Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: February, 2016

D. Savage
J. Ng
S. Moore
Cisco Systems
D. Slice
Cumulus Networks
P. Paluch
University of Zilina
R. White
Ericsson

16 Aug 2015

Enhanced Interior Gateway Routing Protocol

draft-savage-eigrp-04.txt

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 referance in "Loop-Free Routing using Diffusing Computations". The algorithm and procedures were researched, developed, and simulated by SRI International.

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].

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on Febuary 16, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

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 Terminology

The following list describes acronyms and definitions for 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 network layer reachability information associated with the network layer 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 and the cost of the link between the two routers. Exactly one Computed Distance is computed and maintained per the [Destination, Advertising Neighbor] pair.

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 Reported Distance is less than this router's Feasible Distance.

Feasible Distance (FD)
Defined as the lowest 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 Feasible Distance 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 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 Feasibility Condition check. A route in PASSIVE state is considered usable and not in need of a coordinated re-computation.

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. Reported Distance 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 Reported Distance is computed and maintained per destination network.

Sub-Topology
For a given Base Topology, a sub-topology is characterized by an independent set of router 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 Computed Distance 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 Reported Distance

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, Feasible Distance, Reported Distance of each neighbor advertising the destination, Computed Distance 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.

Virtual Routing and Forwarding (VRF)
Independent Virtual Private Network (VPN) routing/forwarding tables which co-exist within the same router at the same time.

3 The DUAL Diffusing Update Algorithm

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 re-convergence. 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 Reported Distance to the affected destination, and 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 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 Computed Distance 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 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 is usable and its next hop is perceived to be a downstream router.

o Active
A route is considered 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 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 Reported Distance until the route enters the PASSIVE state again. Any updated information about this route received during ACTIVE state is reflected only in Computed Distances. Any updates to the Successor, Feasible Distance and Reported Distance 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 Feasible Distance 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 Feasible Distance 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 Feasible Distance 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

The DUAL algorithm 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 Feasibility Condition fails which can happen for reasons like a destination becoming unreachable, or the metric increasing to a value greater than its current Feasible Distance.

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

In addition to these 3 basic types, two addition sub-types have been added to EIGRP:

- o SIA-QUERY - sent when a REPLY has not been received within one half 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, but is only used to indicate 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 finite state machine 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 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 finite state machine, 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 Successor Directed Acyclic Graph (SDAG) is computed for each destination, so loop-free topologies can be maintained for each reachable destination.

