ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Προσομοιώσεις MPLS Δικτύων με χρήση του OMNeT++

Αλέξανδρος Ν. Τσώνης

Επιβλέποντες : **Δημήτριος Βαρουτάς,** Επίκουρος Καθηγητής
Δημήτριος Κατσιάνης, Ε.ΔΙ.Π

ΑΘΗΝΑ

ΑΥΓΟΥΣΤΟΣ 2015

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Προσομοιώσεις MPLS Δικτύων με χρήση του OMNeT++

Αλέξανδρος Ν. Τσώνης
Α.Μ.: Μ1333

ΕΠΙΒΛΕΠΟΝΤΕΣ: **Δημήτριος Βαρουτάς,** Επίκουρος Καθηγητής
Δημήτριος Κατσιάνης, Ε.ΔΙ.Π

Αύγουστος 2015

ΠΕΡΙΛΗΨΗ

Στην παρούσα Διπλωματική Εργασία επιχειρείται μία εκτενής παρουσίαση και ανάλυση όλων των πτυχών της τεχνολογίας Multiprotocol Label Switching (MPLS). Μετά από μία εισαγωγή στις τεχνολογίες πριν το MPLS και μία συνοπτική αναφορά των πλεονεκτημάτων του, εξετάζονται η αρχιτεκτονική και τα βασικά στοιχεία του MPLS. Ειδική αναφορά γίνεται στο πρωτόκολλο διανομής ετικετών LDP. Στη συνέχεια εξηγείται ο τρόπος με τον οποίο προωθούνται τα πακέτα στο MPLS δίκτυο, καθώς και ο ρόλος που παίζουν οι ετικέτες στην προώθηση. Κατόπιν, δίνεται ιδιαίτερη έμφαση στην παρουσίαση των δύο βασικότερων εφαρμογών του MPLS: του Virtual Private Network (VPN) και του Traffic Engineering. Εξετάζονται επίσης δύο διαφορετικά μοντέλα Quality of Service για το MPLS, καθώς και οι προτεινόμενοι τρόποι μεταφοράς IPv6 πακέτων μέσω MPLS δικτύων. Τέλος, αναλύονται οι τομείς στους οποίους αναμένεται να επικεντρωθούν οι προσπάθειες για την εξέλιξη του MPLS. Όσον αφορά την πρακτική επαλήθευση της γνώσης που αποκομίσθηκε από την εργασία, πραγματοποιήθηκαν με χρήση του εργαλείου προσομοιώσεων OMNeT++ case studies MPLS δικτύων. Σε πρώτη φάση μελετώνται δύο υπάρχουσες προσομοιώσεις που παρέχονται στη βιβλιοθήκη INET και αφορούν α) τον τρόπο λειτουργίας και προώθησης πακέτων σε ένα MPLS δίκτυο και β) η λειτουργία του πρωτοκόλλου LDP για την κατανομή των ετικετών. Στη συνέχεια, δημιουργούνται και παρουσιάζονται δύο νέες προσομοιώσεις για τις βασικές εφαρμογές του MPLS. Η πρώτη προσομοίωση αφορά τη λειτουργία ενός MPLS VPN δικτύου και η δεύτερη προσομοίωση αφορά το Traffic Engineering στο MPLS. Ο κώδικας των προσομοιώσεων παρατίθεται στα Παραρτήματα Ι και ΙΙ.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Multiprotocol Label Switching (MPLS)

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Προσομοιώσεις, OMNeT++, VPN, Traffic Engineering, QoS

ABSTRACT

In this Master Thesis a detailed presentation and analysis of all aspects of the Multiprotocol Label Switching (MPLS) technology is provided. After an introduction in the pre-MPLS technologies and a brief report of its advantages, the MPLS architecture and basic network elements are being examined. There is a specific mention on the Label Distribution Protocol (LDP). Subsequently, we explain the packet forwarding in MPLS networks and the role played by the labels. Afterwards, particular emphasis is given in the presentation of the two fundamentals MPLS applications: Virtual Private Network (VPN) and Traffic Engineering. We also examine two different Quality of Service models for MPLS, as well as the suggested ways of carrying IPv6 packets across an MPLS backbone. Finally, we analyze the areas expected to be the main focus of the efforts regarding the development of MPLS. Finally, for the practical verification of the knowledge reflected in our analysis, some case studies of MPLS networks were made using the OMNeT++ simulation tool. Firstly, we study two existing simulations of the INET framework concerning a) the operation and forwarding of packets in MPLS networks and b) the function of the LDP for label distribution. After that, we create and present two new simulations about MPLS basic applications. The first simulation is about the operation of an MPLS VPN network and the second one, about the MPLS Traffic Engineering. The code of the simulations is given in the Annexes I and II.

