           Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)

Abstract

   This document describes the protocol design and architecture for
   Enhanced Interior Gateway Routing Protocol (EIGRP). EIGRP is a
   routing protocol based on Distance Vector technology. The specific
   algorithm used is called "DUAL", a Diffusing Update Algorithm as
   referenced in "Loop-Free Routing Using Diffusing Computations"
   (Garcia-Luna-Aceves 1993). The algorithm and procedures were
   researched, developed, and simulated by SRI International.

Status of This Memo

   This document is not an Internet Standards Track specification; it is
   published for informational purposes.

   This is a contribution to the RFC Series, independently of any other
   RFC stream. The RFC Editor has chosen to publish this document at
   its discretion and makes no statement about its value for
   implementation or deployment. Documents approved for publication by
   the RFC Editor are not a candidate for any level of Internet
   Standard; see Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7868.

Copyright Notice

Table of Contents

1. Introduction

   This document describes the Enhanced Interior Gateway Routing
   Protocol (EIGRP), a routing protocol designed and developed by Cisco
   Systems, Inc. DUAL, the algorithm used to converge the control plane
   to a single set of loop-free paths is based on research conducted at
   SRI International [3]. The Diffusing Update Algorithm (DUAL) is the
   algorithm used to obtain loop freedom at every instant throughout a
   route computation [2]. This allows all routers involved in a
   topology change to synchronize at the same time; the routers not
   affected by topology changes are not involved in the recalculation.
   This document describes the protocol that implements these functions.

2. Conventions

2.1. Requirements Language

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
   document are to be interpreted as described in RFC 2119 [1].

2.2. Terminology

   The following is a list of abbreviations and terms used throughout
   this document:

   ACTIVE State:
      The local state of a route on a router triggered by any event that
      causes all neighbors providing the current least-cost path to fail
      the Feasibility Condition check. A route in Active state is
      considered unusable. During Active state, the router is actively
      attempting to compute the least-cost loop-free path by explicit
      coordination with its neighbors using Query and Reply messages.

   Address Family Identifier (AFI):
      Identity of the network-layer protocol reachability information
      being advertised [12].

   Autonomous System (AS):
      A collection of routers exchanging routes under the control of one
      or more network administrators on behalf of a single
      administrative entity.

Base Topology:
   A routing domain representing a physical (non-virtual) view of the
   network topology consisting of attached devices and network
   segments EIGRP uses to form neighbor relationships. Destinations
   exchanged within the Base Topology are identified with a Topology
   Identifier value of zero (0).

Computed Distance (CD):
   Total distance (metric) along a path from the current router to a
   destination network through a particular neighbor computed using
   that neighbor's Reported Distance (RD) and the cost of the link
   between the two routers. Exactly one CD is computed and
   maintained per the [Destination, Advertising Neighbor] pair.

CR-Mode
   Conditionally Received Mode

Diffusing Computation:
   A distributed computation in which a single starting node
   commences the computation by delegating subtasks of the
   computation to its neighbors that may, in turn, recursively
   delegate sub-subtasks further, including a signaling scheme
   allowing the starting node to detect that the computation has
   finished while avoiding false terminations. In DUAL, the task of
   coordinated updates of routing tables and resulting best path
   computation is performed as a diffusing computation.

Diffusing Update Algorithm (DUAL):
   A loop-free routing algorithm used with distance vectors or link
   states that provides a diffused computation of a routing table.
   It works very well in the presence of multiple topology changes
   with low overhead. The technology was researched and developed at
   SRI International [3].

Downstream Router:
   A router that is one or more hops away from the router in question
   in the direction of the destination.

EIGRP:
   Enhanced Interior Gateway Routing Protocol.

Feasibility Condition:
   The Feasibility Condition is a sufficient condition used by a
   router to verify whether a neighboring router provides a loop-free
   path to a destination. EIGRP uses the Source Node Condition
   stating that a neighboring router meets the Feasibility Condition
   if the neighbor's RD is less than this router's Feasible Distance.

Feasible Distance (FD):
   Defined as the least-known total metric to a destination from the
   current router since the last transition from ACTIVE to PASSIVE
   state. Being effectively a record of the smallest known metric
   since the last time the network entered the PASSIVE state, the FD
   is not necessarily a metric of the current best path. Exactly one
   FD is computed per destination network.

Feasible Successor:
   A neighboring router that meets the Feasibility Condition for a
   particular destination, hence, providing a guaranteed loop-free
   path.

Neighbor/Peer:
   For a particular router, another router toward which an EIGRP
   session, also known as an "adjacency", is established. The
   ability of two routers to become neighbors depends on their mutual
   connectivity and compatibility of selected EIGRP configuration
   parameters. Two neighbors with interfaces connected to a common
   subnet are known as adjacent neighbors. Two neighbors that are
   multiple hops apart are known as remote neighbors.

PASSIVE state:
   The local state of a route in which at least one neighbor
   providing the current least-cost path passes the Feasibility
   Condition check. A route in PASSIVE state is considered usable
   and not in need of a coordinated re-computation.

Network Layer Reachability Information (NLRI):
   Information a router uses to calculate the global routing table to
   make routing and forwarding decisions.

Reported Distance (RD):
   For a particular destination, the value representing the router's
   distance to the destination as advertised in all messages carrying
   routing information. RD is not equivalent to the current distance
   of the router to the destination and may be different from it
   during the process of path re-computation. Exactly one RD is
   computed and maintained per destination network.

Sub-Topology:
   For a given Base Topology, a sub-topology is characterized by an
   independent set of routers and links in a network for which EIGRP
   performs an independent path calculation. This allows each sub-
   topology to implement class-specific topologies to carry class-
   specific traffic.

Successor:
   For a particular destination, a neighboring router that meets the
   Feasibility Condition and, at the same time, provides the least-
   cost path.

Stuck In Active (SIA):
   A destination that has remained in the ACTIVE State in excess of a
   predefined time period at the local router (Cisco implements this
   as 3 minutes).

Successor-Directed Acyclic Graph (SDAG):
   For a particular destination, a graph defined by routing table
   contents of individual routers in the topology, such that nodes of
   this graph are the routers themselves and a directed edge from
   router X to router Y exists if and only if router Y is router X's
   successor. After the network has converged, in the absence of
   topological changes, SDAG is a tree.

Topology Change / Topology-Change Event:
   Any event that causes the CD for a destination through a neighbor
   to be added, modified, or removed. As an example, detecting a
   link-cost change, receiving any EIGRP message from a neighbor
   advertising an updated neighbor's RD.

Topology Identifier (TID):
   A number that is used to mark prefixes as belonging to a specific
   sub-topology.

Topology Table:
   A data structure used by EIGRP to store information about every
   known destination including, but not limited to, network prefix /
   prefix length, FD, RD of each neighbor advertising the
   destination, CD over the corresponding neighbor, and route state.

Type, Length, Value (TLV):
   An encoding format for information elements used in EIGRP messages
   to exchange information. Each TLV-formatted information element
   consists of three generic fields: Type identifying the nature of
   information carried in this element, Length describing the length
   of the entire TLV triplet, and Value carrying the actual
   information. The Value field may, itself, be internally
   structured; this depends on the actual type of the information
   element. This format allows for extensibility and backward
   compatibility.

Upstream Router:
   A router that is one or more hops away from the router in
   question, in the direction of the source of the information.

   VID:
      VLAN Identifier

   Virtual Routing and Forwarding (VRF):
      Independent Virtual Private Network (VPN) routing/forwarding
      tables that coexist within the same router at the same time.

3. The Diffusing Update Algorithm (DUAL)

   The Diffusing Update Algorithm (DUAL) constructs least-cost paths to
   all reachable destinations in a network consisting of nodes and edges
   (routers and links). DUAL guarantees that each constructed path is
   loop free at every instant including periods of topology changes and
   network reconvergence. This is accomplished by all routers, which
   are affected by a topology change, computing the new best path in a
   coordinated (diffusing) way and using the Feasibility Condition to
   verify prospective paths for loop freedom. Routers that are not
   affected by topology changes are not involved in the recalculation.
   The convergence time with DUAL rivals that of any other existing
   routing protocol.

3.1. Algorithm Description

   DUAL is used by EIGRP to achieve fast loop-free convergence with
   little overhead, allowing EIGRP to provide convergence rates
   comparable, and in some cases better than, most common link state
   protocols [10]. Only nodes that are affected by a topology change
   need to propagate and act on information about the topology change,
   allowing EIGRP to have good scaling properties, reduced overhead, and
   lower complexity than many other interior gateway protocols.

   Distributed routing algorithms are required to propagate information
   as well as coordinate information among all nodes in the network.
   Unlike basic Bellman-Ford distance vector protocols that rely on
   uncoordinated updates when a topology change occurs, DUAL uses a
   coordinated procedure to involve the affected part of the network
   into computing a new least-cost path, known as a "diffusing
   computation". A diffusing computation grows by querying additional
   routers for their current RD to the affected destination, and it
   shrinks by receiving replies from them. Unaffected routers send
   replies immediately, terminating the growth of the diffusing
   computation over them. These intrinsic properties cause the
   diffusing computation to self-adjust in scope and terminate as soon
   as possible.

   One attribute of DUAL is its ability to control the point at which
   the diffusion of a route calculation terminates by managing the
   distribution of reachability information through the network.

Controlling the scope of the diffusing process is accomplished by
hiding reachability information through aggregation (summarization),
filtering, or other means. This provides the ability to create
effective failure domains within a single AS, and allows the network
administrator to manage the convergence and processing
characteristics of the network.

3.2. Route States

A route to a destination can be in one of two states: PASSIVE or
ACTIVE. These states describe whether the route is guaranteed to be
both loop free and the shortest available (the PASSIVE state) or
whether such a guarantee cannot be given (the ACTIVE state).
Consequently, in PASSIVE state, the router does not perform any route
recalculation in coordination with its neighbors because no such
recalculation is needed.

In ACTIVE state, the router is actively involved in re-computing the
least-cost loop-free path in coordination with its neighbors. The
state is reevaluated and possibly changed every time a topology
change is detected. A topology change is any event that causes the
CD to the destination over any neighbor to be added, changed, or
removed from EIGRP's topology table.

More exactly, the two states are defined as follows:

o Passive

   A route is considered to be in the Passive state when at least one
   neighbor that provides the current least-total-cost path passes
   the Feasibility Condition check that guarantees loop freedom. A
   route in the PASSIVE state is usable and its next hop is perceived
   to be a downstream router.

o Active

   A route is considered to be in the ACTIVE state if neighbors that
   do not pass the Feasibility Condition check provide lowest-cost
   path, and therefore the path cannot be guaranteed loop free. A
   route in the ACTIVE state is considered unusable and this router
   must coordinate with its neighbors in the search for the new loop-
   free least-total-cost path.

In other words, for a route to be in PASSIVE state, at least one
neighbor that provides the least-total-cost path must be a Feasible
Successor. Feasible Successors providing the least-total-cost path
are also called "successors". For a route to be in PASSIVE state, at
least one successor must exist.

Conversely, if the path with the least total cost is provided by
routers that are not Feasible Successors (and thus not successors),
the route is in the ACTIVE state, requiring re-computation.

Notably, for the definition of PASSIVE and ACTIVE states, it does not
matter if there are Feasible Successors providing a worse-than-least-
total-cost path. While these neighbors are guaranteed to provide a
loop-free path, that path is potentially not the shortest available.

The fact that the least-total-cost path can be provided by a neighbor
that fails the Feasibility Condition check may not be intuitive.
However, such a situation can occur during topology changes when the
current least-total-cost path fails and the next-least-total-cost
path traverses a neighbor that is not a Feasible Successor.

While a router has a route in the ACTIVE state, it must not change
its successor (i.e., modify the current SDAG) nor modify its own
Feasible Distance or RD until the route enters the PASSIVE state
again. Any updated information about this route received during
ACTIVE state is reflected only in CDs. Any updates to the successor,
FD, and RD are postponed until the route returns to PASSIVE state.
The state transitions from PASSIVE to ACTIVE and from ACTIVE to
PASSIVE are controlled by the DUAL FSM and are described in detail in
Section 3.5.

3.3. Feasibility Condition

The Feasibility Condition is a criterion used to verify loop freedom
of a particular path. The Feasibility Condition is a sufficient but
not a necessary condition, meaning that every path meeting the
Feasibility Condition is guaranteed to be loop free; however, not all
loop-free paths meet the Feasibility Condition.

The Feasibility Condition is used as an integral part of DUAL
operation: every path selection in DUAL is subject to the Feasibility
Condition check. Based on the result of the Feasibility Condition
check after a topology change is detected, the route may either
remain PASSIVE (if, after the topology change, the neighbor providing
the least cost path meets the Feasibility Condition) or it needs to
enter the ACTIVE state (if the topology change resulted in none of
the neighbors providing the least cost path to meet the Feasibility
Condition).

The Feasibility Condition is a part of DUAL that allows the diffused
computation to terminate as early as possible. Nodes that are not
affected by the topology change are not required to perform a DUAL
computation and may not be aware a topology change occurred. This
can occur in two cases:

First, if informed about a topology change, a router may keep a route
in PASSIVE state if it is aware of other paths that are downstream
towards the destination (routes meeting the Feasibility Condition).
A route that meets the Feasibility Condition is determined to be loop
free and downstream along the path between the router and the
destination.

Second, if informed about a topology change for which it does not
currently have reachability information, a router is not required to
enter into the ACTIVE state, nor is it required to participate in the
DUAL process.

In order to facilitate describing the Feasibility Condition, a few
definitions are in order.

o A successor for a given route is the next hop used to forward data
   traffic for a destination. Typically, the successor is chosen
   based on the least-cost path to reach the destination.

o A Feasible Successor is a neighbor that meets the Feasibility
   Condition. A Feasible Successor is regarded as a downstream
   neighbor towards the destination, but it may not be the least-cost
   path but could still be used for forwarding data packets in the
   event equal or unequal cost load sharing was active. A Feasible
   Successor can become a successor when the current successor
   becomes unreachable.

o The Feasibility Condition is met when a neighbor's advertised
   cost, (RD) to a destination is less than the FD for that
   destination, or in other words, the Feasibility Condition is met
   when the neighbor is closer to the destination than the router
   itself has ever been since the destination has entered the PASSIVE
   state for the last time.

o The FD is the lowest distance to the destination since the last
   time the route went from ACTIVE to PASSIVE state. It should be
   noted it is not necessarily the current best distance; rather, it
   is a historical record of the best distance known since the last
   diffusing computation for the destination has finished. Thus, the
   value of the FD can either be the same as the current best
   distance, or it can be lower.

A neighbor that advertises a route with a cost that does not meet the
Feasibility Condition may be upstream and thus cannot be guaranteed
to be the next hop for a loop-free path. Routes advertised by
upstream neighbors are not recorded in the routing table but saved in
the topology table.

3.4. DUAL Message Types

   DUAL operates with three basic message types: QUERY, UPDATE, and
   REPLY.

   o UPDATE - sent to indicate a change in metric or an addition of a
      destination.

   o QUERY - sent when the Feasibility Condition fails, which can
      happen for reasons like a destination becoming unreachable or the
      metric increasing to a value greater than its current FD.

   o REPLY - sent in response to a QUERY or SIA-QUERY

   In addition to these three basic types, two additional sub-types have
   been added to EIGRP:

   o SIA-QUERY - sent when a REPLY has not been received within one-
      half of the SIA interval (90 seconds as implemented by Cisco).

   o SIA-REPLY - sent in response to an SIA-QUERY indicating the route
      is still in ACTIVE state. This response does not stratify the
      original QUERY; it is only used to indicate that the sending
      neighbor is still in the ACTIVE state for the given destination.

   When in the PASSIVE state, a received QUERY may be propagated if
   there is no Feasible Successor found. If a Feasible Successor is
   found, the QUERY is not propagated and a REPLY is sent for the
   destination with a metric equal to the current routing table metric.
   When a QUERY is received from a non-successor in ACTIVE state, a
   REPLY is sent and the QUERY is not propagated. The REPLY for the
   destination contains a metric equal to the current routing table
   metric.

3.5. DUAL Finite State Machine (FSM)

   The DUAL FSM embodies the decision process for all route
   computations. It tracks all routes advertised by all neighbors. The
   distance information, known as a metric, is used by DUAL to select
   efficient loop-free paths. DUAL selects routes to be inserted into a
   routing table based on Feasible Successors. A successor is a
   neighboring router used for packet forwarding that has a least-cost
   path to a destination that is guaranteed not to be part of a routing
   loop.

   When there are no Feasible Successors but there are neighbors
   advertising the destination, a recalculation must occur to determine
   a new successor.

The amount of time it takes to calculate the route impacts the
convergence time. Even though the recalculation is not processor
intensive, it is advantageous to avoid recalculation if it is not
necessary. When a topology change occurs, DUAL will test for
Feasible Successors. If there are Feasible Successors, it will use
any it finds in order to avoid any unnecessary recalculation.