Figure 1 illustrates the FSM:

```
 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.
```

```
           +------------+                +-----------+
           |             \              /            |
           |              \            /             |
           |   +=================================+   |
           |   |                                 |   |
           |(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

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 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 Active 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 active. 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 which 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 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 successor if Oij indicates that QUERY was received from 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 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 elects C as its successor to reach node N with a cost of 4.

Notice that node 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 sends QUERY to B. B replies to the QUERY since it is in ACTIVE State.
Once C has received this REPLY, it has heard from all its neighbors, so it can go passive 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 5 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 which 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; IPv4 "IGRP Routers" [13] and IPv6 "EIGRP Routers" [14]. Thesse data link multicast addresses 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 state machine.

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 a single QUERY packet with 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 acknowledgement is accomplished either by sending an ACK packet, or piggybacked the acknowledgment onto another packet already being transmitted.

Each TLV destination will be processed individually through the DUAL state machine. When a QUERY is received for a route that doesn't exist in our topology table, a REPLY with 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.

4.4 Exception Handling

4.4.1 Active Duration (Stuck-in-Active)

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 Stuck-In-Active (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 3 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 which 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.

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 attempt 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 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 which 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, and 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 QUERY is from the successor, transition to the ACTIVE state if and only if feasibility-condition 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 metric for which is stored in its own routing 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, however; 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 Protocol 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 0

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 or 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 insure that convergence time remains low in the presence of varying speed links.

The DUAL Algorithm assumes there is lossless communication between devices and thus must rely upon the transport protocol to guarantee that messages are transmitted reliably. EIGRP implements the Reliable Transport Protocol to ensure ordered delivery and acknowledgement 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 reliable EIGRP 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 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 zero (0)

indicates that sender is not expecting to explicitly acknowledging 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 acknowledgement requirement for the packet from the respective neighbor.

If the required acknowledgement 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 that peer which 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. 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 assure that all multicasts are transmitted in order as well as not mixing the order among unicasts 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                     Receives 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 pkt
                                   <----------------
                                   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                  Receives 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
                                Keeps packet on A's retransmit list
----------------->
ACK (unicast)
SEQ=0, ACK=101                  Receives 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 4 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                  Receives ACK
Process Update                  Delete Packet from B's retransmit list

----------------->
C sends ACK (unicast)
SEQ=0, ACK=100                  Receives ACK
Process Update                  Delete Packet from C's retransmit list

----------------->
D sends ACK (unicast)
SEQ=0, ACK=100                  Receives ACK
Process Update                  Delete Packet from D's retransmit list
```

```
                      An Example QUERY Exchange


                               <----------------
                               A send 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 send 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 send 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 send 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 9.

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


                                   <----------------
                                   A send UPDATE (multicast)
                                   SEQ=100, ACK=0
---------------/LOST/->            Add Packet to B's retransmit list
B send 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 send HELLO (multicast)
                                   SEQ=101, 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 send UPDATE (multicast)
                                   SEQ=101, ACK=0, CR-Flag=1
---------------/LOST/->            Add Packet to B's retransmit list
B send 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 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 pkt from retransmit list
SEQ=0, ACK=100
                                   <----------------
                                   A resends UPDATE (unicast to B)
                                   SEQ=101, ACK=0

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

Figure 9

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 to not receive the next packet or it would receive the retransmitted packet out of order.

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.

Router-C and Router-D process the SEQUENCE_TYPE TLV by looking for its own address in the list. If not found, they put themselves in Conditionally Received (CR-mode) 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 acknowledgement is received, Router-A can remove both packets off 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 rest. 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 3 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

When a router detects a new neighbor by receiving a HELLO packet from a neighbor not presently known, it will send a unicast UPDATE packet to the neighbor with no routing information. The initial UPDATE packet sent MUST have the INIT-flag set. This instructs the neighbor to advertise its routes. The INIT-flag 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.

5.3.3.1 NULL Update

The number of destinations in its routing table will require at a minimum two (2) UPDATE packets to be sent. The first UPDATE packet (referred it as the NULL UPDATE packet) is sent with the INIT-Flag, and containing no topology information. The use of the NULL UPDATE is used to ensure di-directional UNICAST packet delivery.

The second packet is queued, and cannot be sent until the first is acknowledged.

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 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 cannot be sent until the first is acknowledged.

(3) Router A receives first UPDATE packet and processes it as a DUAL event. If the UPDATE contains topology information, the packet will be process and stored in topology table. 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 3-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 neighbor which 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 not send it any a QUERY or UPDATE. When Router-A receives the unicast acknowledgement from Router-B, it will check 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 the pending state, and 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 will not be expected to REPLY to it.

5.4 Topology Table

The Topology Table is populated by the protocol dependent modules (IPv4/IPv6 PDM), and is acted upon by the DUAL finite state machine. Associated with each entry are the destination address and a list of neighbors that have advertised this destination, and the metric associated with the destination. The metric is referred to as the Computed Distance.

The Computed Distance is the best-advertised Reported Distance from all neighbors, plus the link cost between the receiving router and the neighbor.

The Reported Distance is the Computed Distance as advertised by the Feasible Successor for the destination. Said another way, 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 Computed Distance 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.

5.4.7 Route Management

Within the topology table, EIGRP has the notion of internal and external routes. Internal routes MUST be prefered over external routes independent of the metric. I practical terms, if and internal route is received, the Dufusion comoutation will be run considering only the interal routes. Only when no internal routes for a give destination exist, will EIGRP choose the a Successor from the available external routes.

5.4.7.1 Internal Routes

Internal routes are destinations that have been originated within the same EIGRP Autonomous System. 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.

5.4.7.2 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 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 use 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 AS's where EIGRP would typically interact with an inter-domain routing protocol that implements global policies.

5.4.7.3 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.

The split horizon rule states: "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". Again, as with the case of Split Horizon, EIGRP implements rule as it applies to the interface for which the Successor route was learned.

In EIGRP, split horizon suppresses a QUERY, where as Reverse Poison 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

5.4.7.3.1 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).

5.4.7.3.2 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.

5.4.7.3.3 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 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

```
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 links 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 result 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 results 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 currently 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

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.

