

Network Working Group  
Internet-Draft  
Intended status: Informational  
Expires: May, 2014

D. Savage  
D. Slice  
J. Ng  
S. Moore  
Cisco Systems  
7 Oct 2013  
R. White  
VCE  
7 Oct 2013

**Enhanced Interior Gateway Routing Protocol  
draft-savage-eigrp-01.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[4]. The algorithm and procedures were researched, developed, and simulated by SRI International.

#### Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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."

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>.

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

This Internet-Draft will expire on May 7, 2014 .

#### Copyright Notice

Copyright (c) 2013 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.

#### Conventions used in this document

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].

## Table of Contents

<a href="#">1</a>	<a href="#">Introduction .....</a>	<a href="#">5</a>
<a href="#">2</a>	<a href="#">Terminology .....</a>	<a href="#">5</a>
<a href="#">3</a>	<a href="#">The DUAL Diffusing Update Algorithm .....</a>	<a href="#">8</a>
	<a href="#">3.1 Algorithm Description .....</a>	<a href="#">8</a>
	<a href="#">3.2 Route States .....</a>	<a href="#">8</a>
	<a href="#">3.3 Feasibility Condition .....</a>	<a href="#">9</a>
	<a href="#">3.4 DUAL Message Types .....</a>	<a href="#">10</a>
	<a href="#">3.5 Dual Finite State Machine (FSM) .....</a>	<a href="#">10</a>
	<a href="#">3.6 DUAL Operation - Example Topology .....</a>	<a href="#">14</a>
<a href="#">4</a>	<a href="#">EIGRP Packets .....</a>	<a href="#">17</a>
	<a href="#">4.1 UPDATE Packets .....</a>	<a href="#">17</a>
	<a href="#">4.2 QUERY Packets .....</a>	<a href="#">18</a>
	<a href="#">4.3 REPLY Packets .....</a>	<a href="#">18</a>
	<a href="#">4.4 Exception Handling .....</a>	<a href="#">18</a>
	<a href="#">4.4.1 Active Route Duration control .....</a>	<a href="#">18</a>
	<a href="#">4.4.2 Stuck-in-Active .....</a>	<a href="#">19</a>
	<a href="#">4.4.3 SIA-QUERY .....</a>	<a href="#">19</a>
	<a href="#">4.4.4 SIA-REPLY .....</a>	<a href="#">20</a>
<a href="#">5</a>	<a href="#">EIGRP Protocol Operation .....</a>	<a href="#">21</a>
	<a href="#">5.1 Finite State Machine .....</a>	<a href="#">21</a>
	<a href="#">5.2 Reliable Transport Protocol .....</a>	<a href="#">21</a>
	<a href="#">5.2.1 Bandwidth on Low-Speed Links .....</a>	<a href="#">28</a>
	<a href="#">5.3 Neighbor Discovery/Recovery .....</a>	<a href="#">28</a>
	<a href="#">5.3.1 Neighbor HoldTime .....</a>	<a href="#">28</a>
	<a href="#">5.3.2 HELLO Packets .....</a>	<a href="#">28</a>
	<a href="#">5.3.3 UPDATE Packets .....</a>	<a href="#">29</a>
	<a href="#">5.3.4 Initialization Sequence .....</a>	<a href="#">30</a>
	<a href="#">5.3.5 QUERY Packets During Neighbor Formation .....</a>	<a href="#">31</a>
	<a href="#">5.3.6 Neighbor Formation .....</a>	<a href="#">31</a>
	<a href="#">5.3.7 Topology Table .....</a>	<a href="#">32</a>
	<a href="#">5.3.8 Route Management .....</a>	<a href="#">32</a>
	<a href="#">5.4 EIGRP Metric Coefficients .....</a>	<a href="#">33</a>
	<a href="#">5.4.1 Coefficients K1 and K2 .....</a>	<a href="#">34</a>
	<a href="#">5.4.2 Coefficients K3 .....</a>	<a href="#">34</a>
	<a href="#">5.4.3 Coefficients K4 and K5 .....</a>	<a href="#">34</a>
	<a href="#">5.4.4 Coefficients K6 .....</a>	<a href="#">35</a>
	<a href="#">5.4.4.1 Jitter .....</a>	<a href="#">35</a>
	<a href="#">5.4.4.2 Energy .....</a>	<a href="#">35</a>
	<a href="#">5.5 EIGRP Metric Calculations .....</a>	<a href="#">35</a>
	<a href="#">5.5.1 Classic Metrics .....</a>	<a href="#">35</a>
	<a href="#">5.5.1.1 Classic Composite Formulation .....</a>	<a href="#">36</a>
	<a href="#">5.5.1.2 Cisco Interface Delay Compatibility .....</a>	<a href="#">38</a>
	<a href="#">5.5.2 Wide Metrics .....</a>	<a href="#">38</a>
	<a href="#">5.5.2.1 Wide Metric Vectors .....</a>	<a href="#">38</a>
	<a href="#">5.5.2.2 Wide Metric Conversion Constants .....</a>	<a href="#">40</a>
	<a href="#">5.5.2.3 Throughput Formulation .....</a>	<a href="#">40</a>
	<a href="#">5.5.2.4 Latency Formulation .....</a>	<a href="#">40</a>

	<a href="#">5.5.2.5</a>	Composite Formulation .....	<a href="#">41</a>
<a href="#">6</a>		Security Considerations .....	<a href="#">41</a>
<a href="#">7</a>		IANA Considerations .....	<a href="#">41</a>
<a href="#">8</a>		References .....	<a href="#">42</a>
	<a href="#">8.1</a>	Normative References .....	<a href="#">42</a>
	<a href="#">8.2</a>	Informative References .....	<a href="#">42</a>
<a href="#">9</a>		Acknowledgments .....	<a href="#">43</a>

<a href="#">A</a>	EIGRP Packet Formats .....	<a href="#">44</a>
<a href="#">A.1</a>	Protocol Number .....	<a href="#">44</a>
<a href="#">A.2</a>	Protocol Assignment Encoding .....	<a href="#">44</a>
<a href="#">A.3</a>	Destination Assignment Encoding .....	<a href="#">45</a>
<a href="#">A.4</a>	EIGRP Communities Attribute .....	<a href="#">45</a>
<a href="#">A.5</a>	EIGRP Packet Header .....	<a href="#">46</a>
<a href="#">A.6</a>	EIGRP TLV Encoding Format .....	<a href="#">48</a>
<a href="#">A.6.1</a>	Type Field Encoding .....	<a href="#">48</a>
<a href="#">A.6.2</a>	Length Field Encoding .....	<a href="#">48</a>
<a href="#">A.6.3</a>	Value Field Encoding .....	<a href="#">49</a>
<a href="#">A.7</a>	EIGRP Generic TLV Definitions .....	<a href="#">49</a>
<a href="#">A.7.1</a>	0x0001 - PARAMETER_TYPE .....	<a href="#">49</a>
<a href="#">A.7.2</a>	0x0002 - AUTHENTICATION_TYPE .....	<a href="#">50</a>
<a href="#">A.7.3</a>	0x0003 - SEQUENCE_TYPE .....	<a href="#">50</a>
<a href="#">A.7.4</a>	0x0004 - SOFTWARE_VERSION_TYPE .....	<a href="#">51</a>
<a href="#">A.7.5</a>	0x0005 - MULTICAST_SEQUENCE_TYPE .....	<a href="#">51</a>
<a href="#">A.7.6</a>	0x0006 - PEER_INFORMATION_TYPE .....	<a href="#">51</a>
<a href="#">A.7.7</a>	0x0007 - PEER_TERMINATION_TYPE .....	<a href="#">51</a>
<a href="#">A.7.8</a>	0x0008 - TID_LIST_TYPE .....	<a href="#">52</a>
<a href="#">A.8</a>	Classic Route Information TLV Types .....	<a href="#">52</a>
<a href="#">A.8.1</a>	Classic Flag Field Encoding .....	<a href="#">52</a>
<a href="#">A.8.2</a>	Classic Metric Encoding .....	<a href="#">53</a>
<a href="#">A.8.3</a>	Classic Exterior Encoding .....	<a href="#">54</a>
<a href="#">A.8.4</a>	Classic Destination Encoding .....	<a href="#">55</a>
<a href="#">A.8.5</a>	IPv4 Specific TLVs .....	<a href="#">55</a>
<a href="#">A.8.6</a>	IPv6 Specific TLVs .....	<a href="#">58</a>
<a href="#">A.9</a>	Multi-Protocol Route Information TLV Types .....	<a href="#">61</a>
<a href="#">A.9.1</a>	TLV Header Encoding .....	<a href="#">61</a>
<a href="#">A.9.2</a>	Wide Metric Encoding .....	<a href="#">62</a>
<a href="#">A.9.3</a>	Extended Metrics .....	<a href="#">63</a>
<a href="#">A.9.4</a>	Exterior Encoding .....	<a href="#">69</a>
<a href="#">A.9.5</a>	Destination Encoding .....	<a href="#">70</a>
<a href="#">A.9.6</a>	Route Information .....	<a href="#">70</a>

## **1 Introduction**

This document describes the Enhanced Interior Gateway Routing Protocol (EIGRP), routing protocol designed and developed by Cisco Systems. The convergence technology is based on research conducted at SRI International. The Diffusing Update Algorithm (DUAL) is the algorithm used to obtain loop-freedom at every instant throughout a route computation[3]. 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:

### **EIGRP**

Enhanced Interior Gateway Routing Protocol.