The FSM, which applies per destination in the topology table,
operates independently for each destination. It is true that if a
single link goes down, multiple routes may go into ACTIVE state.
However, a separate SDAG is computed for each destination, so loop-
free topologies can be maintained for each reachable destination.

```
          +------------+                 +-----------+
          |           \               /             |
          |            \             /              |
          |       +===========================+     |
          |       |                            |     |
          |(1)|                Passive            |(2)|
          +-->|                            |<--+
                  +===========================+
                   ^     |    ^    ^    ^    |
                 (14)|     | (15)|  (13)|    |
                   |   (4)|    | (16)|   | (3)|
                   |     |    | |    | |    +-----------+
                   |     |    | |    | |   /            \
           +-------+     +    +   |   +------------+    \
          /            /    /    |             \    \
         /            /    /  +----+            \    \
         |            |   |    |                |    |
         |            v   |    |                |    v
      +==========+(11) +==========+   +==========+(12) +==========+
      |  Active  |---->|  Active  |(5) |  Active  |---->|  Active  |
      |          | (9)|          |---->|         | (10)|          |
      |  oij=0   |<----|  oij=1   |    |  oij=2   |<----|  oij=3   |
      +--|        |   +--|        |    +--|        |   +--|        |
      |  +==========+ |  +==========+  |  +==========+ |  +==========+
      |     ^   |(5) |     ^         |     ^    ^    |     ^
      |     |   +-----|------|---------|----+    |     |     |
      +------+       +------+        +--------+    +---------+
      (6,7,8)        (6,7,8)         (6,7,8)        (6,7,8)
```
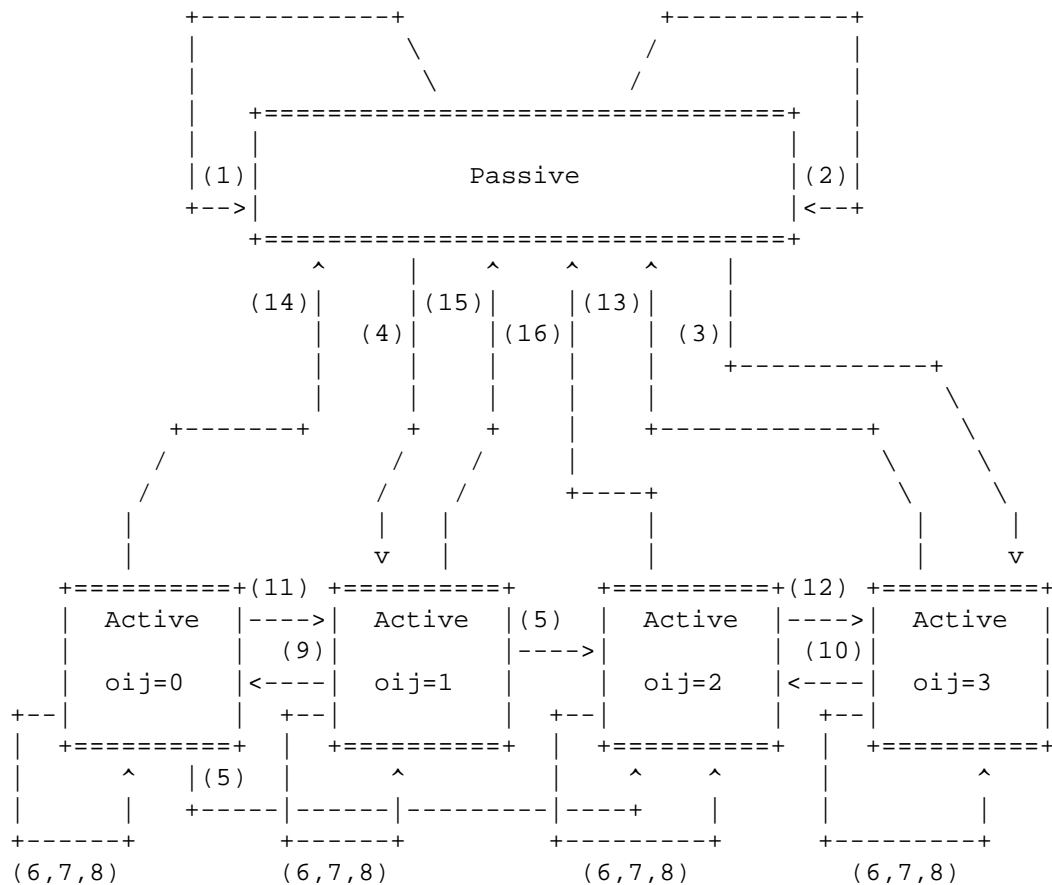
                    Figure 1: DUAL Finite State Machine

   Legend:

     i   Node that is computing route
     j   Destination node or network
     k   Any neighbor of node i
    oij QUERY origin flag
       0 = metric increase during ACTIVE state
       1 = node i originated
       2 = QUERY from, or link increase to, successor during ACTIVE state
       3 = QUERY originated from successor
    rijk REPLY status flag for each neighbor k for destination j
       1 = awaiting REPLY
       0 = received REPLY
    lik = the link connecting node i to neighbor k

The following describes in detail the state/event/action transitions
of the DUAL FSM. For all steps, the topology table is updated with
the new metric information from either QUERY, REPLY, or UPDATE
received.

(1) A QUERY is received from a neighbor that is not the current
     successor. The route is currently in PASSIVE state. As the
     successor is not affected by the QUERY, and a Feasible Successor
     exists, the route remains in PASSIVE state. Since a Feasible
     Successor exists, a REPLY MUST be sent back to the originator of
     the QUERY. Any metric received in the QUERY from that neighbor
     is recorded in the topology table and the Feasibility Check (FC)
     is run to check for any change to current successor.

(2) A directly connected interface changes state (connects,
     disconnects, or changes metric), or similarly an UPDATE or QUERY
     has been received with a metric change for an existing
     destination, the route will stay in the PASSIVE state if the
     current successor is not affected by the change, or it is no
     longer reachable and there is a Feasible Successor. In either
     case, an UPDATE is sent with the new metric information if it
     has changed.

(3) A QUERY was received from a neighbor who is the current
     successor and no Feasible Successors exist. The route for the
     destination goes into ACTIVE state. A QUERY is sent to all
     neighbors on all interfaces that are not split horizon. Split
     horizon takes effect for a query or update from the successor it
     is using for the destination in the query. The QUERY origin
     flag is set to indicate the QUERY originated from a neighbor
     marked as successor for route. The REPLY status flag is set for
     all neighbors to indicate outstanding replies.

(4) A directly connected link has gone down or its cost has
     increased, or an UPDATE has been received with a metric
     increase. The route to the destination goes to ACTIVE state if
     there are no Feasible Successors found. A QUERY is sent to all
     neighbors on all interfaces. The QUERY origin flag is to
     indicate that the router originated the QUERY. The REPLY status
     flag is set to 1 for all neighbors to indicate outstanding
     replies.

(5) While a route for a destination is in ACTIVE state, and a QUERY
     is received from the current successor, the route remains in
     ACTIVE state. The QUERY origin flag is set to indicate that
     there was another topology change while in ACTIVE state. This
     indication is used so new Feasible Successors are compared to
     the metric that made the route go to ACTIVE state with the
     current successor.

(6) While a route for a destination is in ACTIVE state and a QUERY
     is received from a neighbor that is not the current successor, a
     REPLY should be sent to the neighbor. The metric received in
     the QUERY should be recorded.

(7) If a link cost changes, or an UPDATE with a metric change is
     received in ACTIVE state from a non-successor, the router stays
     in ACTIVE state for the destination. The metric information in
     the UPDATE is recorded. When a route is in the ACTIVE state,
     neither a QUERY nor UPDATE are ever sent.

(8) If a REPLY for a destination, in ACTIVE state, is received from
     a neighbor or the link between a router and the neighbor fails,
     the router records that the neighbor replied to the QUERY. The
     REPLY status flag is set to 0 to indicate this. The route stays
     in ACTIVE state if there are more replies pending because the
     router has not heard from all neighbors.

(9) If a route for a destination is in ACTIVE state, and a link
     fails or a cost increase occurred between a router and its
     successor, the router treats this case like it has received a
     REPLY from its successor. When this occurs after the router
     originates a QUERY, it sets the QUERY origin flag to indicate
     that another topology change occurred in ACTIVE state.

(10) If a route for a destination is in ACTIVE state, and a link
     fails or a cost increase occurred between a router and its
     successor, the router treats this case like it has received a
     REPLY from its successor. When this occurs after a successor
     originated a QUERY, the router sets the QUERY origin flag to
     indicate that another topology change occurred in ACTIVE state.

(11) If a route for a destination is in ACTIVE state, the cost of the
     link through which the successor increases, and the last REPLY
     was received from all neighbors, but there is no Feasible
     Successor, the route should stay in ACTIVE state. A QUERY is
     sent to all neighbors. The QUERY origin flag is set to 1.

   (12) If a route for a destination is in ACTIVE state because of a
        QUERY received from the current successor, and the last REPLY
        was received from all neighbors, but there is no Feasible
        Successor, the route should stay in ACTIVE state. A QUERY is
        sent to all neighbors. The QUERY origin flag is set to 3.

   (13) Received replies from all neighbors. Since the QUERY origin
        flag indicates the successor originated the QUERY, it
        transitions to PASSIVE state and sends a REPLY to the old
        successor.

   (14) Received replies from all neighbors. Since the QUERY origin
        flag indicates a topology change to the successor while in
        ACTIVE state, it need not send a REPLY to the old successor.
        When the Feasibility Condition is met, the route state
        transitions to PASSIVE.

   (15) Received replies from all neighbors. Since the QUERY origin
        flag indicates either the router itself originated the QUERY or
        FC was not satisfied with the replies received in ACTIVE state,
        FD is reset to infinite value and the minimum of all the
        reported metrics is chosen as FD and route transitions back to
        PASSIVE state. A REPLY is sent to the old-successor if oij
        flags indicate that there was a QUERY from successor.

   (16) If a route for a destination is in ACTIVE state because of a
        QUERY received from the current successor or there was an
        increase in distance while in ACTIVE state, the last REPLY was
        received from all neighbors, and a Feasible Successor exists for
        the destination, the route can go into PASSIVE state and a REPLY
        is sent to the successor if oij indicates that QUERY was
        received from the successor.

3.6. DUAL Operation -- Example Topology

   The following topology (Figure 2) will be used to provide an example
   of how DUAL is used to reroute after a link failure. Each node is
   labeled with its costs to destination N. The arrows indicate the
   successor (next hop) used to reach destination N. The least-cost
   path is selected.

```
                             N
                             |
                       (1)A ---<--- B(2)
                          |          |
                          ^          |
                          |          |
                       (2)D ---<--- C(3)
```
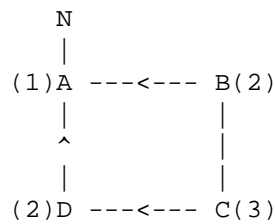
                     Figure 2: Stable Topology

   In the case where the link between A and D fails (Figure 3);

```
       N                                    N
       |                                    |
       A ---<--- B                          A ---<--- B
       |         |                          |         |
       X         |                          ^         |
       |         |                          |         |
       D ---<--- C                          D ---<--- C
         Q->                                      <-R
```

```
                        N
                        |
                  (1)A ---<--- B(2)
                               |
                               ^
                               |
                  (4)D --->--- C(3)
```
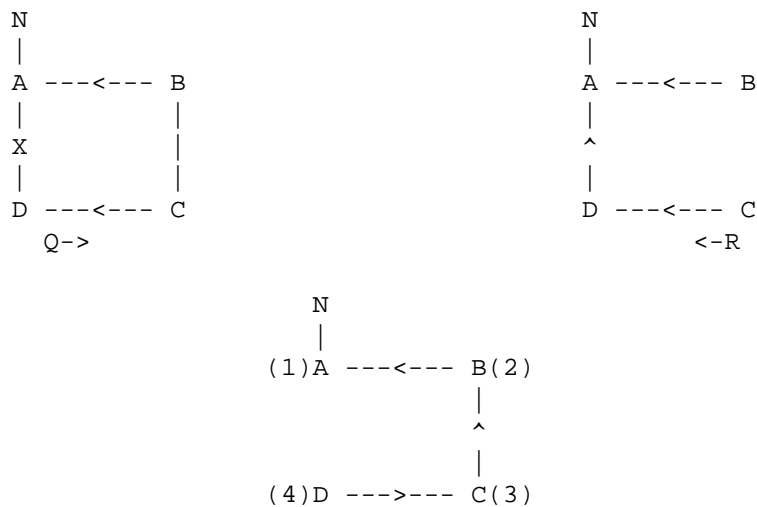
                 Figure 3: Link between A and D Fails

   Only observing the destination provided by node N, D enters the
ACTIVE state and sends a QUERY to all its neighbors, in this case
node C.
   C determines that it has a Feasible Successor and replies
immediately with metric 3.
   C changes its old successor of D to its new single successor B
and the route to N stays in PASSIVE state.
   D receives the REPLY and can transition out of ACTIVE state
since it received replies from all its neighbors.
   D now has a viable path to N through C.
   D selects C as its successor to reach node N with a cost of 4.

Notice that nodes A and B were not involved in the recalculation
since they were not affected by the change.

Let's consider the situation in Figure 4, where Feasible Successors
may not exist. If the link between node A and B fails, B goes into
ACTIVE state for destination N since it has no Feasible Successors.
Node B sends a QUERY to node C. C has no Feasible Successors, so it
goes active for destination N; and since C has no neighbors, it
replies to the QUERY, deletes the destination, and returns to the
PASSIVE state for the unreachable route. As C removes the (now
unreachable) destination from its table, C sends REPLY to its old
successor. B receives this REPLY from C, and determines this is the
last REPLY it is waiting on before determining what the new state of
the route should be; on receiving this REPLY, B deletes the route to
N from its routing table.

Since B was the originator of the initial QUERY, it does not have to
send a REPLY to its old successor (it would not be able to any ways,
because the link to its old successor is down). Note that nodes A
and D were not involved in the recalculation since their successors
were not affected.

```
       N                                N
       |                                |
    (1)A ---<--- B(2)                   A ------- B    Q
       |         |                      |         |   |^         ^
       ^         ^                      ^         |   v|         |
       |         |                      |         |    |         |
    (2)D         C(3)                   D         C   ACK        R
```
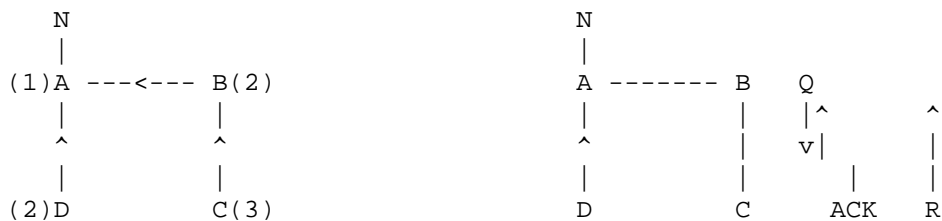
        Figure 4: No Feasible Successors When Link between A and B Fails

4. EIGRP Packets

   EIGRP uses five different packet types to handle session management
   and pass DUAL Message types:

      HELLO Packets (includes ACK)
      QUERY Packets (includes SIA-Query)
      REPLY Packets (includes SIA-Reply)
      REQUEST Packets
      UPDATE Packets

   EIGRP packets are directly encapsulated into a network-layer
   protocol, such as IPv4 or IPv6. While EIGRP is capable of using
   additional encapsulation (such as AppleTalk, IPX, etc.) no further
   encapsulation is specified in this document.

Support for network-layer protocol fragmentation is not supported,
and EIGRP will attempt to avoid a maximum size packets that exceed
the interface MTU by sending multiple packets that are less than or
equal to MTU-sized packets.

Each packet transmitted will use either multicast or unicast network-
layer destination addresses. When multicast addresses are used, a
mapping for the data link multicast address (when available) must be
provided. The source address will be set to the address of the
sending interface, if applicable.

The following network-layer multicast addresses and associated data
link multicast addresses:

    224.0.0.10 for IPv4 "EIGRP Routers" [13]
    FF02:0:0:0:0:0:0:A for IPv6 "EIGRP Routers" [14]

They will be used on multicast-capable media and will be media
independent for unicast addresses. Network-layer addresses will be
used and the mapping to media addresses will be achieved by the
native protocol mechanisms.

4.1. UPDATE Packets

UPDATE packets carry the DUAL UPDATE message type and are used to
convey information about destinations and the reachability of those
destinations. When a new neighbor is discovered, unicast UPDATE
packets are used to transmit a full table to the new neighbor, so the
neighbor can build up its topology table. In normal operation (other
than neighbor startup such as a link cost changes), UPDATE packets
are multicast. UPDATE packets are always transmitted reliably. Each
TLV destination will be processed individually through the DUAL FSM.

4.2. QUERY Packets

A QUERY packet carries the DUAL QUERY message type and is sent by a
router to advertise that a route is in ACTIVE state and the
originator is requesting alternate path information from its
neighbors. An infinite metric is encoded by setting the delay part
of the metric to its maximum value.

If there is a topology change that causes multiple destinations to be
marked ACTIVE, EIGRP will build one or more QUERY packets for all
destinations present. The state of each route is recorded
individually, so a responding QUERY or REPLY need not contain all the
same destinations in a single packet. Since EIGRP uses a reliable
transport mechanism, route QUERY packets are also guaranteed be
reliably delivered.

When a QUERY packet is received, each destination will trigger a DUAL
event, and the state machine will run individually for each route.
Once the entire original QUERY packet is processed, then a REPLY or
SIA-REPLY will be sent with the latest information.

4.3. REPLY Packets

A REPLY packet carries the DUAL REPLY message type and will be sent
in response to a QUERY or SIA-QUERY packet. The REPLY packet will
include a TLV for each destination and the associated vector metric
in its own topology table.

The REPLY packet is sent after the entire received QUERY packet is
processed. When a REPLY packet is received, there is no reason to
process the packet before an acknowledgment is sent. Therefore, an
acknowledgment is sent immediately and then the packet is processed.
The sending of the acknowledgment is accomplished either by sending
an ACK packet or by piggybacking the acknowledgment onto another
packet already being transmitted.

Each TLV destination will be processed individually through the DUAL
FSM. When a QUERY is received for a route that doesn't exist in our
topology table, a REPLY with an infinite metric is sent and an entry
in the topology table is added with the metric in the QUERY if the
metric is not an infinite value.

If a REPLY for a designation not in the Active state, or not in the
topology table, EIGRP will acknowledge the packet and discard the
REPLY.

4.4. Exception Handling

4.4.1. Active Duration (SIA)

When an EIGRP router transitions to ACTIVE state for a particular
destination, a QUERY is sent to a neighbor and the ACTIVE timer is
started to limit the amount of time a destination may remain in an
ACTIVE state.

A route is regarded as SIA when it does not receive a REPLY within a
preset time. This time interval is broken into two equal periods
following the QUERY, and up to three additional "busy" periods in
which an SIA-QUERY packet is sent for the destination.

This process is begun when a router sends a QUERY to its neighbor.
After one-half the SIA time interval (default implementation is 90
seconds), the router will send an SIA-QUERY; this must be replied to
with either a REPLY or SIA-REPLY. Any neighbor that fails to send

either a REPLY or SIA-REPLY with-in one-half the SIA interval will
result in the neighbor being deemed to be "stuck" in the active
state.

Cisco also limits the number of SIA-REPLY messages allowed to three.
Once the timeout occurs after the third SIA-REPLY with the neighbor
remaining in an ACTIVE state (as noted in the SIA-Reply message), the
neighbor being deemed to be "stuck" in the active state.

If the SIA state is declared, DUAL may take one of two actions;

   a) Delete the route from that neighbor, acting as if the neighbor
      had responded with an unreachable REPLY message from the
      neighbor.

   b) Delete all routes from that neighbor and reset the adjacency
      with that neighbor, acting as if the neighbor had responded
      with an unreachable message for all routes.

Implementation note: Cisco currently implements option (b).

4.4.1.1. SIA-QUERY

When a QUERY is still outstanding and awaiting a REPLY from a
neighbor, there is insufficient information to determine why a REPLY
has not been received. A lost packet, congestion on the link, or a
slow neighbor could cause a lack of REPLY from a downstream neighbor.

In order to try to ascertain if the neighboring device is still
attempting to converge on the active route, EIGRP may send an SIA-
QUERY packet to the active neighbor(s). This enables an EIGRP router
to determine if there is a communication issue with the neighbor or
if it is simply still attempting to converge with downstream routers.