SUBJECT AREA: Multiprotocol Label Switching (MPLS)

KEYWORDS: Simulations, OMNeT++, VPN, Traffic Engineering, QoS

Η εργασία αυτή αφιερώνεται στην οικογένειά μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Για τη διεκπεραίωση της παρούσας Διπλωματικής Εργασίας, θα ήθελα να ευχαριστήσω αρχικά τον επιβλέποντα καθηγητή κ. Δημήτρη Βαρουτά για την ευκαιρία που μου έδωσε να εκπονήσω την εργασία αυτή και τη βοήθειά του στην επιλογή ενός θέματος άμεσου ενδιαφέροντος που μου έμαθε πολλά πάνω για την τεχνολογία MPLS, αλλά και για τον προσομοιωτή OMNeT++. Θα ήθελα επίσης να ευχαριστήσω τον επιβλέποντα κ. Δημήτρη Κατσιάνη για τη συμβολή του στην ολοκλήρωση της Διπλωματικής και για τη συνεχή του βοήθεια και υποστήριξη σε όσα προβλήματα αντιμετώπισα. Τέλος, θέλω να ευχαριστήσω την οικογένεια και τους φίλους μου για την αμέριστη υποστήριξη που μου έδειξαν καθόλη τη διάρκεια των σπουδών μου.

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΑΤΑΛΟΓΟΣ ΕΙΚΟΝΩΝ

ΠΡΟΛΟΓΟΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στην Αθήνα από το Δεκέμβριο του 2014 μέχρι και τον Αύγουστο του 2015. Αποτελεί αναπόσπαστο κομμάτι για την απόκτηση του Μεταπτυχιακού Διπλώματος Ειδίκευσης και διεξήχθη κατά το δεύτερο έτος της φοίτησης μου ως μεταπτυχιακός φοιτητής στο Πρόγραμμα Μεταπτυχιακών Σπουδών με ειδίκευση Τηλεπικοινωνιακά Συστήματα και Δικτυακές Τεχνολογίες του Τμήματος Πληροφορικής και Τηλεπικοινωνιών του Εθνικού και Καποδιστριακού Πανεπιστήμιου Αθηνών.

1. Εισαγωγή στο MPLS

1.1 Μία πρώτη ματιά

Το MPLS (Multiprotocol Label Switching) αποτελεί μια δημοφιλή τεχνολογία δικτύωσης που χρησιμοποιεί ετικέτες για να προωθήσει τα πακέτα στο δίκτυο. Οι MPLS ετικέτες διαφημίζονται μεταξύ των δρομολογητών με τρόπο τέτοιο ώστε να οικοδομηθεί μία αντιστοιχία ετικέτας με ετικέτα. Οι ετικέτες επισυνάπτονται στα IP πακέτα, επιτρέποντας στους δορμολογητές να προωθούν την κίνηση του δικτύου ελέγχοντας την ετικέτα και όχι την IP διεύθυνση προορισμού. Τα πακέτα προωθούνται μέσω μεταγωγής ετικέτας αντί IP μεταγωγής.

Η τεχνική της μεταγωγής ετικέτας δεν είναι καινούρια. Τόσο η τεχνολογία του Frame Relay, όσο και του ATM τη χρησιμοποιήσαν για να μετακινήσουν πλαίσια ή κελιά στο δίκτυο. Η κεφαλίδα του ATM κελιού και του Frame Relay πλαισίου υποδεικνύουν το εικονικό κύκλωμα στο οποίο βρίσκεται το κελλί/πλαίσιο. Σε κάθε αναπήδηση από δρομολογητή σε δρομολογητή (εντός του δικτύου), η τιμή της ετικέτας στην κεφαλίδα αλλάζει, στοιχείο που το διαφοροποιεί από την προώθηση των IP πακέτων. Όταν ένας δρομολογητής προωθεί ένα IP πακέτο, δεν αλλάζει η τιμή του προορισμού του πακέτου. Το γεγονός ότι οι MPLS ετικέτες χρησιμοποιούνται για την προώθηση των πακέτων αντί των IP διευθύνσεων συνέβαλλε στη δημοφιλία του πρωτοκόλλου. [1]

1.2 Προ-MPLS πρωτόκολλα

Πριν από το MPLS, τα πλέον δημοφιλή WAN πρωτόκολλα ήταν το ATM και το Frame Relay. Με την αύξηση της δημοτικότητας του Internet, το IP κατέστη το πλέον δημοφιλές πρωτόκολλο, βοηθώντας να δημιουργηθούν VPNs πάνω σε αυτά τα WAN πρωτόκολλα. Οι πελάτες μίσθωναν ATM και Frame relay συνδέσεις ή χρησιμοποιούσαν μισθωμένες γραμμές για να χτίσουν το ιδιωτικό δίκτυό τους. Επειδή οι δρομολογητές του παρόχου πρόσφεραν και μία Layer 2 (L2) υπηρεσία προς τους L3 δρομολογητές του πελάτη, ο διαχωρισμός και η απομόνωση μεταξύ των δικτύων των πελατών ήταν εγγυημένα. Τα δίκτυα αυτά αναφέρονται ως δίκτυα επικάλυψης και χρησιμοποιούνται ακόμα, αν και έχει επικρατήσει η χρήση της υπηρεσίας MPLS VPN. [1]