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

5.6.1.1 Classic Composite Formulation

EIGRP calculates the composite metric with the following formula:

```
metric = {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 is the sum of all outbound interface delays along the path. The router dynamically measures reliability (REL) and load. It expresses 100 percent reliability as 255/255. It expresses load as a fraction of 255. An interface with no load is represented as 1/255.

Bandwidth is the inverse minimum bandwidth (in kbps) of the path in bits per second scaled by a factor of 256 multiplied by 10^7. The formula for bandwidth is

```
(256 x (10^7))/BWmin
```

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

```
[sum of delays] x 256
```

Reliability is a value between 1 and 255. Cisco IOS routers display reliability as a fraction of 255. That is, 255/255 is 100 percent reliability or a perfectly stable link; a value of 229/255 represents a 90 percent 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 percent saturated link.

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

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

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

The calculated EIGRP bandwidth (BW) metric is then:

```
256 x (10^7)/BW = 256 x {(10^7)/10,000}
                = 256 x 10,000
                = 256,00
```

And the calculated EIGRP delay metric is then:

```
256 x sum of delay = 256 x 100 x 10 microseconds
                   = 25,600 (in tens of microseconds)
```

5.5.1.2 Cisco Interface Delay Compatibility

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

Bandwidth (Kbps)	Classic Delay	Wide Metrics Delay	Interface Type
9	500000000	500000000	Tunnel
56	20000000	20000000	56Kb/s
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 155Mb/s
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 accommodate interfaces with high bandwidths, and to allow EIGRP to perform the path selection; the EIGRP packet and composite metric formula has been modified 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 maximum transmission unit (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

To this 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 which either does not monitor these attributes, or which 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 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 Designers 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 effect 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 percent 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:

```
                                           K2 * Max-Throughput
     Net-Throughput = Max-Throughput + ---------------------
                                              256 - 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, or 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:

```
                      Delay * EIGRP_WIDE_SCALE
     Latency = K3 *  --------------------------
                         EIGRP_DELAY_PICO
```

5.6.2.5 Composite Calculation

```
                                                             K5
   metric =[(K1*Net-Throughput) + Latency)+(K6*ExtAttr)] * ------
                                                           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 Autonomous system number and matching K-values.

6.2 Protocol Assignment Encoding

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 on the following types are used:

AFI Designation	AFI Value
IPv4 Address	1
IPv6 Address	2
Service Family Common	16384
Service Family IPv4	16385
Service Family IPv6	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 all three protocols, 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                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Acknowledgement 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 - EIGRP opcode indicating function packet serves. It will be one of the following values:

```
        EIGRP_OPC_UPDATE          1
        EIGRP_OPC_REQUEST         2
        EIGRP_OPC_QUERY           4
        EIGRP_OPC_REPLY           4
        EIGRP_OPC_HELLO           5
        Reserved                  6
        Reserved                  7
        Reserved                  8
        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 check-sum will be the standard ones complement of the ones complement sum. The packet is discarded if the packet checksum fails.

Flags - Defines special handling of the packet. There are currently two defined flag bits.

Init Flag (0x01) - This bit is set in the initial UPDATE sent to a newly discovered neighbor. It requests the neighbor to download a 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 (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 maintains the adjacency, and will set the RS flag in its UPDATE packet to indicated it is doing a soft restart.

EOT (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 state route is any route, which existed before the restart, and was not refreshed by the neighbor via and UPDATE.

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

ACK - The 32-bit sequence number that is being acknowledged with respect to 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 nonzero ACK field should be decoded as an ACK packet rather than a HELLO packet.

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

Value Range	Usage
0000	Unicast Address Family
0001	Multicast Address Family
0002-7FFFF	Reserved
8000	Unicast Service Family
8001-FFFF	Reserved

Autonomous System (AS) - 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.

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 allows multiple network layer protocols to carry independent information. Therefore, later if it is 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 inter-dependencies among them.

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
Multi-Protocol	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 does 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 AppleTalk, the value is 4. For Novell IPX, the value is 10, for IPv6 it is 16

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.