By sending an SIA-QUERY, the originating router may extend the
effective active time by resetting the ACTIVE timer that has been
previously set, thus allowing convergence to continue so long as
neighbor devices successfully communicate that convergence is still
underway.

The SIA-QUERY packet SHOULD be sent on a per-destination basis at
one-half of the ACTIVE timeout period. Up to three SIA-QUERY packets
for a specific destination may be sent, each at a value of one-half
the ACTIVE time, so long as each are successfully acknowledged and
met with an SIA-REPLY.

Upon receipt of an SIA-QUERY packet, an EIGRP router should first
send an ACK and then continue to process the SIA-QUERY information.
The QUERY is sent on a per-destination basis at approximately one-
half the active time.

If the EIGRP router is still active for the destination specified in
the SIA-QUERY, the router should respond to the originator with the
SIA-REPLY indicating that active processing for this destination is
still underway by setting the ACTIVE flag in the packet upon
response.

If the router receives an SIA-QUERY referencing a destination for
which it has not received the original QUERY, the router should treat
the packet as though it was a standard QUERY:

   1) Acknowledge the receipt of the packet

   2) Send a REPLY if a successor exists

   3) If the SIA-QUERY is from the successor, transition to the
      ACTIVE state if and only if a Feasibility Condition check fails
      and send an SIA-REPLY with the ACTIVE bit set

4.4.1.2. SIA-REPLY

An SIA-REPLY packet is the corresponding response upon receipt of an
SIA-QUERY from an EIGRP neighbor. The SIA-REPLY packet will include
a TLV for each destination and the associated vector metric in the
topology table. The SIA-REPLY packet is sent after the entire
received SIA-QUERY packet is processed.

If the EIGRP router is still ACTIVE for a destination, the SIA-REPLY
packet will be sent with the ACTIVE bit set. This confirms for the
neighbor device that the SIA-QUERY packet has been processed by DUAL
and that the router is still attempting to resolve a loop-free path
(likely awaiting responses to its own QUERY to downstream neighbors).

The SIA-REPLY informs the recipient that convergence is complete or
still ongoing; it is an explicit notification that the router is
still actively engaged in the convergence process. This allows the
device that sent the SIA-QUERY to determine whether it should
continue to allow the routes that are not converged to be in the
ACTIVE state or if it should reset the neighbor relationship and
flush all routes through this neighbor.

5. EIGRP Operation

   EIGRP has four basic components:

        o Finite State Machine
        o Reliable Transport Protocol
        o Neighbor Discovery/Recovery
        o Route Management

5.1. Finite State Machine

   The detail of DUAL, the State Machine used by EIGRP, is covered in
   Section 3.5.

5.2. Reliable Transport Protocol

   The reliable transport is responsible for guaranteed, ordered
   delivery of EIGRP packets to all neighbors. It supports intermixed
   transmission of multicast and unicast packets. Some EIGRP packets
   must be transmitted reliably and others need not. For efficiency,
   reliability is provided only when necessary.

   For example, on a multi-access network that has multicast
   capabilities, such as Ethernet, it is not necessary to send HELLOs
   reliably to all neighbors individually. EIGRP sends a single
   multicast HELLO with an indication in the packet informing the
   receivers that the packet need not be acknowledged. Other types of
   packets, such as UPDATE packets, require acknowledgment and this is
   indicated in the packet. The reliable transport has a provision to
   send multicast packets quickly when there are unacknowledged packets
   pending. This helps ensure that convergence time remains low in the
   presence of varying speed links.

   DUAL assumes there is lossless communication between devices and thus
   must depend on the transport protocol to guarantee that messages are
   transmitted reliably. EIGRP implements the reliable transport
   protocol to ensure ordered delivery and acknowledgment of any
   messages requiring reliable transmission. State variables such as a
   received sequence number, acknowledgment number, and transmission
   queues MUST be maintained on a per-neighbor basis.

The following sequence number rules must be met for the EIGRP
reliable transport protocol to work correctly:

o A sender of a packet includes its global sequence number in the
   sequence number field of the fixed header. The sequence number
   wraps around to one when the maximum value is exceeded
   (sequence number zero is reserved for unreliable transmission).
   The sender includes the receivers sequence number in the
   acknowledgment number field of the fixed header.

o Any packets that do not require acknowledgment must be sent
   with a sequence number of 0.

o Any packet that has an acknowledgment number of 0 indicates
   that sender is not expecting to explicitly acknowledge
   delivery. Otherwise, it is acknowledging a single packet.

o Packets that are network-layer multicast must contain
   acknowledgment number of 0.

When a router transmits a packet, it increments its sequence number
and marks the packet as requiring acknowledgment by all neighbors on
the interface for which the packet is sent. When individual
acknowledgments are unicast addressed by the receivers to the sender
with the acknowledgment number equal to the packets sequence number,
the sender SHALL clear the pending acknowledgment requirement for the
packet from the respective neighbor.

If the required acknowledgment is not received for the packet, it
MUST be retransmitted. Retransmissions will occur for a maximum of 5
seconds. This retransmission for each packet is tried 16 times,
after which, if there is no ACK, the neighbor relationship is reset
with the peer that didn't send the ACK.

The protocol has no explicit windowing support. A receiver will
acknowledge each packet individually and will drop packets that are
received out of order.

Implementation note: The exception to this occurs if a duplicate
packet is received, and the acknowledgment for the original packet
has been scheduled for transmission, but not yet sent. In this case,
EIGRP will not send an acknowledgment for the duplicate packet, and
the queued acknowledgment will acknowledge both the original and
duplicate packet.

Duplicate packets are also discarded upon receipt. Acknowledgments
are not accumulative. Therefore, an ACK with a non-zero sequence
number acknowledges a single packet.

   There are situations when multicast and unicast packets are
   transmitted close together on multi-access broadcast-capable
   networks. The reliable transport mechanism MUST ensure that all
   multicasts are transmitted in order and not mix the order among
   unicast and multicast packets. The reliable transport provides a
   mechanism to deliver multicast packets in order to some receivers
   quickly, while some receivers have not yet received all unicast or
   previously sent multicast packets. The SEQUENCE_TYPE TLV in HELLO
   packets achieves this. This will be explained in more detail in this
   section.

   Figure 5 illustrates the reliable transport protocol on point-to-
   point links. There are two scenarios that may occur: an UPDATE-
   initiated packet exchange or a QUERY-initiated packet exchange.

   This example will assume no packet loss.

```
Router A                        Router B

               An Example UPDATE Exchange
                                <---------------
                                UPDATE (multicast)
A receives packet               SEQ=100, ACK=0
                                Add packet to A's retransmit list
--------------->
ACK (unicast)
SEQ=0, ACK=100                  Receive ACK
Process UPDATE                  Delete packet from A's retransmit list

               An Example QUERY Exchange
                                <---------------
                                QUERY (multicast)
A receives packet               SEQ=101, ACK=0
Process QUERY                   Add packet to A's retransmit list

--------------->
REPLY (unicast)
SEQ=201, ACK=101                Process ACK
                                Delete packet from A's retransmit
list
                                Process REPLY packet
                                <---------------
                                ACK (unicast)
A receives packet               SEQ=0, ACK=201
```

        Figure 5: Reliable Transfer on Point-to-Point Links

The UPDATE exchange sequence requires UPDATE packets sent to be
delivered reliably. The UPDATE packet transmitted contains a
sequence number that is acknowledged by a receipt of an ACK packet.
If the UPDATE or the ACK packet is lost on the network, the UPDATE
packet will be retransmitted.

This example will assume there is heavy packet loss on a network.

```
Router A                          Router B
                                  <----------------
                                  UPDATE (multicast)
A receives packet                 SEQ=100, ACK=0
                                  Add packet to A's retransmit list

---------------->
ACK (unicast)
SEQ=0, ACK=100                    Receive ACK
Process UPDATE                    Delete packet from A's retransmit list

                                  <--/LOST/-------------
                                  UPDATE (multicast)
                                  SEQ=101, ACK=0
                                  Add packet to A's retransmit list

                                  Retransmit Timer Expires
                                  <----------------
                                  Retransmit UPDATE (unicast)
                                  SEQ=101, ACK=0
                                  Keep packet on A's retransmit list
---------------->
ACK (unicast)
SEQ=0, ACK=101                    Receive ACK
Process UPDATE                    Delete packet from A's retransmit list
```

           Figure 6: Reliable Transfer on Lossy Point-to-Point Links

   Reliable delivery on multi-access LANs works in a similar fashion to
   point-to-point links. The initial packet is always multicast and
   subsequent retransmissions are unicast addressed. The
   acknowledgments sent are always unicast addressed. Figure 7 shows an
   example with four routers on an Ethernet.

```
        Router B -----------+
                            |
        Router C -----------+----------- Router A
                            |
        Router D -----------+
```

```
                        An Example UPDATE Exchange
                              <----------------
                              A send UPDATE (multicast)
                              SEQ=100, ACK=0
                              Add packet to B's retransmit list
                              Add packet to C's retransmit list
                              Add packet to D's retransmit list
---------------->
B sends ACK (unicast)
SEQ=0, ACK=100                Receive ACK
Process UPDATE               Delete packet from B's retransmit list


---------------->
C sends ACK (unicast)
SEQ=0, ACK=100                Receive ACK
Process UPDATE               Delete packet from C's retransmit list


---------------->
D sends ACK (unicast)
SEQ=0, ACK=100                Receive ACK
Process UPDATE               Delete packet from D's retransmit list

                        An Example QUERY Exchange
                              <----------------
                              A sends UPDATE (multicast)
                              SEQ=101, ACK=0
                              Add packet to B's retransmit list
                              Add packet to C's retransmit list
                              Add packet to D's retransmit list

---------------->
B sends REPLY (unicast)      <----------------
SEQ=511, ACK=101             A sends ACK (unicast to B)
Process UPDATE               SEQ=0, ACK=511
                             Delete packet from B's retransmit list
---------------->
C sends REPLY (unicast)      <----------------
SEQ=200, ACK=101             A sends ACK (unicast to C)
Process UPDATE               SEQ=0, ACK=200
                             Delete packet from C's retransmit list

---------------->
D sends REPLY (unicast)      <----------------
SEQ=11, ACK=101             A sends ACK (unicast to D)
Process UPDATE               SEQ=0, ACK=11
                             Delete packet from D's retransmit list
```

                Figure 7: Reliable Transfer on Multi-Access Links

And finally, a situation where numerous multicast and unicast packets
are sent close together in a multi-access environment is illustrated
in Figure 8.

```
        Router B -----------+
                            |
        Router C -----------+----------- Router A
                            |
        Router D -----------+


                                 <----------------
                                 A sends UPDATE (multicast)
                                 SEQ=100, ACK=0
---------------/LOST/->          Add packet to B's retransmit list
B sends ACK (unicast)            Add packet to C's retransmit list
SEQ=0, ACK=100                   Add packet to D's retransmit list

--------------->
C sends ACK (unicast)
SEQ=0, ACK=100                   Delete packet from C's retransmit list

--------------->
D sends ACK (unicast)
SEQ=0, ACK=100                   Delete packet from D's retransmit list
                                 <----------------
                                 A sends HELLO (multicast)
                                 SEQ=0, ACK=0, SEQ_TLV listing B

B receives Hello, does not set CR-Mode
C receives Hello, sets CR-Mode
D receives Hello, sets CR-Mode


                                 <----------------
                                 A sends UPDATE (multicast)
                                 SEQ=101, ACK=0, CR-Flag=1
---------------/LOST/->          Add packet to B's retransmit list
B sends ACK (unicast)            Add packet to C's retransmit list
SEQ=0, ACK=100                   Add packet to D's retransmit list

B ignores UPDATE 101 because the CR-Flag
is set and it is not in CR-Mode

--------------->
C sends ACK (unicast)
SEQ=0, ACK=101
```

```
---------------->
D sends ACK (unicast)

SEQ=0, ACK=101
                                <----------------
                                A resends UPDATE (unicast to B)
                                SEQ=100, ACK=0
B packet duplicate

--------------->
B sends ACK (unicast)           A removes packet from retransmit list
SEQ=0, ACK=100
                                <----------------
                                A resends UPDATE (unicast to B)
                                SEQ=101, ACK=0

--------------->
B sends ACK (unicast)            A removes packet from retransmit list
SEQ=0, ACK=101
```

                 Figure 8: Reliable Transfer on Multi-Access Links
                           with Conditional Receive

   Initially, Router A sends a multicast addressed UPDATE packet on the
   LAN. B and C receive it and send acknowledgments. Router B receives
   the UPDATE, but the acknowledgment sent is lost on the network.
   Before the retransmission timer for Router B's packet expires, there
   is an event that causes a new multicast addressed UPDATE to be sent.

   Router A detects that there is at least one neighbor on the interface
   with a full queue. Therefore, it MUST signal that neighbor not to
   receive the next packet or it would receive the retransmitted packet
   out of order. If all neighbors on the interface have a full queue,
   then EIGRP should reschedule the transmission of the UPDATE once the
   queues are no longer full.

   Router A builds a HELLO packet with a SEQUENCE_TYPE TLV indicating
   all the neighbors that have full queues. In this case, the only
   neighbor address in the list is Router B. The HELLO packet is sent
   via multicast unreliably out the interface.

   Routers C and D process the SEQUENCE_TYPE TLV by looking for their
   own addresses in the list. If not found, they put themselves in CR-
   Mode.

   Router B does not find its address in the SEQUENCE TLV peer list, so
   it enters CR-Mode. Packets received by Router B with the CR-Flag
   MUST be discarded and not acknowledged.

Later, Router A will unicast transmit both packets 100 and 101
directly to Router B. Router B already has 100, so it discards and
acknowledges it.

Router B then accepts and acknowledges packet 101. Once an
acknowledgment is received, Router A can remove both packets from
Router B's transmission list.

5.2.1. Bandwidth on Low-Speed Links

By default, EIGRP limits itself to using no more than 50% of the
bandwidth reported by an interface when determining packet-pacing
intervals. If the bandwidth does not match the physical bandwidth
(the network architect may have put in an artificially low or high
bandwidth value to influence routing decisions), EIGRP may:

   1. Generate more traffic than the interface can handle, possibly
      causing drops, thereby impairing EIGRP performance.

   2. Generate a lot of EIGRP traffic that could result in little
      bandwidth remaining for user data. To control such
      transmissions, an interface-pacing timer is defined for the
      interfaces on which EIGRP is enabled. When a pacing timer
      expires, a packet is transmitted out on that interface.

5.3. Neighbor Discovery/Recovery

Neighbor Discovery/Recovery is the process that routers use to
dynamically learn of other routers on their directly attached
networks. Routers MUST also discover when their neighbors become
unreachable or inoperative. This process is achieved with low
overhead by periodically sending small HELLO packets. As long as any
packets are received from a neighbor, the router can determine that
neighbor is alive and functioning. Only after a neighbor router is
considered operational can the neighboring routers exchange routing
information.

5.3.1. Neighbor Hold Time

Each router keeps state information about adjacent neighbors. When
newly discovered neighbors are learned the address, interface, and
Hold Time of the neighbor is noted. When a neighbor sends a HELLO,
it advertises its Hold Time. The Hold Time is the amount of time a
router treats a neighbor as reachable and operational. In addition
to the HELLO packet, if any packet is received within the Hold Time
period, then the Hold Time period will be reset. When the Hold Time
expires, DUAL is informed of the topology change.

5.3.2. HELLO Packets

   When an EIGRP router is initialized, it will start sending HELLO
   packets out any interface on which EIGRP is enabled. HELLO packets,
   when used for neighbor discovery, are normally sent multicast
   addressed. The HELLO packet will include the configured EIGRP metric
   K-values. Two routers become neighbors only if the K-values are the
   same. This enforces that the metric usage is consistent throughout
   the Internet. Also included in the HELLO packet is a Hold Time
   value. This value indicates to all receivers the length of time in
   seconds that the neighbor is valid. The default Hold Time will be
   three times the HELLO interval. HELLO packets will be transmitted
   every 5 seconds (by default). There may be a configuration command
   that controls this value and therefore changes the Hold Time. HELLO
   packets are not transmitted reliably, so the sequence number should
   be set to 0.

5.3.3. UPDATE Packets

   A router detects a new neighbor by receiving a HELLO packet from a
   neighbor not presently known. To ensure unicast and multicast packet
   delivery, the detecting neighbor will send a unicast UPDATE packet to
   the new neighbor with no routing information (the NULL UPDATE
   packet). The initial NULL UPDATE packet sent MUST have the INIT-Flag
   set and contain no topology information.

   Implementation note: The NULL UPDATE packet is used to ensure
   bidirectional UNICAST packet delivery as the NULL UPDATE and the ACK
   are both sent unicast. Additional UPDATE packets cannot be sent
   until the initial NULL UPDATE packet is acknowledged.

   The INIT-Flag instructs the neighbor to advertise its routes, and it
   is also useful when a neighbor goes down and comes back up before the
   router detects it went down. In this case, the neighbor needs new
   routing information. The INIT-Flag informs the router to send it.

   Implementation note: When a router sends an UPDATE with the INIT-Flag
   set, and without the Restart (RS) flag set in the header, the
   receiving neighbor must also send an UPDATE with the INIT-Flag.
   Failure to do so will result in a Cisco device posting a "stuck in
   INIT state" error and subsequent discards.

5.3.4. Initialization Sequence

```
         Router A                          Router B
       (just booted)                    (up and running)

    (1)---------------->
        HELLO (multicast)         <---------------     (2)
        SEQ=0, ACK=0               HELLO (multicast)
                                   SEQ=0, ACK=0

                                  <---------------     (3)
                                   UPDATE (unicast)
                                   SEQ=10, ACK=0, INIT
    (4)---------------->           UPDATE 11 is queued
        UPDATE (unicast)
        SEQ=100, ACK=10, INIT     <---------------     (5)
                                   UPDATE (unicast)
                                   SEQ=11, ACK=100
                                   All UPDATES sent
    (6)-------------/lost/->
        ACK (unicast)
        SEQ=0, ACK=11
                                   (5 seconds later)
                                  <---------------     (7)
        Duplicate received,        UPDATE (unicast)
        packet discarded           SEQ=11, ACK=100
    (8)---------------->
        ACK (unicast)
        SEQ=0, ACK=11
```

                    Figure 9: Initialization Sequence

   (1) Router A sends a multicast HELLO and Router B discovers it.

   (2) Router B sends an expedited HELLO and starts the process of
       sending its topology table to Router A. In addition, Router B
       sends the NULL UPDATE packet with the INIT-Flag. The second
       packet is queued, but it cannot be sent until the first is
       acknowledged.