### **Active state**

A route that is currently in an unresolved or un-converged state. The term active is used because the router is actively attempting to compute an SDAG.

### **Address Family Identifier (AFI)**

A term used to describe an address encoding in a packet. An address family currently pertains to an IPv4 or IPv6 address. See [[RFC3232](#)] for details.

### **Autonomous System (AS)**

A routing sub-domain representing a logical set of network segments and attached devices.

### **Base Topology**

The entire network itself, for which the usual set of routes is calculated, is known as the base topology. The base topology (or underlying network) is characterized by the Network Layer Reachability Information (NLRI) that a router uses to calculate the global routing table to make routing and forwarding decisions.

#### Downstream Router

A router that is one or more hops away in the direction of the destination.

#### 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.

#### Feasibility Condition

The feasibility condition is met when the minimum of all total cost found (where total cost is the sum of neighbor's cost and the link cost to that neighbor), and the neighbor's advertise cost is less than the current successors cost. This is the Source Node Condition (SNC) cited in reference [2].

#### Feasible Successor

A route describing reachability through a specific neighbor that meets the feasibility condition.

#### Neighbor / Peer

Two routers that have interfaces connected to a common subnet are known as adjacent neighbors. Two routers that are multiple hops apart on a common subnet are known as remote neighbors. Neighbors dynamically discover each other and exchange EIGRP protocol messages. Each router maintains a topology table containing information learned from each of its neighbors.

#### Passive state

A route is considered in passive state when there are one or more minimal cost feasible successors that can reach a destination. The term passive is used because the router is not actively computing a shortest path SDAG for this destination. A route in passive state is usable for forwarding data packets.

#### PE Router / Provider Edge Router

This is the device that logically sits on the provider side of the provider/customer demarcation in a network topology.

#### Routing Information Base (RIB) / Routing Table

A table where a router stores network destinations associated with a next-hop to reach particular network destinations and the metric associated with the route.





**Subsequent-Address Family Identifier (SAFI)**

Unicast and Multicast are examples of a Subsequent-Address Family Identifier.

**Successor Directed Acyclic Graph (SDAG)**

When a route to a destination becomes unreachable, it is required that a router computes a directed graph with respect to the destination. This decision requires the router to select from the neighbor's topology table a feasible successor.

**Sub-Topology**

A sub-topology is characterized by an independent set of routers and links in a network, for which EIGRP performs an independent path calculations. This allows each sub-topology to implement class-specific topologies to carry class specific traffic.

**Successor**

The unique neighboring router that has met the feasibility condition and has been selected as the next-hop for forwarding packets.

**Topology Identifier (TID)**

A number that is used to mark prefixes as belonging to a specific sub-topology.

**Type, Length, Value (TLV)**

An encoding format used by EIGRP. Each attribute present in a routing packet is tagged. The tag determines the type and length of information in the value portion of the attribute. This format allows extensibility and backward compatibility

**Upstream Router**

Any router that is one or multiple hops in the direction of the source of the information.

**Reported Distance (RD)**

Total metric along a path to a destination network as advertised by an upstream neighbor.

**Feasible Distance (FD)**

Defined as the best metric along a path to a destination network, including the metric to the neighbor advertising the path.

### **3 The DUAL Diffusing Update Algorithm**

The Diffusing Update Algorithm (DUAL) provides a loop-free path through a network made up of nodes and edges (routers and links) at every instant throughout a route computation. This allows all involved in a topology change to compute a best path in a distributed (diffusing) way, so calculations are performed in parallel. 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**

DDUAL is used by EIGRP to achieve fast loop-free convergence with little cost overhead, allowing EIGRP to provide convergence rates comparable, and in some cases better than, most common link state protocols[7]. "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 Bellman-Ford distance vector protocols, DUAL uses an approach to propagation of routing information with feedback known as diffusing computations. The diffusing computation grows by including nodes that are affected by the topology change and shrinks by excluding ones that are not. This allows the computation to dynamically adjust in scope and terminate as soon as possible.

#### **3.2 Route States**

A topology table entry for a destination can have one of two states, Passive and Active. A route transitions its state when there is a topology change in the network. This can be caused by link failure, node failure, or a link cost increase. The two states are as follow:

- o **Passive**  
A route is considered in the Passive state when a router is not performing a route recalculation. When a route is in passive state it is usable and the next hop is perceived to be downstream of the destination.
- o **Active**  
A destination is in Active state when a router is computing a Successor Directed Acyclic Graph (SDAG) for the destination.

While a router has a route in active state, it records the new metric information but does not make any routing decisions until it goes back to passive state. A route goes from active state to passive state when a router receives responses from all of its neighbors and the diffusing computation is complete.

If an alternate loop free path exists for the route, the neighbor WILL NOT go into the Active state avoiding a route recalculation. When there are no feasible successors, a route goes into Active state and a route recalculation must occur.

### **3.3 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. 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.

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.

The Feasibility Condition is met when a neighbor's advertised cost (RD) to a destination is less than the cost of that same destination through the current successor (or best path, FD). 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, Queries, Updates, and Replies:

- 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

When in passive state, a received query may be propagated if there are no feasible successors 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. Figure [1](#) illustrates the FSM:



*i* Node that is computing route.  
*j* Destination node or network.  
*K* Any neighbor of node *i*.  
*oi j* 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.  
*li k* The link connecting node *i* to neighbor *k*.  
*FS* Feasible Successor

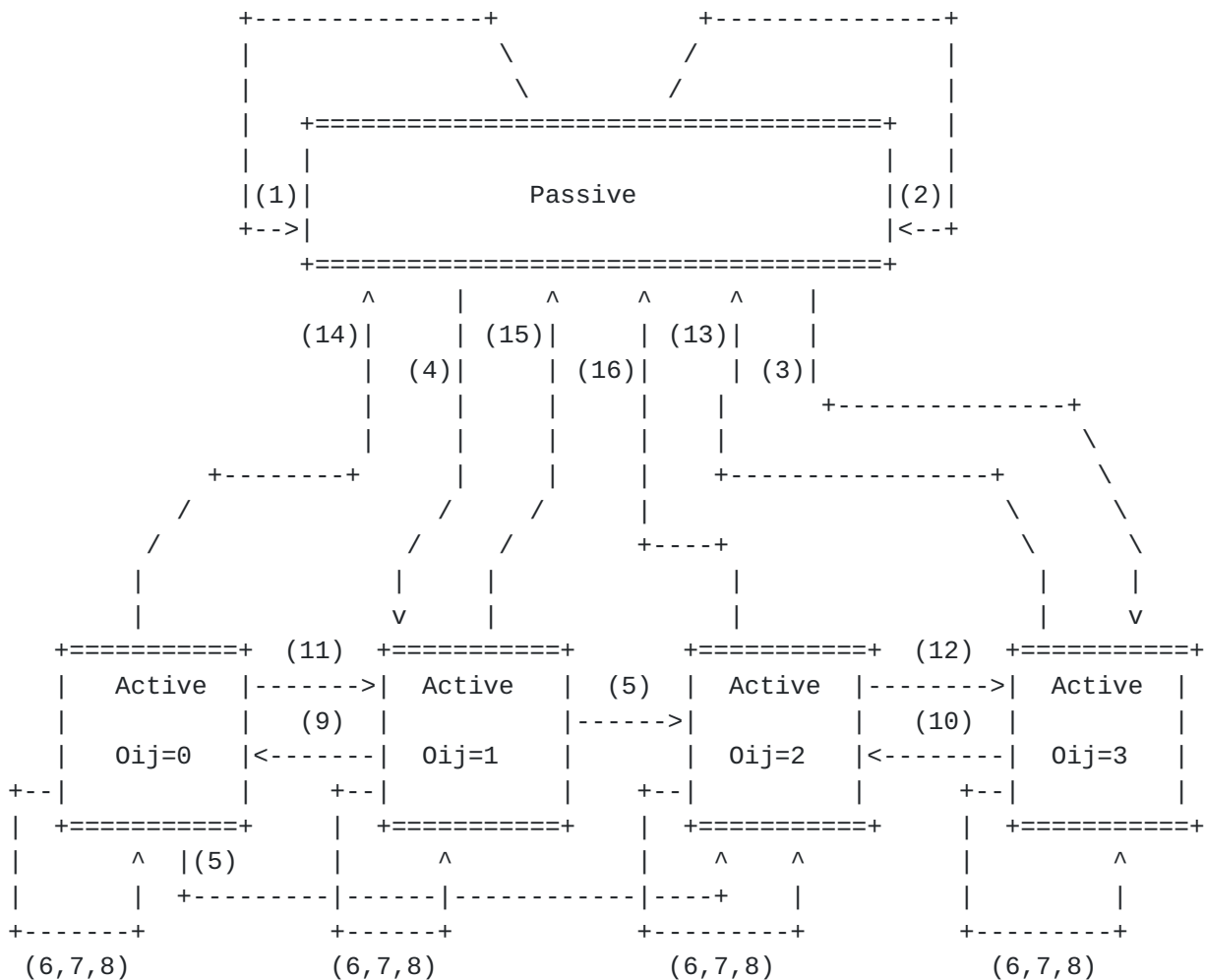


Figure 1 - DUAL Finite State Machine



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. A feasible successor exists since the successor was not affected, so the route remains in passive state. Since a feasible successor exists, a REPLY is required to 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 had 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. 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 advertised in the QUERY should be recorded.