6.7.5 0x0005 - MULTICAST_SEQUENCE_TYPE

The next multicast sequence TLV

6.7.6 0x0006 - PEER_INFORMATION_TYPE

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

6.7.7 0x0007 - PEER_TERMAINATION_TYPE

This TLV is used in HELLO Packets to specify a given neighbor has been reset.

```
 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)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6.7.8 0x0008 - TID_LIST_TYPE

List of sub-topology identifiers, including the base topology, supported but 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 three-way handshake has finished

When the TID list received, it compares the list to the previous list sent. If a TID is found which 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, which 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 is 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 which 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 are 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 a along an unloaded path. This is expressed in units of 10s 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 maximum transmission unit 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.

Flag Field - See Section 6.8.1

6.8.3 Classic Exterior Encoding

Additional routing information so provided for destinations outside of the EIGRP autonomous system 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 Identification (RID)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Autonomous System Number (AS)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   External Protocol Metric                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |Extern Protocol|  Flags Field  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

Autonomous System (AS) - 32-bit number indicating the external autonomous system the sending router is a member of. If the source protocol is EIGRP, this field will be the [VRID|AS] pair.

External Protocol Metric - 32bit 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 - Defines the external protocol that this route was learned. See Section 6.2

Flag 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 o carry the destination address. The length is determined by the number of consecutive bits in the destination address, rounded up to the nearest octet boundary, determines the length of the address.

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 is represented by 4 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.

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in section 6.8.2

Destination Section - The network/subnet/host destination address being requested. See description of destination in section6.8.4

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 Section6.8.3)           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            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 is represented by 4 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

Exterior Section - Additional routing information provide for a destination outside of the autonomous system and that has been redistributed into the EIGRP. See Section 6.8.3

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in Section 6.8.2

Destination Section - The network/subnet/host destination address being requested. See 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)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Destination - The IPv4 address the community information should be stored with.

Community Length - 2 octet unsigned number that indicates the length of the Community List. The length does not includes the IPv4 Address, reserved, or Length fields

Community List - One or more 8 octet EIGRP community as defined in section 6.4

6.8.6 IPv6 Specific TLVs

```
REQUEST_TYPE                 0x0401
INTERNAL_TYPE                0x0402
EXTERNAL_TYPE                0x0403
```

6.8.6.1 IPv6 INTERNAL_TYPE

This TLV conveys 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                      |
|                 IPv4 Address (variable length)                |
|                      (See Section 6.8.4)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Hop Forwarding Address - IPv6 address is represented by 8 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.

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in section 6.8.2

Destination Section - The network/subnet/host destination address being requested. See 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. 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 Section6.8.3)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Vector Metric Section (See Section 6.8.2)         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
|                      Destination Section                      |
|                IPv4 Address (variable length)                 |
|                     (See Section 6.8.4)                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Next Hop Forwarding Address - IPv6 address is represented by 8 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 provide for a destination outside of the autonomous system and that has been redistributed into the EIGRP. See description of exterior encoding in Section 6.8.3

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in section 6.8.2

Destination Section - The network/subnet/host destination address being requested. See 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0x01     |      0x04     |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Destination                          |
|                          (16 octets)                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved           |        Community Length       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Community List                        |
|                       (variable length)                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Destination - The IPv6 address the community information should be stored with.

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

Community List - One or more 8 octet EIGRP community as defined in section 6.4

6.9 Multi-Protocol 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                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Acknowledgement 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 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, 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 32bit number provided by the router sourcing the information to uniquely identify it as the source.

6.9.2 Wide Metric Encoding

Multi-Protocol TLV's 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:

Offset - Number of 16bit words in the Extended Attribute section, 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 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 maximum transmission unit 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 allows for extensibility of the vector metrics in a manor similar to RFC 6390 [11]. Each Extended metric shall consist of a standard Type-Length header followed by application-specific information. Extended metrics 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 16bit words in the sub-field. Offset does not include the length of the opcode or offset fields)

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 Delay - An administrative parameter assigned statically on a per interface type basis to represent the time it takes a along an unloaded path. This is expressed in units of 10s of microseconds divvied by 256. A delay of 0xFFFFFFFF indicates an unreachable route.

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.

6.9.3.3 0x02 - Administrator Tag
This is used to provide and administrative tags for specific topology entries. It 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      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
This is used to provide communities for specific topology entries. It 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      0x03     |    Offset     |         Community List        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               |
|                         (variable length)                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Offset - Number of 16bit words in the sub-field. Currently transmitted as 4

Community List - One or more community values as defined in section 6.4

6.9.3.5 0x04 - Jitter