   (3) Router A receives the first UPDATE packet and processes it as a
       DUAL event. If the UPDATE contains topology information, the
       packet will be processed and stored in a topology table. Router
       B sends its first and only UPDATE packet with an accompanied ACK.

   (4) Router B receives UPDATE packet 100 from Router A. Router B can
       dequeue packet 10 from A's transmission list since the UPDATE
       acknowledged 10. It can now send UPDATE packet 11 and with an
       acknowledgment of Router A's UPDATE.

   (5) Router A receives the last UPDATE packet from Router B and
       acknowledges it. The acknowledgment gets lost.

   (6) Router B later retransmits the UPDATE packet to Router A.

   (7) Router A detects the duplicate and simply acknowledges the
       packet. Router B dequeues packet 11 from A's transmission list,
       and both routers are up and synchronized.

5.3.5. Neighbor Formation

   To prevent packets from being sent to a neighbor prior to verifying
   multicast and unicast packet delivery is reliable, a three-way
   handshake is utilized.

   During normal adjacency formation, multicast HELLOs cause the EIGRP
   process to place new neighbors into the neighbor table. Unicast
   packets are then used to exchange known routing information and
   complete the neighbor relationship (Section 5.2).

   To prevent EIGRP from sending sequenced packets to neighbors that
   fail to have bidirectional unicast/multicast, or one neighbor
   restarts while building the relationship, EIGRP MUST place the newly
   discovered neighbor in a "pending" state as follows:

      when Router A receives the first multicast HELLO from Router B, it
      places Router B in the pending state and transmits a unicast
      UPDATE containing no topology information and SHALL set the
      initialization bit. While Router B is in this state, A will send
      it neither a QUERY nor an UPDATE. When Router A receives the
      unicast acknowledgment from Router B, it will change the state
      from "pending" to "up".

5.3.6. QUERY Packets during Neighbor Formation

   As described above, during the initial formation of the neighbor
   relationship, EIGRP uses a form of three-way handshake to verify both
   unicast and multicast connectivity are working successfully. During
   this period of neighbor creation, the new neighbor is considered to
   be in the pending state, and it is not eligible to be included in the
   convergence process.

Because of this, any QUERY received by an EIGRP router would not
cause a QUERY to be sent to the new (and pending) neighbor. It would
perform the DUAL process without the new peer in the conversation.
To do this, when a router in the process of establishing a new
neighbor receives a QUERY from a fully established neighbor, it
performs the normal DUAL Feasible Successor check to determine
whether it needs to REPLY with a valid path or whether it needs to
enter the ACTIVE process on the prefix.

If it determines that it must go active, each fully established
neighbor that participates in the convergence process will be sent a
QUERY packet, and REPLY packets are expected from each. Any pending
neighbor will not be expected to REPLY and will not be sent a QUERY
directly. If it resides on an interface containing a mix of fully
established neighbors and pending neighbors, it might receive the
QUERY, but it will not be expected to REPLY to it.

5.4. Topology Table

The topology table is populated by the Protocol-Dependent Modules
(PDMs) (IPv4/IPv6), and it is acted upon by the DUAL finite state
machine. Associated with each entry are the destination address, a
list of neighbors that have advertised this destination, and the
metric associated with the destination. The metric is referred to as
the "CD".

The CD is the best-advertised RD from all neighbors, plus the link
cost between the receiving router and the neighbor.

The "RD" is the CD as advertised by the Feasible Successor for the
destination. In other words, the Computed Distance, when sent by a
neighbor, is referred to as the "Reported Distance" and is the metric
that the neighboring router uses to reach the destination (its CD as
described above).

If the router is advertising a destination route, it MUST be using
the route to forward packets; this is an important rule that distance
vector protocols MUST follow.

5.4.1. Route Management

Within the topology table, EIGRP has the notion of internal and
external routes. Internal routes MUST be preferred over external
routes, independent of the metric. In practical terms, if an
internal route is received, the diffusing computation will be run
considering only the internal routes. Only when no internal routes
for a given destination exist will EIGRP choose the successor from
the available external routes.

5.4.1.1. Internal Routes

   Internal routes are destinations that have been originated within the
   same EIGRP AS. Therefore, a directly attached network that is
   configured to run EIGRP is considered an internal route and is
   propagated with this information throughout the network topology.

   Internal routes are tagged with the following information:

      o Router ID of the EIGRP router that originated the route.
      o Configurable administrator tag.

5.4.1.2. External Routes

   External routes are destinations that have been learned from another
   source, such as a different routing protocol or static route. These
   routes are marked individually with the identity of their
   origination. External routes are tagged with the following
   information:

      o Router ID of the EIGRP router that redistributed the route.
      o AS number where the destination resides.
      o Configurable administrator tag.
      o Protocol ID of the external protocol.
      o Metric from the external protocol.
      o Bit flags for default routing.

   As an example, suppose there is an AS with three border routers: BR1,
   BR2, and BR3. A border router is one that runs more than one routing
   protocol. The AS uses EIGRP as the routing protocol. Two of the
   border routers, BR1 and BR2, also use Open Shortest Path First (OSPF)
   [10] and the other, BR3, also uses the Routing Information Protocol
   (RIP).

   Routes learned by one of the OSPF border routers, BR1, can be
   conditionally redistributed into EIGRP. This means that EIGRP
   running in BR1 advertises the OSPF routes within its own AS. When it
   does so, it advertises the route and tags it as an OSPF-learned route
   with a metric equal to the routing table metric of the OSPF route.
   The router-id is set to BR1. The EIGRP route propagates to the other
   border routers.

   Let's say that BR3, the RIP border router, also advertises the same
   destinations as BR1. Therefore, BR3, redistributes the RIP routes
   into the EIGRP AS. BR2, then, has enough information to determine
   the AS entry point for the route, the original routing protocol used,
   and the metric.

   Further, the network administrator could assign tag values to
   specific destinations when redistributing the route. BR2 can utilize
   any of this information to use the route or re-advertise it back out
   into OSPF.

   Using EIGRP route tagging can give a network administrator flexible
   policy controls and help customize routing. Route tagging is
   particularly useful in transit ASes where EIGRP would typically
   interact with an inter-domain routing protocol that implements global
   policies.

5.4.2. Split Horizon and Poison Reverse

   In some circumstances, EIGRP will suppress or poison QUERY and UPDATE
   information to prevent routing loops as changes propagate though the
   network.

   Within Cisco, the split horizon rule suggests: "Never advertise a
   route out of the interface through which it was learned". EIGRP
   implements this to mean, "if you have a successor route to a
   destination, never advertise the route out the interface on which it
   was learned".

   The poison reverse rule states: "A route learned through an interface
   will be advertised as unreachable through that same interface". As
   with the case of split horizon, EIGRP applies this rule only to
   interfaces it is using for reaching the destination. Routes learned
   though interfaces that EIGRP is NOT using to reach the destination
   may have the route advertised out those interfaces.

   In EIGRP, split horizon suppresses a QUERY, where as poison reverse
   advertises a destination as unreachable. This can occur for a
   destination under any of the following conditions:

      o two routers are in startup or restart mode
      o advertising a topology table change
      o sending a query

5.4.2.1. Startup Mode

   When two routers first become neighbors, they exchange topology
   tables during startup mode. For each destination a router receives
   during startup mode, it advertises the same destination back to its
   new neighbor with a maximum metric (Poison Route).

5.4.2.2. Advertising Topology Table Change

   If a router uses a neighbor as the successor for a given destination,
   it will send an UPDATE for the destination with a metric of infinity.

5.4.2.3. Sending a QUERY/UPDATE

   In most cases, EIGRP follows normal split-horizon rules. When a
   metric change is received from the successor via QUERY or UPDATE that
   causes the route to go ACTIVE, the router will send a QUERY to
   neighbors on all interfaces except the interface toward the
   successor.

   In other words, the router does not send the QUERY out of the inbound
   interface through which the information causing the route to go
   ACTIVE was received.

   An exception to this can occur if a router receives a QUERY from its
   successor while already reacting to an event that did not cause it to
   go ACTIVE, for example, a metric change from the successor that did
   not cause an ACTIVE transition, but was followed by the UPDATE/QUERY
   that does result the router to transition to ACTIVE.

5.5. EIGRP Metric Coefficients

   EIGRP allows for modification of the default composite metric
   calculation (see Section 5.6) through the use of coefficients (K-
   values). This adjustment allows for per-deployment tuning of network
   behavior. Setting K-values up to 254 scales the impact of the scalar
   metric on the final composite metric.

   EIGRP default coefficients have been carefully selected to provide
   optimal performance in most networks. The default K-values are as
   follows:

                 K1 == K3 == 1
                 K2 == K4 == K5 == 0
                 K6 == 0

   If K5 is equal to 0, then reliability quotient is defined to be 1.

5.5.1. Coefficients K1 and K2

   K1 is used to allow path selection to be based on the bandwidth
   available along the path. EIGRP can use one of two variations of
   Throughput-based path selection.

   o Maximum Theoretical Bandwidth: paths chosen based on the highest
      reported bandwidth

   o Network Throughput: paths chosen based on the highest "available"
      bandwidth adjusted by congestion-based effects (interface reported
      load)

   By default, EIGRP computes the Throughput using the maximum
   theoretical Throughput expressed in picoseconds per kilobyte of data
   sent. This inversion results in a larger number (more time)
   ultimately generating a worse metric.

   If K2 is used, the effect of congestion as a measure of load reported
   by the interface will be used to simulate the "available Throughput"
   by adjusting the maximum Throughput.

5.5.2. Coefficient K3

   K3 is used to allow delay or latency-based path selection. Latency
   and delay are similar terms that refer to the amount of time it takes
   a bit to be transmitted to an adjacent neighbor. EIGRP uses one-way-
   based values either provided by the interface or computed as a factor
   of the link s bandwidth.

5.5.3. Coefficients K4 and K5

   K4 and K5 are used to allow for path selection based on link quality
   and packet loss. Packet loss caused by network problems results in
   highly noticeable performance issues or Jitter with streaming
   technologies, voice over IP, online gaming and videoconferencing, and
   will affect all other network applications to one degree or another.

   Critical services should pass with less than 1% packet loss. Lower
   priority packet types might pass with less than 5% and then 10% for
   the lowest of priority of services. The final metric can be weighted
   based on the reported link quality.

   The handling of K5 is conditional. If K5 is equal to 0, then
   reliability quotient is defined to be 1.

5.5.4. Coefficient K6

   K6 has been introduced with Wide Metric support and is used to allow
   for Extended Attributes, which can be used to reflect in a higher
   aggregate metric than those having lower energy usage. Currently
   there are two Extended Attributes, Jitter and energy, defined in the
   scope of this document.

5.5.4.1. Jitter

   Use of Jitter-based Path Selection results in a path calculation with
   the lowest reported Jitter. Jitter is reported as the interval
   between the longest and shortest packet delivery and is expressed in
   microseconds. Higher values result in a higher aggregate metric when
   compared to those having lower Jitter calculations.

   Jitter is measured in microseconds and is accumulated along the path,
   with each hop using an averaged 3-second period to smooth out the
   metric change rate.

   Presently, EIGRP does not have the ability to measure Jitter, and, as
   such, the default value will be zero (0). Performance-based
   solutions such as PfR could be used to populate this field.

5.5.4.2. Energy

   Use of Energy-based Path Selection results in paths with the lowest
   energy usage being selected in a loop-free and deterministic manner.
   The amount of energy used is accumulative and has results in a higher
   aggregate metric than those having lower energy.

   Presently, EIGRP does not report energy usage, and as such the
   default value will be zero (0).

5.6. EIGRP Metric Calculations

5.6.1. Classic Metrics

   The composite metric is based on bandwidth, delay, load, and
   reliability. MTU is not an attribute for calculating the composite
   metric, but carried in the vector metrics.

   One of the original goals of EIGRP was to offer and enhance routing
   solutions for IGRP. To achieve this, EIGRP used the same composite
   metric as IGRP, with the terms multiplied by 256 to change the metric
   from 24 bits to 32 bits.

5.6.1.1. Classic Composite Formulation

   EIGRP calculates the composite metric with the following formula:

   metric = 256 * ({(K1*BW) + [(K2*BW)/(256-LOAD)] + (K3*DELAY)} *
           (K5/(REL+K4)))

   In this formula, Bandwidth (BW) is the lowest interface bandwidth
   along the path, and delay (DELAY) is the sum of all outbound
   interface delays along the path. Load (LOAD) and reliability (REL)
   values are expressed percentages with a value of 1 to 255.

   Implementation note: Cisco IOS routers display reliability as a
   fraction of 255. That is, 255/255 is 100% reliability or a perfectly
   stable link; a value of 229/255 represents a 90% reliable link. Load
   is a value between 1 and 255. A load of 255/255 indicates a
   completely saturated link. A load of 127/255 represents a 50%
   saturated link. These values are not dynamically measured; they are
   only measured at the time a link changes.

   Bandwidth is the inverse minimum bandwidth (in kbps) of the path in
   bits per second scaled by a factor of 10^7. The formula for
   bandwidth is as follows:

                    $(10^7)/BWmin$

   Implementation note: When converting the real bandwidth to the
   composite bandwidth, truncate before applying the scaling factor.
   When converting the composite bandwidth to the real bandwidth, apply
   the scaling factor before the division and only then truncate.

   The delay is the sum of the outgoing interface delay (in tens of
   microseconds) to the destination. A delay set to it maximum value
   (hexadecimal 0xFFFFFFFF) indicates that the network is unreachable.
   The formula for delay is as follows:

                    [sum of delays]

   The default composite metric, adjusted for scaling factors, for EIGRP
   is:

           metric = 256 * { [(10^7)/ BWmin] + [sum of delays]}

   Minimum Bandwidth (BWmin) is represented in kbps, and the "sum of
   delays" is represented in tens of microseconds. The bandwidth and
   delay for an Ethernet interface are 10 Mbps and 1 ms, respectively.

The calculated EIGRP bandwidth (BW) metric is then:

$$256 * (10^7)/BW = 256 * \{(10^7)/10{,}000\}$$
$$= 256 * 1000$$
$$= 256{,}000$$

And the calculated EIGRP delay metric is then:

$$256 * \text{sum of delay} = 256 * 100 * 10 \text{ microseconds}$$
$$= 25{,}600 \text{ (in tens of microseconds)}$$

5.6.1.2. Cisco Interface Delay Compatibility

   For compatibility with Cisco products, the following table shows the
   times in nanoseconds EIGRP uses for bandwidth and delay.

Bandwidth (kbps)	Classic Delay	Wide Metrics Delay	Interface Type
9	500000000	500000000	Tunnel
56	20000000	20000000	56 kbps
64	20000000	20000000	DS0
1544	20000000	20000000	T1
2048	20000000	20000000	E1
10000	1000000	1000000	Ethernet
16000	630000	630000	TokRing16
45045	20000000	20000000	HSSI
100000	100000	100000	FDDI
100000	100000	100000	FastEthernet
155000	100000	100000	ATM 155 Mbps
1000000	10000	10000	GigaEthernet
2000000	10000	5000	2 Gig
5000000	10000	2000	5 Gig
10000000	10000	1000	10 Gig
20000000	10000	500	20 Gig
50000000	10000	200	50 Gig
100000000	10000	100	100 Gig
200000000	10000	50	200 Gig
500000000	10000	20	500 Gig

5.6.2. Wide Metrics

   To enable EIGRP to perform the path selection for interfaces with
   high bandwidths, both the EIGRP packet and composite metric formula
   have been modified. This change allows EIGRP to choose paths based
   on the computed time (measured in picoseconds) information takes to
   travel though the links.

5.6.2.1. Wide Metric Vectors

   EIGRP uses five "vector metrics": minimum Throughput, latency, load,
   reliability, and MTU. These values are calculated from destination
   to source as follows:

                o Throughput   - Minimum value
                o Latency   - accumulative
                o Load   - maximum
                o Reliability   - minimum
                o MTU   - minimum
                o Hop count   - accumulative

   There are two additional values: Jitter and energy. These two values
   are accumulated from destination to source:

                o Jitter - accumulative
                o Energy - accumulative

   These Extended Attributes, as well as any future ones, will be
   controlled via K6. If K6 is non-zero, these will be additive to the
   path's composite metric. Higher Jitter or energy usage will result
   in paths that are worse than those that either do not monitor these
   attributes or that have lower values.

   EIGRP will not send these attributes if the router does not provide
   them. If the attributes are received, then EIGRP will use them in
   the metric calculation (based on K6) and will forward them with those
   routers values assumed to be "zero" and the accumulative values are
   forwarded unchanged.

   The use of the vector metrics allows EIGRP to compute paths based on
   any of four (bandwidth, delay, reliability, and load) path selection
   schemes. The schemes are distinguished based on the choice of the
   key-measured network performance metric.

   Of these vector metric components, by default, only minimum
   Throughput and latency are traditionally used to compute the best
   path. Unlike most metrics, minimum Throughput is set to the minimum
   value of the entire path, and it does not reflect how many hops or
   low Throughput links are in the path, nor does it reflect the
   availability of parallel links. Latency is calculated based on one-
   way delays and is a cumulative value, which increases with each
   segment in the path.

   Network Designer note: When trying to manually influence EIGRP path
   selection though interface bandwidth/delay configuration, the
   modification of bandwidth is discouraged for following reasons:

The change will only affect the path selection if the configured
value is the lowest bandwidth over the entire path. Changing the
bandwidth can have impact beyond affecting the EIGRP metrics. For
example, Quality of Service (QoS) also looks at the bandwidth on an
interface.

EIGRP throttles its packet transmissions so it will only use 50% of
the configured bandwidth. Lowering the bandwidth can cause EIGRP to
starve an adjacency, causing slow or failed convergence and control-
plane operation.

Changing the delay does not impact other protocols, nor does it cause
EIGRP to throttle back; changing the delay configured on a link only
impacts metric calculation.

5.6.2.2. Wide Metric Conversion Constants

EIGRP uses a number of defined constants for conversion and
calculation of metric values. These numbers are provided here for
reference

```
        EIGRP_BANDWIDTH                       10,000,000
        EIGRP_DELAY_PICO                       1,000,000
        EIGRP_INACCESSIBLE      0xFFFFFFFFFFFFFFFFLL
        EIGRP_MAX_HOPS                               100
        EIGRP_CLASSIC_SCALE                          256
        EIGRP_WIDE_SCALE                           65536
```

When computing the metric using the above units, all capacity
information will be normalized to kilobytes and picoseconds before
being used. For example, delay is expressed in microseconds per
kilobyte, and would be converted to kilobytes per second; likewise,
energy would be expressed in power per kilobytes per second of usage.

