

## ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

### EIGRP – Μελέτη και υλοποίηση



Του φοιτητή

Καραχατζή Παρασκευά

Αρ. Μητρώου: 09/3464

Επιβλέπων καθηγητής

Ψαρράς Νικόλαος

Θεσσαλονίκη 2015

### ΠΡΟΛΟΓΟΣ

Η παρούσα πτυχιακή εργασία με τίτλο < EIGRP- Μελέτη και υλοποίηση > ανατέθηκε από τον Καθηγητή Εφαρμογών Νικόλαο Ψαρρά, της Σχολής Τεχνολογικών Εφαρμογών του τμήματος πληροφορικής. Τα κίνητρα που με οδήγησαν στην επιλογή του συγκεκριμένου θέματος προς διερεύνηση είναι η

πρόκληση των δυνατοτήτων μου καθώς η ανάπτυξη του πρωτοκόλλου είναι μια δύσκολη διαδικασία. Επιπλέον, είναι μια καλή αρχή για να εμβαθύνω τις γνώσεις μου στην γλώσσα προγραμματισμού C καθώς και την δημιουργία package σε Debian.

Ο βασικός στόχος της πτυχιακής εργασίας είναι η κατασκευή ενός routing daemon για το EIGRP που είναι ένα Distance Vector πρωτόκολλο δρομολόγησης. Η εφαρμογή είναι μια διεργασία όπου τρέχει στο background και είναι επιφορτισμένη με διαδικασίες όπως την εύρεση διαδρομών δικτύου, την συντήρηση γειτόνων και την προσθαφαίρεση διαδρομών στο πίνακα δρομολόγησης. Στην ανάπτυξη της εφαρμογής έλαβε μέρος ο συνδυασμός γνώσεων με αποτέλεσμα την εύκολη ρύθμιση του πρωτοκόλλου όπως διεξάγετε στις συσκευές CISCO. Επίσης απαραίτητη ήταν η μελέτη της βασικής έκδοσης του πρωτοκόλλου EIGRP, για την υλοποίηση της εφαρμογής.

Η εφαρμογή παρέχει στους χρήστες της, την επικοινωνία με τις συσκευές του δικτύου που χρησιμοποιούν το πρωτόκολλο EIGRP για την εύρεση και προσθήκη διαδρομών δικτύου στον πίνακα δρομολόγησης του υπολογιστή. Συνεπώς, η εφαρμογή μπορεί να επιλύσει την αριθμητική έλλειψη συσκευών CISCO router σε ένα δίκτυο με χαμηλό κόστος και άμεσο τρόπο. Τέλος η εφαρμογή έχει την δυνατότητα να προσφέρει στο δίκτυο τη δημιουργία σύνθετων ρυθμίσεων, η οποία υπό άλλες συνθήκες δεν θα μπορούσε.

## ΠΕΡΙΛΗΨΗ

Στην πτυχιακή εργασία με θέμα την μελέτη και υλοποίηση του πρωτοκόλλου EIGRP ασχολούμαστε με την δημιουργία μιας εφαρμογής δρομολόγησης. Αυτή η εφαρμογή δρομολόγησης είναι υπεύθυνη για την διαχείριση του πίνακα δρομολόγησης και την εύρεση γειτόνων. Σκοπός του πρωτοκόλλου δρομολόγησης

είναι η εύρεση και εγκατάσταση της βέλτιστης διαδρομής για κάθε προορισμό στο δίκτυο. Έτσι ώστε να μην χρειάζεται ο διαχειριστής του δικτύου να περνάει τις διαδρομές στους πίνακες δρομολόγησης με το χέρι, για να μπορούν να επικοινωνήσουν οι συσκευές σε ένα δίκτυο.

Αρχικά πρέπει να αναφέρουμε ότι η εφαρμογή είναι ένα routing daemon για το πρωτόκολλο EIGRP που είναι ένα Distance Vector πρωτόκολλο δρομολόγησης ιδιοκτησία της CISCO. Υλοποιήθηκε σε λειτουργικό σύστημα Debian OS (Kernel Linux 3.2.0), χρησιμοποιώντας C για γλώσσα προγραμματισμού καθώς και το packaging του. Πρέπει να τονιστεί ότι η βασική έκδοση του πρωτοκόλλου δόθηκε το 2013 με την μορφή Informational RFC. Αρχικά οι αλγόριθμοι δρομολόγησης χωρίζονται σε διάφορα είδη με βάση τον τρόπο λειτουργίας τους. Για την ανάπτυξη της εφαρμογής χρειάστηκαν οι αλγόριθμοι διανυσμάτων απόστασης και κατάστασης συνδέσεων καθώς το πρωτόκολλο EIGRP δανείζεται χαρακτηριστικά και των δύο. Επίσης, το πρωτόκολλο EIGRP σε αντίθεση με τα άλλα πρωτόκολλα δρομολόγησης δεν χρησιμοποιεί τα πρωτοκολλά TCP ή UDP για την μεταφορά δεδομένων αλλά έχει υλοποιήσει το δικό του πρωτόκολλο RTP(Reliable Transport Protocol), το οποίο συνδυάζει αξιοπιστία και ταχύτητα. Επίσης, το πρωτόκολλο EIGRP χαρακτηρίζει τις διαδρομές με τις καταστάσεις PASSIVE και ACTIVE. Όταν το δρομολόγιο είναι σε κατάσταση PASSIVE προωθεί πακέτα για τον προορισμό ενώ όταν βρίσκετε στην κατάσταση ACTIVE ψάχνει να βρει ένα καινούργιο δρομολόγιο γιατί το προηγούμενο δεν είναι πια εφικτό. Για να θεωρηθεί ότι μια διαδρομή είναι έγκυρη, πρέπει η αναφερόμενη απόσταση του γείτονα που την έστειλε να είναι μικρότερη από την ελάχιστη απόσταση που γνωρίζει ο δρομολογητής. Αυτό ονομάζετε Feasible Condition. Όταν η διαδρομή ενός γείτονα ικανοποιεί αυτήν την προϋπόθεση ο γείτονας ονομάζετε feasible successor. Από όλους τους feasible successor θα επιλεγθεί ένας successor μέσω του οποίου θα γίνετε η προώθηση πακέτων. Το DUAL Finite State Machine χρησιμοποιείτε για την διαχείριση και τον υπολογισμό των διαδρομών. Είναι ένας από τους μηχανισμούς στον οποίον οφείλει την ταχύτητα του το πρωτόκολλο EIGRP. Αυτό επιτυγχάνετε υπολογίζοντας τα δρομολόγια ξεχωριστά το καθένα καθώς και στο ότι αν και το πρωτόκολλο EIGRP είναι πρωτόκολλο διανύσματος απόστασης εμπεριέχει ένα πίνακα τοπολογίας από τον οποίο μπορεί να ανατρέξει πληροφορίες για την εύρεση καινούργιας διαδρομής. Για την απόκτηση των απαραίτητων δεδομένων για τον υπολογισμό των διαδρομών το DUAL Finite State Machine χρησιμοποιεί τα πακέτα UPDATE, QUERY και REPLY. Η κεφαλίδα των πακέτων είναι μικρή και πέρα από τα βασικά στοιχεία για την επικοινωνία εμπεριέχει και το πρωτόκολλο RTP (Reliable Transport Protocol). Στο σύνολο των πακέτων επικοινωνίας του πρωτοκόλλου EIGRP εμπεριέχονται πέρα από τα

προαναφερόμενα πακέτα και τα παρακάτω. Το πακέτο HELLO το οποίο έχει διάφορες χρίσεις όπου η πιο γνωστή είναι για την διαφήμιση του δρομολογητή στους γείτονες του. Το πακέτο SIA-QUERY το οποίο στέλνεται όταν δεν ληφθεί απάντηση από ένα QUERY. Και το πακέτο SIA-REPLY που είναι η απάντηση σε στο πακέτο SIA-QUERY. Επίσης, οι διαδρομές διαιρούνται σε internal και external. Τα internal είναι διαδρομές που προέρχονται από το πρωτόκολλο EIGRP ενώ οι external διαδρομές έχουν γνωστοποιηθεί από άλλα πρωτόκολλα δρομολόγησης. Ακόμα για να γλυτώσει την αποστολή πολλαπλών ίδιων πακέτο το πρωτόκολλο EIGRP στέλνει πακέτα multicast. Παρόλα αυτά αυτό δημιουργεί προβλήματα καθώς η αποστολή των πακέτων πρέπει να γίνεται σε σειρά, ωστόσο έχει αναπτυχθεί και ο μηχανισμός για την αντιμετώπιση αυτού του προβλήματος. Καλό θα ήταν να αναφερθεί ότι όταν μια διαδρομή μεταβεί σε ACTIVE κατάσταση ενεργοποιείται ένας μηχανισμός ο οποίος έχει σκοπό να μην επιτρέψει στην διαδρομή να μείνει σε ACTIVE κατάσταση για πάντα λόγο προβλήματος. Αναγκαίο είναι δυο δρομολογητές για να ανταλλάξουν πληροφορίες μεταξύ τους πρέπει να γίνουν πρώτα γείτονες στέλνοντας κάποια πακέτα. Για να πληρούν τις προϋποθέσεις για να είναι γείτονες πρέπει η K μεταβλητές να είναι όμοιες. Οι K μεταβλητές είναι τα βάρη για των υπολογισμό της απόστασης των διαδρομών. Για των υπολογισμό τις απόστασης χρησιμοποιείτε είτε ο κλασσικός τύπος είτε ο Wide Metric τύπος. Πιο συγκεκριμένα τα Wide Metric δημιουργήθηκαν από το πρόβλημα που προέκυψε από την εμφάνιση καρτών δικτύου μεγαλύτερου εύρους ζώνης των 10 Gbps. Τέλος, η μεταφορά των πληροφοριών μέσα στα πακέτο γίνεται μέσω των TLV τα οποία είναι παρόμοια με δομές που κρατάν δεδομένα.

Για την ανάπτυξη του κώδικα χρησιμοποιήθηκε το Incremental μοντέλο. Οι βασικότεροι κίνδυνοι τις εφαρμογής ήταν η εύρεση των μηχανισμών ή εργαλείων για την επικοινωνία της εφαρμογής με τον πυρήνα ή άλλες εφαρμογές. Έτσι για την υλοποίηση τις εφαρμογής χρησιμοποιήθηκαν τα δυο πρωτόκολλα, telnet και netlink. Οι βιβλιοθήκες τους βρέθηκαν από την κοινότητα του Debian και υπάρχουν σε πακέτα στο repository. Για την συγγραφή, μεταγλώττιση, αποσφαλμάτωση και την δημιουργία του πακέτου χρησιμοποιήθηκαν τα προγράμματα gedit, gcc, gdb, Valgrind, Wireshark και το πακέτο debhelper. Ακόμα σημαντικό ρόλο έπαιξαν οι Cisco δρομολογητές.

## ABSTRACT

This thesis is deals with the study and implementation of the EIGRP protocol routing protocol. The purpose of a routing protocol is to keep track of the changes of the network topology and adjust the routes to the destination networks, on top of discovering these networks. EIGRP is a distance vector routing protocol,

property of Cisco. At 2013 Cisco released an Informational RFC describing the basic version of the EIGRP protocol. The program was developed using C as a programming language and it is used as a daemon.

There are two basic categories of routing algorithms based on how they work. Even though EIGRP is described as a distance vector routing protocol, it uses features of link-state protocols as well. Also unlike other routing protocols it does not use TCP or UDP to transport packets through the network but it has implemented a transport layer protocol called RTP (Reliable Transport Protocol) which is a mix of both TCP and UDP and offers both speed and reliability. The routes are defined as either in PASSIVE state or ACTIVE state. When a route is in PASSIVE state the router will forward packet to that destination in contrast when the route goes into ACTIVE state it will stop forwarding packets and start searching for a new valid route. A valid route is considered a route that meets the Feasible Condition. This condition is met when the advertised distance from the destination is less than the feasible distance the router learned since the last time the route went into ACTIVE state. The routes that meet this condition are called feasible successor. Each route can have multiple feasible successor but one of them will be chosen as the successor which will be the neighbor that the packets will be forwarded to. We should also mention that when a router changes to ACTIVE state a mechanism is triggered to limit the time the route stays in the ACTIVE state. Both the DUAL algorithm and state machine provide fast convergence time by recalculated only the affected routes while also providing loop free paths. And by using the topology table the protocol can recalculate the route without having to change the state of the route. The information needed by the DUAL algorithm is conveyed with the packets UPDATE, QUERY and REPLY. On top of these EIGRP also uses the packets HELLO, SIA-QUERY and SIA-REPLY. The data transferred in these packets are encapsulated into TLV which are similar to structures and allow different types of data to be transferred using the same packet. Also routes are distinguished as internal and external routes. Internal routes are the routes created or found by the EGRP protocol while external routes are those that were discovered from other sources such as static routes and other routing protocols. To reduce traffic at the network the EIGRP protocol will send multicast packets whenever such an option is available. Even though this has created a problem since the packets much to read in order, the solution to this problem is also implemented. Lastly the calculation of the route metric can be done by either the classic formula or the wide metric formula. Wide metrics were introduced as the classic formula could not keep up with the fast network interfaced. Because EIGRP uses the inverted bandwidth of the link to

calculate the bandwidth based on a theoretical maximum bandwidth, all the network interfaces that pass this limits have trouble calculating a correct or more likely an accurate metric. To solve this problem wide metrics were introduced solving this problem and most importantly are compatible with the classic metrics.

For the development of the code the incremental model was used. It allows easy debugging as the parts that are made on each iteration are functional. At the same time it allows the programmer to break down the problem into smaller pieces and work them stage by stage. The drawback of this model is that you can not calculate the cost of the project, which in this case is the time consumed to develop the software. The program uses two more protocols telnet and netlink to interact with other parts of the system and the user. Both their libraries were found at the Debian repository and are contributed by the community. Software used for the project are gedit, gcc, gdb, valgrind, wireshark and the libraries from the debhelper package. Lastly an important role was played by the Cisco routers at the end of each iteration for debugging.

## **ΕΥΧΑΡΙΣΤΙΕΣ**

Με την ολοκλήρωση της πτυχιακής εργασίας θα ήθελα να ευχαριστήσω τους ανθρώπους που συνέλαβαν στη διεκπεραίωση της. Ιδιαίτερες ευχαρίστησες θα ήθελα να απευθύνω στον Καθηγητή Εφαρμογών κύριο Ψαρρά Νικόλαο για την άμεση ανταπόκριση του στις απορίες που δημιουργήθηκαν κατά την εκπόνησης της πτυχιακής εργασίας καθώς και για την βοήθεια που πρόσφερε καθ' όλη την διάρκεια ολοκληρώσεις της.

Ακόμα θα ήθελα να δώσω τις ευχαριστίες μου στον Paul Wise μέλος του Debian mentor mailing list για την βοήθεια του στην εύρεση των βιβλιοθηκών που χρησιμοποιήθηκαν στην πτυχιακή. Τέλος, θα ήθελα να δώσω τις θερμές μου ευχαριστίες στους φίλους μου και στην οικογένεια μου για την υποστήριξη τούς και την βοήθεια τους κατά την διάρκεια της εκπόνησης της πτυχιακής εργασίας.