```
 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

```
 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

```
 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 4 AddPath 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 16bit words in the sub-field. Currently transmitted as 4

AddPath - Length of this field will vary in length based on weather 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 Address(Upper 2 byes)|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| IPv4 Address (Lower 2 byes)   |       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 - IPv4 address is represented by 4 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 32bit 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 2 addition bytes will be add as follows:
External Protocol - Defines the external protocol that this route was learned. See Section 6.2

Flag 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 - IPv6 address is represented by 8 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 32bit 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 2 addition bytes will be add as follows:

External Protocol - Defines the external protocol that this route was learned. See Section 6.2

Flag Field - See Section 6.8.1

6.9.4 Exterior Encoding

Additional routing information so provided for destinations outside of the EIGRP autonomous system 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 Identification (RID)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Autonomous System Number (AS)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   External Protocol Metric                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Reserved            |Extern Protocol|  Flags Field  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

Autonomous System (AS) - 32-bit number indicating the external autonomous system the sending router is a member of. If the source protocol is EIGRP, this field will be the [VRID|AS] pair.

External Protocol Metric - 32bit 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 - Defines the external protocol that this route was learned. See Section 6.2

Flag Field - See Section 6.8.1

6.9.5 Destination Encoding

Destination information is encoded in Multi-Protocol packets in the same manner as 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 o carry the destination address. The length is determined by the number of consecutive bits in the destination address, rounded up to the nearest octet boundary, determines the length of the address.

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

By the nature of 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 more heavy weight 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 though the use of MD5 or SHA2-256 [6].

8 IANA Considerations

This document serves as the sole reference for two multicast addresses; IGRP Routers [13], EIGRP Routers [14] and 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, April 1997.

[2] J.J. Garcia-Luna-Aceves, "A Unified Approach to Loop-Free Routing using Distance Vectors or Link States", 1989 ACM 089791-332-9/89/0009/0212, pages 212-223.

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

[4] Rosen, E., "IANA Registries for BGP Extended Communities", RFC 7153, March 2014.

[5] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", RFC 3692, January 2004

[6] Kelly, S., Frankel, S., "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec", RFC 4868, May 2007.

[7] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989

[8] "DARPA Internet Protocol Specification", RFC 791, Sept 1981

[9] [7] Deering, S., Hinden, R., "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998

9.2 Informative References

[10] Moy, J., "OSPF Version 2" RFC 2328, 1998

[11] Clark, A., Claise, B., "Guidelines for Considering New Performance Metric Development", RFC 6390, October 2011

[12] Address Family Numbers, http://www.iana.org/assignments/address-family-numbers/address-family-numbers.xhtml

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

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

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

10 Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

An initial thank you goes to Dino Farinacci, Bob Albrightson, and Dave Katz. Their significant accomplishments towards the design and development of the EIGRP protocol 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 lead the evolution of EIGRP over the following 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, I would like to personally recognise the team what crated the first Open Source verison of EIGRP. This team comprises of: Jan Janovic, Matej Perina, Peter Orsag, and Peter Paluch who made it all possible.

Author's Address

Donnie V Savage
Cisco Systems, Inc
7025 Kit Creed Rd, RTP, NC

Phone: 919-392-2379
Email: dsavage@cisco.com

Donald Slice
Cumulus Networks
Apex, NC

Phone:
Email: dslice@cumulusnetworks.com

James Ng
Cisco Systems, Inc
7025 Kit Creed Rd, RTP, NC

Phone: 919-392-2582
Email: jamng@cisco.com

Peter Paluch
University of Zilina
Univerzitna 8215/1, Zilina 01026, Slovakia

Phone: 421-905-164432
Email: Peter.Paluch@fri.uniza.sk

Steven Moore
Cisco Systems, Inc
7025 Kit Creed Rd, RTP, NC

Phone: 919-392-2674
Email: smoore@cisco.com

Russ White
Ericsson
Apex, NC

Phone: 1-877-308-0993
Email: russw@riw.us