5.6.2.3. Throughput Calculation

The formula for the conversion for Max-Throughput value directly from
the interface without consideration of congestion-based effects is as
follows:

```
                         (EIGRP_BANDWIDTH * EIGRP_WIDE_SCALE)
        Max-Throughput = K1 *   ------------------------------------
                               Interface Bandwidth (kbps)
```

If K2 is used, the effect of congestion as a measure of load reported
by the interface will be used to simulate the "available Throughput"
by adjusting the maximum Throughput according to the formula:

$$\text{Net-Throughput} = \text{Max-Throughput} + \frac{\text{K2} * \text{Max-Throughput}}{256 - \text{Load}}$$

K2 has the greatest effect on the metric occurs when the load
increases beyond 90%.

5.6.2.4. Latency Calculation

Transmission times derived from physical interfaces MUST be n units
of picoseconds, converted to picoseconds prior to being exchanged
between neighbors, or used in the composite metric determination.

This includes delay values present in configuration-based commands
(i.e., interface delay, redistribute, default-metric, route-map,
etc.).

The delay value is then converted to a "latency" using the formula:

$$\text{Latency} = \text{K3} * \frac{\text{Delay} * \text{EIGRP\_WIDE\_SCALE}}{\text{EIGRP\_DELAY\_PICO}}$$

5.6.2.5. Composite Calculation

$$\text{metric} = [(\text{K1}*\text{Net-Throughput}) + \text{Latency})+(\text{K6}*\text{ExtAttr})] * \frac{\text{K5}}{\text{K4+Rel}}$$

By default, the path selection scheme used by EIGRP is a combination
of Throughput and Latency where the selection is a product of total
latency and minimum Throughput of all links along the path:

    metric = (K1 * min(Throughput)) + (K3 * sum(Latency)) }

6. EIGRP Packet Formats

6.1. Protocol Number

The IPv6 and IPv4 protocol identifier number spaces are common and
will both use protocol identifier 88 [8] [9].

EIGRP IPv4 will transmit HELLO packets using either the unicast
destination of a neighbor or using a multicast host group address [7]
with a source address EIGRP IPv4 multicast address [13].

EIGRP IPv6 will transmit HELLO packets with a source address being
the link-local address of the transmitting interface. Multicast
HELLO packets will have a destination address of EIGRP IPv6 multicast
address [14]. Unicast packets directed to a specific neighbor will
contain the destination link-local address of the neighbor.

There is no requirement that two EIGRP IPv6 neighbors share a common
prefix on their connecting interface. EIGRP IPv6 will check that a
received HELLO contains a valid IPv6 link-local source address.
Other HELLO processing will follow common EIGRP checks, including
matching AS number and matching K-values.

6.2. Protocol Assignment Encoding

The External Protocol field is an informational assignment to
identify the originating routing protocol that this route was learned
by. The following values are assigned:

```
      Protocols          Value
      IGRP                 1
      EIGRP                2
      Static               3
      RIP                  4
      HELLO                5
      OSPF                 6
      ISIS                 7
      EGP                  8
      BGP                  9
      IDRP                10
      Connected           11
```

6.3. Destination Assignment Encoding

Destinations types are encoded according to the IANA address family
number assignments. Currently only the following types are used:

```
      AFI Description            AFI Number
      -------------------------------------
      IP (IP version 4)               1
      IP6 (IP version 6)              2
      EIGRP Common Service Family   16384
      EIGRP IPv4 Service Family     16385
      EIGRP IPv6 Service Family     16386
```

6.4. EIGRP Communities Attribute

   EIGRP supports communities similar to the BGP Extended Communities
   RFC 4360 [4] extended type with Type field composed of 2 octets and
   Value field composed of 6 octets. Each Community is encoded as an
   8-octet quantity, as follows:

          - Type field: 2 octets
          - Value field: Remaining octets

```
 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 high     | Type low      |                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+             Value              |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
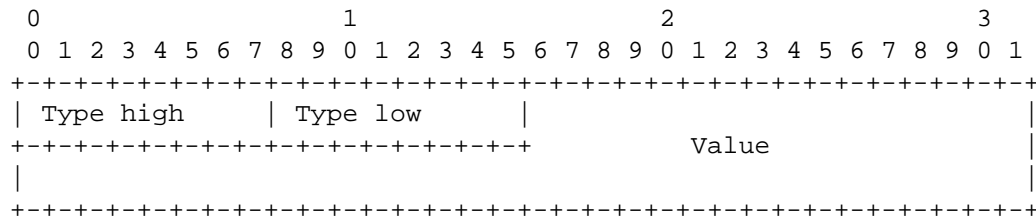
   In addition to well-known communities supported by BGP (such as Site
   of Origin), EIGRP defines a number of additional Community values in
   the "Experimental Use" [5] range as follows:

     Type high: 0x88
     Type low:

        Value   Name   Description
        ----
          00   EXTCOMM_EIGRP   EIGRP route information appended
          01   EXTCOMM_DAD   Data: AS + Delay
          02   EXTCOMM_VRHB   Vector: Reliability + Hop + BW
          03   EXTCOMM_SRLM   System: Reserve + Load + MTU
          04   EXTCOMM_SAR   System: Remote AS + Remote ID
          05   EXTCOMM_RPM   Remote: Protocol + Metric
          06   EXTCOMM_VRR   Vecmet: Rsvd + RouterID

6.5. EIGRP Packet Header

   The basic EIGRP packet payload format is identical for both IPv4 and
   IPv6, although there are some protocol-specific variations. Packets
   consist of a header, followed by a set of variable-length fields
   consisting of Type/Length/Value (TLV) triplets.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Header Version | Opcode        |            Checksum           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Flags                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Acknowledgment Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Virtual Router ID             |   Autonomous System Number    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
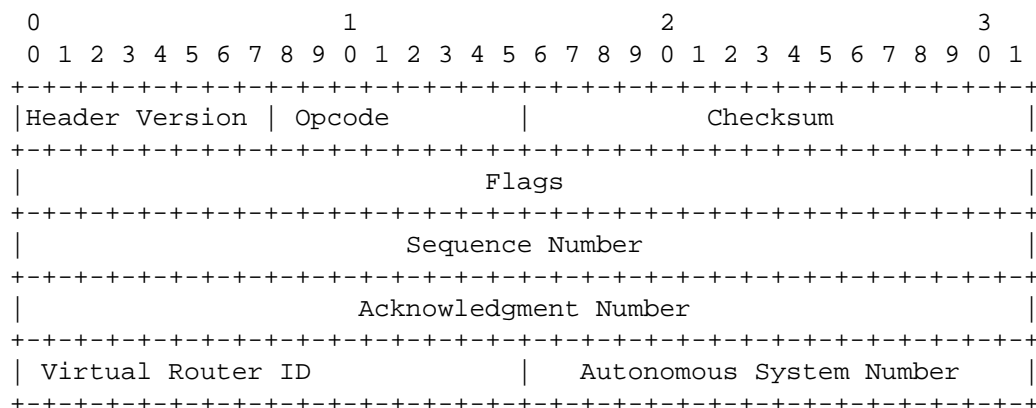
   Header Version: EIGRP Packet Header Format version. Current Version
      is 2. This field is not the same as the TLV Version field.

   Opcode: Indicates the type of the message. It will be one of the
      following values:

```
        EIGRP_OPC_UPDATE             1
        EIGRP_OPC_REQUEST            2
        EIGRP_OPC_QUERY              3
        EIGRP_OPC_REPLY              4
        EIGRP_OPC_HELLO              5
        Reserved                     6     (EIGRP_OPC_IPXSAP)
        Reserved                     7     (EIGRP_OPC_PROBE)
        Reserved                     8     (EIGRP_OPC_ACK)
        Reserved                     9
        EIGRP_OPC_SIAQUERY          10
        EIGRP_OPC_SIAREPLY          11
```

   Checksum: Each packet will include a checksum for the entire contents
      of the packet. The checksum will be the standard ones' complement
      of the ones' complement sum. For purposes of computing the
      checksum, the value of the checksum field is zero. The packet is
      discarded if the packet checksum fails.

   Flags: Defines special handling of the packet. There are currently
      four defined flag bits.

INIT-Flag (0x01): This bit is set in the initial UPDATE sent to a
   newly discovered neighbor. It instructs the neighbor to advertise
   its full set of routes.

CR-Flag (0x02): This bit indicates that receivers should only accept
   the packet if they are in Conditionally Received mode. A router
   enters Conditionally Received mode when it receives and processes
   a HELLO packet with a SEQUENCE TLV present.

RS-Flag (0x04): The Restart flag is set in the HELLO and the UPDATE
   packets during the restart period. The router looks at the RS-
   Flag to detect if a neighbor is restarting. From the restarting
   routers perspective, if a neighboring router detects the RS-Flag
   set, it will maintain the adjacency, and will set the RS-Flag in
   its UPDATE packet to indicated it is doing a soft restart.

EOT-Flag (0x08): The End-of-Table flag marks the end of the startup
   process with a neighbor. If the flag is set, it indicates the
   neighbor has completed sending all UPDATEs. At this point, the
   router will remove any stale routes learned from the neighbor
   prior to the restart event. A stale route is any route that
   existed before the restart and was not refreshed by the neighbor
   via and UPDATE.

Sequence Number: Each packet that is transmitted will have a 32-bit
   sequence number that is unique with respect to a sending router.
   A value of 0 means that an acknowledgment is not required.

Acknowledgment Number: The 32-bit sequence number that is being
   acknowledged with respect to the receiver of the packet. If the
   value is 0, there is no acknowledgment present. A non-zero value
   can only be present in unicast-addressed packets. A HELLO packet
   with a non-zero ACK field should be decoded as an ACK packet
   rather than a HELLO packet.

Virtual Router Identifier (VRID): A 16-bit number that identifies the
   virtual router with which this packet is associated. Packets
   received with an unknown, or unsupported, value will be discarded.

```
        Value Range          Usage
          0x0000               Unicast Address Family
          0x0001               Multicast Address Family
          0x0002-0x7FFF        Reserved
          0x8000               Unicast Service Family
          0x8001-0xFFFF        Reserved
```

   Autonomous System Number: 16-bit unsigned number of the sending
      system. This field is indirectly used as an authentication value.
      That is, a router that receives and accepts a packet from a
      neighbor must have the same AS number or the packet is ignored.
      The range of valid AS numbers is 1 through 65,535.

6.6. EIGRP TLV Encoding Format

   The contents of each packet can contain a variable number of fields.
   Each field will be tagged and include a length field. This allows
   for newer versions of software to add capabilities and coexist with
   old versions of software in the same configuration. Fields that are
   tagged and not recognized can be skipped over. Another advantage of
   this encoding scheme is that it allows multiple network-layer
   protocols to carry independent information. Therefore, if it is
   later decided to implement a single "integrated" protocol, this can
   be done.

   The format of a {type, length, value} (TLV) is encoded as follows:

```
 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 high     | Type low      |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Value (variable length)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
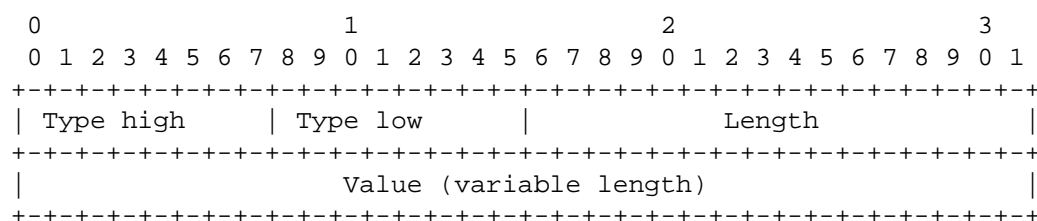
   The type values are the ones defined below. The length value
   specifies the length in octets of the type, length, and value fields.
   TLVs can appear in a packet in any order, and there are no
   interdependencies among them.

   Malformed TLVs contained in EIGRP messages are handled by silently
   discarding the containing message. A TLV is malformed if the TLV
   Length is invalid or if the TLV extends beyond the end of the
   containing message.

6.6.1. Type Field Encoding

   The type field is structured as follows: Type High: 1 octet that
   defines the protocol classification:

                Protocol   ID   VERSION
                General   0x00   1.2
                IPv4   0x01   1.2
                IPv6   0x04   1.2
                SAF   0x05   3.0
                Multiprotocol   0x06   2.0

   Type Low: 1 octet that defines the TLV Opcode; see TLV Definitions in
      Section 3.

6.6.2. Length Field Encoding

   The Length field is a 2-octet unsigned number, which indicates the
   length of the TLV. The value includes the Type and Length fields.

6.6.3. Value Field Encoding

   The Value field is a multi-octet field containing the payload for the
   TLV.

6.7. EIGRP Generic TLV Definitions

                                   Ver 1.2   Ver 2.0
      PARAMETER_TYPE   0x0001   0x0001
      AUTHENTICATION_TYPE   0x0002   0x0002
      SEQUENCE_TYPE   0x0003   0x0003
      SOFTWARE_VERSION_TYPE   0x0004   0x0004
      MULTICAST_SEQUENCE_TYPE   0x0005   0x0005
      PEER_INFORMATION_TYPE   0x0006   0x0006
      PEER_TERMINATION_TYPE   0x0007   0x0007
      PEER_TID_LIST_TYPE   ---   0x0008

6.7.1. 0x0001 - PARAMETER_TYPE

   This TLV is used in HELLO packets to convey the EIGRP metric
   coefficient values: noted as "K-values" as well as the Hold Time
   values. This TLV is also used in an initial UPDATE packet when a
   neighbor is discovered.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            0x0001             |             0x000C            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      K1       |      K2       |      K3       |      K4       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      K5       |      K6       |           Hold Time           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
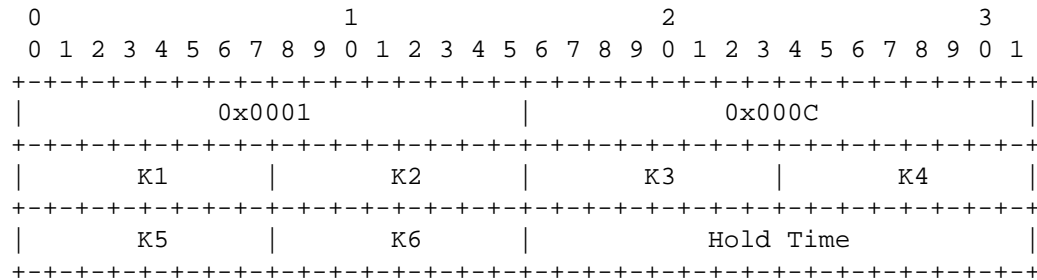
   K-values: The K-values associated with the EIGRP composite metric
      equation. The default values for weights are:

                     K1 - 1
                     K2 - 0
                     K3 - 1
                     K4 - 0
                     K5 - 0
                     K6 - 0

   Hold Time: The amount of time in seconds that a receiving router
      should consider the sending neighbor valid. A valid neighbor is
      one that is able to forward packets and participates in EIGRP. A
      router that considers a neighbor valid will store all routing
      information advertised by the neighbor.

6.7.2. 0x0002 - AUTHENTICATION_TYPE

   This TLV may be used in any EIGRP packet and conveys the
   authentication type and data used. Routers receiving a mismatch in
   authentication shall discard the packet.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             0x0002            |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   Auth Type   |  Auth Length  |      Auth Data (Variable)     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
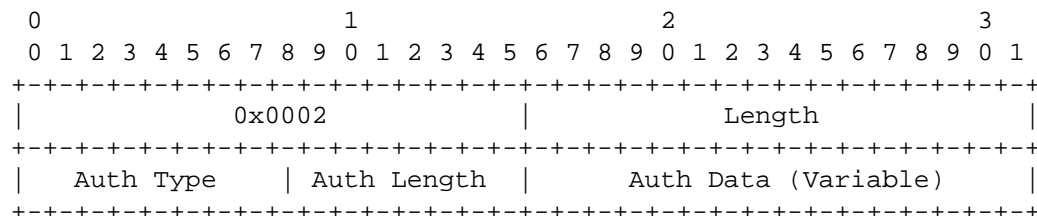
   Authentication Type: The type of authentication used.

   Authentication Length: The length, measured in octets, of the
      individual authentication.

   Authentication Data: Variable-length field reflected by "Auth
      Length", which is dependent on the type of authentication used.
      Multiple authentication types can be present in a single
      AUTHENTICATION_TYPE TLV.

6.7.2.1. 0x02 - MD5 Authentication Type

   MD5 Authentication will use Auth Type code 0x02, and the Auth Data
   will be the MD5 Hash value.

6.7.2.2. 0x03 - SHA2 Authentication Type

   SHA2-256 Authentication will use Type code 0x03, and the Auth Data
   will be the 256-bit SHA2 [6] Hash value.

6.7.3. 0x0003 - SEQUENCE_TYPE

   This TLV is used for a sender to tell receivers to not accept packets
   with the CR-Flag set. This is used to order multicast and unicast
   addressed packets.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            0x0003             |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Address Length |              Protocol Address                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
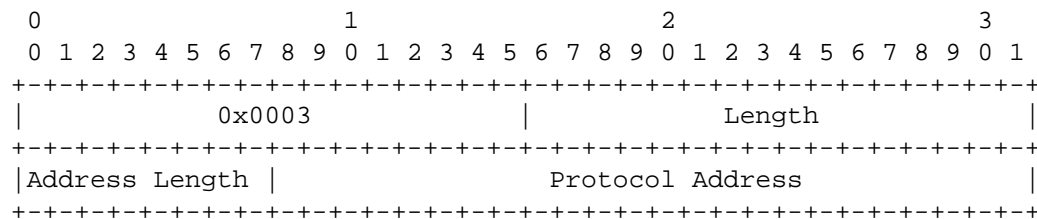
   The Address Length and Protocol Address will be repeated one or more
   times based on the Length field.

   Address Length: Number of octets for the address that follows. For
      IPv4, the value is 4. For IPv6, it is 16. For AppleTalk, the
      value is 4; for Novell IPX, the value is 10 (both are no longer in
      use).

   Protocol Address: Neighbor address on interface in which the HELLO
      with SEQUENCE TLV is sent. Each address listed in the HELLO
      packet is a neighbor that should not enter Conditionally Received
      mode.

6.7.4. 0x0004 - SOFTWARE_VERSION_TYPE

```
        Field                     Length
        Vender OS major version      1
        Vender OS minor version      1
        EIGRP major revision         1
        EIGRP minor revision         1
```

The EIGRP TLV Version fields are used to determine TLV format
versions. Routers using Version 1.2 TLVs do not understand Version
2.0 TLVs, therefore Version 2.0 routers must send the packet with
both TLV formats in a mixed network.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            0x0004             |             0x000C            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Vendor Major V.|Vendor Minor V.| EIGRP Major V.| EIGRP Minor V.|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.7.5. 0x0005 - MULTICAST_SEQUENCE_TYPE

The next multicast SEQUENCE TLV.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            0x0005             |             0x0008            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.7.6. 0x0006 - PEER_INFORMATION_TYPE

This TLV is reserved, and not part of this document.