## Κατάλογος περιεχομένων

ΠΡΟΛΟΓΟΣ.....	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT.....	5
ΕΥΧΑΡΙΣΤΙΕΣ.....	7
ΕΙΣΑΓΩΓΗ.....	10
ΚΕΦΑΛΑΙΟ 1.....	11
Μελέτη και επεξήγηση του πρωτοκόλλου EIGRP.....	11
1.1 Πρωτόκολλα δρομολόγησης.....	11
1.2 Είδη πρωτοκόλλων δρομολόγησης.....	12
1.3 Πρωτόκολλο EIGRP.....	15
1.4 Reliable Transport Protocol.....	16
1.5 Route States.....	17
1.6 Feasibility Condition.....	18
1.7 Feasible Successor και Successor.....	18
1.8 Dual Finite State Machine.....	18
1.9 Η κεφαλίδα πακέτου EIGRP.....	21
1.10 Πακέτα επικοινωνίας.....	23
1.11 Είδη διαδρομών.....	26
1.12 Topology Table.....	27
1.13 Security.....	27
1.14 Multicast Packets.....	28
1.15 Χειρισμός ACTIVE διαδρομών.....	28



1.16 Διαδικασία Γειτνίασης Δρομολογητών.....	30
1.16 Κ Μεταβλητές.....	31
1.17 Classic Metric.....	31
1.18 Wide Metric.....	32
1.19 TVL.....	34
ΕΠΙΛΟΓΟΣ.....	41
ΚΕΦΑΛΑΙΟ 2.....	42
Ανάπτυξη της εφαρμογής για το πρωτόκολλο δρομολόγησης EIGRP.....	42
ΕΙΣΑΓΩΓΗ.....	42
2.1 Μοντέλο ανάπτυξης.....	42
2.2 Οι Βασικότεροι κίνδυνοι.....	42
2.3 Πρωτόκολλα που χρησιμοποιήθηκαν.....	43
2.4 Βιβλιοθήκες που χρησιμοποιήθηκαν.....	44
2.5 Εργαλεία για την ανάπτυξη.....	44
ΕΠΙΛΟΓΟΣ.....	45
ΣΥΜΠΕΡΑΣΜΑΤΑ.....	46
ΑΝΑΦΟΡΕΣ.....	47
ΒΙΒΛΙΟΓΡΑΦΙΑ.....	48
ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ.....	49

## Ευρετήριο πινάκων

## Ευρετήριο σχημάτων

Σχήμα 1: Classic Vector Metric Encoding.....	39
--	----

Πτυχιακή εργασία του φοιτητή Καραχατζή Παρασκευά

Σχήμα 2: Classic Destination Encoding.....	39
Σχήμα 3: Classic Exterior Encoding.....	40
Σχήμα 4: IPv4 Internal route.....	41
Σχήμα 5: IPv4 External route.....	42

## ΕΙΣΑΓΩΓΗ

Στόχος της πτυχιακής είναι η μελέτη και υλοποίηση του πρωτοκόλλου EIGRP. Η πτυχιακή εργασίας περιλαμβάνει δύο κεφάλαια, όπου το πρώτο αναφέρεται στην μελέτη του πρωτοκόλλου EIGRP ενώ το δεύτερο περιγράφει την διαδικασία υλοποίησης του. Αυτό που κινεί το ενδιαφέρον, για το πρωτόκολλο EIGRP, είναι ότι ο μηχανισμός λειτουργίας του δεν είχε δοθεί στο κοινό μέχρι πρόσφατα, όπου έγινε η γνωστοποίηση της βασικής έκδοσης του πρωτοκόλλου. Τελικός σκοπός της πτυχιακής είναι η δημιουργία μίας εφαρμογής δρομολόγησης για το πρωτόκολλο EIGRP για τη χρήση του σε λειτουργικό Debian.

Αρχικά θα εξηγηθούν τα πιο δημοφιλή είδη αλγορίθμων δρομολόγησης και στην συνέχεια θα δοθούν οι ιδιαιτερότητες του πρωτοκόλλου EIGRP. Έπειτα θα γίνει αναλυτική παρουσίαση του τρόπου λειτουργίας του πρωτοκόλλου EIGRP καθώς και των επιμέρους μηχανισμών του, όπως η μονάδα μέτρησης απόστασης, η διαχείριση και ο υπολογισμός των διαδρομών, τα πακέτα και ο τρόπος που χρησιμοποιείτε για την μεταφορά δεδομένων στο δίκτυο και την διαδικασία δημιουργίας των γειτόνων.

Τέλος, για την ανάπτυξη και υλοποίηση της εφαρμογής θα αναφερθεί το μοντέλο που εφαρμόστηκε για την ανάπτυξη του κώδικα, οι βασικότεροι κίνδυνοι που λάβαμε υπόψιν μας για την αποφυγή αποτυχίας του έργου, τα περαιτέρω πρωτόκολλα που χρησιμοποιήθηκαν, οι απαραίτητες βιβλιοθήκες που χρειάστηκαν για την υλοποίηση, καθώς και τα εργαλεία που χρησιμοποιήσαμε.

## ΚΕΦΑΛΑΙΟ 1

### Μελέτη και επεξήγηση του πρωτοκόλλου EIGRP

#### ΕΙΣΑΓΩΓΗ

Στο πρώτο κεφάλαιο θα αναφερθούμε στο θεωρητικό υπόβαθρο του πρωτοκόλλου EIGRP. Το πρωτόκολλο EIGRP θεωρείτε ένα από τα καλύτερα πρωτόκολλα δρομολόγησης λόγω της ταχύτητας του σε σχέση με άλλα πρωτόκολλα δρομολόγησης. Ενδιαφέρον παρουσιάζει η πρόσφατη γνωστοποίηση της βασικής έκδοσης του πρωτοκόλλου, το οποίο εκδόθηκε για πρώτη φορά το 2013. Πιο αναλυτικά θα αναφερθούμε, στην κατανόηση της λειτουργίας του πρωτοκόλλου, στους μηχανισμούς που χρησιμοποιήθηκαν, τους τρόπους επικοινωνίας μεταξύ των δρομολογητών και την διαχείριση των διαδρομών.

#### **1.1 Πρωτόκολλα δρομολόγησης**

Τα δίκτυα επικοινωνούν μεταξύ τους μέσω των δρομολογητών. Ο κάθε δρομολογητής γνωρίζει μόνο τα δίκτυα που είναι απευθείας συνδεδεμένα σε αυτόν. Ωστόσο, για να λειτουργήσει ένα σύνολο δικτύων πρέπει οι δρομολογητές να ενημερωθούν για την ύπαρξη των δικτύων του συνόλου. Η ενημέρωση αυτή επιτυγχάνεται με την καταχώρηση των διαδρομών στους πίνακες δρομολόγησης, είτε μεσώ των πρωτοκόλλων δρομολόγησης είτε περνώντας χειροκίνητα τις διαδρομές. Επειδή η χειροκίνητη καταχώρηση των διαδρομών από τον διαχειριστή του δικτύου είναι χρονοβόρα διαδικασία, προτιμάτε η χρήση των πρωτοκόλλων δρομολόγησης. Τα οποία εκτελούν αυτόματα την διεργασία αυτήν, δηλαδή καταχωρούν τις διαδρομές στους πίνακες δρομολόγησης προς διάφορους προορισμούς στο δίκτυο. Ακόμα πρέπει να αναφέρουμε ότι το πρωτόκολλο μπορεί να εντοπίζει τότε κάποια διαδρομή δεν είναι πλέον ενεργή και να αφαιρεί την εγγραφή της από τον πίνακα δρομολόγησης.

Το πρωτόκολλο δρομολόγησης είναι υπεύθυνο για τον καθορισμό του προορισμού των πακέτων, τα οποία στέλνονται μέσω των δρομολογητών. Για να

γίνει η προώθηση των πακέτων σε άλλα δίκτυα πρέπει να έχουν καταχωρηθεί στον πίνακα δρομολόγησης, οι διαδρομές προς διάφορους προορισμούς στο δίκτυο.

Το κάθε πρωτόκολλο δρομολόγησης έχει δυο βασικά χαρακτηριστικά, μια μονάδα μέτρησης κόστους και έναν αλγόριθμο για την εύρεση της καλύτερης διαδρομής. Ωστόσο, η διαδικασία εύρεσης και ο τρόπος υπολογισμού διαφέρουν από πρωτόκολλο σε πρωτόκολλο. Όπως προαναφέραμε δεν μπορούμε να συγκρίνουμε αποστάσεις διαδρομών διαφορετικών πρωτοκόλλων γιατί το κάθε πρωτόκολλο δρομολόγησης μετράει διαφορετικά μεγέθη. Το πρωτόκολλο RIP για κόστος διαδρομής μετράει πόσους κόμβους θα αλλάξει ένα πακέτο για να φτάσει στον προορισμό. Ενώ το OSPF μετράει σαν κόστος την επιβάρυνση που θα έχει στο δίκτυο η αποστολή του πακέτου.

## **1.2 Είδη πρωτοκόλλων δρομολόγησης**

Τα πρωτόκολλα δρομολόγησης χωρίζονται σε κατηγορίες με βάση την ζώνη λειτουργίας τους, στα πρωτόκολλα εσωτερικών πυλών και στα πρωτόκολλα εξωτερικών πυλών. Τα πρωτόκολλα εσωτερικών πυλών (Interior Gateway Protocol - IGP) λειτουργούν εντός ενός αυτόνομου συστήματος, σε αυτή την κατηγορία ανήκουν τα πρωτόκολλα δρομολόγησης RIP, OSPF, HELLO και EIGRP. Ενώ τα πρωτόκολλα εξωτερικών πυλών δικτύου (Exterior Gateway Protocol - EGP) βρίσκουν διαδρομές μεταξύ των παραπάνω αυτόνομων συστημάτων για την επικοινωνία και ανταλλαγή πληροφοριών. Το πιο γνωστό πρωτόκολλο εξωτερικής πύλης δικτύου είναι το Border Gateway Protocol (BGP).

Ένα αυτόνομο σύστημα καλύπτει το σύνολο δικτύων που βρίσκετε υπό τον έλεγχο μιας διαχειριστικής αρχής. Μέσα στο αυτόνομο σύστημα η διαχειριστική αρχή μπορεί να επιλέξει τους μηχανισμούς διάδοσης των διαδρομών. Τα πρωτόκολλα δρομολόγησης αυτόνομου συστήματος χωρίζονται σε κατηγορίες με βάση τον τρόπο λειτουργίας τους. Διαιρούνται σε αλγορίθμους διανυσμάτων απόστασης (distance vector) και σε κατάσταση συνδέσεων (link-state).

Οι αλγόριθμοι διανυσμάτων απόστασης είναι μια κατηγορία αλγορίθμων την οποία χρησιμοποιούν οι δρομολογητές για την διάδοση των πληροφοριών δρομολόγησης. Η ιδέα στην οποία στηρίζετε είναι απλή. Ο δρομολογητής διατηρεί σε ένα πίνακα όλα τα γνωστά δρομολόγια. Κατά την εκκίνηση του δίνει αρχικές τιμές στα δρομολόγια για τα άμεσα συνδεδεμένα δίκτυα. Για κάθε δρομολόγιο καταγράφεται η απόσταση του δικτύου. Κατά περιόδους κάθε δρομολογητής στέλνει ένα πλήρη αντίγραφο του πίνακα δρομολόγησης στους δρομολογητές που είναι άμεσα συνδεδεμένοι. Όταν ένας δρομολογητής λάβει το αντίγραφο θα εξετάσει για κάθε δρομολόγιο την απόσταση που γνωρίζει αυτός για την διαδρομή. Άμα βρει κάποια εγγραφή που δεν γνωρίζει, θα την προσθέσει ενώ άμα υπάρχει συντομότερη διαδρομή μέσω του δρομολογητή που έστειλε το αντίγραφο, θα την αντικαταστήσει.

Οι αλγόριθμοι διανυσμάτων απόστασης είναι εύκολο να υλοποιηθούν, ωστόσο έχουν ορισμένα μειονεκτήματα. Ένα από αυτά είναι ότι σε περιβάλλον στο οποίο υπάρχουν συνεχής αλλαγές στο δίκτυο οι πίνακες δρομολόγησης δεν μπορούν να σταθεροποιηθούν λόγω του χρόνου αποστολής των αντιγράφων. Έτσι κατά την διάρκεια σύγκλισης ορισμένοι δρομολογητές ενδέχεται να περιέχουν εσφαλμένες διαδρομές. Ακόμα μειονέκτημα είναι ότι εκτελούν τον υπολογισμό των διαδρομών χρησιμοποιώντας τα δρομολόγια που λαμβάνουν από τους γείτονες. Όταν υπάρξει μια αλλαγή στο δίκτυο, για να μάθει ο δρομολογητής για την αλλαγή θα πρέπει ο γείτονας του να διορθώσει πρώτα τον πίνακα δρομολόγησης του και να τον στείλει στους επόμενους. Για ένα δρομολογητή που βρίσκετε σε μια Χ θέση μακριά από αυτήν την αλλαγή θα πρέπει πρώτα όλοι οι ενδιάμεσοι δρομολογητές να ενημερώσουνε τους πίνακές δρομολόγησης τους και με την σειρά τους να τους στείλουν. Για αυτό τον λόγο οι αλγόριθμοι διανυσμάτων απόστασης έχουν μεγαλύτερο χρόνο σύγκλισης του δικτύου. Τέλος, μειονέκτημα αποτελεί ότι τα μηνύματα των αλγορίθμων διανυσμάτων απόστασης γίνονται μεγαλύτερα όσο μεγαλώνει το δίκτυο.

Οι αλγόριθμοι κατάστασης συνδέσεων (link-state) λειτουργούν με διαφορετικό τρόπο από τους αλγόριθμους διανυσμάτων απόστασης. Αντί να στέλνουν μεγάλους πίνακες με δρομολόγια ανά χρονικά διαστήματα, κρατάνε ένα πίνακα με πληροφορίες τοπολογίας με σκοπό να μην χρειάζεται να ανταλλάσσουν διαρκώς πληροφορίες. Κατά την εκκίνηση οι δρομολογητές πλημμυρίζουν το δίκτυο με μηνύματα, τα οποία αναφέρουν τους δρομολογητές που είναι συνδεδεμένοι. Ο κάθε δρομολογητής επιλέγει την καλύτερη διαδρομή για κάθε προορισμό. Από εκεί και πέρα, ότι περαιτέρω μηνύματα σταλούν θα περιέχουν πληροφορίες που σχετίζονται με την κατάσταση λειτουργίας των γειτονικών δρομολογητών. Το πλεονέκτημα του αλγορίθμου είναι ότι αποστέλλονται μηνύματα, μικρού όγκου και δεν επιβαρύνουν το δίκτυο κατά την λειτουργία του. Έτσι ο κάθε δρομολογητής μπορεί ταχύτερα να υπολογίσει τις διαδρομές. Αυτό συμβαίνει επειδή γνωρίζει την εικόνα του δικτύου, ως προς την συνδεσιμότητα των δρομολογητών χωρίς την αποστολή περαιτέρω μηνυμάτων. Αυτός ο μηχανισμός επιτρέπει την γρήγορη σύγκλιση του αυτόνομου συστήματος αφού οι δρομολογητές εκτελούν τοπικά τον υπολογισμό των διαδρομών. Το μειονέκτημα είναι ότι κάθε φορά που υπάρχει μια αλλαγή στο αυτόνομο σύστημα πρέπει να υπολογιστούν πάλι όλα τα δρομολόγια.

### **1.3 Πρωτόκολλο EIGRP**

Το πρωτόκολλο EIGRP είναι ένα ιδιωτικό πρωτόκολλο δρομολόγησης της CISCO. Το 2013 η CISCO έβγαλε την βασική έκδοση που πρωτοκόλλου σε μορφή Information RFC, με αυτό των τρόπο μπορούν οι πελάτες της να εφαρμόζουν το συγκεκριμένο πρωτόκολλο στον δικό τους εξοπλισμό. Η διαφορά του EIGRP πρωτοκόλλου από τα υπόλοιπα πρωτόκολλα δρομολόγησης είναι ότι δανείζετε τα πλεονεκτήματα των αλγορίθμων διανυσμάτων απόστασης (distance vector) και των αλγορίθμων κατάστασης συνδέσεων (link-state). Έτσι μπορεί να υπολογίσει τυχόν αλλαγές στο αυτόνομο σύστημα είτε απευθείας είτε σε μικρο χρονικό διάστημα. Επειδή το πρωτόκολλο δρομολόγησης EIGRP χρησιμοποιεί το

δικό του πρωτόκολλο μεταφοράς Reliable Transport Protocol(RTP) για το επίπεδο μεταφοράς στο μοντέλο του OSI του επιτρέπει να κάνει δρομολόγηση σε πολλαπλά πρωτόκολλα δικτύου. Τα πρωτόκολλα δικτύου για τα οποία έχει υλοποιηθεί το EIGRP είναι τα IPv4, IPv6, Appletalk και IPX.

Όπως προαναφέραμε με τις διαφορές του πρωτοκόλλου EIGRP σε σχέση με τα υπόλοιπα πρωτόκολλα δρομολόγησης, το EIGRP χαρακτηρίζεται ως πρωτόκολλο διανύσματος απόστασης, περιέχει κομμάτια και από τα δύο είδη αλγορίθμων, διανυσμάτων απόστασης (distance vector) και κατάστασης συνδέσεων (link-state), δηλαδή το πρωτόκολλο EIGRP χρησιμοποιεί μέρη και των δυο αλγορίθμων.

Πιο συγκεκριμένα ένα από τα χαρακτηριστικά που έχει δανειστεί το πρωτόκολλο EIGRP από τον αλγόριθμο καταστάσεις συνδέσεων είναι ότι δημιουργεί γειτνιάσεις μεταξύ των δρομολογητών με σκοπό την ταχύτερη διαπίστωση αποχωρήσεις ενός γείτονα. Στέλνοντας πακέτα ανά μικρά χρονικά διαστήματα μπορεί να αντιληφθεί άμα ο γειτονικός δρομολογητής είναι ενεργός. Αυτό επιτυγχάνεται καθώς άμα δεν λάβει απάντηση, σημαίνει αυτόματα ότι έχει παρουσιαστεί κάποιο πρόβλημα μεταξύ της επικοινωνίας τους. Αντίθετα, τα πλεονεκτήματα που παίρνει από τον αλγόριθμο διανύσματος απόστασης είναι ο απλός και γρήγορος υπολογισμός των διαδρόμων.