(7) If a link cost change 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, a QUERY and UPDATE is never 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. 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 and a link cost increase to the successor occurred, 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.

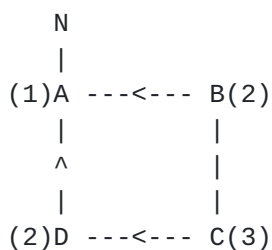


Figure 2 - Stable Topology

Now consider the case where the link between A and D fails (Figure 3). 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. Note that node A and B were not involved in the recalculation since they were not affected by the change.

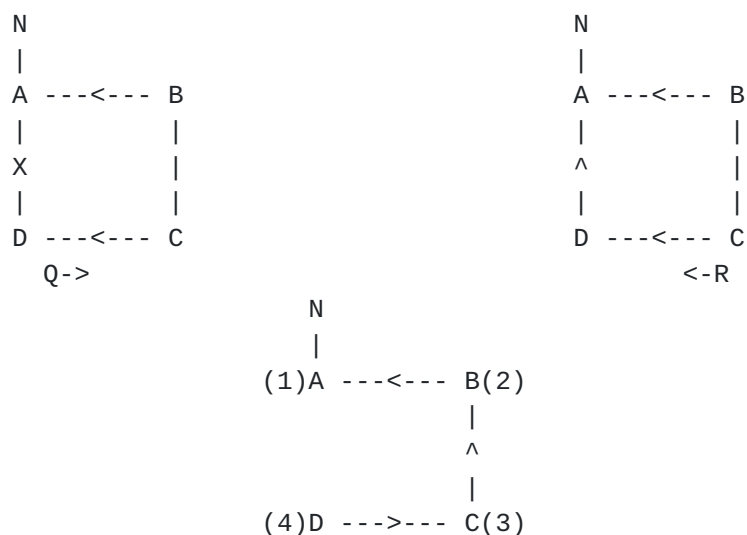


Figure 3 - Link between A and D fails

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.



Figure 4  
No Feasible Successors when link between A and B fails

## **4 EIGRP Packets**

EIGRP uses 5 different packet types to operate.

- o HELLO/Ack Packets
- o QUERY Packets
- o UPDATE Packets
- o REPLY Packets

EIGRP packets will be encapsulated in the respective network layer protocol that it is supporting. Since EIGRP is potentially capable of running in an integrated mode the encapsulation is not specified.

Support for network layer protocol fragmentation is supported, though EIGRP will attempt to avoid 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 will be used.

- IPv4 - 224.0.0.10
- IPv6 - FF02:0:0:0:0:0:0:A

The above 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 are used to convey destinations, and the reachability of the 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 sent by a router advertises 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 go unreachable, 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 the packets are guaranteed reliable all route QUERY packets are guaranteed reliable.

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 will be sent in response to a QUERY or SIA-QUERY packet, if the router believes it has an alternate feasible successor. The REPLY packet will include a TLV for each destination and the associated vectorized 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 Ack packet is sent immediately and then the packet is processed.

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 Route Duration control**

When an EIGRP router transitions to ACTIVE state for a particular destination a QUERY is sent to all neighbors and the ACTIVE timer is started to limit the amount of time a destination may remain in an active state. The default time DUAL is allowed to stay active, trying to resolve a path to a destination, is a maximum of six (6) minutes. This is broken into an initial 90 seconds period following the QUERY, and up to 3 additional "busy" periods in which a SIA-QUERY is sent. Failure to respond to a SIA-QUERY within the 90 second will result in the neighbor being declared in the Stuck In Active(SIA) state.

#### **4.4.2 Stuck-in-Active**

A route is regarded as Stuck-In-Active(SIA) when DUAL does not receive a reply to the active process. This process is begun when a QUERY is sent by. After the initial 90 seconds, the router will send a SIA-QUERY, this must be replied to with either a REPLY or SIA-REPLY. Failure of a neighbor to send either a REPLY or SIA-REPLY with-in the **90 seconds will result in the neighbor being deemed to be in an SIA** state. If the SIA state is declared, DUAL will then delete all routes from that neighbor and resets adjacency with that neighbor, acting as if the neighbor had responded with an unreachable message for all routes.

#### **4.4.3 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 neighbor device is still attempting to converge on the active route, an EIGRP router MAY send a SIA-QUERY packet to the active neighbors. 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 a SIA-QUERY, the originating router may extend the effective active time by resetting the Active timer which has been previously set and thus allow 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 a SIA-REPLY.

Upon receipt of a 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 a 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 a SIA-REPLY with the Active bit set

#### **4.4.4 SIA-REPLY**

A SIA-REPLY packet is the corresponding response upon receipt of a 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 3](#)

### **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, neighborhood 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.

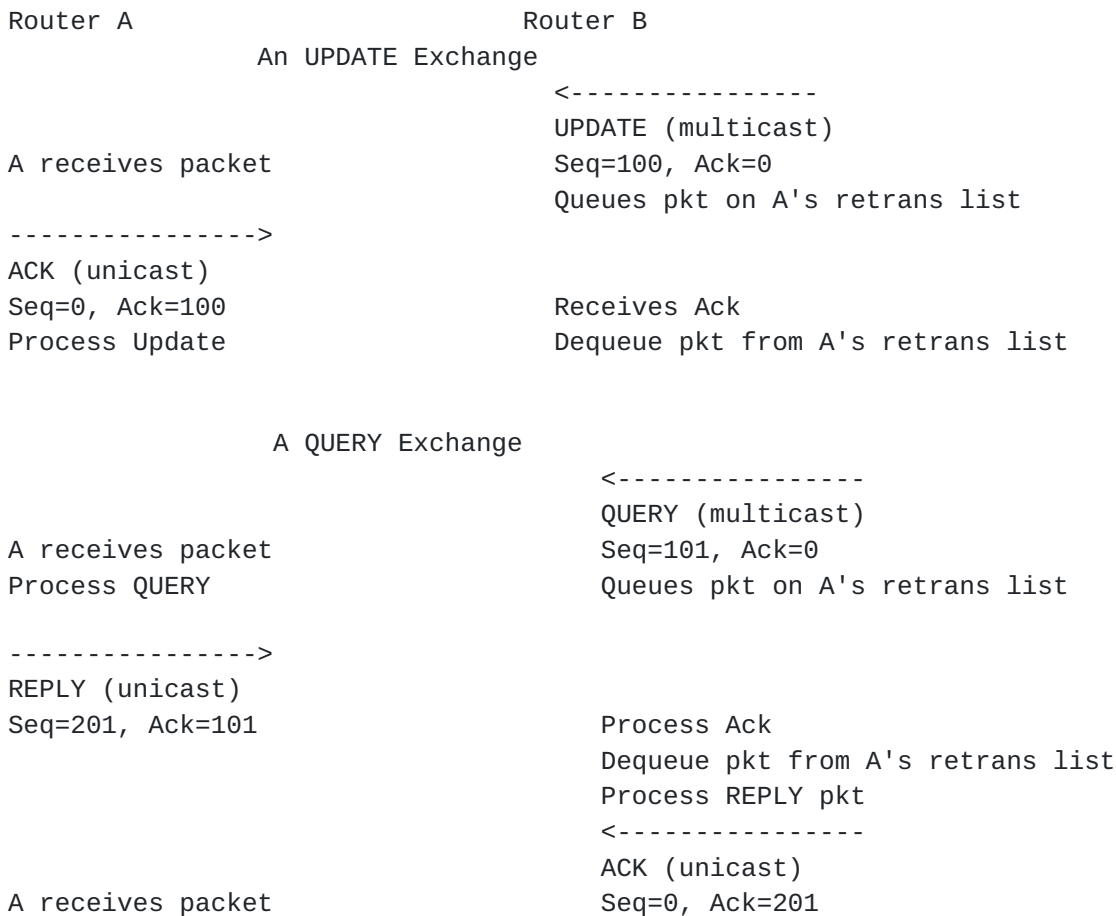


Figure 5 - Reliable Transfer on point-to-point links

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.

Figure 6 illustrates the situation where there is heavy packet loss on a network.