6.7.7. 0x0007 - PEER_ TERMINATION_TYPE

   This TLV is used in HELLO packets to notify the list of neighbor(s)
   the router has reset the adjacency. This TLV is used in HELLO
   packets to notify the list of neighbors that the router has reset the
   adjacency. This is used anytime a router needs to reset an
   adjacency, or signal an adjacency it is going down.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             0x0007            |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Address List (variable)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Implementation note: Older Cisco routers implement this using the
   "Parameters TLV" with all K-values set to 255 (except K6).

6.7.8. 0x0008 - TID_LIST_TYPE

   List of sub-topology identifiers, including the Base Topology,
   supported by the router.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             0x0008            |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Topology Identification List (variable)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   If this information changes from the last state, it means either a
   new topology was added or an existing topology was removed. This TLV
   is ignored until the three-way handshake has finished

   When the TID list is received, it compares the list to the previous
   list sent. If a TID is found that does not previously exist, the TID
   is added to the neighbor's topology list, and the existing sub-
   topology is sent to the peer.

   If a TID that was in a previous list is not found, the TID is removed
   from the neighbor's topology list and all routes learned though that
   neighbor for that sub-topology are removed from the topology table.

6.8. Classic Route Information TLV Types

6.8.1. Classic Flag Field Encoding

   EIGRP transports a number of flags with in the TLVs to indicate
   addition route state information. These bits are defined as follows:

   Flags Field

   Source Withdraw (Bit 0) - Indicates if the router that is the
   original source of the destination is withdrawing the route from the
   network or if the destination is lost due as a result of a network
   failure.

   Candidate Default (CD) (Bit 1) - Set to indicate the destination
   should be regarded as a candidate for the default route. An EIGRP
   default route is selected from all the advertised candidate default
   routes with the smallest metric.

   ACTIVE (Bit 2) - Indicates if the route is in the ACTIVE State.

6.8.2. Classic Metric Encoding

   The handling of bandwidth and delay for Classic TLVs is encoded in
   the packet "scaled" form relative to how they are represented on the
   physical link.
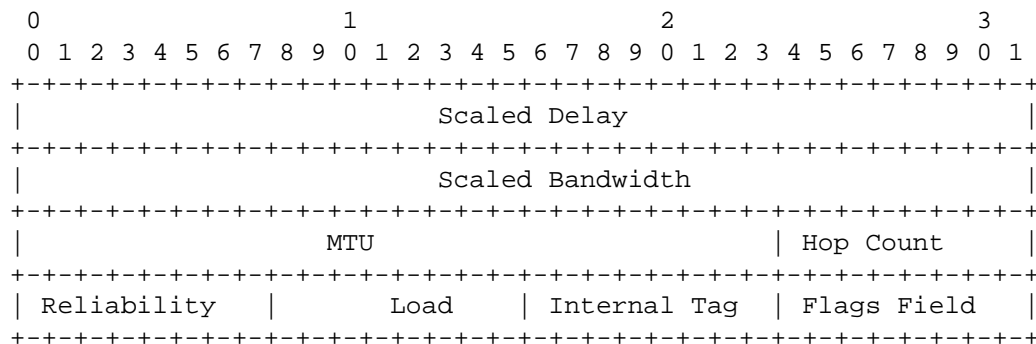
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Scaled Delay                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Scaled Bandwidth                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 MTU                        |    Hop Count     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Reliability   |      Load     |  Internal Tag | Flags Field   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Scaled Delay: An administrative parameter assigned statically on a
      per-interface-type basis to represent the time it takes along an
      unloaded path. This is expressed in units of tens of microseconds
      divvied by 256. A delay of 0xFFFFFFFF indicates an unreachable
      route.

   Scaled Bandwidth: The path bandwidth measured in bits per second. In
      units of 2,560,000,000/kbps.

MTU: The minimum MTU size for the path to the destination.

Hop Count: The number of router traversals to the destination.

Reliability: The current error rate for the path, measured as an
   error percentage. A value of 255 indicates 100% reliability

Load: The load utilization of the path to the destination, measured
   as a percentage. A value of 255 indicates 100% load.

Internal-Tag: A tag assigned by the network administrator that is
   untouched by EIGRP. This allows a network administrator to filter
   routes in other EIGRP border routers based on this value.

Flags Field: See Section 6.8.1.

6.8.3. Classic Exterior Encoding

   Additional routing information so provided for destinations outside
   of the EIGRP AS as follows:
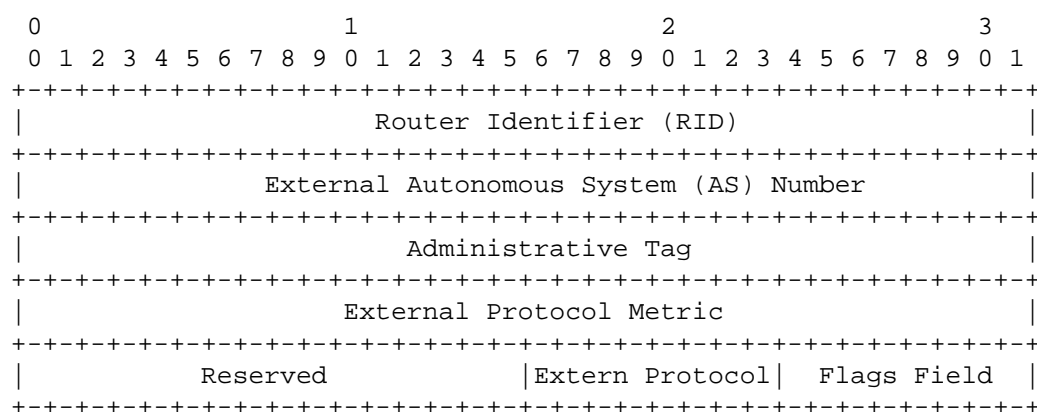
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Router Identifier (RID)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               External Autonomous System (AS) Number          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Administrative Tag                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      External Protocol Metric                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Reserved              |Extern Protocol|  Flags Field |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Router Identifier (RID): A 32-bit number provided by the router
   sourcing the information to uniquely identify it as the source.

External Autonomous System (AS) Number: A 32-bit number indicating
   the external AS of which the sending router is a member. If the
   source protocol is EIGRP, this field will be the [VRID, AS] pair.
   If the external protocol does not have an AS, other information
   can be used (for example, Cisco uses process-id for OSPF).

Administrative Tag: A tag assigned by the network administrator that
   is untouched by EIGRP. This allows a network administrator to
   filter routes in other EIGRP border routers based on this value.

   External Protocol Metric: 32-bit value of the composite metric that
      resides in the routing table as learned by the foreign protocol.
      If the External Protocol is IGRP or another EIGRP routing process,
      the value can optionally be the composite metric or 0, and the
      metric information is stored in the metric section.

   External Protocol: Contains an enumerated value defined in Section
      6.2 to identify the routing protocol (external protocol)
      redistributing the route.

   Flags Field: See Section 6.8.1

6.8.4. Classic Destination Encoding

   EIGRP carries destination in a compressed form, where the number of
   bits significant in the variable-length address field are indicated
   in a counter.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Subnet Mask   |     Destination Address (variable length)     |
| Bit Count     |          ((Bit Count - 1) / 8) + 1            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Subnet Mask Bit Count: 8-bit value used to indicate the number of
      bits in the subnet mask. A value of 0 indicates the default
      network, and no address is present.

   Destination Address: A variable-length field used to carry the
      destination address. The length is determined by the number of
      consecutive bits in the destination address. The formula to
      calculate the length is address-family dependent:

      IPv4: ((Bit Count - 1) / 8) + 1
      IPv6: (Bit Count == 128) ? 16 : ((x / 8) + 1)

6.8.5. IPv4-Specific TLVs

```
      INTERNAL_TYPE        0x0102
      EXTERNAL_TYPE        0x0103
      COMMUNITY_TYPE       0x0104
```

6.8.5.1. IPv4 INTERNAL_TYPE

   This TLV conveys IPv4 destination and associated metric information
   for IPv4 networks. Routes advertised in this TLV are network
   interfaces that EIGRP is configured on as well as networks that are
   learned via other routers running EIGRP.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     0x01      |      0x02      |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                 Next-Hop Forwarding Address                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Vector Metric Section (see Section 6.8.2)          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
   |                     Destination Section                      |
   |                 IPv4 Address (variable length)               |
   |                       (see Section 6.8.4)                     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
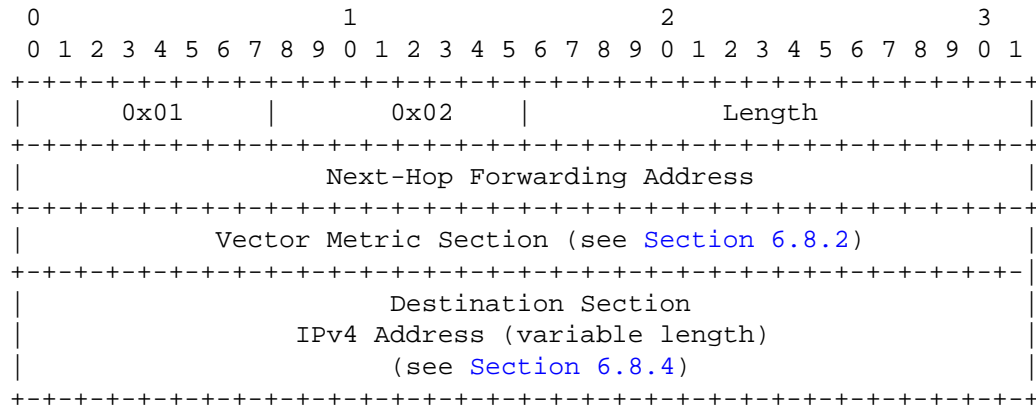
   Next-Hop Forwarding Address: IPv4 address represented by four 8-bit
      values (total 4 octets). If the value is zero (0), the IPv4
      address from the received IPv4 header is used as the next hop for
      the route. Otherwise, the specified IPv4 address will be used.

   Vector Metric Section: The vector metrics for destinations contained
      in this TLV. See the description of "metric encoding" in Section
      6.8.2.

   Destination Section: The network/subnet/host destination address
      being requested. See the description of "destination" in Section
      6.8.4.

6.8.5.2. IPv4 EXTERNAL_TYPE

   This TLV conveys IPv4 destination and metric information for routes
   learned by other routing protocols that EIGRP injects into the AS.
   Available with this information is the identity of the routing
   protocol that created the route, the external metric, the AS number,
   an indicator if it should be marked as part of the EIGRP AS, and a
   network-administrator tag used for route filtering at EIGRP AS
   boundaries.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0x01     |      0x03     |              Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Next-Hop Forwarding Address                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Exterior Section (see Section 6.8.3)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Vector Metric Section (see Section 6.8.2)          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|                      Destination Section                     |
|                 IPv4 Address (variable length)               |
|                       (see Section 6.8.4)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next-Hop Forwarding Address: IPv4 address represented by four 8-bit
   values (total 4 octets). If the value is zero (0), the IPv4
   address from the received IPv4 header is used as the next hop for
   the route. Otherwise, the specified IPv4 address will be used.

Exterior Section: Additional routing information provided for a
   destination that is outside of the AS and that has been
   redistributed into the EIGRP. See the description of "exterior
   encoding" in Section 6.8.3.

Vector Metric Section: Vector metrics for destinations contained in
   this TLV. See the description of "metric encoding" in Section
   6.8.2.

Destination Section: The network/subnet/host destination address
   being requested. See the description of "destination" in Section
   6.8.4.

6.8.5.3. IPv4 COMMUNITY_TYPE

   This TLV is used to provide community tags for specific IPv4
   destinations.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     0x01      |     0x04      |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       IPv4 Destination                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Reserved           |       Community Length        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Community List                         |
   |                       (variable length)                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
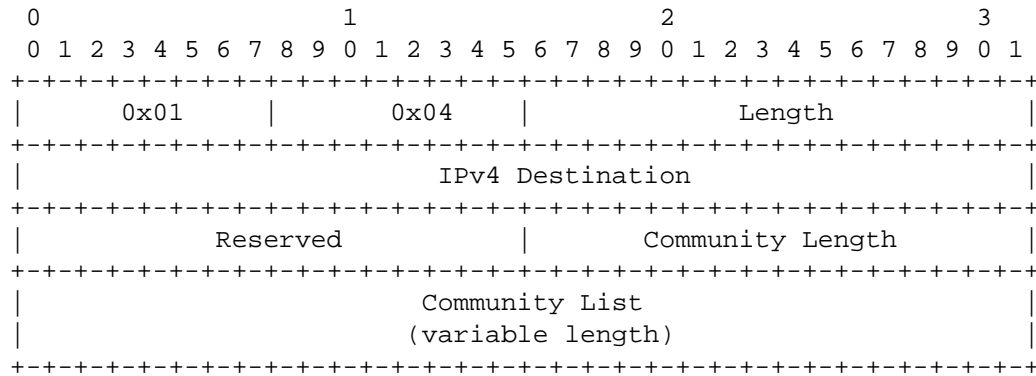
   IPv4 Destination: The IPv4 address with which the community
      information should be stored.

   Community Length: A 2-octet unsigned number that indicates the length
      of the Community List. The length does not include the IPv4
      Address, Reserved, or Length fields.

   Community List: One or more 8-octet EIGRP communities, as defined in
      Section 6.4.

6.8.6. IPv6-Specific TLVs

```
        INTERNAL_TYPE                 0x0402
        EXTERNAL_TYPE                 0x0403
        COMMUNITY_TYPE                0x0404
```

6.8.6.1. IPv6 INTERNAL_TYPE

   This TLV conveys the IPv6 destination and associated metric
   information for IPv6 networks. Routes advertised in this TLV are
   network interfaces that EIGRP is configured on as well as networks
   that are learned via other routers running EIGRP.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x04     |      0x02      |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                   Next-Hop Forwarding Address                 |
   |                          (16 octets)                          |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |             Vector Metric Section (see Section 6.8.2)         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
   |                       Destination Section                     |
   |                   IPv6 Address (variable length)              |
   |                          (see Section 6.8.4)                  |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Next-Hop Forwarding Address: This IPv6 address is represented by
      eight groups of 16-bit values (total 16 octets). If the value is
      zero (0), the IPv6 address from the received IPv6 header is used
      as the next hop for the route. Otherwise, the specified IPv6
      address will be used.

   Vector Metric Section: Vector metrics for destinations contained in
      this TLV. See the description of "metric encoding" in Section
      6.8.2.

   Destination Section: The network/subnet/host destination address
      being requested. See the description of "destination" in Section
      6.8.4.

6.8.6.2. IPv6 EXTERNAL_TYPE

   This TLV conveys IPv6 destination and metric information for routes
   learned by other routing protocols that EIGRP injects into the
   topology. Available with this information is the identity of the
   routing protocol that created the route, the external metric, the AS
   number, an indicator if it should be marked as part of the EIGRP AS,
   and a network administrator tag used for route filtering at EIGRP AS
   boundaries.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x04     |      0x03     |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                              |
   |                 Next-Hop Forwarding Address                  |
   |                        (16 octets)                           |
   |                                                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Exterior Section (see Section 6.8.3)              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Vector Metric Section (see Section 6.8.2)           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
   |                    Destination Section                       |
   |              IPv6 Address (variable length)                  |
   |                    (see Section 6.8.4)                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
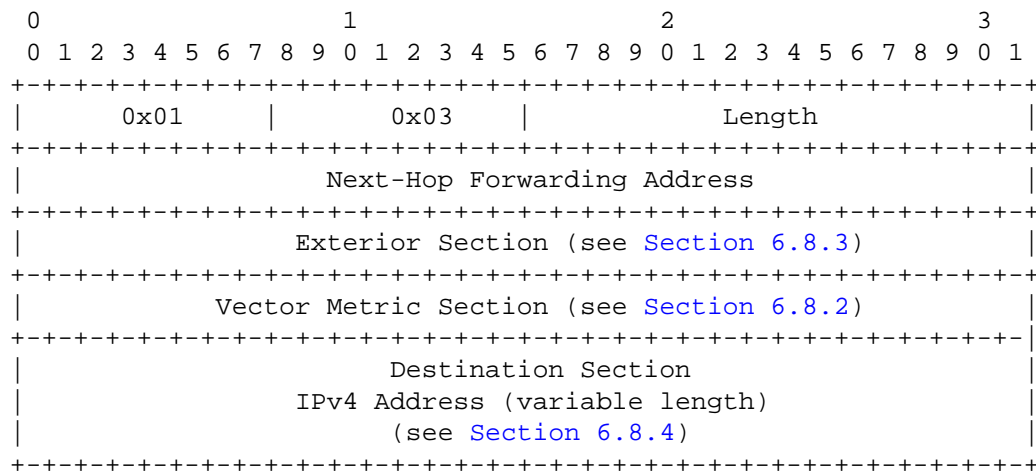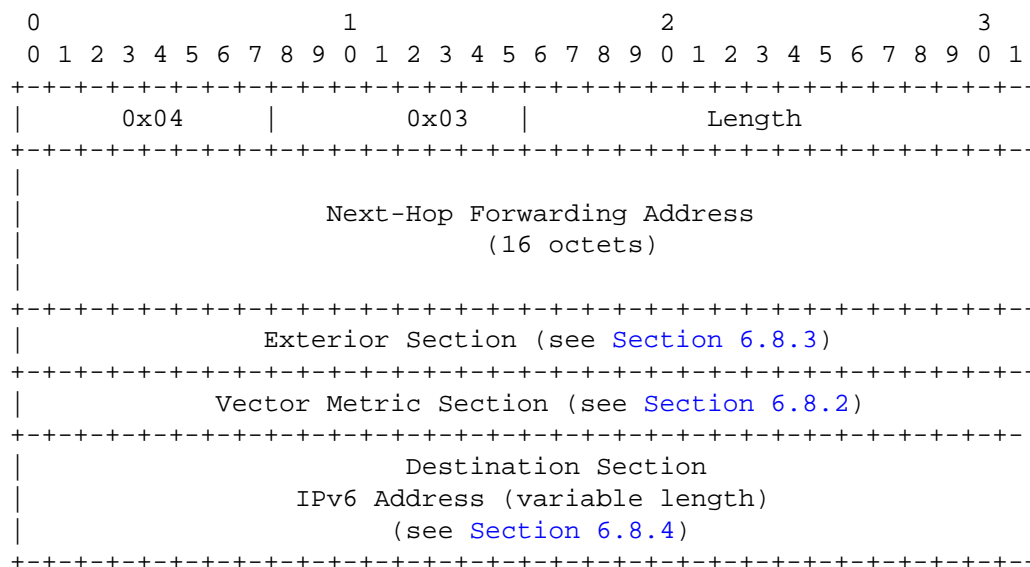
   Next-Hop Forwarding Address: IPv6 address is represented by eight
      groups of 16-bit values (total 16 octets). If the value is zero
      (0), the IPv6 address from the received IPv6 header is used as the
      next hop for the route. Otherwise, the specified IPv6 address
      will be used.