Σαν αποτέλεσμα έχουμε δημιουργία καινούργιων χαρακτηριστικών όπως, στα μηνύματα να μην στέλνονται όλες οι εγγραφές από τον πίνακα δρομολόγησης παρά μόνο οι εγγραφές που έχουν υποστεί αλλαγές, αυτό οφείλεται στο ότι ο δρομολογητής κρατάει πληροφορίες από όλους τους γείτονες και όχι μόνο από την βέλτιστη διαδρομή. Ακόμα λόγω της ύπαρξη γειτνιάσεων δεν χρειάζεται να στέλνονται μηνύματα ανά τακτά χρονικά διαστήματα αλλά μόνο όταν υπάρχει αλλαγή σε κάποια διαδρομή. Αυτό έχει σαν αποτέλεσμα τα μηνύματα να είναι μικρότερα σε όγκο από τα μηνύματα του αλγορίθμου διανυσμάτων απόστασης. Επίσης, ο τελικός υπολογισμός της βέλτιστης διαδρομής γίνεται πολύ εύκολα, συγκρίνοντας τις τιμές για την διαδρομή που έχει υποστεί αλλαγές και διαλέγοντας



αυτήν με το μικρότερο κόστος. Έτσι επιτυγχάνετε γρηγορότερη σύγκληση του δικτύου σε σχέση με τα link-state πρωτόκολλα.

Ωστόσο, σαν αποτέλεσμα του συνδυασμού των δύο αλγορίθμων αυξήθηκε η πολυπλοκότητα του αλγορίθμου. Τέλος, ένα κύριο χαρακτηριστικό του πρωτοκόλλου EIGRP είναι η σύνθετη μονάδα μέτρησης για τον υπολογισμό της απόστασης του δικτύου η οποία βγαίνει από το άθροισμα του εύρους ζώνης, της χρονικής καθυστέρησης του πακέτου, του φόρτου του συνδέσεων και την αξιοπιστία των συνδέσεων, το οποίο αναφέρετε στην απώλεια πακέτων. Λόγω του σύνθετου υπολογισμού του κόστους θεωρείτε πιο αξιόπιστο από άλλα πρωτόκολλα δρομολόγησης.

#### **1.4 Reliable Transport Protocol**

Το πρωτόκολλο EIGRP σε αντίθεση με τα υπόλοιπα πρωτόκολλα δρομολόγησης δεν χρησιμοποιεί το πρωτόκολλο TCP ή UDP για την μεταφορά των πακέτων, αλλά έχει το δικό του πρωτόκολλο με την ονομασία Reliable Transport Protocol (RTP), το οποίο είναι υπεύθυνο για την εγγυημένη παράδοση των πακέτων και την σωστή σειρά παραλαβής. Λόγω των εξωτερικών θορύβων, την βλάβη συσκευών ή την συμφόρηση του δικτύου μπορεί να χαθούν ορισμένα πακέτα. Το RTP μπορεί να εντοπίσει τέτοιες περιπτώσεις και να ζητήσει από τον γείτονα να του στείλει το πακέτο ξανά. Στην κεφαλίδα που πακέτου (header) υπάρχουν δυο πεδία που επιτυγχάνουν τον σκοπό της εγγυημένης παράδοσης των πακέτων και την σωστή σειρά παραλαβής, το sequence number και το acknowledgment number.

Το sequence number δείχνει τον αριθμό του πακέτου, δίνοντας μοναδικές τιμές κατά την αποστολή του. Δίνεται ένας μοναδικός αύξοντα αριθμός ώστε να αναγνωρίζει ο παραλήπτης την σειρά των πακέτων. Άμα σταλθεί ένα πακέτο με ίδιο sequence number ο δρομολογητής το αγνοεί επειδή θεωρεί ότι το έχει λάβει ήδη. Ωστόσο, ξαναστέλνεται η επιβεβαίωση παραλαβής αυτού του πακέτου, ώστε να ενημερωθεί ο γείτονας ότι το έλαβε και να στείλει το επόμενο.

Το acknowledgment number λειτουργεί σαν μηχανισμός του πρωτοκόλλου, που είναι υπεύθυνο για την ενημέρωση του αποστολέα ότι το μήνυμα έφτασε επιτυχώς στον παραλήπτη. Ο μηχανισμός αυτός δουλεύει με τον παρακάτω τρόπο. Πριν σταλθεί ένα πακέτο από τον αποστολέα, διαλέγετε ένα sequence number μεγαλύτερου του προηγούμενου κατά μία μονάδα, ώστε να μην υπάρχουν πακέτα με το ίδιο sequence number. Στην συνέχεια το πακέτο αποθηκεύετε σε μια ουρά όπου από εκεί ο δρομολογητής διαλέγει το παλαιότερο σε σειρά και αποστέλλετε μέχρι να του σταλθεί επιβεβαίωση παραλαβής από τον παραλήπτη. Όταν ο παραλήπτης λάβει acknowledgment number που έχει ίδιο αριθμό με το sequence number, του πακέτου που αποθηκεύτηκε στην ουρά, αφαιρεί το πακέτο από την ουρά και στέλνεται το επόμενο.

Εξαίρεσή αποτελεί, όταν το πακέτο που στάλθηκε στον παραλήπτη έχει sequence number 0, και δεν χρειάζεται να σταλθεί acknowledgment number στον αποστολέα. Σαν πακέτο ACK στέλνεται ένα άδειο πακέτο HELLO με τον αριθμό επιβεβαίωσης λήψης, acknowledgment number ,του πακέτου. Ωστόσο, δεν είναι απαραίτητο πάντα να στέλνουμε ACK πακέτα χωριστά. Αλλά μπορούμε να υλοποιήσουμε την διαδικασία αυτήν με άλλον τρόπο, στέλνονταν το acknowledgment number μέσα σε πακέτο που ετοιμάζεται προς αποστολή.

### **1.5 Route States**

Το Route State είναι μια τιμή που δείχνει την κατάσταση που βρίσκετε μια διαδρομή. Η κάθε διαδρομή χαρακτηρίζεται από μια κατάσταση PASSIVE ή μια κατάσταση ACTIVE. Όταν η διαδρομή βρίσκετε σε PASSIVE κατάσταση μεταγίνει τα πακέτα για το συγκεκριμένο δίκτυο ενώ όταν η διαδρομή μεταβεί στην κατάσταση ACTIVE λόγω μιας αλλαγής στο δίκτυο, σταματάει η μεταγωγή των πακέτων. Ωστόσο, συνεχίζει να καταγράφει αλλαγές για το δίκτυο ώστε να μπορέσει να βρει μια καινούργια διαδρομή για το συγκεκριμένο προορισμό. Άμα βρεθεί καινούργια διαδρομή θα μεταβεί πάλι σε κατάσταση PASSIVE και θα συνεχίσει την μεταγωγή των πακέτων για αυτόν τον προορισμό. Πρέπει να τονιστεί ότι κατά τις αλλαγές του δικτύου δεν είναι απαραίτητο μια διαδρομή να μεταβεί σε

κατάσταση ACTIVE άμα σταματήσει να λειτουργεί ο δρομολογητής στον οποίο μεταγάγει τα πακέτα. Σε αυτήν την περίπτωση ο αλγόριθμος θα ψάξει πρώτα να βρει άμα υπάρχει ήδη κάποιος δρομολογητής που θα φτάνει τον συγκεκριμένο προορισμό, θα τον αντικαταστήσει και η διαδρομή δεν θα μεταβεί σε κατάσταση ACTIVE αλλά θα παραμείνει σε κατάσταση PASSIVE.

### **1.6 Feasibility Condition**

Το Feasibility Condition είναι ένας μηχανισμός του πρωτοκόλλου EIGRP, ο οποίος ελέγχει αν ο γειτονικός δρομολογητής φτάνει στον προορισμό με μικρότερο κόστος από τον ίδιο. Μια διαδρομή που πληρεί τις προϋποθέσεις για το feasibility condition δεν θα είναι ποτέ βρόχος. Όταν υπάρξει κάποια αλλαγή το δίκτυο, χάρη του μηχανισμού Feasibility Condition, μπορούμε να διαλέξουμε έναν καινούργιο δρομολογητή για την μεταγωγή των πακέτων, εφόσον πλήρη τις προϋποθέσεις. Γνωρίζοντας ότι η επιλογή του καινούργιο δρομολογητή δεν θα δημιουργήσει βρόχο.

Reported Distance είναι η απόσταση που χρειάζεται ο γείτονας για να φτάσει σε κάποιο προορισμό ενώ Feasible Distance είναι το άθροισμα της απόστασης που χρειάζεται ο δρομολογητής για φτάσει στον γείτονα και το Reported Distance. Η προϋπόθεση που πρέπει να πληρεί για να μπορεί μια διαδρομή που στάλθηκε από κάποιο γείτονα πρέπει η αναφερόμενη απόσταση (reported distance) να είναι μικρότερη από την εφικτή απόσταση της μικρότερης διαδρομής (feasible distance) που γνωρίζει ο δρομολογητής για αυτόν τον προορισμό.

### **1.7 Feasible Successor και Successor**

Feasible Successor είναι ένας γειτονικός δρομολογητής, για ένα προορισμό ο οποίος πληρεί το feasibility condition. Δηλαδή πρέπει να βρίσκετε πιο κοντά στον προορισμό από ότι ο εαυτός του. Για ένα προορισμό μπορούν να υπάρχουν πολλοί feasible successor. Ένας από αυτούς θα επιλεγεί έως successor και θα

μεταγάγει τα πακέτα σε ένα συγκεκριμένο προορισμό. Συνήθως επιλέγετε σαν successor ο γείτονας με το μικρότερο κόστος για τον προορισμό.

### **1.8 Dual Finite State Machine**

Το Dual Finite State Machine ένας μηχανισμός που διαχειρίζεται το Route State καθώς και άλλες αποφάσεις για τον υπολογισμό των διαδρομών. Ο υπολογισμός αυτός εκτελείται από τον αλγόριθμο Diffusing Update Algorithm (DUAL) ο οποίος παρέχει διαδρομές που δεν είναι ατέρμονες βρόχοι. Έχει σκοπό την γρήγορη σύγκληση του δικτύου απευθείας ή σε σύντομο χρονικό διάστημα. Επίσης, κατά τις αλλαγές του δικτύου μόνο οι επηρεαζόμενες διαδρομές υπολογίζονται με αποτέλεσμα να χρειάζεται λιγότερη υπολογιστική ισχύς και μεταφορά δεδομένων.

Ως προς την διαχείριση του Route State ορίζει την κατάσταση των διαδρόμων σε κατάσταση PASSIVE ή σε κατάσταση ACTIVE. Αυτό επιτυγχάνεται συλλέγοντας τις πληροφορίες από τα πακέτα που ανταλλάσσονται. Έπειτα ανατρέχει τις ήδη γνωστές πληροφορίες και αποφασίζει ποιους μηχανισμούς θα ξεκινήσει ή θα τερματίσει για την εύρεση των διαδρόμων. Συνεπώς, είναι εφικτό να αποφεύγετε η αλλαγή κατάστασης μίας διαδρομής σε ACTIVE, με σκοπό να μην σταματήσει η μεταγωγή πακέτων. Με αυτό τον τρόπο γλυτώσουμε επεξεργαστική ισχύ και έχουμε γρηγορότερη σύγκληση δικτύου.

Ο Dual Finite State Machine χρησιμοποιεί τα πακέτα UPDATE, QUERY και REPLY για την λήψη πληροφοριών σχετικά με την τοπολογία του δικτύου. Τα πακέτα UPDATE στέλνονται για ενημερώσουν τους γείτονες για αλλαγές, προσθήκη ή αφαίρεση διαδρόμων. Τα πακέτα QUERY στέλνονται όταν ένας δρομολογητής θέλει να ζητήσει πληροφορίες για διαδρομές από έναν άλλο δρομολογητή. Τέλος, το πακέτο REPLY στέλνετε σαν απάντηση στο πακέτο QUERY.

Όταν ο Dual Finite State Machine δεχτεί ένα πακέτο QUERY, UPDATE ή αντιληφθεί την αποχώρηση ενός γείτονα και κάποιο από τα δρομολόγια δεν είναι

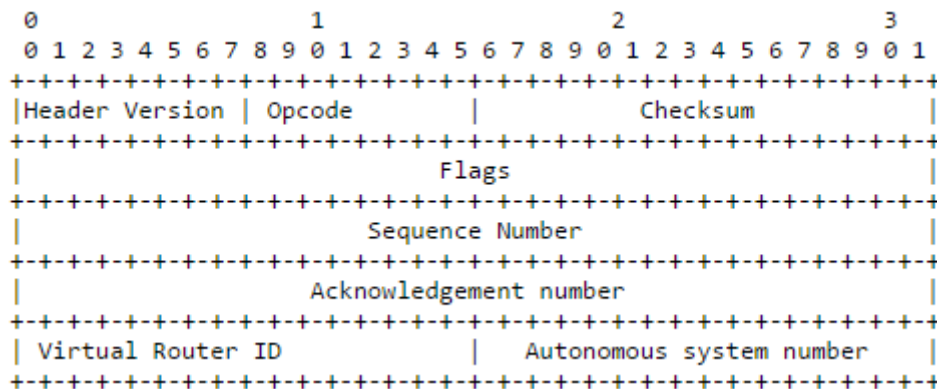
πια εφικτό να χρησιμοποιηθεί, θα κοιτάξει τον πίνακα τοπολογίας για να βρει άμα υπάρχει άλλος feasible successor. Σε περίπτωση που βρει feasible successor θα επιλέξει έναν από αυτούς. Συνήθως θα επιλέξει αυτόν με το μικρότερο κόστος για τον προορισμό και θα γίνει ο καινούργιος successor. Στην συνέχεια θα στείλει ένα UPDATE για να ενημερώσει τους γείτονες του για την αλλαγή της διαδρομής. Το πλεονέκτημα του μηχανισμού αυτού είναι ότι ενώ υπήρξε ήδη αλλαγή στο δίκτυο, δεν χρειάστηκε να γίνει ξανά ο υπολογισμός των διαδρομών. Ωστόσο, δεν τελειώνει εκεί, όταν οι γείτονες δεχτούν το πακέτο θα ελέγξουν άμα πληρείτε ακόμα το feasibility condition και θα επαναληφθεί η διαδικασία στους επόμενους δρομολογητές. Αν δεν βρεθεί κάποιος feasible successor η διαδρομή θα αλλάξει σε κατάσταση ACTIVE και θα σταματήσει να μεταγωγή των πακέτων προς εκείνο τον προορισμό.

Όταν μια διαδρομή μεταβεί σε ACTIVE κατάσταση ο δρομολογητής θα στείλει ένα πακέτο QUERY σε όλους τους διαθέσιμους γείτονες ζητώντας μια διαδρομή για αυτόν τον προορισμό. Οι γείτονες που θα παραλάβουν το πακέτο QUERY θα πρέπει να απαντήσουν με ένα πακέτο REPLY, όπου κάθε γείτονας περιλαμβάνει τις πληροφορίες για την βέλτιστη διαδρομή που γνωρίζει για τον προορισμό. Όταν ο δρομολογητής που έστειλε το QUERY λάβει τις απαντήσεις από όλους τους γείτονες θα διαλέξει την διαδρομή που θα μεταγάγει τα πακέτα για τον συγκεκριμένο προορισμό. Στην συνέχεια θα στείλει ένα πακέτο UPDATE για να ενημερώσει τους γείτονες για την αλλαγή και θα αλλάξει την κατάσταση της διαδρομής σε PASSIVE ώστε να συνεχίσει τη μεταγωγή των πακέτων σε αυτόν τον προορισμό. Σε περίπτωση που δεν βρεθεί κάποια έγκυρη διαδρομή ο δρομολογητής θα αφαιρέσει τον προορισμό από τον πίνακα δρομολόγησης.

Αυτή είναι μια σύντομη περιγραφή πως λειτουργεί ο Dual Finite State Machine. Στην πραγματικότητα η διαδικασία είναι πιο περίπλοκη. Για παράδειγμα, την ώρα του υπολογισμού μιας διαδρομής μπορεί υπάρξει και άλλη αλλαγή, αυτό έχει σαν αποτέλεσμα να ξαναστέλνονται τα QUERY και να ξεκινάει η διαδικασία από την αρχή.

## 1.9 Η κεφαλίδα του πακέτου EIGRP

Η κεφαλίδα ενός πακέτου αναφέρεται στον αποστολέα που το έστειλε, στον παραλήπτη που το έλαβε και το περιεχόμενο που περιέχει. Μέσα στην κεφαλίδα του EIGRP εμπεριέχεται και το πρωτόκολλο RTP (Reliable Transport Protocol). Η κεφαλίδα του πακέτου είναι όσο το δυνατόν μικρότερη, για να χρησιμοποιεί όσο τον δυνατόν λιγότερους πόρους του δικτύου και απλή, για να γίνεται γρήγορα η επεξεργασία της. Η κεφαλίδα (header) του πακέτου EIGRP περιέχει τα παρακάτω στοιχεία.



Σχήμα 1: EIGRP Header Format

- Header Version

Είναι η έκδοση της κεφαλίδας. Η τωρινή έκδοση είναι η 2.