Router A	Router B
	<-----
A receives packet	UPDATE (multicast)
	Seq=100, Ack=0
	Queues pkt on A's retrans list
----->	
ACK (unicast)	
Seq=0, Ack=100	Receives Ack
Process Update	Dequeue pkt from A's retrans list
	<--/LOST/-----
	UPDATE (multicast)
	Seq=101, Ack=0
	Queues pkt on A's retrans list
	Retransmit Timer Expires
	<-----
	Retransmit UPDATE (unicast)
	Seq=101, Ack=0
	Keeps pkt on A's retrans list
----->	
ACK (unicast)	
Seq=0, Ack=101	Receives Ack
Process Update	Dequeue pkt from A's retrans 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 UPDATE Exchange

```

<-----
A send UPDATE (multicast)
Seq=100, Ack=0
Queues pkt on B's retrans list
Queues pkt on C's retrans list
Queues pkt on D's retrans list

----->
B sends ACK (unicast)
Seq=0, Ack=100
Process Update

Receives Ack
Dequeue pkt from B's retrans list

----->
C sends ACK (unicast)
Seq=0, Ack=100
Process Update

Receives Ack
Dequeue pkt from C's retrans list

----->
D sends ACK (unicast)
Seq=0, Ack=100
Process Update

Receives Ack
Dequeue pkt from D's retrans list

```

## A QUERY Exchange

```

<-----
A send UPDATE (multicast)
Seq=101, Ack=0
Queues pkt on B's retrans list
Queues pkt on C's retrans list
Queues pkt on D's retrans list

----->
B send REPLY (unicast)
Seq=511, Ack=101
Process Update

<-----
A sends Ack (unicast to B)
Seq=0, Ack=511
Dequeue pkt from B's retrans list

----->
C send REPLY (unicast)
Seq=200, Ack=101
Process Update

<-----
A sends Ack (unicast to C)
Seq=0, Ack=200
Dequeue pkt from C's retrans list

----->
D send REPLY (unicast)
Seq=11, Ack=101
Process Update

<-----
A sends Ack (unicast to D)
Seq=0, Ack=11
Dequeue pkt from D's retrans 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
Queues pkt on B's retrans list
Queues pkt on C's retrans list
Queues pkt on D's retrans list

-----/LOST/->
B send ACK (unicast)
Seq=0, Ack=100

----->
C sends ACK (unicast)
Seq=0, Ack=100
Dequeue pkt from C's retrans list

----->
D sends ACK (unicast)
Seq=0, Ack=100
Dequeue pkt from D's retrans 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
Queues pkt on B's retrans list
Queues pkt on C's retrans list
Queues pkt on D's retrans list

-----/LOST/->
B send ACK (unicast)
Seq=0, Ack=100

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)
Seq=0, Ack=100

                                     A removes pkt from retrans list

                                     <-----
                                     A resends UPDATE (unicast to B)
                                     Seq=101, Ack=0

----->
B sends ACK (unicast)
Seq=0, Ack=101

                                     A removes pkt from retrans list

```

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 is REQUIRED to tell that neighbor to not receive the next packet or it would receive it 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 multicasted unreliably out the interface. Router-C and Router-D process the SEQUENCE\_TYPE TLV by looking for its own address in the list. If it is not found, they put themselves in Conditionally Received(CR-mode) mode. Any subsequent packets received that have the CR-flag set can be received. Router-B does not put itself in CR-mode because it finds itself in the list. 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 packet 101 and acknowledges it too. Router-A can remove both packets off Router-B's transmission list. Next time when Router-A has an update to be sent to its neighbors, it sees that B is up to date w.r.t the updates it has to receive and it wouldn't get any Unicast packets(CR-Mode).



### **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 HoldTime**

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 HoldTime. The HoldTime is the amount of time a router treats a neighbor as reachable and operational. In other words, if a HELLO packet isn't heard within the HoldTime, then the HoldTime expires. When the HoldTime 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 for 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 HoldTime value. This value indicates to all receivers the length of time in seconds that the neighbor is valid. The default HoldTime 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 HoldTime. 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 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.

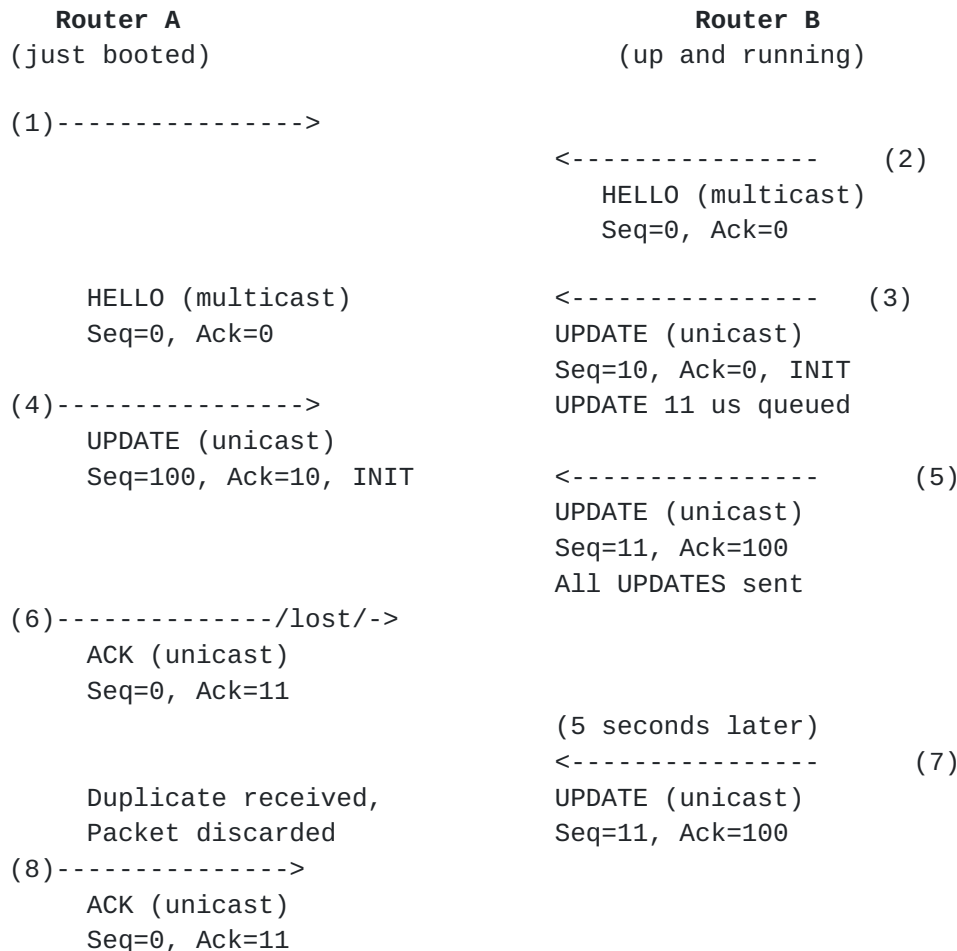
**5.3.4 Initialization Sequence**

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. The number of destinations in its routing table will require at least 2 UPDATE packets to be sent. The first UPDATE (referred it as the NULL UPDATE) is sent with the INIT-Flag, and congaing no topology information. The second packet is queued, and cannot be sent until the first is acknowledged.

(3) Router A receives first UPDATE 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 from Router B and acknowledges it. The acknowledgment gets lost.

(6) Router B later retransmits the UPDATE 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 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.

### **5.3.6 Neighbor Formation**

To prevent packets from being sent to a neighbor prior to the multicast and unicast delivery has been verified as 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 SHALL place the newly discovered neighbor in a "pending" state as follows:

- o 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
- o While Router-B is in this state, A will not send it any a QUERY or UPDATE
- o When Router-A receives the unicast acknowledgement from Router-B, it will check the state from pending to up

### **5.3.7 Topology Table**

The Topology Table is populated by the protocol dependent modules and acted upon by the DUAL finite state machine. It contains all destinations advertised by neighboring routers. Associated with each entry are the destination address and a list of neighbors that have advertised this destination. For each neighbor, the advertised metric is recorded. This is the metric that the neighbor stores in its routing table. If the neighbor is advertising this destination, it must be using the route to forward packets. This is an important rule that distance vector protocols MUST follow.

Also associated with the destination is the metric that the router uses to reach the destination. This is the sum of the best-advertised metric from all neighbors plus the link cost to the best neighbor. This is the metric that the router uses in the routing table and to advertise to other routers.

### **5.3.8 Route Management**

EIGRP has the notion of internal and external routes. Internal routes are ones that have been originated within an EIGRP Autonomous System(AS). Therefore, a directly attached network that is configured to run EIGRP is considered an internal route and is propagated with this information throughout the network topology.

External routes are destinations that have been learned though another source, such as a 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 The router ID of the EIGRP router that redistributed the route.
- o The AS number where the destination resides.
- o A configurable administrator tag.
- o Protocol ID of the external protocol.
- o The 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) 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 more global policies.

#### **5.4 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.

#### **5.4.1 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.

#### **5.4.2 Coefficients 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.

#### **5.4.3 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.

#### **5.4.4 Coefficients 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.

##### **5.4.4.1 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.

##### **5.4.4.2 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).

## **5.5 EIGRP Metric Calculations**

### **5.5.1 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.

#### **5.5.1.1 Classic Composite Formulation**

EIGRP calculates the composite metric with the following formula:

$$\text{metric} = \{K1 \cdot BW + [(K2 \cdot BW) / (256 - \text{load})] + (K3 \cdot \text{delay})\} \cdot \{K5 / (\text{reliability} + K4)\}$$

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 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

$$(256 \times (10^7)) / BW_{\min}$$

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

$$[\text{sum of delays}] \times 256$$

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:

$$\text{metric} = 256 \times \{ [(10^7) / BW_{\min}] + [\text{sum of delays}] \}$$

Minimum Bandwidth ( $BW_{\min}$ ) 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 BW metric is then:

$$\begin{aligned} 256 \times (10^7)/BW &= 256 \times \{(10^7)/10,000\} \\ &= 256 \times 10,000 \\ &= 256,000 \end{aligned}$$

And the calculated EIGRP delay metric is then:

$$\begin{aligned} 256 \times \text{sum of delay} &= 256 \times 1 \text{ ms} \\ &= 256 \times 100 \times 10 \text{ microseconds} \\ &= 25,600 \text{ (in tens of microseconds)} \end{aligned}$$



### **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

### **5.5.2 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.

#### **5.5.2.1 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 through interface bandwidth/delay configuration, the modification of bandwidth is discouraged for following reasons:

- 1. The change will only effect the path selection if the configured value is the lowest bandwidth over the entire path.**
- 2. 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.
- 3. EIGRP throttles to use 50 percent of the configured bandwidth.** Lowering the bandwidth can cause problems like starving EIGRP neighbors from getting packets because of the throttling back.

