from disrupting active routes between communicating nodes.

If the mobile nodes in the ad hoc network have pre-established security associations, the purposes for which the security associations are created should include that of authorizing the processing of DYMO control packets. Given this understanding, the mobile nodes should be able to use the same authentication mechanisms based on their IP addresses as they would have used otherwise.

8. Acknowledgments

DYMO is a descendant of the design of previous MANET reactive protocols, especially AODV [2] and DSR [4]. Changes to previous MANET reactive protocols stem from research and implementation experiences. Thanks to Luke Klein-Berndt, Pedro Ruiz, Fransisco Ros and Koojana Kuladinithi for reviewing of DYMO, as well as several specification suggestions.

9. References

9.1 Normative References

[1] T. Narten and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 2434, BCP 26, October 1998.

[2] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

9.2 Informative References

[3] C. Perkins and E. Belding-Royer, "Ad hoc On-Demand Distance Vector (AODV) Routing", Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, pp. 90-100, February 1999.

[4] D. Johnson and D. Maltz, "Dynamic Source Routing (DSR) in Ad hoc Networks", In Mobile Computing, Chapter 5, pp. 153-181, 1996.

Authors' Addresses

Ian Chakeres
Boeing Phantom Works
The Boeing Company
P.O. Box 3707 Mailcode 7L-49
Seattle, WA 98124-2207
USA

Email: ian.chakeres@gmail.com

Elizabeth Belding-Royer
University of California Santa Barbara
Dept. of Computer Science
Santa Barbara, CA 93106-5110
USA

Phone: +1-805-893-3411
Fax: +1-805-893-8553
Email: ebelding@cs.ucsb.edu

Charlie Perkins
Nokia Research Center
313 Fairchild Drive
Mountain View, CA 94043
USA

Phone: +1-650-625-2986
Fax: +1-650-625-2502
Email: charlie.perkins@nokia.com

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.