- Opcode

Είναι ο αριθμός που υποδηλώνει τον τύπο του πακέτου (HELLO, QUERY, REPLY)

- Checksum

Είναι ένας ακαριαίος αριθμός, που χρησιμοποιείτε για την εύρεση σφαλμάτων κατά την μετάδοση του πακέτου. Όταν εντοπιστούν λάθη το πακέτο απλώς απορρίπτεται. Η τιμή βγαίνει από τα περιεχόμενα όλου του πακέτου και η διαδικασία υπολογισμού είναι παρόμοια με το IP.

- Flags

Τα flag χρησιμοποιούνται για να δείξουν ειδικές καταστάσεις.

- ➔ INIT (0x01) μπαίνει στο αρχικό UPDATE όταν ξεκινάει η γειτνίαση.
- ➔ CR (0x02) χρησιμοποιείται για να μαρκάρει ένα πακέτο ώστε μην διαβαστεί από συγκεκριμένους γείτονες. Χρησιμοποιείται σε συνδυασμό με ένα πακέτο HELLO κατά την αποστολή multicast πακέτων όποτε χρειάζεται.
- ➔ RS (0x04) χρησιμοποιείται κατά την προγραμματισμένη επανεκκίνηση ενός δρομολογητή.
- ➔ EOT (0x08) χρησιμοποιείται για να δείξει ο γείτονας ότι αυτό είναι το τελευταίο πακέτο με πληροφορίες για το δίκτυο.

- Sequence number

Είναι ο αύξων αριθμός του πακέτου. Κάθε πακέτο προς αποστολή, με εξαίρεση τα πακέτα που στέλνονται επανειλημμένα και η απώλεια τους δεν θα παρουσίαση πρόβλημα (επειδή στέλνονται επανειλημμένα), παίρνει έναν αύξων μοναδικό αριθμό ο οποίος δείχνει την σειρά με την οποία στάλθηκαν τα πακέτα. Όταν ανιχνευθεί ότι έφτασαν δυο πακέτα με ίδιο αριθμό (sequence number) σημαίνει ότι έγινε διπλή αποστολή και ένα από τα δυο πακέτα θα πεταχτεί. Το ίδιο θα συμβεί (θα πεταχτεί το πακέτο) και σε περίπτωση που το sequence number είναι μικρότερο από το sequence number του τελευταίου πακέτου που δέχθηκε καθώς το πακέτο έχει ήδη σταλεί και επεξεργαστεί (με εξαίρεση όταν το πακέτο έχει sequence number 0). Όταν ο αριθμός είναι 0 σημαίνει ότι δεν χρειάζεται να επιβεβαιωθεί η άφιξη αυτού του πακέτου.

- Acknowledgment number

Είναι η επιβεβαίωση ότι ο δρομολογητής έχει παραλάβει το πακέτο με τον συγκεκριμένο αριθμό. Όταν ο αριθμός είναι 0 σημαίνει ότι έχει σταλεί η τελευταία επιβεβαίωση λήψης και ο δρομολογητής μπορεί να παραλάβει το επόμενο πακέτο.

- Virtual Router ID

Το πεδίο Virtual Router ID αναφέρετε ότι υπάρχει στο draft του EIGRP ωστόσο δεν δίνονται πληροφορίες για την χρήση του.

- Autonomous System Number

Είναι ένας αριθμός, όπου δίνεται από τον διαχειριστή του αυτόνομου συστήματος. Οι τιμές του είναι από 0 μέχρι 65.535, η επιλογή εξαρτάται από την κρίση του διαχειριστή του αυτόνομου συστήματος. Ο σκοπός του είναι να διαχωρίζει πακέτα για διαφορετικά αυτόνομα συστήματα που υπάρχουν στον ίδιο δρομολογητή.

### **1.10 Πακέτα επικοινωνίας**

Για την επικοινωνία μεταξύ δυο δρομολογητών χρησιμοποιούνται πακέτα τα οποία μεταφέρουν πληροφορίες. Οι πληροφορίες αυτές ενημερώνουν τους δρομολογητές για αλλαγές στα δίκτυα ή τις γειτνιάσεις. Το είδος των πακέτων ξεχωρίζετε από το πεδίο opcode του EIGRP header. Αυτά τα πακέτα είναι το HELLO/Ack, QUERY, UPDATE, REPLY, SIA-QUERY, SIA-REPLY

- HELLO/Ack

Το πακέτο HELLO έχει πολλαπλές χρήσεις. Τα πακέτα HELLO στέλνονται για να ανακαλύψουν άλλους δρομολογητές που χρησιμοποιούν το πρωτόκολλο EIGRP στο δίκτυο. Από την έναρξη μέχρι την λήξη λειτουργίας του πρωτοκολλήσου δρομολόγησης στέλνονται πακέτα HELLO με πληροφορίες για τα τις K μεταβλητές καθώς και την έκδοση του πρωτοκόλλου. Άμα οι K μεταβλητές των δρομολογητών είναι ίσες τότε μπορούν να γίνουν γείτονες, ώστε να ξεκινήσουν να ανταλλάσσουν πληροφορίες για τους προορισμούς που γνωρίζει ο καθένας. Τα πακέτα HELLO χρησιμοποιούνται επίσης για να δείξει ένας



δρομολογητής στους γείτονες του ότι είναι ακόμα ενεργός. Το πρωτόκολλο EIGRP χρησιμοποιεί τον χρόνο για να εντοπίσει άμα κάποιος γείτονας έχει πρόβλημα και δεν μπορεί να στήσει μηνύματα, είτε επειδή υπάρχει πρόβλημα στο δίκτυο είτε επειδή παρουσιάστηκε κάποιο άλλο πρόβλημα. Έπειτα από κάποιο χρονικό διάστημα αν δεν λάβει πακέτο από κάποιον γείτονα, τότε θα τον θεωρήσει ανενεργό. Θα θέσει όλες τις έγγραφες του δρομολογητή σαν μη έγκυρες και θα ψάξει μια καινούργια εγγραφή. Ο χρόνος αυτός ορίζεται από την τιμή HOLDOUT. Στέλνονται σε τακτά χρονικά διαστήματα χρόνου 1/3 του HOLDOUT πακέτα HELLO για να ανανεώνεται ο μετρητής. Όταν ο δρομολογητής θέλει να στείλει μια επιβεβαίωση λήψης ενός πακέτου, αν δεν υπάρχει ήδη στη λίστα άλλο διαθέσιμο πακέτο για αποστολή, στέλνει ένα κενό HELLO με το acknowledge number του πακέτου. Τέλος, πριν την αποστολή multicast πακέτων στέλνεται ένα HELLO πακέτο για να ενημερώσει ποιοι δρομολογητές πρέπει να αγνοήσουν το συγκεκριμένο πακέτο.

- QUERY

Τα πακέτα QUERY στέλνονται όταν ένας δρομολογητής θέλει να ζητήσει από ένα γείτονα πληροφορίες για διαδρομές, στις οποίες υπήρξαν αλλαγές ή για να ζητήσει πληροφορίες για διαδρομές που δεν γνωρίζει. Τέτοια πακέτα στέλνονται συνήθως μετά από την διαπίστωση της απώλειας ενός γείτονα. Για την εξοικονόμηση του εύρους ζώνης, στέλνεται ένα πακέτο με όλες τις πληροφορίες των διαδρομών που ζητά ο δρομολογητής.

- UPDATE

Τα πακέτα UPDATE στέλνονται για να ενημερώσουν οι δρομολογητές στους γείτονες για αλλαγές στον πίνακα δρομολόγησης. Όπως και στα πακέτα QUERY έτσι και στα πακέτα UPDATE στέλνεται ένα πακέτο που περιέχει πολλαπλές διαδρομές. Επίσης, το πακέτο UPDATE χρησιμοποιείται κατά την γεινίαση δυο δρομολογητών. Όταν ένας δρομολογητής διαβάσει το πακέτο HELLO θα απαντήσει με ένα αρχικό πακέτο UPDATE, το οποίο δεν θα έχει καμία πληροφορία

για τον πίνακα δρομολόγησης αλλά θα έχει το flag INIT, το οποίο επισημαίνει την αρχή της γειννίασης. Έπειτα γίνεται η αποστολή του πακέτου UPDATE με το flag INIT μαζί με την επιβεβαίωση αρχικής λήψης του πακέτου. Στην συνέχεια ο γείτονας στέλνει όσα πακέτα UPDATE είναι απαραίτητα ώστε να μεταφέρει πλήρως τον πίνακα δρομολόγησης του. Στο τελευταίο πακέτο UPDATE θα βάλει το flag ENDOFTABLE, το οποίο σηματοδοτεί την λήξη του πίνακα δρομολόγησης. Το flag INIT έχει μια ακόμα χρήση. Ότι υπάρχει περίπτωση να μην αντιληφθεί ο δρομολογητής την απώλεια του γείτονα επειδή δεν έληξε το HOLDOUT. Οπότε όταν διαβάσει το flag INIT, ο δρομολογητής, θα καταλάβει ότι δεν είναι παλιός γείτονας αλλά καινούργιος.

- REPLY

Τα REPLY είναι πακέτα που στέλνονται σαν απάντηση στα QUERY, η διαφορά τους με τα UPDATE είναι ο τρόπος που τα διαχειρίζεται το DUAL. Τα REPLY είναι απαραίτητα να σταλούν σε απάντηση ενός QUERY καθώς ο δρομολογητής που στέλνει τα QUERY κρατά ένα πίνακα όπου αναγράφει τους γείτονες που έχει στείλει τα πακέτα και περιμένει απάντηση. Καθώς δέχεται πακέτα θα παραλαμβάνει ταυτόχρονα και τα πακέτα REPLY όπου θα αφαιρεί τον γείτονα από τον παραπάνω πίνακα. Όταν του έχουν στείλει όλοι οι γείτονες απαντήσεις μπορεί να ξεκινήσει τον υπολογισμό της διαδρομής.

- SIA-QUERY

Τα πακέτα SIA-QUERY χρησιμοποιούνται όπως τα πακέτα QUERY με τη διαφορά ότι στέλνονται μόνο όταν έχει ήδη σταλεί ένα πακέτο QUERY και δεν έχει ληφθεί απάντηση. Αυτό συμβαίνει επειδή κατά την διάρκεια του υπολογισμού μπορεί να χάθηκε το πακέτο. Οπότε με τα SIA-QUERY μπορούν να ξαναρωτήσουν για την διαδρομή. Ο λόγος ύπαρξής τους είναι ότι δεν μπορεί να στείλει ξανά πακέτο QUERY γιατί αυτό θα οδηγήσει σε εσφαλμένο συμπέρασμα από τον γειτονικό δρομολογητή, δηλαδή ότι υπήρξε αλλαγή της τοπολογίας ενώ η διαδρομή ήταν σε

κατάσταση ACTIVE. Όταν ένας δρομολογητής λάβει ένα πακέτο SIA-QUERY πρώτα στέλνει το acknowledge number και μετά επεξεργάζεται τις πληροφορίες του πακέτου. Σε περίπτωση που ένας δρομολογητής λάβει ένα πακέτο SIA-QUERY χωρίς να έχει λάβει προηγουμένως ένα πακέτο QUERY θα το επεξεργαστή σαν να ήταν απλό QUERY.

- SIA-REPLY

Τα πακέτα SIA-REPLY όπως και τα πακέτα REPLY στέλνονται σαν απάντηση στα SIA-REQUERY. Άμα η διαδρομή είναι ακόμα σε κατάσταση ACTIVE τότε θα έχει το flag ACTIVE. Το πακέτο SIA-REPLY επιτρέπει στον δρομολογητή που έστειλε το SIA-QUERY να επιλέξει άμα θα περιμένει τον γείτονα για τον υπολογισμό των διαδρομών του, δηλαδή να μεταβεί η κατάσταση τους σε PASSIVE, αλλιώς θα ξεκινήσει την εύρεση καινούργιας διαδρομής. Στην περίπτωση που δεν απαντήσει ο γείτονας, ο δρομολογητής θα διαγράψει τα δρομολόγια του γείτονα.

### **1.11 Είδη διαδρομών**

Στο πρωτόκολλο EIGRP διακρίνονται δυο είδη διαδρομών, τα internal και external routes. Τα internal routes είναι διαδρομές οι οποίες προέρχονται από το πρωτόκολλο EIGRP και η μεταφορά τους είναι πιο εύκολη επειδή στέλνουν μόνο τις βασικές πληροφορίες. Σε αντίθεση τα external routes είναι διαδρομές που προέρχονται από εξωτερικές πηγές όπως static routes ή άλλα πρωτόκολλα δρομολόγησης. Για την μεταφορά τους χρειάζονται περισσότερα bytes καθώς μεταφέρουν περισσότερα δεδομένα.

Κατά την επιλογή διαδρομών τα internal routes θεωρούνται πιο έμπιστα και προτιμούνται από τα external. Πρέπει να αναφερθεί, ότι εφόσον υπάρχουν internal route που φτάνουν στον προορισμό δεν θα χρησιμοποιηθούν τα external. Ωστόσο, είναι αρκετά χρήσιμα σε αυτόνομα συστήματα τα όποια έχουν border router τα οποία χρησιμοποιούν διαφορετικά πρωτόκολλα.

### **1.12 Topology Table**

Στον πίνακα τοπολογίας αποθηκεύονται δεδομένα για όλους τους προορισμούς που έχουν στείλει οι γείτονες. Κάθε προορισμός εμπεριέχει όλες τις εγγραφές των διαδρομών που έχουν σταλεί από τους γείτονες. Κατά των υπολογισμό των διαδρομών οι δρομολογητές ανατρέχουν τα δεδομένα για να επιλεγεί ο καινούργιος successor.

### **1.13 Security**

Το πρωτόκολλο EIGRP θα δημιουργήσει γειτνιάσεις με οποιονδήποτε δρομολογητή στείλει ένα πακέτο HELLO. Η προστασία από επιθέσεις επιτυγχάνετε με απλή πιστοποίηση των γειτόνων μέσω των αλγορίθμων κρυπτογράφησης SHA2-256 ή MD5. Όσο για τις επιθέσεις DOS θα πρέπει να καταπολεμηθούν βάζοντας όρια στην αποστολή πακέτων. Όταν το πρωτόκολλο ρυθμιστεί να ζητά πιστοποίηση από τους γείτονες του, στα πακέτα θα προστεθούν οι απαραίτητες πληροφορίες για την πιστοποίηση. Επίσης, κατά την λήψη ενός πακέτου ο δρομολογητής θα κοιτάξει άμα υπάρχουν τα δεδομένα για την πιστοποίηση. Σε περίπτωση που λείπουν τα δεδομένα ή υπάρχει κάποιο σφάλμα θα πετάξει το πακέτο χωρίς να το επεξεργαστεί. Ο λόγος που δεν χρησιμοποιούνται πολύ συνθέτη αλγόριθμοι κρυπτογράφησης είναι ότι η επεξεργαστική ισχύ για την αποκρυπτογράφηση δεν επιτρέπει τη γρήγορη σύγκληση δικτύου.

Αν και η ασφάλεια του μπορεί να χαρακτηριστεί ελλιπής, χρησιμοποιώντας αλγορίθμους κρυπτογράφησης δημοσίου κλειδιού ή κάποιων άλλων πιο βαρύ αλγόριθμο κρυπτογράφησης, απλώς θα επιβαρύνουμε τον δρομολογητή με υπολογισμούς.

### **1.14 Αποστολή πακέτων μέσω multicast**

Όταν είναι εφικτό, το πρωτόκολλο EIGRP, στέλνει τα πακέτα μέσω multicast ώστε να μειώσει το απαιτούμενου εύρος ζώνης. Έτσι αντί να στέλνει το ίδιο πακέτο ξεχωριστά σε κάθε παραλήπτη, το στέλνει σε όλους με ένα πακέτο. Η αποστολή των multicast πακέτων γίνεται στην διεύθυνση 224.0.0.10 για το IPv4 και στην

διεύθυνση FF02::A για το IPv6 και σαν αποστολέας μπαίνει η διεύθυνση του δρομολογητή σε αυτό το δίκτυο. Για να λειτουργήσει σωστά η αποστολή πακέτων μέσω multicast θα πρέπει πριν την αποστολή, ο δρομολογητής να ελέγχει άμα οι γείτονες σε αυτό το δίκτυο έχουν στείλει τις επιβεβαιώσεις λήψης προηγούμενων πακέτων. Σε περίπτωση που όλοι έχουν στείλει τις επιβεβαιώσεις τότε απλώς στέλνει το πακέτο χωρίς καμιά περαιτέρω διαδικασία. Όταν ένας ή περισσότεροι γείτονες δεν έχουν στείλει επιβεβαιώσεις τότε πριν την αποστολή του multicast πακέτου στέλνετε ένα πακέτο HELLO, το οποίο εμπεριέχει τις διευθύνσεις των γειτόνων που δεν έχουν στείλει επιβεβαιώσεις. Ακόμα εμπεριέχει το sequence number του πακέτου που πρόκειται να στείλει μαζί με το flag CR. Με αυτόν το τρόπο ενημερώνει τους γείτονες να μην επεξεργαστούν το πακέτο και απλώς να το αγνοήσουν. Στην συνέχεια αποθηκεύει αντίγραφο του multicast πακέτου στην ουρά αποστολής πακέτων του κάθε γείτονα. Έτσι μπορεί να γίνει επαναποστολή του πακέτου άμα χαθεί. Η επαναποστολή του πακέτου προς ένα γείτονα δεν θα γίνει μέσω multicast αλλά μέσω unicast για να μην το παραλάβουν ξανά οι υπόλοιποι γείτονες. Συνεπώς με αυτόν τον τρόπο τα πακέτα θα φτάσουν με την σωστή σειρά υπό όλες τις συνθήκες.