   Exterior Section: Additional routing information provided for a
      destination that is outside of the AS and that has been
      redistributed into the EIGRP. See the description of "exterior
      encoding" in Section 6.8.3.

   Vector Metric Section: vector metrics for destinations contained in
      this TLV. See the description of "metric encoding" in Section
      6.8.2.

   Destination Section: The network/subnet/host destination address
      being requested. See the description of "destination" in Section
      6.8.4.

6.8.6.3 IPv6 COMMUNITY_TYPE

   This TLV is used to provide community tags for specific IPv4
   destinations.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |     0x04      |     0x04      |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   |                         Destination                           |
   |                         (16 octets)                           |
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |            Reserved          |       Community Length         |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Community List                          |
   |                      (variable length)                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
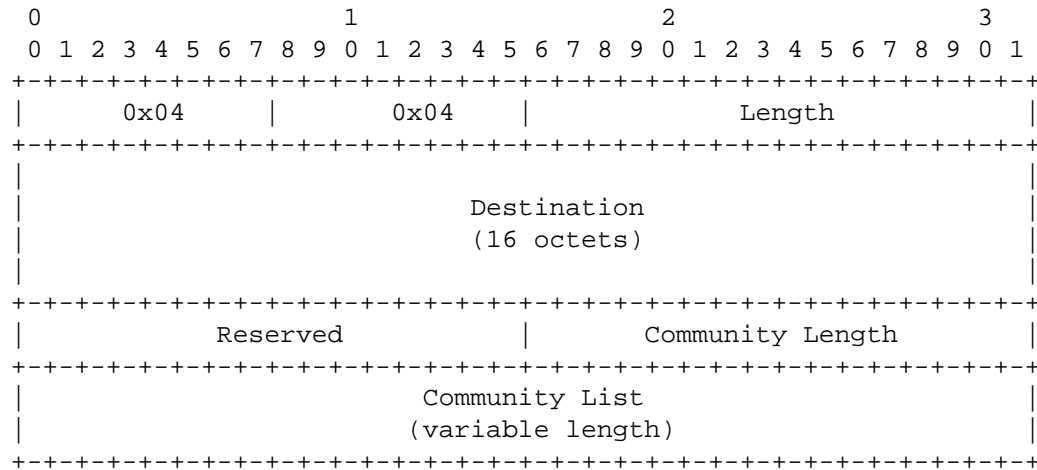
   Destination: The IPv6 address with which the community information
      should be stored.

   Community Length: A 2-octet unsigned number that indicates the length
      of the Community List. The length does not include the IPv6
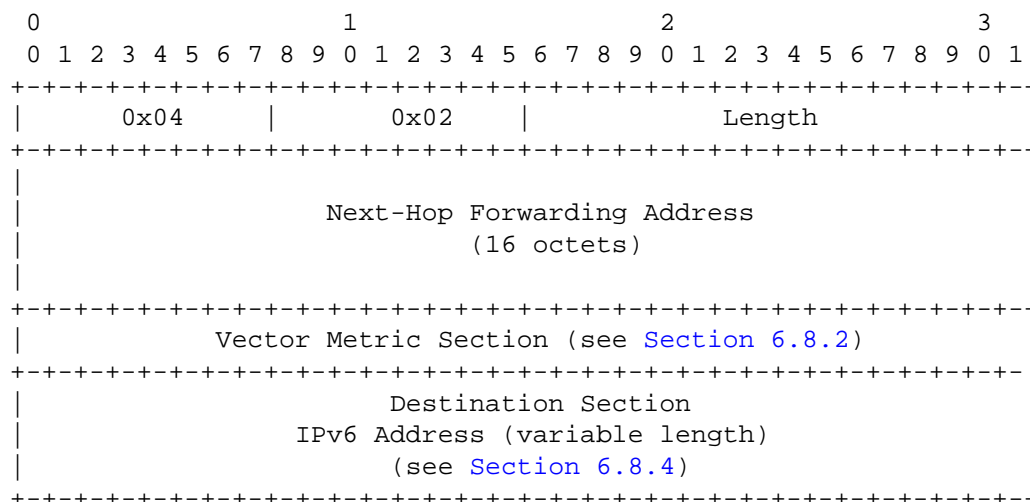   Address, Reserved, or Length fields.

   Community List: One or more 8-octet EIGRP communities, as defined in
      Section 6.4.

6.9. Multiprotocol Route Information TLV Types

   This TLV conveys topology and associated metric information.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |Header Version |    Opcode     |             Checksum          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                              Flags                            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                         Sequence Number                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Acknowledgment Number                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Virtual Router ID           |     Autonomous System Number    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        TLV Header Encoding                    |
   |                        (see Section 6.9.1)                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Wide Metric Encoding                   |
   |                        (see Section 6.9.2)                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                       Destination Descriptor                  |
   |                          (variable length)                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
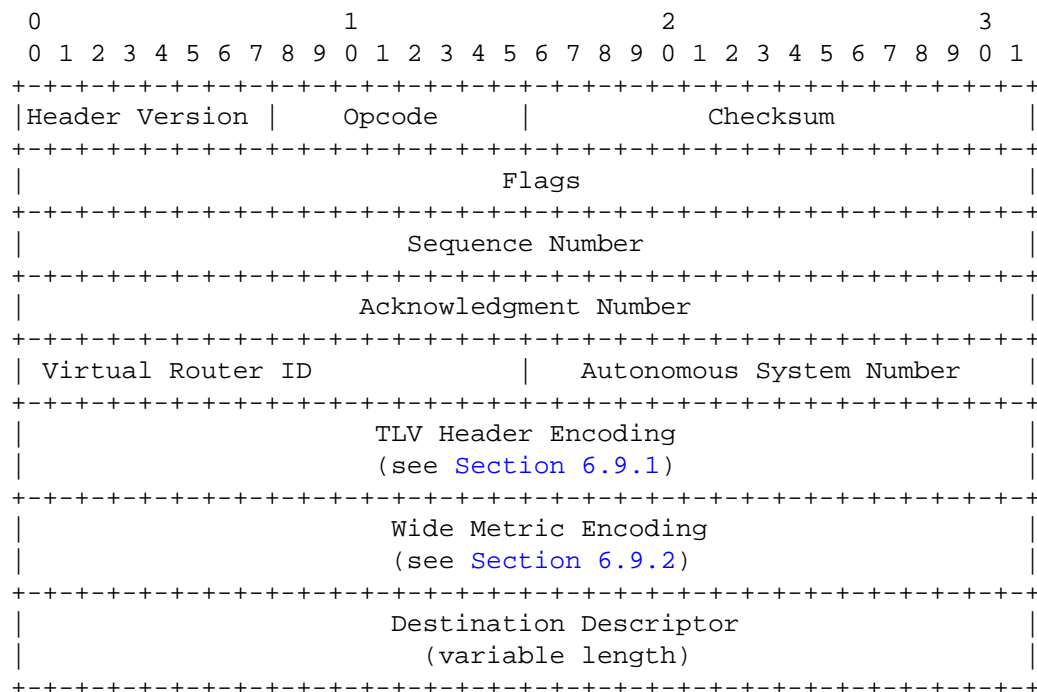
6.9.1. TLV Header Encoding

   There has been a long-standing requirement for EIGRP to support
   routing technologies, such as multi-topologies, and to provide the
   ability to carry destination information independent of the
   transport. To accomplish this, a Vector has been extended to have a
   new "Header Extension Header" section. This is a variable-length
   field and, at a minimum, it will support the following fields:

```
    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 High     | Type Low      |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              AFI              |              TID               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Router Identifier (RID)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Value (variable length)                   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
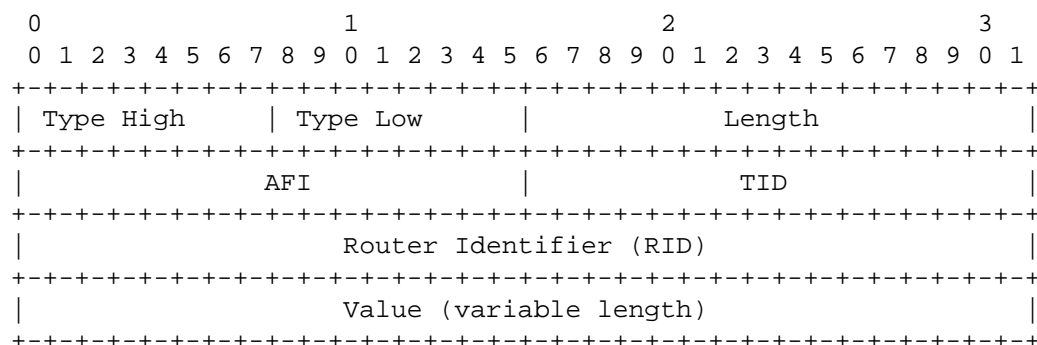
The available fields are:

TYPE - Topology TLVs have the following TYPE codes:
    Type High: 0x06
    Type Low:
        REQUEST_TYPE   0x01
        INTERNAL_TYPE   0x02
        EXTERNAL_TYPE   0x03

Router Identifier (RID): A 32-bit number provided by the router
    sourcing the information to uniquely identify it as the source.

6.9.2. Wide Metric Encoding

Multiprotocol TLVs will provide an extendable section of metric
information, which is not used for the primary routing compilation.
Additional per-path information is included to enable per-path cost
calculations in the future. Use of the per-path costing along with
the VID/TID will prove a complete solution for multidimensional
routing.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Offset     |    Priority    |  Reliability  |     Load      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               MTU                             |   Hop Count   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Delay                             |
|                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   |                             |
+-+-+-+-+-+-+-+-+-+-+-+                             |
|                          Bandwidth                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved          |          Opaque Flags           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Extended Attributes                      |
|                       (variable length)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
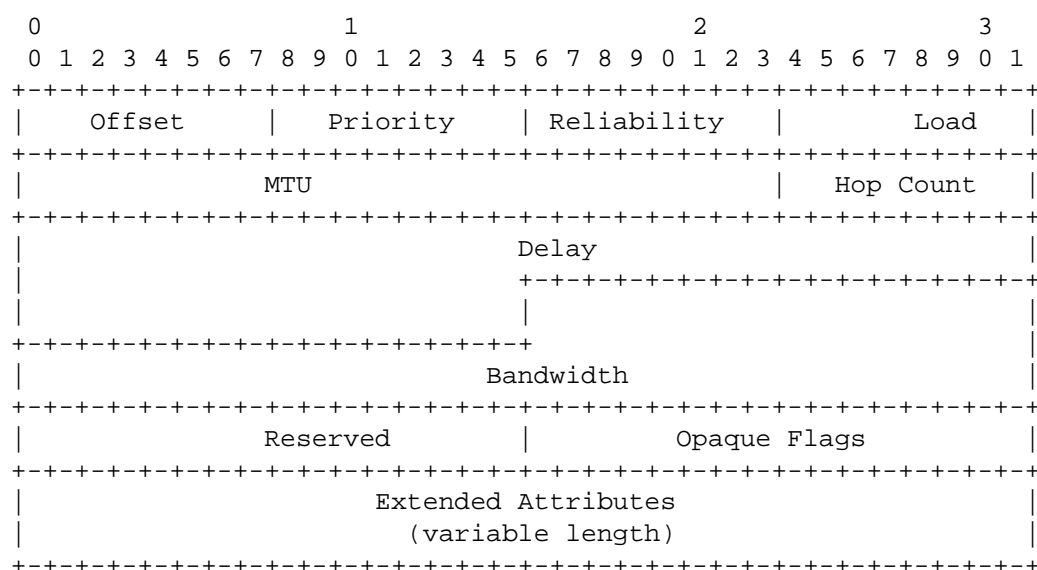
The fields are as follows:

Offset: Number of 16-bit words in the Extended Attribute section that
    are used to determine the start of the destination information. A
    value of zero indicates no Extended Attributes are attached.

   Priority: Priority of the prefix when processing a route. In an AS
      using priority values, a destination with a higher priority
      receives preferential treatment and is serviced before a
      destination with a lower priority. A value of zero indicates no
      priority is set.

   Reliability: The current error rate for the path. Measured as an
      error percentage. A value of 255 indicates 100% reliability

   Load: The load utilization of the path to the destination, measured
      as a percentage. A value of 255 indicates 100% load.

   MTU: The minimum MTU size for the path to the destination. Not used
      in metric calculation but available to underlying protocols

   Hop Count: The number of router traversals to the destination.

   Delay: The one-way latency along an unloaded path to the destination
      expressed in units of picoseconds per kilobit. This number is not
      scaled; a value of 0xFFFFFFFFFFFF indicates an unreachable route.

   Bandwidth: The path bandwidth measured in kilobit per second as
      presented by the interface. This number is not scaled; a value of
      0xFFFFFFFFFFFF indicates an unreachable route.

   Reserved: Transmitted as 0x0000.

   Opaque Flags: 16-bit protocol-specific flags. Values currently
      defined by Cisco are:

         OPAQUE_SRCWD   0x01   Route Source Withdraw
         OPAQUE_CD   0x02   Candidate default route
         OPAQUE_ACTIVE   0x04   Route is currently in active state
         OPAQUE_REPL   0x08   Route is replicated from another VRF

   Extended Attributes (Optional): When present, defines extendable per-
      destination attributes. This field is not normally transmitted.

6.9.3. Extended Metrics

   Extended metrics allow for extensibility of the vector metrics in a
   manner similar to RFC 6390 [11]. Each Extended metric shall consist
   of a header identifying the type (Opcode) and the length (Offset)
   followed by application-specific information. Extended metric values
   not understood must be treated as opaque and passed along with the
   associated route.

The general formats for the Extended Metric fields 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Opcode    |     Offset    |              Data             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
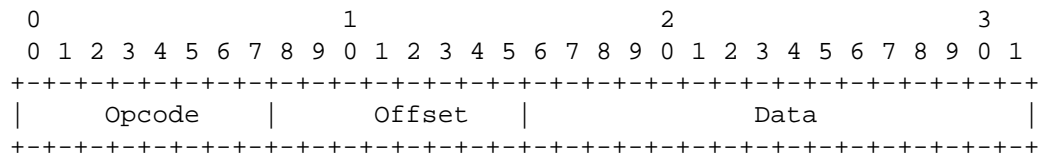
Opcode: Indicates the type of Extended Metric.

Offset: Number of 16-bit words in the application-specific
   information. Offset does not include the length of the Opcode or
   Offset.

Data: Zero or more octets of data as defined by Opcode.

6.9.3.1. 0x00 - NoOp

This is used to pad the attribute section to ensure 32-bit alignment
of the metric encoding section.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     0x00      |     0x00      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
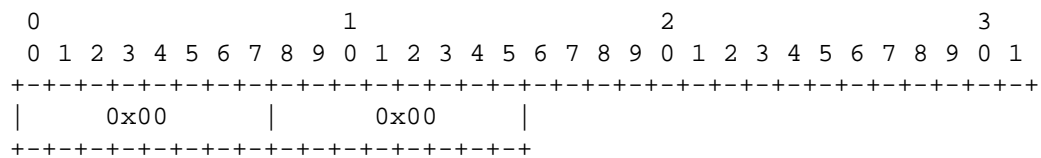
The fields are:

Opcode: Transmitted as zero (0).

Offset: Transmitted as zero (0) indicating no data is present.

Data: No data is present with this attribute.

6.9.3.2. 0x01 - Scaled Metric

   If a route is received from a back-rev neighbor, and the route is
   selected as the best path, the scaled metric received in the older
   UPDATE may be attached to the packet. If received, the value is for
   informational purposes and is not affected by K6.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     0x01      |     0x04      |            Reserved            |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                      Scaled Bandwidth                         |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                        Scaled Delay                           |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
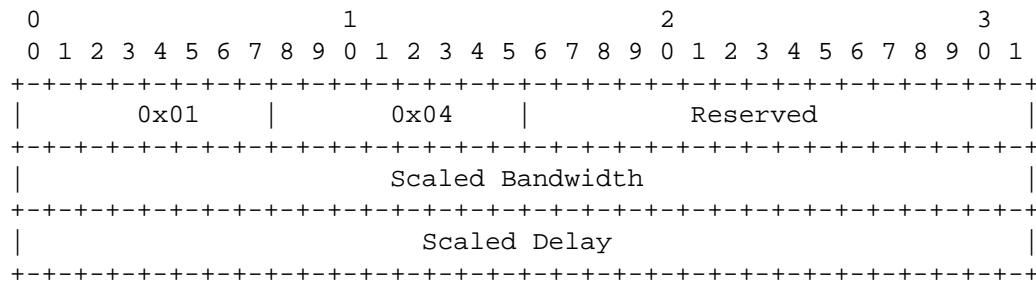
   Reserved: Transmitted as 0x0000

   Scaled Bandwidth: The minimum bandwidth along a path expressed in
      units of 2,560,000,000/kbps. A bandwidth of 0xFFFFFFFF indicates
      an unreachable route.

   Scaled Delay: An administrative parameter assigned statically on a
      per-interface-type basis to represent the time it takes along an
      unloaded path. This is expressed in units of tens of microseconds
      divvied by 256. A delay of 0xFFFFFFFF indicates an unreachable
      route.

6.9.3.3. 0x02 - Administrator Tag

   EIGRP administrative tag does not alter the path decision-making
   process. Routers can set a tag value on a route and use the flags to
   apply specific routing polices within their network.

```
  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
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |     0x02      |     0x02      |        Administrator Tag       |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | Administrator Tag (cont.)     |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
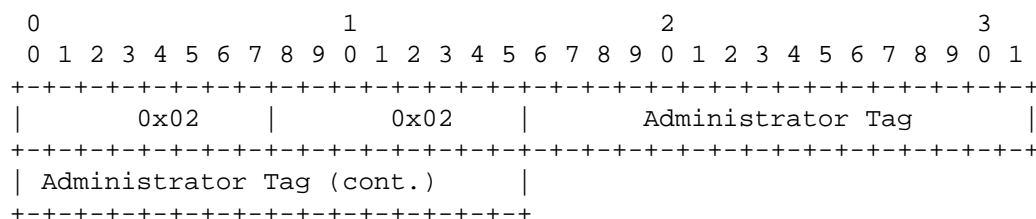
   Administrator Tag: A tag assigned by the network administrator that
      is untouched by EIGRP. This allows a network administrator to
      filter routes in other EIGRP border routers based on this value.