Changing the delay does not impact other protocols nor does it cause EIGRP to throttle back, and because, as it's the sum of all delays, has a direct effect on path selection.

### 5.5.2.2 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
EIGRP_RIB_SCALE	128

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.

### 5.5.2.3 Throughput Formulation

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

$$\text{Max-Throughput} = K1 * \frac{(\text{EIGRP\_BANDWIDTH} * \text{EIGRP\_WIDE\_SCALE})}{\text{Interface Bandwidth (kbps)}}$$

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

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

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

### 5.5.2.4 Latency Formulation

Transmission times derived from physical interfaces MUST be in 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:

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

#### **5.5.2.5 Composite Formulation**

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

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:

$$\text{metric} = (K1 * \min(\text{Throughput})) + (K3 * \sum(\text{Latency})) \}$$

## **6 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, DoS attack prevention will depend on implementations rate-limiting packets to the control plane as well as authentication of the neighbor through the use of SHA2-256

## **7 IANA Considerations**

This document has no actions for IANA.

## **8 References**

### **8.1 Normative References**

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [[RFC2119](#)], April 1997.
- [2] Crocker, D. and Overell, P.(Editors), "Augmented BNF for Syntax Specifications: ABNF", [[RFC2234](#)], Internet Mail Consortium and Demon Internet Ltd., November 1997.
- [3] A Unified Approach to Loop-Free Routing using Distance Vectors or Link States, J.J. Garcia-Luna-Aceves, 1989 ACM 089791-332-9/89/0009/0212, pages 212-223.
- [4] Loop-Free Routing using Diffusing Computations, J.J. Garcia-Luna-Aceves, Network Information Systems Center, SRI International to appear in IEEE/ACM Transactions on Networking, Vol. 1, No. 1, 1993.
- [5] BGP Extended Communities Attribute [[RFC4360](#)]
- [6] HMAC-SHA256, SHA384, SHA512 in IPsec [[RFC4868](#)]

### **8.2 Informative References**

- [7] OSPF Version 2, Network Working Group [[RFC1247](#)], J. Moy, July 1991.
- [8] Guidelines for Considering New Performance Metric Development [[RFC6390](#)]

## **9 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.

## A EIGRP Packet Formats

### [A.1 Protocol Number](#)

The IPv6 and IPv4 protocol identifier number spaces are common and will both use protocol identifier 88.

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 FF02::A (the EIGRP IPv6 multicast address). 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.

### [A.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

A.3 Destination Assignment Encoding

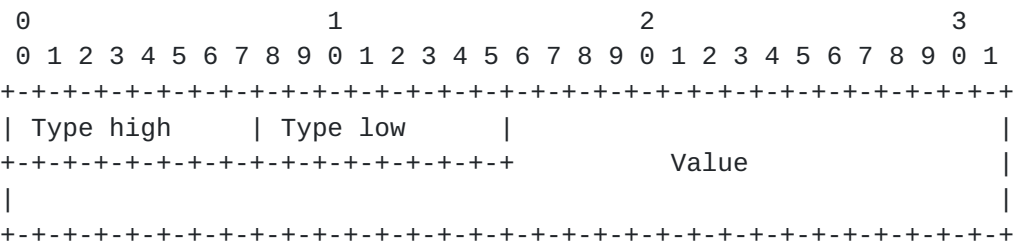
Destinations types are encoded according to the IANA address family number assignments. Currently on the following types are used:

AFI Designation	AFI Value
-----	
IPv4 Address	1
IPv6 Address	2
Service Family Common	16384
Service Family IPv4	16385
Service Family IPv6	16386

A.4 EIGRP Communities Attribute

EIGRP supports communities similar to the BGP Extended Communities [5] 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: 1 or 2 octets
- Value Field: Remaining octets



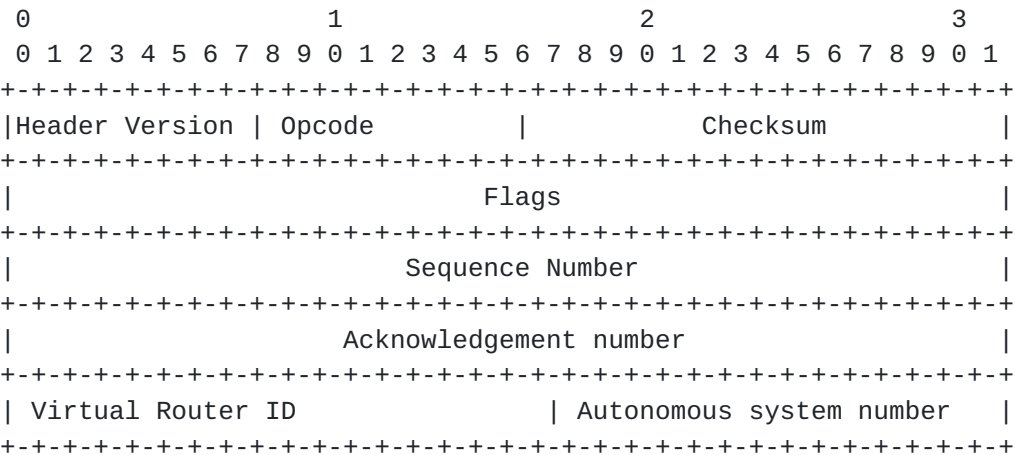
In addition to well-known communities supported by BGP (such as Site of Origin), EIGRP defines a number of additional defined Community values as follows:

Value	Name	Description
-----		
8800	EXTCOMM_EIGRP	EIGRP route information appended
8801	EXTCOMM_DAD	Data: AS + Delay
8802	EXTCOMM_VRHB	Vector: Reliability + Hop + BW
8803	EXTCOMM_SRLM	System: Reserve +Load + MTU
8804	EXTCOMM_SAR	System: Remote AS + Remote ID
8805	EXTCOMM_RPM	Remote: Protocol + Metric
8806	EXTCOMM_VRR	Vecmet: Rsvd + RouterID



A.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.



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
EIGRP_OPC_PROBE	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 packet 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 init UPDATE packets during the signaling period. The router looks at the RS flag to detect if a neighbor is restarting and maintain the adjacency. A restarting router looks at this flag to determine if the neighbor is helping out with the restart.

EOT (0x08) - The End-of-Table flag marks the end of the startup process with a new neighbor. A restarting router looks at this flag to determine if it has finished receiving the startup UPDATE packets from all neighbors, before cleaning up the stale routes from the restarting neighbor.

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.

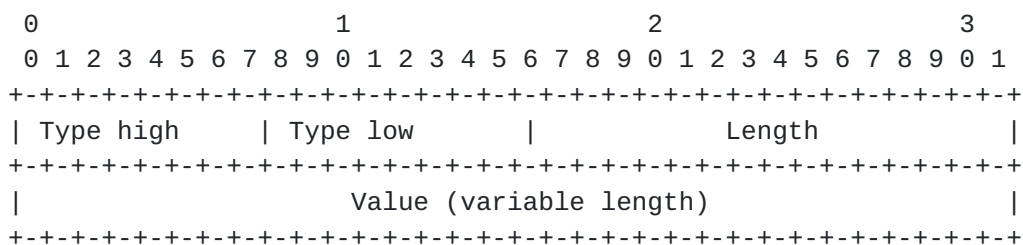
Value Range	Usage
0000	Unicast Address Family
0001	Multicast Address Family
0002-7FFFF	Reserved
8000	Unicast Service Family
8001-FFFF	Reserved

AS number - Autonomous System - 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.

## A.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:



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.

### A.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](#)

### A.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

**A.6.3 Value Field Encoding**

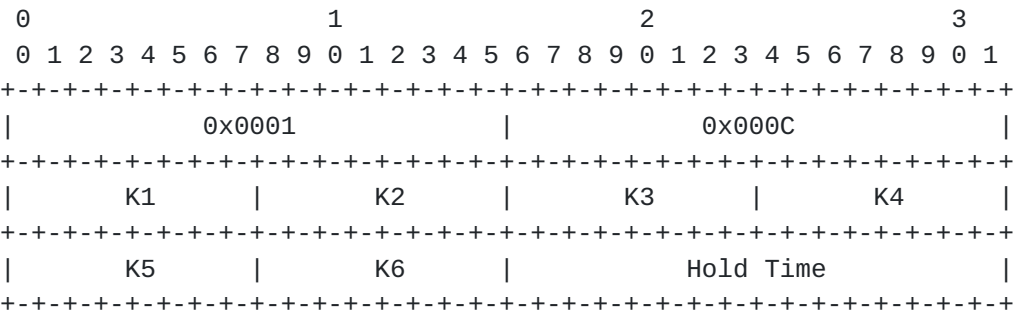
The Value field is a multi-octet field containing the payload for the TLV.

**A.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

**A.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 Holdtime values. This TLV is also used in an initial UPDATE packet when a neighbor is discovered.



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.