Αν δεν είχαμε αυτόν τον μηχανισμό τότε κατά την χρήση αποστολής πακέτων μέσω multicast οι δρομολογητές μπορεί να παρελυπταν άθελα τους την σειρά αποστολής πακέτων, διότι θα γινότανε αποστολή χωρίς να ζητείτε επιβεβαίωση της λήψης των προηγούμενων πακέτων. Οπότε θα υπήρχε πιθανότητα να παραλάβει το πακέτο χωρίς να έχει παραλάβει τα προηγούμενα.

### **1.15 Χειρισμός ACTIVE διαδρομών**

Όταν μια διαδρομή μεταβεί σε κατάσταση ACTIVE ξεκινάει ένας μηχανισμός που ονομάζετε Active Route Duration Control. Ο μηχανισμός αυτός έχει σκοπό να οριοθέτηση του χρόνου στον οποίο η διαδρομή θα παραμείνει σε κατάσταση ACTIVE και θα περιμένει για απάντηση από τους γείτονες της. Ο προκαθορισμένος χρόνος για την διαδρομή να μείνει σε κατάσταση ACTIVE είναι έξι λεπτά. Αυτός ο χρόνος είναι διαιρεμένος σε τέσσερα ισόχρονα στάδια. Στα

πρώτα 90 δευτερόλεπτα είναι το πρώτο στάδιο στο οποίο ο δρομολογητής περιμένει το πακέτο REPLY από τους γείτονες. Και σε τρεις ακόμα περιόδους των 90 δευτερολέπτων στις οποίες ο δρομολογητής θα στείλει ένα SIA-QUERY. Άμα οι γείτονες που δεν έχουν στείλει REPLY, δεν καταφέρουν να απαντήσουν είτε με SIA-REPLY είτε με REPLY, θα μπουν σε κατάσταση stack-in-active, θα διαγραφούν όλες οι διαδρομές και θα αφαιρεθεί ο γείτονας.

### **1.16 Διαδικασία Γειννίασης Δρομολογητών**

Σε ένα πρωτόκολλο δρομολόγησης είναι απαραίτητο να μπορεί ο δρομολογητής να βρίσκει τους γείτονες του δυναμικά χωρίς την παρεμβολή του χρήστη. Επίσης, πρέπει εντοπίζει πότε οι γείτονες δεν είναι πια διαθέσιμη για οποιονδήποτε λόγο, όπως βλάβη σε συσκευή ή πρόβλημα στο δίκτυο. Αυτό πρέπει να επιτυγχάνετε με μικρή επιβάρυνση στο δίκτυο και στον δρομολογητή. Για αυτό το πρωτόκολλο EIGRP χρησιμοποιεί μικρά πακέτα HELLO για να δείξει ότι ο δρομολογητής είναι ενεργός. Όταν σε ένα δρομολογητή λειτουργεί το πρωτόκολλο EIGRP, ο δρομολογητής A στέλνει ανά τακτά χρονικά διαστήματα ένα πακέτο HELLO για να κάνει αντιληπτή την ύπαρξη του στους άλλους δρομολογητές. Ένας δρομολογητής B που θα παραλάβει αυτό το πακέτο θα ελέγξει άμα πληρούνται οι προϋποθέσεις, δηλαδή οι K μεταβλητές να είναι ίσες, να είναι ρυθμισμένα να στέλνουν και τα δύο πιστοποιήσεις ή αλλιώς κανένας, και το HOLDOUT να είναι ίδιο. Εφόσον πληρούνται οι προϋποθέσεις, θα γίνουν γείτονες και θα στείλει ένα άδειο πακέτο UPDATE με το flag INIT για ξεκινήσει η γειννίαση. Έπειτα ο δρομολογητής A θα παραλάβει το πακέτο UPDATE και αυτός με την σειρά του θα στείλει ένα άδειο πακέτο UPDATE με το flag INIT. Στην συνέχεια ο δρομολογητής B θα στείλει πακέτα UPDATE με τον πίνακα δρομολόγησης του και στο τελευταίο θα βάλει το flag EOT (End of table) για να δείξει ότι τελείωσε με η μεταφορά δεδομένων. Πριν τελειώσει η γειννίαση οι δρομολογητές δεν θα στείλουν πακέτα QUERY ή REPLY μεταξύ τους. Η κατάσταση αυτή χαρακτηρίζετε ως PENDING και συμβαίνει αποκλειστικά κατά την γειννίαση. Κατά την διάρκεια δημιουργίας μιας γειννίασης ένας δρομολογητής

μπορεί να λάβει ένα πακέτο QUERY και ξεκινήσει τον υπολογισμό μιας διαδρομής, τότε ο δρομολογητής δεν θα στείλει πακέτα QUERY στους PENDING γείτονες και θα συνεχίσει την διαδικασία εύρεσης της διαδρομής χωρίς να περιμένει απάντηση. Ακόμα και σε περίπτωση που σταλθεί ένα πακέτο QUERY μέσω multicast ο δρομολογητής που θα το παραλάβει, κατά το στάδιο της γειτνίασης, δεν θα στείλει απάντηση.

### **1.17 Κ Μεταβλητές**

Οι Κ μεταβλητές χρησιμοποιούνται σαν βάρη για τον υπολογισμό του κόστους της διαδρομής. Ο διαχειριστής του δικτύου μπορεί να αλλάξει τα βάρη ώστε να επηρεάσει τα κριτήρια για την επιλογή των διαδρομών. Όσο πιο μικρή είναι το κόστος τόσο πιο καλή είναι και η διαδρομή. Το πρωτόκολλο EIGRP χρησιμοποιεί πέντε διαφορετικές μεταβλητές τις K1, K2, K3, K4 και K5. Για να μπορούν δυο δρομολογητές να γίνουν γείτονες θα πρέπει αυτές οι μεταβλητές να είναι ίσες και στους δύο. Η K1 χρησιμοποιείται για να δώσει βάρος στο εύρος ζώνης των συνδέσεων. Όπου το εύρος ζώνης υπολογίζετε από το πηλίκο του μέγιστου θεωρητικά εύρους ζώνης ( $10^7$ kbps) πολλαπλασιασμένο με το 256 προς το μικρότερο εύρος ζώνης συνδέσμου από όλους του κόμβους προς τον προορισμό. Έτσι σύνδεσμοι μεγαλύτερου εύρους ζώνης βγάζουν μικρότερα αποτελέσματα. Η K2 χρησιμοποιείται για να δώσει βάρος στο φόρτο του δικτύου. Η K3 χρησιμοποιείται για να δώσει βάρος στην χρονική καθυστέρηση. Όπου η χρονική καθυστέρηση παράγεται από το άθροισμα των χρόνων για να φτάσει ένα bit στον προορισμό. Μετρείται σε δέκα μονάδες του μικρο δευτερολέπτου πολλαπλασιασμένο με το 256. Και η K4 και K5 χρησιμοποιούνται για να δίνουν βάρος στην ποιότητα και την απώλεια πακέτων. Οι προεπιλεγμένες τιμές των Κ μεταβλητών είναι:

$$K1 == K3 == 1$$

$$K2 == K4 == K5 == 0$$

Με την προσθήκη του Wide Metric προστέθηκε και η μεταβλητή K6 οι οποία χρησιμοποιείται για να δίνει βάρος στα επιπλέον στοιχεία όπως την κατανάλωση ρεύματος και την απόκλιση του κύματος (jitter).

### **1.18 Classic Metric**

Το classic metric αναφέρεται στον τύπο για την εύρεση της απόστασης. Για τον υπολογισμό της απόστασης χρησιμοποιείται ο παρακάτω τύπος.

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

Όπου για εύρος ζώνης χρησιμοποιείται το μικρότερο εύρος ζώνης, κατά μήκος όλης της διαδρομής. Για την χρονική καθυστέρηση χρησιμοποιείται το άθροισμα των χρονικών καθυστερήσεων, όλων των διαδικτυακών διασυνδέσεων. Το load και reliability παίρνουν τιμές μεταξύ του 1 και 255.

$$BW = (256 * (10^7)) / BW_{\min}$$

$$\text{delay} = [\text{sum of delays}] * 256$$

Χρησιμοποιώντας τις προκαθορισμένες K μεταβλητές, ο τύπος απλοποιείται στην μορφή:

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

### **1.19 Wide Metric**

Με την εμφάνιση διαδικτυακών διασυνδέσεων με ταχύτητες μεγαλύτερες των 10Gbps δημιουργήθηκε καινούργιο πρόβλημα. Λόγο της μέγιστης τιμής εύρους ζώνης ( $10^7$ ) που έχει δοθεί κατά των υπολογισμό του εύρους ζώνης, το αποτέλεσμα που θα δώσει για οποιοδήποτε εύρος ζώνης μεγαλύτερο των 10Gbps θα είναι ίδιο. Αυτό συμβαίνει διότι η ακέραιη διαίρεση του 10.000.000 με μεγαλύτερο αριθμό γυρνά πάντα αποτέλεσμα 0. Για παράδειγμα, δυο διαδικτυακές



διασυνδέσεις 10Gbps και 20Gbps χρησιμοποιώντας τις προκαθορισμένες Κ μεταβλητές θα δώσουν το ίδιο αποτέλεσμα.

Για παράδειγμα:

11 GigabitEthernet:

$$\text{Scaled Bandwidth} = 10,000,000 / 11000000$$

$$\text{Scaled Delay} = 10 / 10$$

$$\text{Composite Metric} = (0 + 1) * 256 = 256$$

20 GigabitEthernet:

$$\text{Scaled Bandwidth} = 10,000,000 / 20000000$$

$$\text{Scaled Delay} = 10 / 10$$

$$\text{Composite Metric} = (0 + 1) * 256 = 256$$

Οι wide metric διαδρομές είναι συμβατές με τις classic metric. Για τον υπολογισμό των wide metric διαδρομών χρησιμοποιείτε διαφορετικός τύπος ο οποίος εισήγαγε καινούργιες σταθερές.

EIGRP_DELAY_PICO	1.000.000
EIGRP_WIDE_SCALE	65.536
EIGRP_RIB_SCALE	128

Για τον υπολογισμό του εύρους ζώνης, ο παρονομαστής πολλαπλασιάζεται με την σταθερά EIGRP\_WIDE\_SCALE για μην γυρνάει μηδενικό πηλίκο. Ενώ ο υπολογισμός της χρονικής καθυστέρησης, μετατρέπεται σε υπολογισμό λανθάνουσας περιόδου και βγαίνει από τον τύπο:

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

$$\text{Latency} = K3 * (\text{Delay} * \text{EIGRP\_WIDE\_METRIC}) / \text{EIGRP\_DELAY\_PICO}$$

Ο τύπος για τον υπολογισμό του metric είναι:

$$\text{metric} = [((K1 * \min(\text{Throughput})) + \text{sum}(\text{Latency}) + (K6 * \text{ExtAttr})) * (K5 / K4)]$$

Με την χρήση των προκαθορισμένων τιμών ο τύπος απλοποιήτε στην μορφή

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

Τέλος, επειδή ο πίνακας δρομολόγησης δέχεται τιμές έως 32 bit πρέπει να διαιρούμαι το metric με την σταθερά EIGRP\_RIB\_SCALE, γιατί αλλιώς τα αποτελέσματα δεν θα χωρούσαν να αποθηκευτούν σε μεταβλητές των 32 bit. Για να μην υπάρξει πρόβλημα μεταξύ wide και classic metric όταν προστεθεί η πρώτη wide διαδρομή, τότε αδειάζει όλος ο πίνακας δρομολόγησης και ξαναμπαίνουν όλες οι διαδρομές.

## **1.20 TVL**

Οι πληροφορίες των πακέτων μεταφέρονται με την μορφή των TLV τα οποία ενθυλακώνονται μέσα στο πακέτο μετά το EIGRP header. Η ονομασία TLV προέρχεται από το {type, length, value} που είναι η δομή τους. Τα TLV είναι μεταβλητού μήκους για μην στέλνουμε περιττά δεδομένα, αν και καθιστούν την επεξεργασία τους δυσκολότερη στην υλοποίηση σε συγκεκριμένα σημεία. Για την

ανάγνωση των TLV φτάνει να δούμε τον τύπο και το μήκος τους ώστε να μπορέσουμε να το επεξεργαστούμε. Κάθε είδος πακέτου φέρει συγκεκριμένα είδη TVL, αν και από προσωπική εμπειρία ο δρομολογητής θα αγνοήσει TLV τα οποία βρίσκονται σε λάθος πακέτα.

Με το TLV μπορεί το EIGRP να στείλει μια παρτίδα διαφορετικών δεδομένων με την αποστολή ενός πακέτου αντί να στέλνει τα δεδομένα σε ξεχωριστά πακέτα. Για παράδειγμα άμα είχε internal άλλα και external route θα έπρεπε να στείλει δυο διαφορετικά πακέτα. Οπότε θα χρειαζόμασταν περισσότερα είδη πακέτων ή κάποιον άλλο μηχανισμό για να διαχωρίσουμε τις διαδρομές. Αυτό όχι μόνο επιβαρύνει το δίκτυο περισσότερο αλλά κάνει και το πρωτόκολλο δυσκολότερο στην υλοποίηση και συντήρηση του. Ακόμα γίνεται πιο αισθητό στην περίπτωση του πακέτου HELLO καθώς χρησιμοποιείτε για πολλούς διαφορετικούς λόγους.

Παρακάτω θα εξετασθή η δομή των πιο σημαντικών TLV που χρησιμοποιούνται κατά την λειτουργία του πρωτοκόλλου:

1) 0x0001 – PARAMETER\_TYPE

Το PARAMETER\_TYPE χρησιμοποιείτε στα HELLO πακέτα για μεταφέρει τις K μεταβλητές και το HOLDDTIME. Άμα είναι ίσες οι τιμές και στους δύο δρομολογητές μπορούν να γίνουν γείτονες. Το TLV είναι σταθερού μήκους, 12bit.

2) 0x0002 – AUTHENTICATION\_TYPE

Το AUTHENTICATION\_TYPE χρησιμοποιείτε για την πιστοποίηση του γείτονα. Χρησιμοποιείτε αλγόριθμος MD5 ή SHA2-256. Στο RFC δεν δίνονται αρκετές πληροφορίες για τον τρόπο με τον οποίο υπολογίζετε το hash. Στο auth type μπαίνει η τιμή 0x02 όταν χρησιμοποιείτε MD5 ή 0x03 όταν χρησιμοποιείτε ο αλγόριθμος SHA2. Σε ένα πακέτο μπορούν να εμπεριέχονται πολλαπλά AUTHENTICATION\_TYPE.

3) 0x0003 – SEQUENCE\_TYPE

Το SEQUENCE\_TYPE χρησιμοποιείτε μαζί με το MULTICAST\_SEQUENCE\_TYPE κατά την αποστολή multicast πακέτων για να ενημερώσουν ποιοι γείτονες δεν πρέπει να παραλάβουν το πακέτο. Το address length δείχνει το μέγεθος της διεύθυνσης που θα ακολουθήσει και στο protocol address μπαίνει η διεύθυνση του γείτονα που πρέπει να αγνοήσει το πακέτο.

#### 4) 0x0004 – SOFTWARE\_VERSION\_TYPE

Το SOFTWARE\_VERSION\_TYPE χρησιμοποιείτε για να αναγνωρίσει ο δρομολογητής την έκδοση των TLV που χρησιμοποιούν οι γείτονες. Ένας δρομολογητής που χρησιμοποιεί την έκδοση 1.2 δεν μπορεί να διαβάσει TLV της έκδοσης 2, οπότε σε μεικτά δίκτυα οι δρομολογητές που χρησιμοποιούν την έκδοσή 2 πρέπει να στέλνουν τα πακέτα χρησιμοποιώντας και της δύο εκδόσεις TLV.

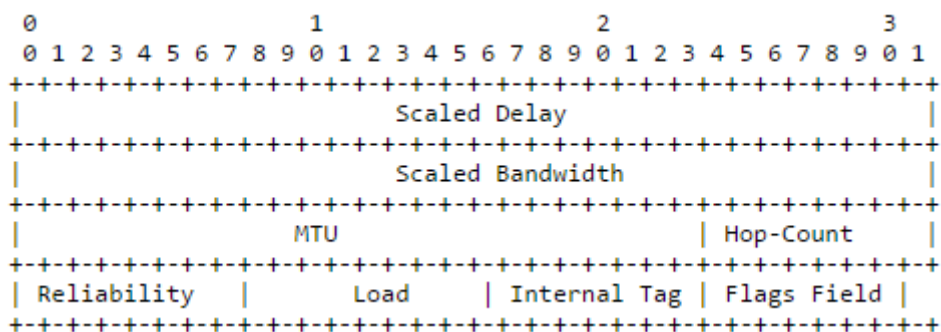
#### 5) 0x0005– MULTICAST\_SEQUENCE\_TYPE

Το MULTICAST\_SEQUENCE\_TYPE περιέχει τον αριθμό του multicast πακέτου που πρέπει να απορριφθεί. Χρησιμοποιείτε σε συνδυασμό με το SEQUENCE\_TYPE TLV.