6.9.3.4. 0x03 - Community List

   EIGRP communities themselves do not alter the path decision-making
   process, communities can be used as flags in order to mark a set of
   routes. Upstream routers can then use these flags to apply specific
   routing polices within their network.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      0x03     |     Offset    |         Community List        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                         (variable length)                    |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
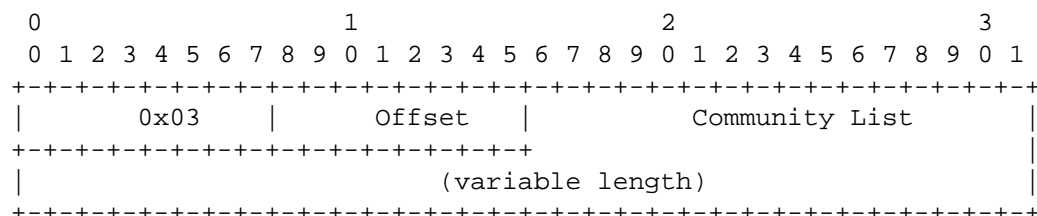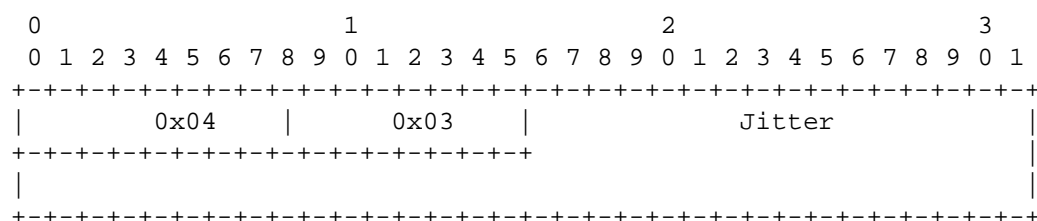
   Offset: Number of 16-bit words in the sub-field.

   Community List: One or more 8-octet EIGRP communities, as defined in
      Section 6.4.

6.9.3.5. 0x04 - Jitter

   (Optional) EIGRP can carry one-way Jitter in networks that carry UDP
   traffic if the node is capable of measuring UDP Jitter. The Jitter
   reported to will be averaged with any existing Jitter data and
   include in the route updates. If no Jitter value is reported by the
   peer for a given destination, EIGRP will use the locally collected
   value.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |      0x04     |      0x03     |             Jitter            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
    |                                                              |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Jitter: The measure of the variability over time of the latency
      across a network measured in measured in microseconds.

6.9.3.6. 0x05 - Quiescent Energy

   (Optional) EIGRP can carry energy usage by nodes in networks if the
   node is capable of measuring energy. The Quiescent Energy reported
   will be added to any existing energy data and include in the route
   updates. If no energy data is reported by the peer for a given
   destination, EIGRP will use the locally collected value.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         0x05      |         0x02      |      Q-Energy (high)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Q-Energy (low)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Q-Energy: Paths with higher idle (standby) energy usage will be
      reflected in a higher aggregate metric than those having lower
      energy usage. If present, this number will represent the idle
      power consumption expressed in milliwatts per kilobit.

6.9.3.7. 0x06 - Energy

   (Optional) EIGRP can carry energy usage by nodes in networks if the
   node is capable of measuring energy. The active Energy reported will
   be added to any existing energy data and include in the route
   updates. If no energy data is reported by the peer for a given
   destination, EIGRP will use the locally collected value.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         0x06      |         0x02      |       Energy (high)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Energy (low)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Energy: Paths with higher active energy usage will be reflected in a
      higher aggregate metric than those having lower energy usage. If
      present, this number will represent the power consumption
      expressed in milliwatts per kilobit.

6.9.3.8. 0x07 - AddPath

   The Add Path enables EIGRP to advertise multiple best paths to
   adjacencies. There will be up to a maximum of four AddPaths
   supported, where the format of the field will be as follows.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         0x07      |      Offset       |  AddPath (Variable Length) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Offset: Number of 16-bit words in the sub-field.

   AddPath: Length of this field will vary in length based on whether it
      contains IPv4 or IPv6 data.

6.9.3.8.1. AddPath with IPv4 Next Hop

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |      0x07     |      Offset    | Next-Hop Addr. (Upper 2 bytes)|
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv4 Address (Lower 2 bytes)  |       RID (Upper 2 bytes)     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |        RID (Upper 2 bytes)    | Admin Tag (Upper 2 bytes)     |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Admin Tag (Upper 2 bytes)     |Extern Protocol| Flags Field   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Next-Hop Address: An IPv4 address represented by four 8-bit values
      (total 4 octets). If the value is zero (0), the IPv6 address from
      the received IPv4 header is used as the next hop for the route.
      Otherwise, the specified IPv4 address will be used.

   Router Identifier (RID): A 32-bit number provided by the router
      sourcing the information to uniquely identify it as the source.

   Admin Tag: A 32-bit administrative tag assigned by the network. This
      allows a network administrator to filter routes based on this
      value.

   If the route is of type external, then two additional bytes will be
   added as follows:

   External Protocol: Contains an enumerated value defined in Section
      6.2 to identify the routing protocol (external protocol)
      redistributing the route.

   Flags Field: See Section 6.8.1.

6.9.3.8.2. AddPath with IPv6 Next Hop

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0x07       |     Offset    |       Next-Hop Address       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                              |
|                                                               |
|                                                               |
|                          (16 octets)                          |
|                                   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|                                   |       RID (Upper 2 byes)    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        RID (Upper 2 byes)        |    Admin Tag (Upper 2 byes)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Admin Tag (Upper 2 byes)        | Extern Protocol | Flags Field |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Next-Hop Address: An IPv6 address represented by eight groups of
      16-bit values (total 16 octets). If the value is zero (0), the
      IPv6 address from the received IPv6 header is used as the next hop
      for the route. Otherwise, the specified IPv6 address will be
      used.

   Router Identifier (RID): A 32-bit number provided by the router
      sourcing the information to uniquely identify it as the source.
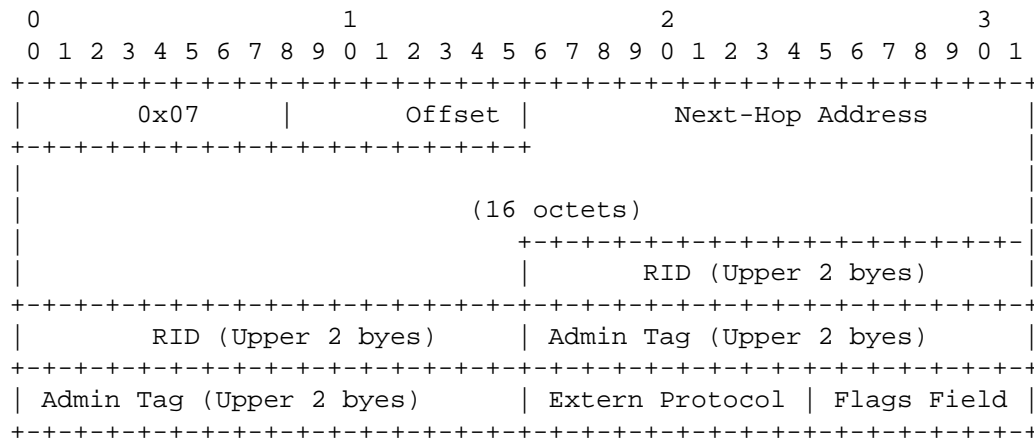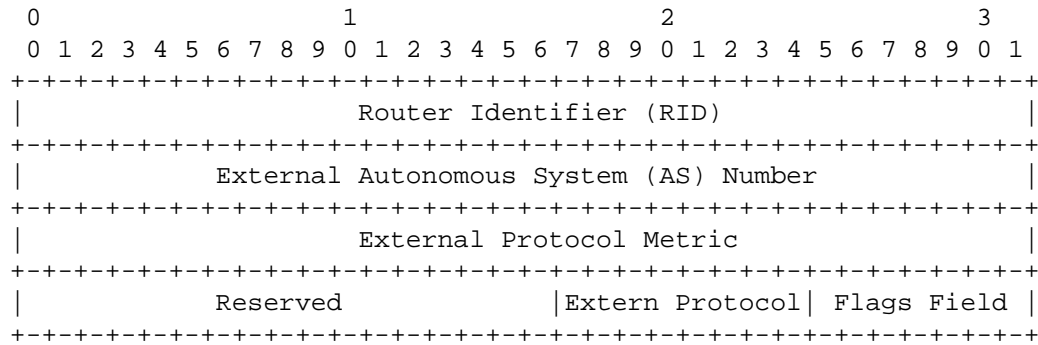
   Admin Tag: A 32-bit administrative tag assigned by the network. This
      allows a network administrator to filter routes based on this
      value. If the route is of type external, then two addition bytes
      will be added as follows:

   External Protocol: Contains an enumerated value defined in Section
      6.2 to identify the routing protocol (external protocol)
      redistributing the route.

   Flags Field: See Section 6.8.1.

6.9.4. Exterior Encoding

   Additional routing information provided for destinations outside of
   the EIGRP AS as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Router Identifier (RID)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             External Autonomous System (AS) Number            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    External Protocol Metric                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved            |Extern Protocol| Flags Field  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Router Identifier (RID): A 32-bit number provided by the router
      sourcing the information to uniquely identify it as the source.

   External Autonomous System (AS) Number: A 32-bit number indicating
      the external AS of which the sending router is a member. If the
      source protocol is EIGRP, this field will be the [VRID, AS] pair.
      If the external protocol does not have an AS, other information
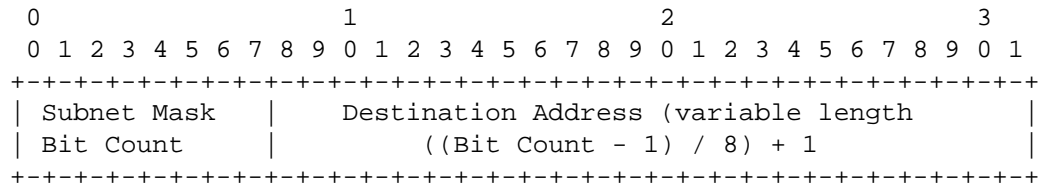   can be used (for example, Cisco uses process-id for OSPF).

   External Protocol Metric: A 32-bit value of the metric used by the
      routing table as learned by the foreign protocol. If the External
      Protocol is IGRP or EIGRP, the value can (optionally) be 0, and
      the metric information is stored in the metric section.

   External Protocol: Contains an enumerated value defined in Section
      6.2 to identify the routing protocol (external protocol)
      redistributing the route.

   Flags Field: See Section 6.8.1.

6.9.5. Destination Encoding

   Destination information is encoded in Multiprotocol packets in the
   same manner used by Classic TLVs. This is accomplished by using a
   counter to indicate how many significant bits are present in the
   variable-length address field.

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | Subnet Mask   |    Destination Address (variable length      |
   | Bit Count     |         ((Bit Count - 1) / 8) + 1            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Subnet Mask Bit Count: 8-bit value used to indicate the number of
      bits in the subnet mask. A value of 0 indicates the default
      network and no address is present.

   Destination Address: A variable-length field used to carry the
      destination address. The length is determined by the number of
      consecutive bits in the destination address. The formula to
      calculate the length is address-family dependent:

      IPv4: ((Bit Count - 1) / 8) + 1
      IPv6: (Bit Count == 128) ? 16 : ((x / 8) + 1)

6.9.6. Route Information

6.9.6.1. INTERNAL TYPE

   This TLV conveys destination information based on the IANA AFI
   defined in the TLV Header (see Section 6.9.1), and associated metric
   information. Routes advertised in this TLV are network interfaces
   that EIGRP is configured on as well as networks that are learned via
   other routers running EIGRP.

6.9.6.2. EXTERNAL TYPE

   This TLV conveys destination information based on the IANA AFI
   defined in the TLV Header (see Section 6.9.1), and metric information
   for routes learned by other routing protocols that EIGRP injects into
   the AS. Available with this information is the identity of the
   routing protocol that created the route, the external metric, the AS
   number, an indicator if it should be marked as part of the EIGRP AS,
   and a network administrator tag used for route filtering at EIGRP AS
   boundaries.

7. Security Considerations

   Being promiscuous, EIGRP will neighbor with any router that sends a
   valid HELLO packet. Due to security considerations, this
   "completely" open aspect requires policy capabilities to limit
   peering to valid routers.

   EIGRP does not rely on a PKI or a heavyweight authentication system.
   These systems challenge the scalability of EIGRP, which was a primary
   design goal.

   Instead, Denial-of-Service (DoS) attack prevention will depend on
   implementations rate-limiting packets to the control plane as well as
   authentication of the neighbor through the use of MD5 or SHA2-256
   [6].

8. IANA Considerations

   This document serves as the sole reference for two multicast
   addresses: 224.0.0.10 for IPv4 "EIGRP Routers" [13] and
   FF02:0:0:0:0:0:0:A for IPv6 "EIGRP Routers" [14]. It also serves as
   assignment for protocol number 88 (EIGRP) [15].

9. References

9.1. Normative References

   [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement
        Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,
        <http://www.rfc-editor.org/info/rfc2119>.

   [2] Garcia-Luna-Aceves, J.J., "A Unified Approach to Loop-Free
        Routing Using Distance Vectors or Link States", SIGCOMM '89,
        Symposium proceedings on Communications architectures &
        protocols, Volume 19, pages 212-223, ACM
        089791-332-9/89/0009/0212, DOI 10.1145/75247.75268, 1989.

   [3] Garcia-Luna-Aceves, J.J., "Loop-Free Routing using Diffusing
        Computations", Network Information Systems Center, SRI
        International, appeared in IEEE/ACM Transactions on Networking,
        Vol. 1, No. 1, DOI 10.1109/90.222913, 1993.

   [4] Rosen, E. and Y. Rekhter, "IANA Registries for BGP Extended
        Communities", RFC 7153, DOI 10.17487/RFC7153, March 2014,
        <http://www.rfc-editor.org/info/rfc7153>.

   [5] Narten, T., "Assigning Experimental and Testing Numbers
        Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692,
        January 2004, <http://www.rfc-editor.org/info/rfc3692>.

   [6] Kelly, S. and S. Frankel, "Using HMAC-SHA-256, HMAC-SHA-384, and
        HMAC-SHA-512 with IPsec", RFC 4868, DOI 10.17487/RFC4868, May
        2007, <http://www.rfc-editor.org/info/rfc4868>.

   [7] Deering, S., "Host extensions for IP multicasting", STD 5,
        RFC 1112, DOI 10.17487/RFC1112, August 1989,
        <http://www.rfc-editor.org/info/rfc1112>.

   [8] Postel, J., "Internet Protocol", STD 5, RFC 791,
        DOI 10.17487/RFC0791, September 1981,
        <http://www.rfc-editor.org/info/rfc791>.

   [9] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6)
        Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998,
        <http://www.rfc-editor.org/info/rfc2460>.

9.2. Informative References

   [10] Moy, J., "OSPF Version 2", STD 54, RFC 2328,
        DOI 10.17487/RFC2328, April 1998,
        <http://www.rfc-editor.org/info/rfc2328>.

   [11] Clark, A. and B. Claise, "Guidelines for Considering New
        Performance Metric Development", BCP 170, RFC 6390,
        DOI 10.17487/RFC6390, October 2011,
        <http://www.rfc-editor.org/info/rfc6390>.

   [12] IANA, "Address Family Numbers",
        <http://www.iana.org/assignments/address-family-numbers>.

   [13] IANA, "IPv4 Multicast Address Space Registry",
        <http://www.iana.org/assignments/multicast-addresses>.

   [14] IANA, "IPv6 Multicast Address Space Registry",
        <http://www.iana.org/assignments/ipv6-multicast-addresses>.

   [15] IANA, "Protocol Numbers",
        <http://www.iana.org/assignments/protocol-numbers>.

Acknowledgments

   Thank you goes to Dino Farinacci, Bob Albrightson, and Dave Katz.
   Their significant accomplishments towards the design and development
   of the EIGRP provided the bases for this document.

   A special and appreciative thank you goes to the core group of Cisco
   engineers whose dedication, long hours, and hard work led the
   evolution of EIGRP over the past decade. They are Donnie Savage,
   Mickel Ravizza, Heidi Ou, Dawn Li, Thuan Tran, Catherine Tran, Don
   Slice, Claude Cartee, Donald Sharp, Steven Moore, Richard Wellum, Ray
   Romney, Jim Mollmann, Dennis Wind, Chris Van Heuveln, Gerald Redwine,
   Glen Matthews, Michael Wiebe, and others.

   The authors would like to gratefully acknowledge many people who have
   contributed to the discussions that lead to the making of this
   proposal. They include Chris Le, Saul Adler, Scott Van de Houten,
   Lalit Kumar, Yi Yang, Kumar Reddy, David Lapier, Scott Kirby, David
   Prall, Jason Frazier, Eric Voit, Dana Blair, Jim Guichard, and Alvaro
   Retana.

   In addition to the tireless work provided by the Cisco engineers over
   the years, we would like to personally recognize the teams that
   created open source versions of EIGRP:

   o Linux implementation developed by the Quagga team: Jan Janovic,
      Matej Perina, Peter Orsag, and Peter Paluch.

   o BSD implementation developed and released by Renato Westphal.

Authors' Addresses

   Donnie V. Savage
   Cisco Systems, Inc.
   7025 Kit Creek Rd., RTP,
   Morrisville, NC 27560
   United States
   Phone: 919-392-2379
   Email: dsavage@cisco.com

   James Ng
   Cisco Systems, Inc.
   7025 Kit Creek Rd., RTP,
   Morrisville, NC 27560
   United States
   Phone: 919-392-2582
   Email: jamng@cisco.com

   Steven Moore
   Cisco Systems, Inc.
   7025 Kit Creek Rd., RTP,
   Morrisville, NC 27560
   United States
   Phone: 408-895-2031
   Email: smoore@cisco.com

   Donald Slice
   Cumulus Networks
   Apex, NC
   United States
   Email: dslice@cumulusnetworks.com

   Peter Paluch
   University of Zilina
   Univerzitna 8215/1, Zilina 01026
   Slovakia
   Phone: 421-905-164432
   Email: Peter.Paluch@fri.uniza.sk

   Russ White
   LinkedIn
   Apex, NC
   United States
   Phone: 1-877-308-0993
   Email: russw@riw.us