**A.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.

**A.7.2.1 0x02 - Authentication Type - MD5**

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

**A.7.2.2 0x03 -Authentication Type - SHA2**

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

**A.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.

#### A.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.

#### A.7.5 0x0005 - MULTICAST\_SEQUENCE\_TYPE

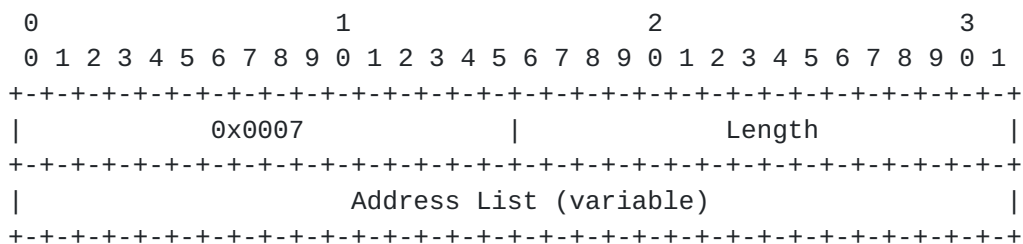
The next multicast sequence TLV

#### A.7.6      0x0006 - PEER\_ INFORMATION \_TYPE

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

#### A.7.7 0x0007 - PEER\_TERMINATION\_TYPE

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



### [A.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 through that neighbor for that sub-topology is removed from the topology table.

## [A.8](#) Classic Route Information TLV Types

### [A.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.

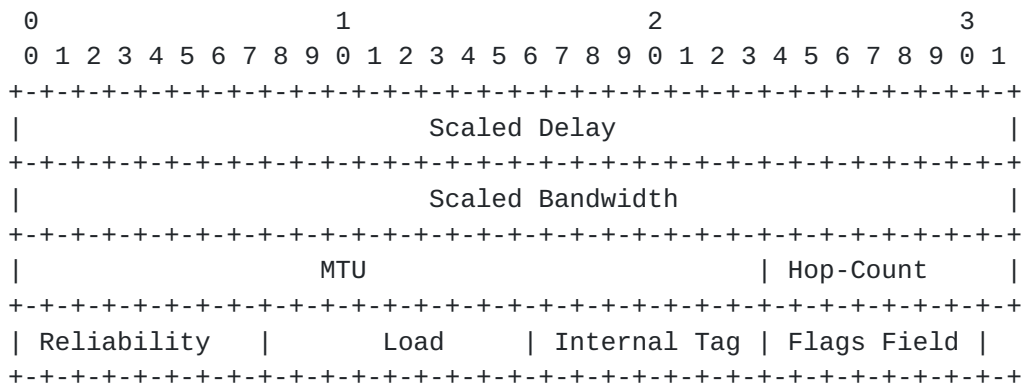
Candidate Default (CD) (Bit 1) - If set, this destination should be regarded as a candidate for the default route. An EIGRP default route is selected from all the advertised candidate default routes with the smallest metric.

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



### A.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.



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 divided 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

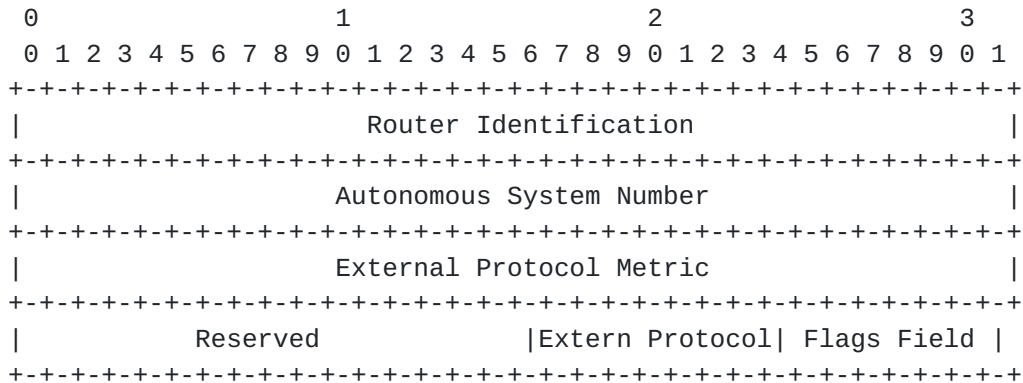
Load - The load utilization of the path to the destination, measured as a percentage. A value of 255 indicates 100% load.

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 A.8.1

### A.8.3      **Classic Exterior Encoding**

Additional routing information so provided for destinations outside of the EIGRP autonomous system as follows:



Router Identifier (RID) - A unique 32bit number that identifies the router sourcing the route that has redistributed this external route into the EIGRP autonomous system. If an IPv4 address is used, the address SHOULD be the largest unsigned address of any inter-face IPv4 address.

AS Number - The autonomous system number that the route resides in.

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.

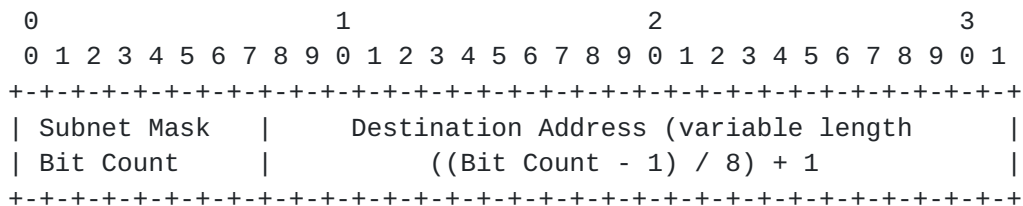
External Protocol Metric - 32bit value of the composite metric that resides in the routing table as learned by the foreign protocol. If the External Protocol is IGRP or another EIGRP routing process, the value can optionally be the composite metric or 0, and the metric information is stored in the metric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

External Protocol - Defines the external protocol that this route was learned. See Section A.2

Flag Field - See Section A.8.1

#### A.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



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

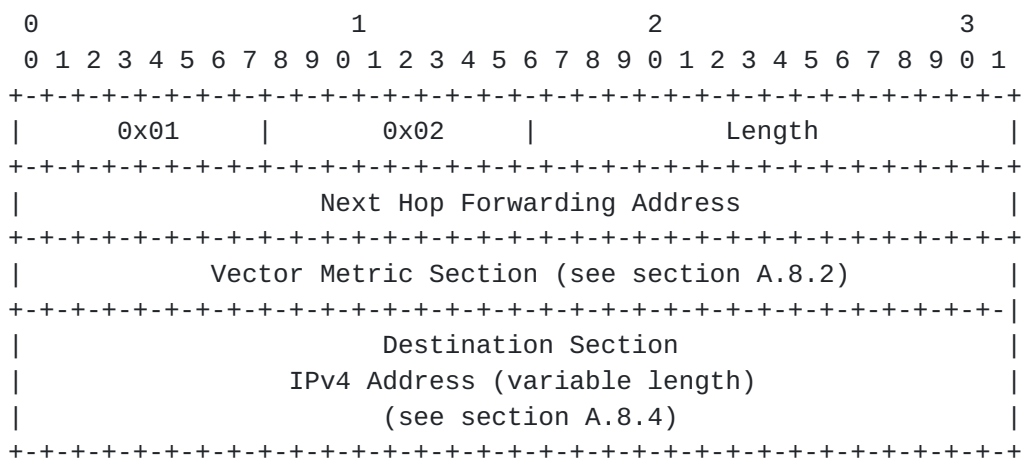
Destination Address - A variable length field used to carry the destination address. The length is determined by the number of consecutive bits in the destination address, rounded up to the nearest octet boundary, determines the length of the address.

### A.8.5 IPv4 Specific TLVs

INTERNAL_TYPE	0x0102
EXTERNAL_TYPE	0x0103
COMMUNITY_TYPE	0x0104

#### A.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.



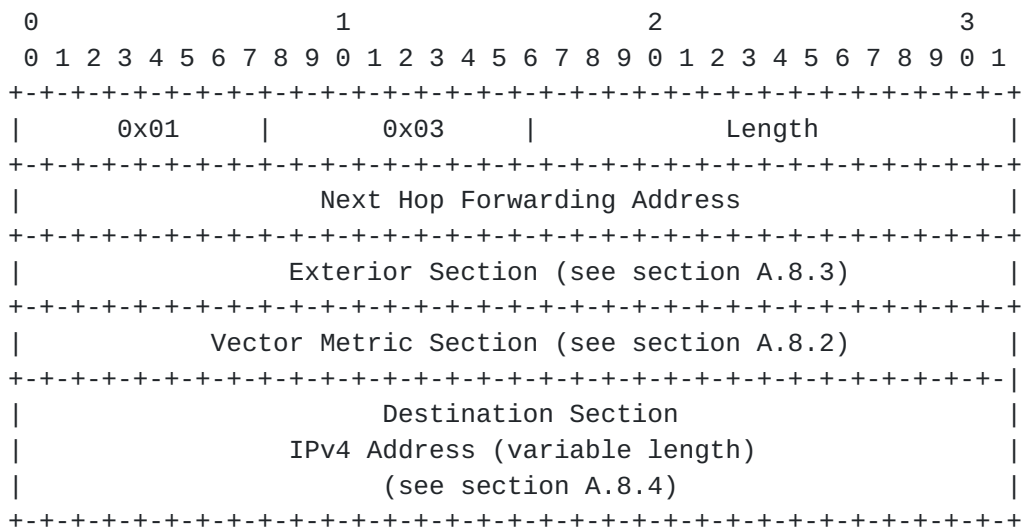
Next Hop Forwarding Address - IPv4 address is represented by 4 8-bit values (total 4 octets). If the value is zero (0), the IPv6 address from the received IPv4 header is used as the next-hop for the route. Otherwise, the specified IPv4 address will be used.