Για την περιγραφή των internal και external route θα χρειαστεί να τα χωρίσουμε σε κομμάτια και να τα εξηγήσουμε ξεχωριστά. Για αρχή θα εξηγηθούν τα flags. Στα external και internal route TLV μπορεί να συναντήσουμε τρία διαφορετικά flags. Το Source Withdraw (Bit 0) χρησιμοποιείτε όταν η αρχική πηγή της διαδρομής αφαιρεί μια διαδρομή. Το Candidate Default (Bit 1) χρησιμοποιείτε για να δείξει ότι η διαδρομή μπορεί να είναι υποψήφια για να μεταγάγει τα πακέτα όταν ο δρομολογητής δεν ξέρει που να τα στείλει. Από όλες τις διαδρομές που περιέχουν το flag θα επιλεγεί αυτή με τη μικρότερη μονάδα απόστασης. Το Active

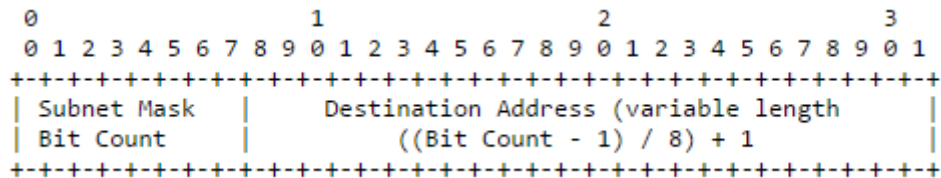
(Bit 2) χρησιμοποιείτε όταν η κατάσταση της διαδρομής για την οποία στέλνουμε δεδομένα είναι σε ACTIVE κατάσταση.

Περαιτέρω τα δεδομένα που μεταφέρονται και χρησιμοποιούνται για τον υπολογισμό της διαδρομής είναι τα εξής. Το άθροισμα της χρονικής καθυστέρησης κατά μήκος όλης της διαδρομής, σε μονάδες των 10 μικροδευτερολέπτων διαιρεμένες με την σταθερά 256. Όταν η χρονική καθυστέρηση έχει την τιμή 0xFFFFFFFF σημαίνει ότι η διαδρομή δεν είναι εφικτή. Επιπλέον είναι το μικρότερο εύρος ζώνης κατά μήκος της διαδρομής, διαιρεμένο με την σταθερά 256. Οι παραπάνω τιμές διαιρούνται με την σταθερά 256 για να μειώσουν τον όγκο τους. Ακόμα είναι η μέγιστη μονάδα μετάδοσης (MTU) όπου διαλέγετε η μικρότερη τιμή κατά μήκος όλης της διαδρομής. Το hop count που δείχνει πόσα δίκτυα χρειάζεται να μεταβεί η διαδρομή. Το reliability που δείχνει την αξιοπιστία της διαδρομής, παίρνει τιμές στο διάστημα 0 και 255, όπου 255 είναι μια τελείως αξιόπιστη διαδρομή. Το load που δείχνει το φόρτο του δικτύου, παίρνει τιμές στο διάστημα 0 και 255, όπου 0 είναι μια διαδρομή που δεν έχει κίνηση. Και τέλος μια τιμή internal tag η οποία δίνεται από τον διαχειριστή και τα flag που προαναφέρθηκαν. Τα παραπάνω έχουν την μορφή:



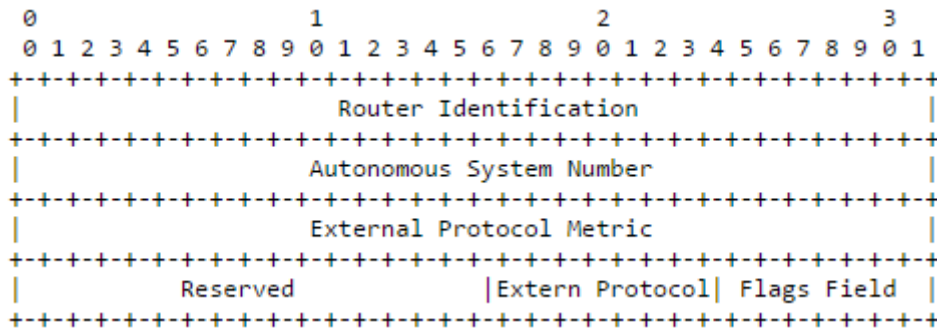
Σχήμα 1: Classic Vector Metric Encoding

Για την μεταφορά της διεύθυνσης του προορισμού στέλνεται μόνο το network κομμάτι στρογγυλοποιημένο στην κοντινότερη οκτάδα. Στην αρχή μπαίνει το subnet mask με μορφή prefix. Για παράδειγμα, το subnet mask 255.255.255.0 έχει prefix 24. Και στην συνέχεια ακολουθεί η διεύθυνση μήκους 3 οκτάδων. Για παράδειγμα, άμα είχαμε το δίκτυο 10.11.3.0 θα παίρναμε μόνο το κομμάτι 10.11.3



Σχήμα 2: Classic Destination Encoding

Για την μεταφορά των external διαδρομών χρειαζόμαστε και επιπλέον πληροφορίες σχετικά με την πηγή από την οποία προήλθε η διαδρομή. Το όνομα της πηγής, για την αναγνώριση της, δίνεται από την μεγαλύτερη διεύθυνση που έχει στις διαδικτυακές διασυνδέσεις της στην περίπτωση του IPv4. Ακόμα εμπεριέχει το αυτόνομο σύστημα το οποίο βρίσκεται η διαδρομή. Καθώς και την μονάδα μέτρησης που έχει δοθεί από το άλλο πρωτόκολλο. Τον αριθμό του πρωτοκόλλου που δημιούργησε την διαδρομή. Και τέλος περιλαμβάνει τα flags που αναφέρθηκαν στην αρχή.



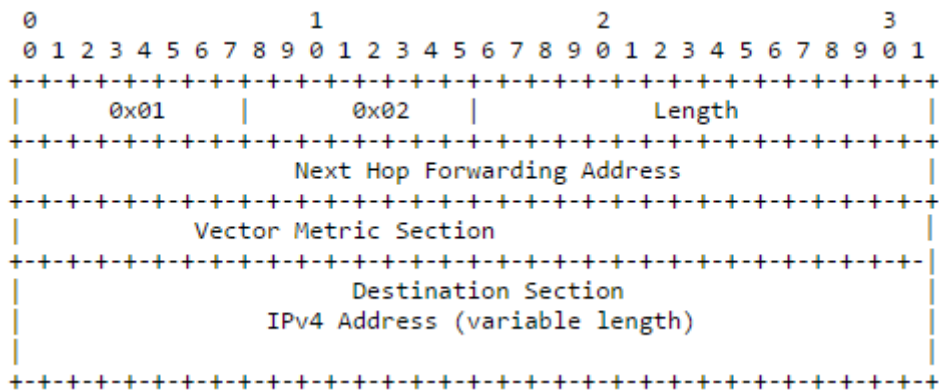
Σχήμα 3: Classic Exterior Encoding

Για την ανταλλαγή διαδρομών μεταξύ των δρομολογητών χρησιμοποιούνται τα Internal και External Route TLV. Διαδρομές που δημιουργήθηκαν από το πρωτόκολλο EIGRP μεταφέρονται με τα Internal Route TLV ενώ διαδρομές που δημιουργήθηκαν από άλλα πρωτόκολλα μεταφέρονται με τα External Route TLV. Κάθε TVL μεταφέρει πληροφορίες για μία διαδρομή, ωστόσο κάθε πακέτο μπορεί να περιέχει πολλαπλά TLV. Με βάση αυτά που ειπώθηκαν προηγουμένως θα περιγράψουμε την δομή των παρακάτω TVL.

#### 0x0102 – IPv4 Internal Route

Τα TLV του τύπου IPv4 Internal Route μεταφέρουν δεδομένα για διαδρομές του πρωτοκόλλου IPv4 τα οποία προήλθαν από το πρωτόκολλο EIGRP. Το TVL περιέχει τη διεύθυνση του δρομολογητή για την μεταγωγή των πακέτων. Όταν αυτή η τιμή είναι 0, η διεύθυνση παίρνεται από την κεφαλίδα του IP header από το πεδίο αποστολέα του πακέτου. Στην συνέχεια ακολουθούν οι πληροφορίες της διαδρομής όπως περιγράφηκαν προηγουμένως. Και τέλος μπαίνει η διεύθυνση του προορισμού.

Πτυχιακή εργασία του φοιτητή Καραχατζή Παρασκευά

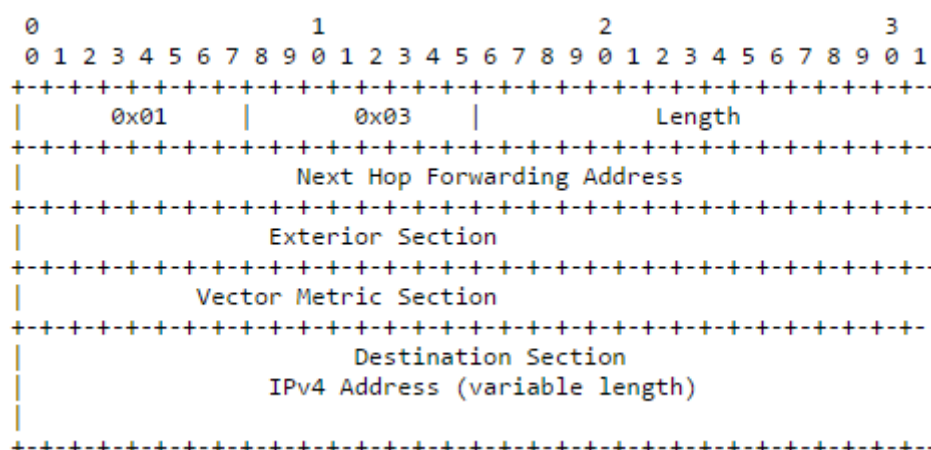


Σχήμα 4: IPv4 Internal route



## 6) 0x0103 – IPv4 External Route

Τα TLV του τύπου IPv4 External Route είναι διαδρομές του πρωτοκόλλου IPv4 οι οποίες προέρχονται από διαφορετικά πρωτόκολλα δρομολόγησης. Η δομή τους είναι παρόμοια με τα TLV Internal Route με την διαφορά ότι περιέχουν πληροφορίες για το πρωτόκολλο που δημιούργησε αυτή την διαδρομή. Πριν από τις πληροφορίες της διαδρομής προηγούνται οι πληροφορίες για το πρωτόκολλο που δημιούργησε την διαδρομή.



Σχήμα 5: IPv4 External route

## ΕΠΙΛΟΓΟΣ

Ανακεφαλαιώνοντας έγινε γνωστή η λειτουργία του πρωτοκόλλου EIGRP και των επιμέρους κομματιών του. Όστε να μπορέσουμε να υιοθετήσουμε την ιδέα και τον τρόπο του μηχανισμού του πρωτοκόλλου. Ο σκοπός της μελέτης του πρωτοκόλλου EIGRP είναι για να μπορέσουμε να υλοποιήσουμε την εφαρμογή δρομολόγησης. Συμπερασματικά, θα μπορούσαμε να πούμε ότι έχουμε το

θεωρητικό υπόβαθρο για να μπορέσουμε να αναπτύξουμε και να ολοκληρώσουμε τον πηγαίο κώδικα.

## ΚΕΦΑΛΑΙΟ 2

### Ανάπτυξη της εφαρμογής για το πρωτόκολλο δρομολόγησης EIGRP

#### **ΕΙΣΑΓΩΓΗ**

Η ανάπτυξη του πηγαίου κώδικα έγινε σταδιακά κομμάτι-κομμάτι. Αρχικά μελετήσαμε τα κομμάτια υψηλού κινδύνου για αποφυγή αποτυχίας του έργου. Πιο συγκεκριμένα αυτοί οι κίνδυνοι αποτελούν την προσθαφαίρεση των διαδρομών στον πίνακα δρομολόγησης του λειτουργικού συστήματος. Επίσης, απαραίτητη είναι η δημιουργία του Transport Layer πρωτοκόλλου, ώστε να μεταφέρονται τα πακέτα με την σωστή σειρά κατά την διάρκεια της εκτέλεσης. Για την ολοκλήρωση της εφαρμογής χρησιμοποιήθηκαν πέρα από τα πρωτόκολλα που προαναφέρθηκαν στο πρώτο κεφάλαιο και μερικά ακόμα πρωτόκολλα. Τέλος, θα αναφέρουμε τα εργαλεία που μας βοήθησαν για την ολοκλήρωση της εφαρμογής.

#### **2.1 Μοντέλο ανάπτυξης**

Η ανάπτυξη του κώδικα του πρωτοκόλλου EIGRP έγινε με το Incremental μοντέλο. Το συγκεκριμένο μοντέλο δανείζετε στοιχεία από το “μοντέλο του Καταρράχτη” και το “Επαναληπτικό μοντέλο”. Η ανάπτυξη γίνεται σε στάδια, προσθέτοντας λειτουργικότητα στην εφαρμογή σε κάθε επανάληψη. Επιλέχτηκε το μοντέλο αυτό επειδή επιτρέπει την εύκολη αποσφαλμάτωση καθώς μπορούν να ελεγχθούν τα κομμάτια ξεχωριστά κατά την διάρκεια της υλοποίησής τους. Επομένως, αυτή η μέθοδος επικεντρώνετε στην ανάπτυξη και τελειοποίηση κομματιών ένα προς ένα ξεχωριστά χωρίς την έγνοια της πολυπλοκότητας του προβλήματος. Ένα από τα αρνητικά του μοντέλου είναι ότι δεν μπορεί να υπολογιστεί το κόστος ενός έργου, που σε αυτή την περίπτωση συνεπάγεται σε χρόνο ανάπτυξης.

## **2.2 Οι Βασικότεροι κίνδυνοι**

Ξεκινώντας την πτυχιακή εργασία απαραίτητη είναι η εύρεση εργαλείων και βιβλιοθηκών που είναι διαθέσιμα από το λειτουργικό ή από την κοινότητα του Debian. Ένα από τα πρώτα προβλήματα που εμφανίστηκαν κατά την μελέτη ήταν η προσθήκη και η αφαίρεση διαδρομών από τον πίνακα δρομολόγησης. Ωστόσο, βρέθηκαν δυο λύσεις με τις οποίες γίνεται η αντιμετώπιση αυτού του προβλήματος, το πρωτόκολλο netlink και η μέθοδος ioctl(). Παρόλα αυτά για την λύση του προβλήματος επιλέχθηκε το πρωτόκολλο netlink. Με το πρωτόκολλο netlink γίνεται εύκολη η πρόσθεση ή αφαίρεση διαδρομών από τον πίνακα δρομολόγησης. Επίσης, μέσω αυτού του πρωτοκόλλου είναι εφικτό να ενημερωθούμε για τυχόν αλλαγές στην κατάσταση των διαδικτυακών διασυνδέσεων του υπολογιστή. Έτσι είναι δυνατόν να ανιχνευτεί ένα σφάλμα σε μια ζεύξη.

Ένα δεύτερο πρόβλημα είναι ότι το πρωτόκολλο EIGRP σε αντίθεση με τα περισσότερα πρωτόκολλα δρομολόγησης, δεν διακινεί πακέτα μέσω του πρωτοκόλλου TCP ή UDP αλλά χρησιμοποιεί ένα δικό του πρωτόκολλο, το RTP (Reliable Transport Protocol). Επομένως, αυτό δημιουργεί πρόβλημα, στο ότι το λειτουργικό σύστημα δεν γνωρίζει πως να χειριστεί το συγκεκριμένο πρωτόκολλο. Ωστόσο, με την χρήση των raw socket μπορούν να προωθηθούν όλα τα πακέτα του συγκεκριμένου πρωτοκόλλου στην εφαρμογή μας όπου εκεί τα επεξεργαζόμαστε εμείς.

## **2.3 Πρωτόκολλα που χρησιμοποιήθηκαν**

Για την ανάπτυξη και ολοκλήρωση της πτυχιακής εργασίας χρησιμοποιήθηκαν δυο πρωτόκολλα, το netlink και το telnet. Το netlink χρησιμοποιήθηκε για την προσθήκη ή αφαίρεση διαδρομών στον πίνακα δρομολόγησης ενώ το telnet για να μπορεί ο διαχειριστής να ελέγχει την λειτουργία του πρωτοκόλλου EIGRP.

Πιο συγκεκριμένα το netlink είναι ένα πρωτόκολλο που επιτρέπει την επικοινωνία μεταξύ του πυρήνα και τις εφαρμογές χρήστη. Μέσω του netlink

μπορούμε να ζητήσουμε από τον πυρήνα να μας δώσει ή και να τροποποιήσει πληροφορίες σχετικά με τον πίνακα δρομολόγησης, των διαδικτυακών διασυνδέσεων και των κανόνων firewall. Οι παραπάνω διαδικασίες είναι εφικτές και χωρίς την χρήση του πρωτοκόλλου καλώντας τις κατάλληλες μεθόδους. Παρόλα αυτά προτιμάτε η χρήση του πρωτόκολλο netlink για την αποφυγή των προβλημάτων συμβατότητας.

Το telnet είναι ένα πρωτόκολλο επικοινωνίας. Αν και δεν χρησιμοποιείται τόσο πολύ στις μέρες μας επειδή έχει αντικατασταθεί με άλλα πρωτόκολλα όπως το SSH που προσφέρει κρυπτογράφηση σε αντίθεση με το telnet. Επιπλέον, το telnet είναι πιο απλό στην υλοποίηση του από ότι το πρωτόκολλο SSH. Τέλος, με το telnet μπορούμε να στέλνουμε και να δεχόμαστε απλό κείμενο που μπορεί να χρησιμοποιηθεί σαν να δουλεύει ο διαχειριστής πίσω από ένα τερματικό.

#### **2.4 Βιβλιοθήκες που χρησιμοποιήθηκαν**

Για την υλοποίηση του πρωτοκόλλου EIGRP χρησιμοποιήθηκαν δυο βιβλιοθήκες, η libnetlink και η libtelnet. Ακόμα πρέπει να αναφερθεί ότι οι βιβλιοθήκες αυτές πάρθηκαν από την αποθήκη πακέτων του Debian.

Η πρώτη βιβλιοθήκη που χρησιμοποιήθηκε είναι libnetlink από το πακέτο iproute2. Είναι μια βιβλιοθήκη που παρέχει μεθόδους για την εύκολη χρήση του πρωτοκόλλου netlink. Αυτό το επιτυγχάνει παρέχοντας μεθόδους που απλοποιούν τη διαδικασία κατασκευής των πακέτων netlink για την επικοινωνία καθώς και μεθόδους για την αποστολή τους.

Η δεύτερη βιβλιοθήκη είναι η libtelnet από τον Sean Middleditch. Χρησιμοποιήθηκε για τον έλεγχο του πρωτοκόλλου EIGRP. Όπως με τη βιβλιοθήκη του netlink και αυτή παρέχει μεθόδους για την μεταφορά δεδομένων (plain text) και την δομή για την δημιουργία της συνόδου. Ακόμα ο διαχειριστής μπορεί να ξεκινήσει μια συνεδρία (session) και να δει πληροφορίες για την κατάσταση των διαδρομών ή των γειτόνων. Τέλος, για την αποσφαλμάτωση του δικτύου μπορεί να

ενεργοποιήσει το debugging για να βλέπει τις αλλαγές στο δίκτυο ώστε να εντοπίζει τα προβλήματα.

## **2.5 Επαναλήψεις ανάπτυξη στην εφαρμογή**

Μετά την εύρεση λύσεων για τους κινδύνους που αναφέρθηκαν ξεκίνησε η σταδιακή συγγραφή του πηγαίου κώδικα και αποσφαλμάτωση της εφαρμογής σε κάθε επανάληψη. Πριν την αρχή της κάθε επανάληψης γινότανε σχεδιασμός της δομής για τον τρόπο υλοποίησης με σκοπό την αποφυγή αδιεξόδων.

Κατά την πρώτη επανάληψη δημιουργήθηκε το πρωτόκολλο RTP (Reliable Transport Protocol), τα νήματα, για την αποστολή και λήψη των πακέτων για το πρωτόκολλο EIGRP. Για την αποσφαλμάτωση δημιουργήθηκαν στατικά πακέτα HELLO με σκοπό τον έλεγχο της επικοινωνίας με τους Cisco δρομολογητές.

Στην δεύτερη επανάληψη φτιάχτηκε η δομή του πρωτοκόλλου. Αναλυτικότερα δομήθηκαν τα βασικά structures για την λειτουργία της εφαρμογής τα οποία στην συνέχεια εμπλουτίζονταν σταδιακά με της απαραίτητες μεταβλητές. Επίσης, υλοποιήθηκαν και οι αλγόριθμοι διαχείρισης των πακέτων HELLO, UPDATE και των TVL. Ύστερα υλοποιήθηκε ο τρόπος δημιουργίας γειτόνων καθώς και η συντήρησή τους. Σε αυτό το στάδιο έγινε ο έλεγχος γειτνίασης των δρομολογητών καθώς και η ανάγνωση διαδρομών από τα πακέτα κατά την γειτνίαση. Ακόμα πρέπει να αναφέρουμε ότι υπήρξαν αλλαγές στις μεθόδους αποστολής ώστε στέλνονται τα πακέτα μέσα από λίστες και όχι στατικά όπως γινόταν στην πρώτη επανάληψη. Τέλος πρέπει να τονιστεί ότι υλοποιήθηκε μόνο η ανάγνωση και όχι ο υπολογισμός των διαδρομών.

Κατά την τρίτη επανάληψη έχοντας πλέον πετύχει την επικοινωνία με τους δρομολογητές ξεκίνησε η υλοποίηση της ανάγνωσης των ρυθμίσεων από ένα αρχείο. Με την ανάγνωση των ρυθμίσεων στήθηκαν οι βάσεις για την επίτευξη του υπολογισμού των διαδρομών που θα γίνει παρακάτω. Έγιναν έλεγχοι για την επαλήθευση της σωστής ανάγνωσης του αρχείου.

Στην τέταρτη επανάληψη έγινε ανασχεδιασμός της δομής του προγράμματος λόγω των πολλών νημάτων. Κάθε διεργασία αυτόνομου συστήματος δημιουργούσε τα δικά της νήματα για την ανάγνωση των πακέτων, καθώς και για κάθε διαδικτυακή διασύνδεση. Με τον ανασχεδιασμό τα νήματα των διαδικτυακών διασυνδέσεων μεταφέρθηκαν στο κυρίως πρόγραμμα όπου αυτά προωθούσαν τα πακέτα στις διεργασίες μειώνοντας σημαντικά τον αριθμό των νημάτων.

Στην πέμπτη επανάληψη υλοποιήθηκε ο υπολογισμός διαδρομών. Η διαδικασία ήταν επίπονη λόγω της ελλείψεως επεξήγησης για την αρχικοποίηση των διαδρομών καθώς και των ασαφειών που υπήρχαν. Μετά από πολλά λάθη και τροποποιήσεις, ολοκληρώθηκε η επανάληψη.

Στην έκτη επανάληψη προστέθηκε το DUAL Finite State Machine. Όπως και στην προηγούμενη επανάληψη άργησε η ολοκλήρωση του για τους ίδιους λόγους. Κυρίως η έλλειψη επεξήγησης για την αρχικοποίηση διαδρομών είχε σαν αποτέλεσμα την πολλαπλή τροποποίηση του πηγαίου κώδικα μέχρι να δώσει τα επιθυμητά αποτελέσματα. Πρέπει να αναφερθεί ότι αυτές η δυο επαναλήψεις σπατάλησαν ένα μεγάλο κομμάτι του χρόνου ανάπτυξης.

Στην έβδομη επανάληψη υλοποιήθηκαν οι μέθοδοι για την προσθήκη και αφαίρεση των διαδρομών από τον πίνακα δρομολόγησης του λειτουργικού συστήματος. Αυτό επιτεύχθηκε χρησιμοποιώντας το πρωτόκολλο netlink. Αν και η ανάπτυξη του δεν ήταν εύκολη, δεν παρουσιάστηκαν σημαντικά προβλήματα.

Στην όγδοη επανάληψη προστέθηκε το telnet για τον έλεγχο των διαδρομών κατά την λειτουργία της εφαρμογής. Η υλοποίηση δεν παρουσίασε προβλήματα καθώς λόγω της βιβλιοθήκης το μεγαλύτερο κομμάτι ήταν ήδη έτοιμο.

Στην ένατη επανάληψη προστέθηκε ο μηχανισμός για το stuck-in-active ο οποίος ενεργοποιείται όταν μια διαδρομή μεταβεί σε ACTIVE κατάσταση. Καθώς και η υλοποίηση και επεξεργασία των πακέτων SIA-QUERY και SIA-REPLY. Για τον έλεγχο του μηχανισμού έπρεπε να χαλάσουμε εσκεμμένα την επεξεργασία των πακέτων REPLY.

Στις επαναλήψεις αναφέρθηκαν συνοπτικά οι κυριότεροι στόχοι τους και όχι αναλυτικά η διαδικασία που ακολουθήθηκε. Σημαντικό είναι να αναφερθεί πως το έγγραφο γράφτηκε μετά από μεγάλο χρονικό διάστημα από την έναρξη της ανάπτυξης του κώδικα οπότε ενδέχεται να έχουν ξεχαστεί επαναλήψεις ή να έχουν αναφερθεί σε λάθος σειρά. Τέλος αν και η γλώσσα προγραμματισμού C δεν έχει κλάσεις, η ανάπτυξη της εφαρμογής έγινε όσο το δυνατό ίδια με την ανάπτυξη εφαρμογών σε αντικειμενοστραφής γλώσσα προγραμματισμού.

## **2.6 Εργαλεία για την ανάπτυξη**

Για την ανάπτυξη και ολοκλήρωση της πτυχιακής εργασίας ήταν απαραίτητα διάφορων ειδών εργαλεία. Τα βασικότερα από αυτά ήταν τα υλικά και τα προγράμματα. Με την συνδυασμένη λειτουργία αυτών των εργαλείων επιτεύχθηκε ο σκοπός μας.

Για την ανάπτυξη του προγράμματος από υλικά εργαλεία που χρησιμοποιήθηκαν ήτανε δυο δρομολογητές της εταιρίας Cisco 1700, ο υπολογιστής που έτρεχε το πρωτόκολλο δρομολόγησης και ένα hub. Η τοπολογία του δικτύου έγινε με εναλλασσόμενες θέσης σύνδεσης μεταξύ των συσκευών ανάλογα με τις ανάγκες κάθε φορά της ανάπτυξης της εφαρμογής. Όστε να παρομοιαστούν όλα τα δυνατά σενάρια που μπορεί να προκύψουν σε ένα δίκτυο.

Απαραίτητα προγράμματα για την υλοποίηση της πτυχιακής εργασίας ήτανε το gedit, το gcc, το gdb, το Valgrind, το Wireshark καθώς και η μηχανή αναζήτησης πακέτων του Debian. Για την συγγραφή του κώδικα χρησιμοποιήθηκε το gedit το οποίο είναι ένας κειμενογράφος. Με την εγκατάσταση plugins προσφέρει μερικά από τα βασικά εργαλεία που μπορεί να προσφέρει και ένας IDE. Για την αποσφαλμάτωση του κώδικα χρησιμοποιήθηκαν δυο προγράμματα. Το gdb είναι ένας από τους πιο γνωστούς και κάλους debugger με τον οποίο μπορεί κάποιος να ανιχνεύσει από που προέρχεται κάποιο σφάλμα κατά την εκτέλεση του προγράμματος. Όταν εμφανιστεί ένα πρόβλημα και η εφαρμογή σταματήσει να

τρέχει, το gdb εμφανίζει την σειρά του κώδικα από την οποία προήλθε. Για τον έλεγχο διαρροών μνήμης χρησιμοποιήθηκε το Valgrind. Είναι ένα framework με διάφορα εργαλεία ανάλυσης, ένα από τα πιο διάσημα εργαλεία του είναι το memcheck το οποίο ανιχνεύει εγγραφές και αναγνώσεις του προγράμματος από μη κατανοημένες θέσεις μνήμης. Για την δειγματοληψία των πακέτων χρησιμοποιήθηκε το Wireshark το οποίο καταγράφει πλαίσια. Ο λόγος για τον οποίο χρησιμοποιήθηκε είναι για τον έλεγχο της ακεραιότητας των πακέτων που στέλνονται από την εφαρμογή και την κατανόηση λειτουργίας του πρωτοκόλλου EIGRP.

## **2.7 Packaging για Debian**

Το packaging έγινε με την βοήθεια των εργαλείων που παρέχει το debhelper. Τα conf και settings αρχεία μπαίνουν στο path /etc/eigrp/ ενώ το εκτελέσιμο αρχείο μπαίνει στο path /usr/bin. Για την εφαρμογή χρειαζόμαστε τα package libtelnet-dev και debhelper.

## **ΕΠΙΛΟΓΟΣ**

Ανακεφαλαιώνοντας η ανάπτυξη του πηγαίου κώδικα αν και ήταν επίπονη και χρονοβόρα καταφέραμε την συγγραφή του με την βοήθεια των εργαλείων. Ακόμα απαραίτητο είναι να αναφέρουμε ότι κατά την συγγραφή του πηγαίου κώδικα χρειάστηκε να κάνουμε αλλαγές σε τμήματα του που είχαν ήδη υλοποιηθεί ώστε να είναι συμβατά με τα καινούργια κομμάτια. Τέλος καταφέραμε την δημιουργία ενός κλώνου του πρωτοκόλλου δρομολόγησης EIGRP.



## ΣΥΜΠΕΡΑΣΜΑΤΑ

Συνοψίζοντας από τα προαναφερόμενα της πτυχιακής εργασίας και της μελέτης που διεξαγάγαμε, μπορούμε να βγάλουμε το συμπέρασμά ότι το πρωτόκολλο EIGRP είναι ένα αξιόπιστο πρωτόκολλο δρομολογήσεις με πολλές δυνατότητες χάρις του σύνθετου μηχανισμού λειτουργίας του συστήματος και της γρήγορης ανταπόκρισης στις αλλαγές του δικτύου. Γιαυτό το λόγω προσφέρει γρήγορη σύγκλιση του δικτύου ακόμα και σε δίκτυα τα οποία έχουν αυξημένο αριθμό πακέτων που χάνονται ή αλλοιώνονται κατά την μετάδοση τους.

Πρέπει να τονιστεί όσο αναφορά την συγγραφή του πηγαίου κώδικα ότι ήταν μια επίπονη και χρονοβόρα εργασία λόγω του ότι η γλώσσα προγραμματισμού που χρησιμοποιήθηκε δεν είναι αντικειμενοστραφής, δηλαδή δεν υποστηρίζει κλάσης οπότε καθιστά την συγγραφή των μεθόδων και την συντήρηση πιο δύσκολη. Ωστόσο η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είχε περισσότερα θετικά παρά αρνητικά. Για παράδειγμα η ανάγνωση και εγγραφή πακέτων σε μορφή byte stream χρησιμοποιώντας τα pointer γίνεται υπερβολικά απλή σε βαθμό που απλώς δίνεις τιμές, σε σχέση με την java στην οποία χρειαζότανε να αναπτυχθούν μέθοδοι για την σωστή εγγραφή σε byte stream. Ωστόσο οι διαρροές μνήμης είναι αρκετά κακό μειονέκτημα της γλώσσας αν και με την χρήση των κατάλληλων εργαλείων μπορεί να καταπολεμηθεί αυτό το πρόβλημα.

Χρειάζεται επίσης να σημειωθεί ότι η εφαρμογή που δημιουργήσαμε δεν μπορεί να φτάσει σε πολυπλοκότητα ρυθμίσεων το κανονικό πρωτόκολλο EIGRP. Αυτό οφείλετε εν μέρη στο ότι δεν υπήρχε η απαραίτητη γνώση διότι η CISCO δεν γνωστοποίησε τη λειτουργία όλου του πρωτοκόλλου για την περαιτέρω ανάπτυξη της εφαρμογής. Ωστόσο, η εφαρμογή μπορεί να λειτουργήσει ικανοποιητικά σε μεγάλο ποσοστό και να δίνει τα ίδια αποτελέσματα με το πρωτότυπο πρωτόκολλο EIGRP. Κατά την άποψη μου είναι μια ενδιαφέρον εφαρμογή διότι έχει προοπτικές για περαιτέρω βελτίωση και ανταποκρίνεται αξιόπιστα στον σκοπό του.

Ο λόγος για τον οποίον δημιουργήθηκε αυτή η εφαρμογή είναι η ελαχιστοποίηση του κόστους κεφαλαίου, όπου χρειάζεται για την δημιουργία και συντήρηση ενός δικτύου, σε αντίθεση με τις συσκευές CISCO router που έχουν υψηλότερο κόστος. Συνεπώς η εφαρμογή μπορεί να επιλύσει την αριθμητική έλλειψη συσκευών CISCO router σε ένα δίκτυο με χαμηλό κόστος και άμεσο τρόπο. Επιπροσθέτως η εφαρμογή έχει την δυνατότητα να προσφέρει στο δίκτυο τη δημιουργία σύνθετων ρυθμίσεων, η οποία υπό άλλες συνθήκες δεν θα μπορούσε να είναι δυνατή. Τέτοιες σύνθετες ρυθμίσεις είναι ο συνδυασμός χρήσης ενός άλλου firewall και της εφαρμογής σε ένα δίκτυο το οποίο λειτουργεί με το πρωτόκολλο EIGRP. Επειδή η εφαρμογή είναι ανοιχτού κώδικα συνεπώς ο καθένας μπορεί να την προσάρμοση στις ανάγκες και το σκοπό χρήσης του. Συμπερασματικά μπορούμε να δούμε ότι η εφαρμογή προσφέρει στους χρήστες της ανάλογα με τις ανάγκες τους τη δημιουργία σύνθετων ρυθμίσεων στο δίκτυο.