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in See Section A.8.2

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

#### [A.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.



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 EIGRP autonomous system and that has been redistributed into the EIGRP. See Section A.8.3

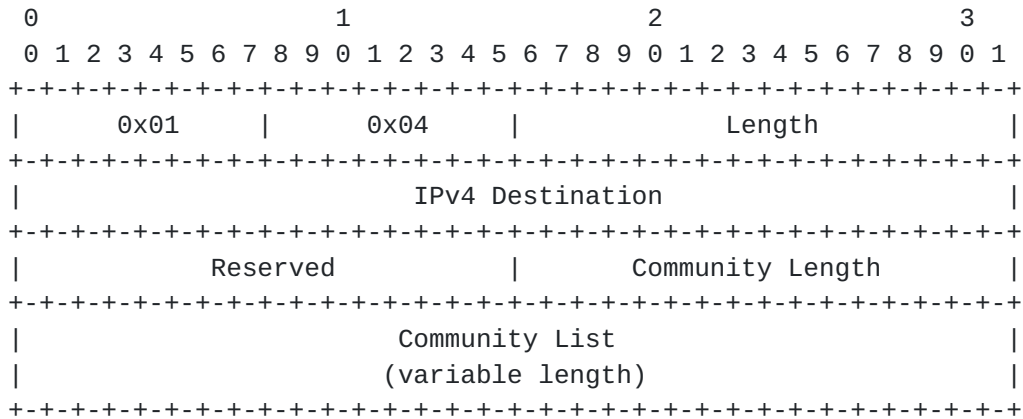
Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in See Section A.8.2

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



### [A.8.5.3](#)    **IPv4 COMMUNITY\_TYPE**

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



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

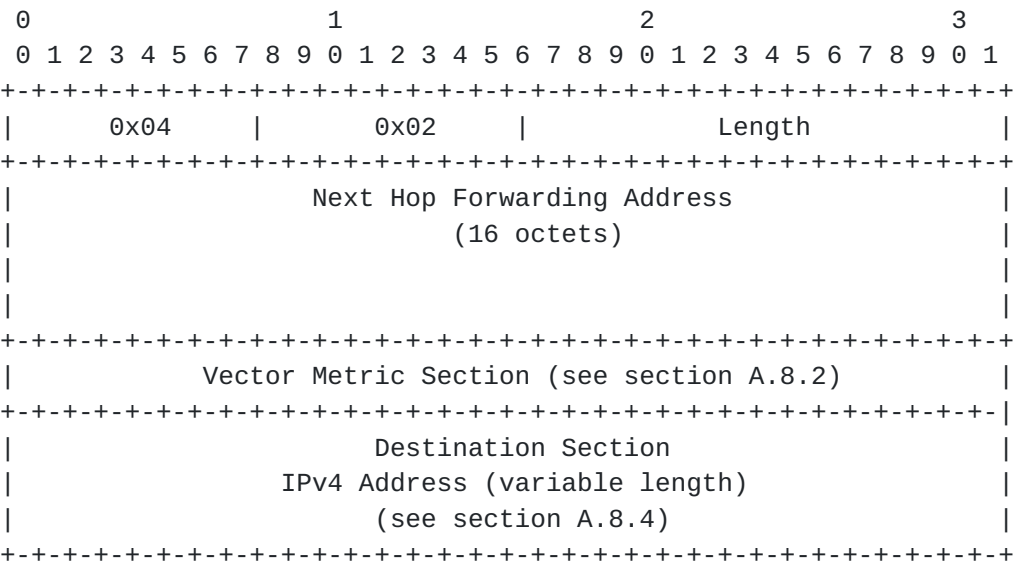
Community List - One or more 8 octet EIGRP community as defined in section A.4

**A.8.6 IPv6 Specific TLVs**

REQUEST_TYPE	0x0401
INTERNAL_TYPE	0x0402
EXTERNAL_TYPE	0x0403

**A.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.



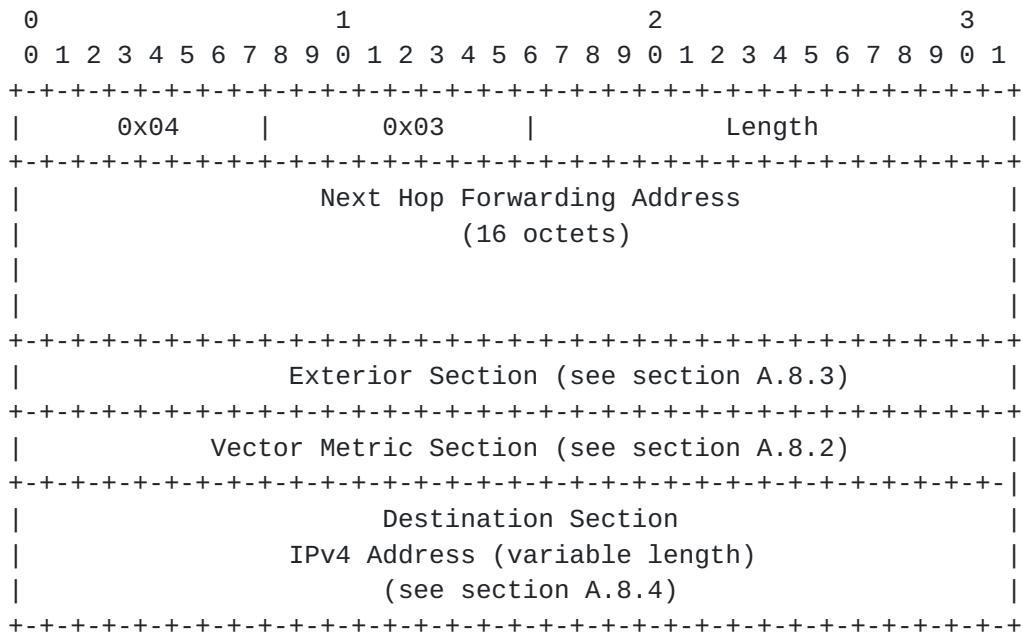
Next Hop Forwarding Address - IPv6 address is represented by 8 groups of 16-bit values (total 16 octets). If the value is zero (0), the IPv6 address from the received IPv6 header is used as the next-hop for the route. Otherwise, the specified IPv6 address will be used.

Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in See Section A.8.2

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

**A.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 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.



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 EIGRP autonomous system and that has been redistributed into the EIGRP. See Section A.8.3

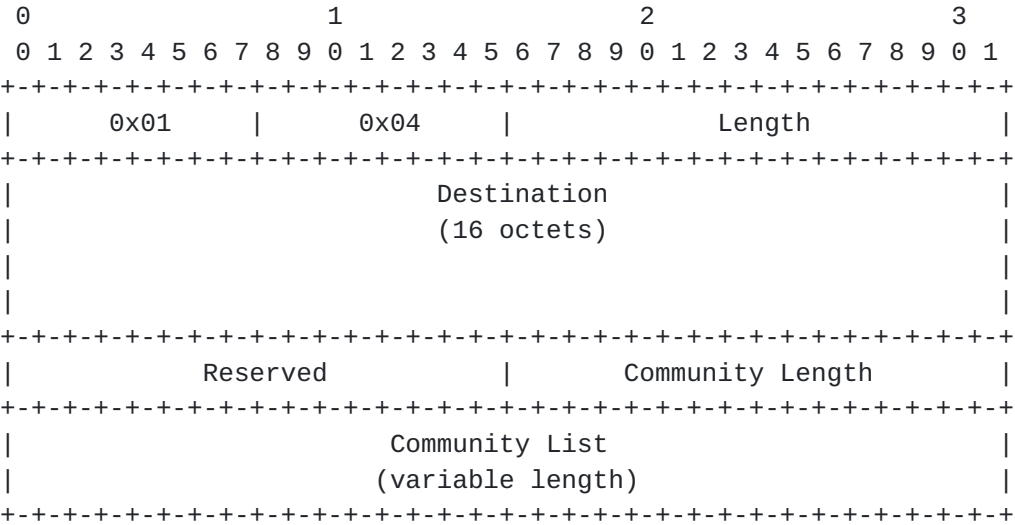
Metric Section - vector metrics for destinations contained in this TLV. See description of metric encoding in See Section A.8.2

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



**A.8.6.3 IPv6 COMMUNITY\_TYPE**

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



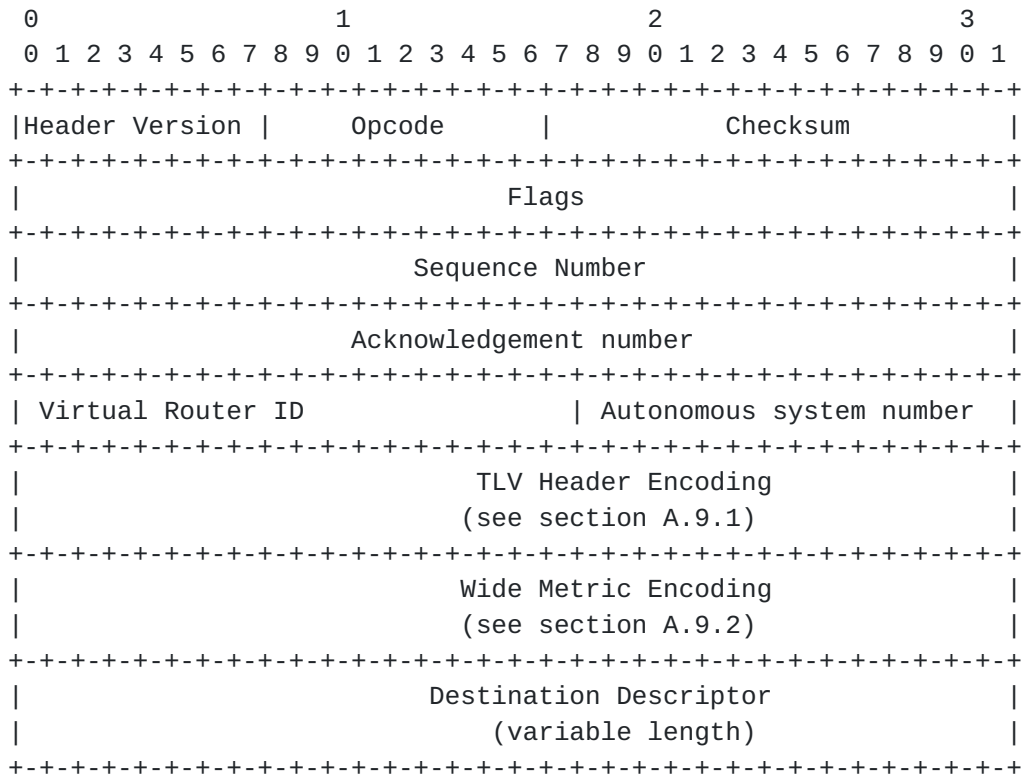
Destination - The IPv6 address the community information should be stored with.

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 A.4

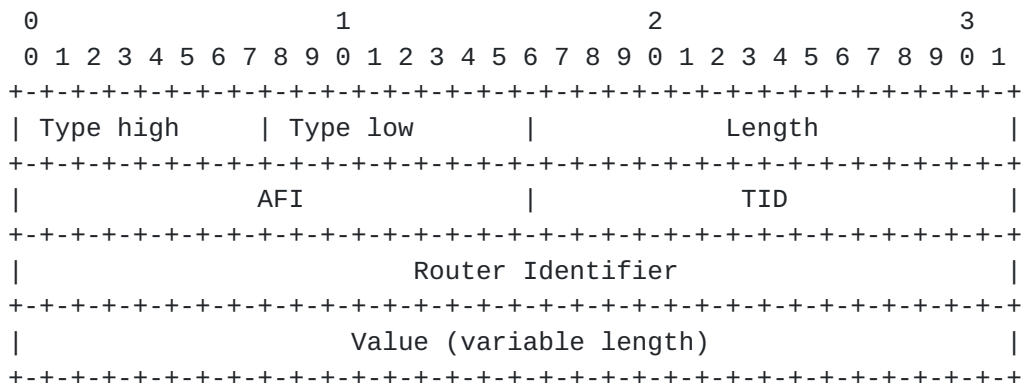
## [A.9](#) Multi-Protocol Route Information TLV Types

This TLV conveys topology and associated metric information



### [A.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:





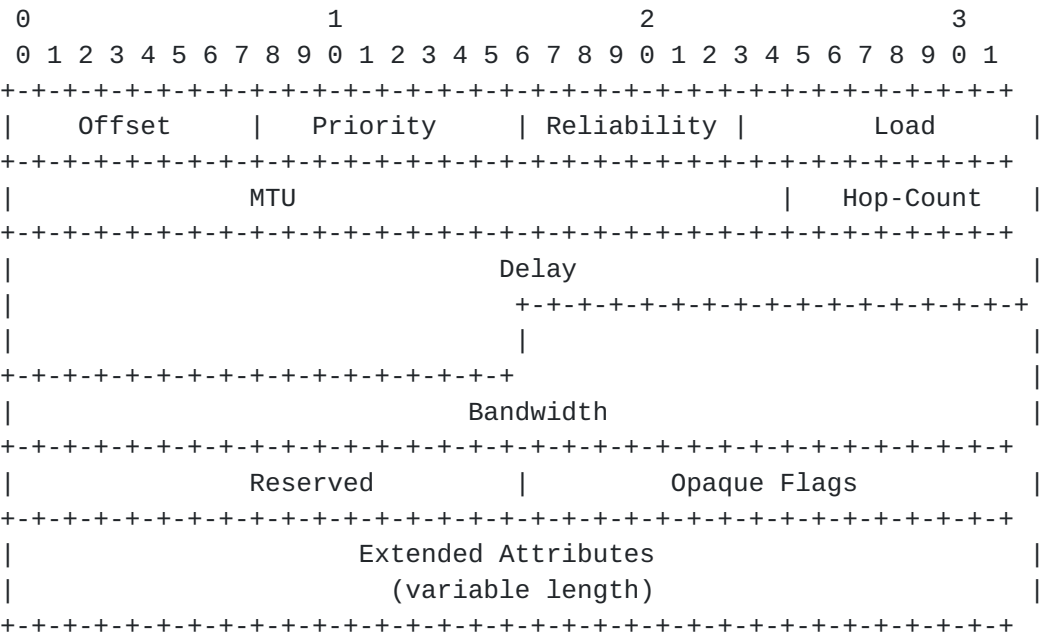
The available fields are:  
TYPE - Topology TLVs have the following TYPE codes:

REQUEST_TYPE	0x0601
INTERNAL_TYPE	0x0602
EXTERNAL_TYPE	0x0603

Address Family Identifier (AFI) - defines the type and format for the destination data. In EIGRP, each address family is implemented as a Protocol Dependent Module.  
Topology Identifier (TID) - The Service specific prefixes from the service specific topology tables will be tagged with a number known as the Topology Identifier (TID). This value was originally introduced with MTR.  
Router Identifier (RID) - A unique 32bit number that identifies the router sourcing the route into this EIGRP autonomous system.

**A.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.



The fields are:

Offset - Number of 16bit words in the Extended Attribute section, used to determine the start of the destination information. If no Extended Attributes are attached, offset will be zero.

Priority: Priority of the prefix when transmitting a group of destination addresses to neighboring routers. A priority of zero indicates no priority is set. Currently transmitted as 0

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, as is the case with "scaled delay". A delay of 0xFFFFFFFF indicates an unreachable route.

Bandwidth - The path bandwidth measured in kilobit per second as presented by the interface. This number is not scaled, as is the case with "scaled bandwidth". A bandwidth of 0xFFFFFFFF indicates an unreachable route.

Reserved - Transmitted as 0x0000

Opaque Flags - 16 bit protocol specific flags.

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

### [A.9.3](#) Extended Metrics

Extended metrics allows for extensibility of EIGRP metrics in a manor similar to [RFC-6390](#)[8]. Each Extended metric shall consist of a standard Type-Length header followed by application-specific information. The information field may be used directly by EIGRP, or by other applications. 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

#### [A.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.

#### [A.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. This value 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.

**A.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.

**A.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 A.4

**A.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.



**A.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.

**A.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.

### [A.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 AddPaths supported, where the format of the field will be as follows;

```

0                               1                               2                               3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      0x07      |      Offset |      AddPath (Variable Length) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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

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

#### [A.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 bytes)|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv4 Address (Lower 2 bytes) |      RID (Upper 2 bytes)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|      RID (Upper 2 bytes)    | Admin Tag (Upper 2 bytes)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Admin Tag (Upper 2 bytes)   | Extern Protocol | Flags Field |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Next Hop Address - 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.

RID - A unique 32bit number that identifies the router sourcing the route that has redistributed this external route into the EIGRP autonomous system. The address should be the largest unsigned address of any inter-face IPv4 address.

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.

If the route is of type external, then 2 additional bytes will be added as follows:

External Protocol - Defines the external protocol that this route was learned. See Section A.2

Flag Field - See Section A.8.1

#### A.9.3.8.2 Addpath with IPv6 Next-hop

[illegible]

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.

RID - A unique 32bit number that identifies the router sourcing the route that has redistributed this external route into the EIGRP autonomous system. The address should be the largest unsigned address of any inter-face IPv4 address.

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.

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 A.2

Flag Field - See Section A.8.1

#### A.9.4 Exterior Encoding

Additional routing information so provided for destinations outside of the EIGRP autonomous system as follows:

[illegible]

Router Identifier(RID) - A unique 32bit number that identifies the router sourcing the route that has redistributed this external route into the EIGRP autonomous system.

AS Number - The autonomous system number that the route resides in.

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.

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 A.2

Flag Field - See Section A.8.1

### A.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

[illegible]

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

Destination Address - A variable length field used to carry the destination address. The length is determined by the number of consecutive bits in the destination address, rounded up to the nearest octet boundary, determines the length of the address.

### A.9.6 Route Information

### A.9.6.1 INTERNAL TYPE

This TLV conveys destination information based on the IANA AFI defined in the TLV Header (See Section A.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.

### A.9.6.2 EXTERNAL TYPE

This TLV conveys destination information based on the IANA AFI defined in the TLV Header (See Section A.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.

## 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  
Cisco Systems, Inc  
[7025 Kit Creed Rd, RTP, NC](#)

Phone: 919-392-2539  
Email: dslice@cisco.com

Steven Moore  
Cisco Systems, Inc  
[7025 Kit Creed Rd, RTP, NC](#)

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

James Ng  
Cisco Systems, Inc  
[7025 Kit Creed Rd, RTP, NC](#)

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

Russ White  
VCE  
RTP, NC

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