## ΑΝΑΦΟΡΕΣ

## ΒΙΒΛΙΟΓΡΑΦΙΑ

Douglas, C. E. (2001). ΔΙΑΔΙΚΤΥΑ με TCP/IP, 4<sup>th</sup> ed. Vol 1. Αθήνα: Εκδόσης Κλειδάριθμος

Enhanced Interior Gateway Routing Protocol (2014, April 10)

Ανακτήθηκε June 25, 2015, από <http://tools.ietf.org/html/draft-savage-eigrp-02>

EIGRP Wide Metrics (χωρίς ημερομηνία)

Ανακτήθηκε June 25, 2015, από [http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute\\_eigrp/configuration/xr-3s/ire-xe-3s-book/ire-wid-met.html](http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xr-3s/ire-xe-3s-book/ire-wid-met.html)

EIGRP Commands (χωρίς ημερομηνία)

Ανακτήθηκε June 25, 2015, από [http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute\\_eigrp/configuration/xr-3s/ire-xe-3s-book/ire-wid-met.html](http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/iproute_eigrp/configuration/xr-3s/ire-xe-3s-book/ire-wid-met.html)

NETLINK(7) (χωρίς ημερομηνία)

Ανακτήθηκε June 25, 2015, από <http://man7.org/linux/man-pages/man7/netlink.7.html>

## ΟΔΗΓΟΣ ΧΡΗΣΗΣ ΛΟΓΙΣΜΙΚΟΥ

Για την ρύθμιση της εφαρμογής υπάρχουν δυο αρχεία, το settings και το conf. Το αρχείο settings είναι προαιρετικό, περιέχει ρυθμίσεις για την εκτέλεση της εφαρμογής. Οι ρυθμίσεις έχουν την μορφή keyword=value, χωρίς κενά. Η έναρξη της επόμενης εντολής στο αρχείο settings σηματοδοτείτε από καινούργια γραμμή και ο χαρακτήρας ';' χρησιμοποιείτε για τα σχόλια. Επειδή αυτές οι ρυθμίσεις είναι προαιρετικές αν δεν υπάρχει η κατάλληλη εντολή θα χρησιμοποιηθεί η προεπιλεγμένη τιμή.

Οι ρυθμίσεις που εμπεριέχονται στο αρχείο settings είναι η επιλογή θύρας για το πρωτόκολλο telnet, ρυθμίζετε με την λέξη κλειδί telnet\_port=<value>, η προεπιλεγμένη θύρα είναι η 11203. Για την πρόσθεση των διαδρομών στον πίνακα δρομολόγησης χρειάζονται δυο μεταβλητές. Η πρώτη μεταβλητή είναι ο αριθμός του πίνακα που θα μπουν οι διαδρομές. Στο Linux ο πίνακας δρομολόγησης αποτελείτε από πολλαπλούς πίνακες. Έτσι ο διαχειριστής μπορεί να φτιάξει κανόνες για την επιλογή των διαδρομών. Αυτό συμβένει καθώς ομαδοποιώντας τις διαδρομές των πρωτοκόλλων μπορεί να ορίσει την προτερότητα για την επιλογή του πίνακα. Η ρύθμιση του γίνεται με την λέξη κλειδί eigrp\_routing\_table=<value> και η προεπιλεγμένη τιμή είναι το RT\_TABLE\_MAIN. Η εφαρμογή δεν μπορεί να διαβάσει κείμενο οπότε πρέπει να δωθεί αναγκαστικά αριθμητική τιμή, για την χρήση του RT\_TABLE\_MAIN απλώς βάζουμε την εντολή σε σχόλιο. Η δεύτερη μεταβλητή είναι ο αριθμός του πρωτοκόλλου που πέρασε τις διαδρομές. Με βάση το rtnetlink(7), μεταβλητές μεγαλύτερες του RTPROT\_STATIC δεν ερμηνεύονται από τον πυρήνα του Linux και χρησιμοποιούνται μόνο σαν πληροφορίες για το χρήστη. Στην έκδοση kernel (2.6.31.13) η τιμή αυτή είναι 4. Η εφαρμογή χρησιμοποιεί σαν προεπιλεγμένη τιμή το 20. Και η λέξη κλειδί για την ρύθμιση του είναι το eigrp\_rtprot=<value>.

Το αρχείο conf είναι το αντίστοιχο startup\_config που υπάρχει στους Cisco δρομολογητές. Κατά την εκκίνηση της εφαρμογής η διεργασία θα διαβάσει το αρχείο conf από το οποίο θα φορτώσει τις ρυθμίσεις. Για μην υπάρξει σύγχυση δεν

υπάρχει το αρχείο `running_config` καθώς δεν έχει γίνει υλοποίηση για αλλαγές την ώρα που εκτελείτε η διεργασία. Αν και είναι τεράστιο μειονέκτημα η υλοποίηση του είναι δύσκολη και χρονοβόρα. Η διαδικασία ρύθμισης του πρωτοκόλλου είναι κατά ένα μεγάλο κομμάτι ίδια με την ρύθμιση του πρωτοκόλου EIGRP σε ένα Cisco δρομολογητή.

Για την ρύθμιση του πρωτοκόλου EIGRP χρησιμοποιούνται οι ίδιες εντολές με το πρωτότυπο πρωτόκολλο EIGRP. Οι εντολές χωρίζονται σε `root commands`, `global commands` και `node commands`, ο διαχωρισμός έγινε με βάση την έκταση της μεταβλητής.

`Root commands` είναι οι εντολές οι οποίες μπορούν να χρησιμοποιηθούν όταν δεν βρισκόμαστε σε κάποιο κόμβο. Τέτοιες εντολές είναι `hostname` και `ip route`. Η εντολή `'hostname'` παίρνει σαν ορίσματα μια συμβολοσειρά, η προεπιλεγμένη τιμή της είναι `'router'` χρησιμοποιείτε για να ονομαστεί ο δρομολογητής, είναι καθαρά για τον χρήστη και δεν επηρεάζει την λειτουργία του προγράμματος. Η εντολή `'ip route'` δέχεται τρία ορίσματα σε συγκεκριμένη σειρά. Η εντολή χρησιμοποιείτε σε συνδυασμό με την εντολή `redistribute static` που βρίσκετε στον κόμβο `router eigrp`. Πρέπει να δωθεί προσοχή στο ότι η εντολή αυτή δεν προσθέτει την διαδρομή στον πίνακα δρομολόγησης αλλά ξεκινά να διαφημίζει την διαδρομή στους γείτονες του. Η διαδρομή θα πρέπει να προστεθεί από τον διαχειριστή του δικτύου. Σαν ορίσματα πρώτα δίνουμε τον προορισμό του δικτύου, το `subnet mask` και την διεύθυνση του γείτονα ή το όνομα του `interface` από την οποία θα γίνετε η μεταγωγή των πακέτων. Για παράδειγμα άμα θέλουμε να κάνουμε μεταγωγή πακέτων για το δίκτυο `10.11.3.0` με `subnet mask 255.255.255.0` προς τον γείτονα `192.168.1.1` ή το `interface eth0`. Η εντολή θα πάρει την μορφή `'ip route 10.11.3.0 255.255.255.0 192.168.1.1'` ή αντίστοιχα `'ip route 10.11.3.0 255.255.255.0 eth0'`.

Τα `global commands` μπορούν να χρησιμοποιηθούν ανεξάρτητα απο το άμα βρισκόμαστε μέσα σε κάποιο `node`. Στα `global command` εμπεριέχονται οι εντολές `'exit'`, `'router eigrp'` και `'interface'`.

Η εντολή 'exit', δεν δέχεται ορίσματα. Χρησιμοποιείτε για να βγούμε από το node εντολών που έχουμε επιλεγμένο.

Η εντολή 'router eigrp', δέχεται σαν ορίσματα τον αριθμό του αυτόνομου συστήματος. Η τιμή πρέπει να βρίσκετε μεταξύ του διαστήματος 1 και 65535. Η διεργασία μπορεί να τρέχει παράλληλα για πολλά αυτόνομα συστήματα. Χρησιμοποιώντας την εντολή μπαίνουμε στον κόμβο εντολών για την ρύθμιση του EIGRP πρωτοκόλλου στο συγκεκριμένο αυτόνομο σύστημα, πιο κάτω θα δωθούν οι εντολές αυτού του κόμβου.

Η εντολή 'interface', δέχεται σαν ορίσματα το όνομα του interface, το οποίο μπορεί βρεθεί εκτελώντας την εντολή ifconfig σε ένα terminal. Σε αντίθεση με τις ρυθμίσεις των CISCO δρομολογητών που ορίζουμε την διεύθυνση του interface, εδώ ορίζουμε μόνο το εύρος ζώνης(bandwidth) και την χρονική καθυστέρηση(delay). Αν και είναι προαιρετική η ρύθμιση των παραπάνω, προτείνεται να γίνετε αλλιώς θα χρησιμοποιηθούν οι τιμές που βγάζει ένα Ethernet interface. Επίσης είναι υποχρεωτική τουλάχιστον η αναφορά του interface αλλιώς η διεργασία θα αγνοήσει το interface.

Τα node commands είναι εντολές που μπορούν να χρησιμοποιηθούν αφότου έχουμε διαλέξει τον κατάλληλο κόμβο. Αυτή την στιγμή υπάρχουν μόνο δυο κόμβοι. Ο κόμβος 'router eigrp' και 'interface', επιλέγονται με την αντίστοιχη εντολή. Αφού επιλέξουμε τον κόμβο 'router eigrp' μπορούμε να δώσουμε τις εντολές για την ρύθμιση του. Οι εντολές που περιέχει αυτός ο κόμβος είναι οι παρακάτω.

Η εντολή 'network', δέχεται δυο ορίσματα το network address του δικτύου που θέλουμε να διαφημίσουμε και τα wildcard bit του (π.χ. network 192.168.1.0 0.0.0.255). Το wildcard δείχνει το host κομμάτι του δικτύου. Σε περίπτωση που δώσουμε subnet mask αντί για wildcard δεν θα υπάρξει πρόβλημα καθώς η εφαρμογή από μόνη της θα το μετατρέψει σε wildcard.

Η εντολή 'redistribute static', δεν δέχεται ορίσματα, θα ξεκινήσει το συγκεκριμένο αυτόνομο σύστημα να διαφημίζει τις 'ip route' διαδρομές που έχουν δοθεί.

Η εντολή 'passive-interface', δέχεται ένα όρισμα το οποίο είναι το όνομα ενός interface, ορίζουμε από ποιο interface να μην διαβάζει πακέτα EIGRP. Δηλαδή δεν θα κάνει γειτνιάσεις με γείτονες από εκείνο το interface.

Η εντολή 'traffic-share', δέχεται ένα από τα δύο προκαθορισμένα ορίσματα 'balanced' ή 'min across-interface' και χρησιμοποιείτε για multipath. Όταν έχει δοθεί η εντολή και υπάρχουν παραπάνω από ένα έγκυρα δίκτυα για την προώθηση πακέτων στον πίνακα δρομολόγησης, θα εγκατασταθούν όλες ή μερικές από αυτές τις διαδρομές και όχι μόνο η καλύτερη. Με την χρήση της 'min across-interface' θα εγκατασταθούν μόνο οι διαδρομές που έχουν ίσο metric με τον successor. Ενώ με την χρήση της 'balanced' θα εγκατασταθούν οι διαδρομές που έχουν μικρότερο metric από το metric του successor επι το variance.

Η εντολή 'variance', δέχεται ένα όρισμα το οποίο είναι ένας αριθμός, μεταξύ του 1 και 128. Όπως αναφέρθηκε και προηγούμενος χρησιμοποιείτε για να ορίσει ο διαχειριστής του δικτύου ποιές διαδρομές θεωρούνται έγκυρες.

Η εντολή 'metric weights', δέχεται επτά ορίσματα. Αρχικά το TOS (Type of Service) η οποία πρέπει πάντα να είναι 0 και η K μεταβλητές σε αριθμητική σειρά από το 1 μέχρι το 5. Αν και η K5 δεν επηρεάζει την λειτουργία της εφαρμογής, προστέθηκε για να είναι συμβατή με τις εντολές του EIGRP. Η K μεταβλητές παίρνουν τιμές από 0 έως 255.

Χρησιμοποιώντας τον κόμβο 'interface' μπορούμε να ρυθμίσουμε το ευρος ζώνης και την χρονική καθυστέρηση. Όπως αναφέρθηκε προηγουμένως άμα δεν οριστούν αυτές οι εντολές θα χρησιμοποιηθούν οι τιμές σαν να ήταν Ethernet interface. Οι εντολές είναι η 'bandwidth' και η 'delay', όπου και οι δυο δέχονται ένα αριθμητικό όρισμα. Το 'bandwidth' δίνετε στην μονάδα τον kilobits ενώ το 'speed' δίνετε σε μονάδες των 10 microsecond.



Για την εκτέλεση της εφαρμογής μπορούμε να δώσουμε παραμέτρους. Δίνετε η δυνατότητα και είναι προτεινόμενο να τρέχουμε την εφαρμογή σαν daemon δίνοντας το όρισμα '--daemon', με αποτέλεσμα η εφαρμογή να τρέχει χωρίς να βλέπουμε κάποιο τερματικό. Για τον έλεγχο της λειτουργίας του πρωτοκόλλου μπορούμε να χρησιμοποιήσουμε το telnet. Μπορούμε να κρατήσουμε log του πρωτοκόλλου δίνοντας το όρισμα '--log=<path>', το path πρέπει να δείχνει σε directory όπου θα δημιουργούνται αρχεία ονοματισμένα με την ώρα έναρξης του πρωτοκόλλου. Άμα δοθεί μόνο το όρισμα '-log' τα αρχεία θα αποθηκεύονται στο directory '/etc/eigrp/logs/'. Η έξοδος των παραπάνω αρχείων δεν έχουν ίδια μορφή με της CISCO αλλά είναι εύκολη στην κατανόηση τους. Η διεργασία θα ψάξει για το conf αρχείο στο directory '/etc/eigrp/', για να διαλέξουμε άλλο directory χρησιμοποιούμε το όρισμα '-conf=<path>'. Το path θα πρέπει να δείχνει το αρχείο και όχι το directory. Με το όρισμα '-var=<path>' ορίζουμε που να ανατρέξει το αρχείο settings, αλλιώς θα ψάξει το αρχείο στο directory '/etc/eigrp/'. Όπως και με το conf το path πρέπει να δείχνει το αρχείο. Δίνοντας το όρισμα '--clear-routes' θα διαγράψει τις διαδρομές από τον πίνακα δρομολόγησης τις οποίες έχει προσθέσει το πρωτόκολλο. Προσοχή καθώς οι διαδρομές που θα σβηστούν είναι οι διαδρομές που έχουν τον αριθμό του πρωτοκόλλου που δόθηκε από το settings ή τις διαδρομές με αριθμό πρωτοκόλλου 20 σε περίπτωση που δεν υπάρχει το αρχείο (λόγο προεπιλεγμένης μεταβλητής) και στην συνέχεια η διεργασία θα τερματίσει. Κατά την εκκίνηση και τον τερματισμό του πρωτοκόλλου διαγράφονται από μόνες τους οι διαδρομές που έχουν προστεθεί από την εφαρμογή.

Για τον έλεγχο της λειτουργίας του πρωτοκόλλου χρησιμοποιούμε το πρωτόκολλο telnet. Η προεπιλεγμένη πύλη σύνδεσης είναι η 11203 αλλά όπως αναφέρθηκε προηγουμένως μπορεί ο διαχειριστής να την αλλάξει. Με παρόμοιο τρόπο που δουλεύει το SSH στους CISCO δρομολογητές μπορούμε να ανατρέξουμε πληροφορίες για την αποσφαλμάτωση του δικτύου. Όταν η γραμμή εντολής τελειώνει με τον χαρακτήρα '?' το telnet θα μας γυρίσει το σύνολο των εντολών που ταιριάζουν με ότι έχει γραφτεί πριν από τον χαρακτήρα.

Οι εντολές που έχουν προστεθεί είναι περιορισμένες, περιλαμβάνει το set για την αποσφαλμάτωση του EIGRP και τις πιο βασικές debug εντολές. Τέλος, στο telnet δεν μπόρεσε να προσομοιαστεί το auto-complete λόγω του τρόπου λειτουργίας του πρωτοκόλλου telnet.