1.3 Πλεονεκτήματα MPLS

1.3.1 Ψευδο-όφελος

Ένας από τους πρώτους λόγους για ένα πρωτόκολλο εναλλαγής ετικέτας ήταν η ανάγκη για υψηλές ταχύτητες. Παρότι υπήρχε η πεποίθηση ότι η αναζήτηση μιας απλής ετικέτας σε έναν πίνακα θα ήταν ταχύτερη από την αναζήτηση της IP διεύθυνσης, η πρόοδος που έγινε στο hardware για τη μεταγωγή IP πακέτων, κατέστησε αυτό το επιχείρημα αμφισβητήσιμο. Πλέον, οι σύνδεσμοι ενός δρομολογητή έχουν εύρος ζώνης 40 Gbps και ένας δρομολογητής με αρκετές συνδέσεις υψηλής ταχύτητας δεν θα μπορούσε να φέρει εις πέρας τη μεταγωγή όλων των IP πακέτων χρησιμοποιώντας αποκλειστικά τη CPU για να πάρει τις αποφάσεις προώθησης. Αντίθετα, η CPU υπάρχει κυρίως για να διαχειρίζεται το επίπεδο ελέγχου.

Το επίπεδο ελέγχου είναι το σύνολο των πρωτοκόλλων που βοηθά στη δημιουργία των επιπέδων δεδομένων και προώθησης. Τα κύρια συστατικά του επιπέδου ελέγχου είναι τα πρωτόκολλα δρομολόγησης, ο πίνακας δρομολόγησης, καθώς και άλλα πρωτόκολλα ελέγχου και σηματοδοσίας που χρησιμοποιούνται για τη σύσταση του επιπέδου δεδομένων. Το επίπεδο δεδομένων είναι το μονοπάτι προώθησης των πακέτων

διαμέσου ενός δρομολογητή ή μεταγωγέα. Η μεταγωγή των πακέτων – ή επίπεδο προώθησης – γίνεται σε ειδικά σχεδιασμένο hardware ή ολοκληρωμένα κυκλώματα ειδικών εφαρμογών (ASIC). Η χρήση των ASICs στο επίπεδο προώθησης ενός δρομολογητή έχει οδηγήσει τη μεταγωγή IP πακέτων να είναι το ίδιο γρήγορη με των επισημασμένων πακέτων. Συνεπώς, η ταχύτητα δεν μπορεί να είναι ο μόνος λόγος για την υλοποίηση του MPLS σε ένα δίκτυο. [1]

1.3.2 Χρήση ενιαίας δικτυακής υποδομής

Βασική ιδέα του MPLS είναι η επισήμανση των πακέτων με βάση τη διεύθυνση προορισμού ή άλλα προκαθορισμένα κριτήρια και η μεταγωγή της κίνησης πάνω από μία κοινή υποδομή.

Ένας από τους λόγους που το IP έγινε το κυρίαρχο πρωτόκολλο στο δικτυακό κόσμο είναι ότι μπορεί να εξυπηρετήσει πολλές τεχνολογίες. Χρησιμοποιώντας το MPLS μαζί με το IP, μπορεί να επεκταθούν οι δυνατότητες μεταφοράς. Προσθέτοντας ετικέτες στα πακέτα γίνεται δυνατή η μεταφορά πρωτοκόλλων όπως IPv4, IPv6, Ethernet, HDLC, PPP και άλλες L2 τεχνολογίες. Το χαρακτηριστικό όπου κάθε L2 πλαίσιο μεταφέρεται μέσω της MPLS υποδομής λέγεται Any Transport over MPLS (AToM). Οι δρομολογητές δε χρειάζεται να γνωρίζουν το MPLS φορτίο, πρέπει απλά να μπορούν να μεταβιβάσουν τα πακέτα εξετάζοντας την ετικέτα τους. Στην ουσία, η μεταγωγή ετικέτας του MPLS είναι μία απλή μέθοδος μεταγωγής πολλαπλών πρωτοκόλλων σε ένα δίκτυο. Το AToM επιτρέπει στον πάροχο να παρέχει την ίδια L2 υπηρεσία όπως με κάθε μη-MPLS πρωτόκολλο, ενώ ταυτόχρονα χρειάζεται μία ενιαία δικτυακή υποδομή για να μεταφέρει όλη την κίνηση. [1]

1.3.3 Καλύτερη IP over ATM ενσωμάτωση

Η ενσωμάτωση του IP μέσω ATM ήταν από τους βασικούς λόγους ανάπτυξης του MPLS. Η προϋπόθεση για το MPLS στους ATM μεταγωγείς ήταν να γίνουνπιο “έξυπνοι”. Οι ATM μεταγωγείς έπρεπε να τρέξουν ένα IP πρωτόκολλο δρομολόγησης και να υλοποιήσουν ένα πρωτόκολλο διανομής ετικετών. [1]

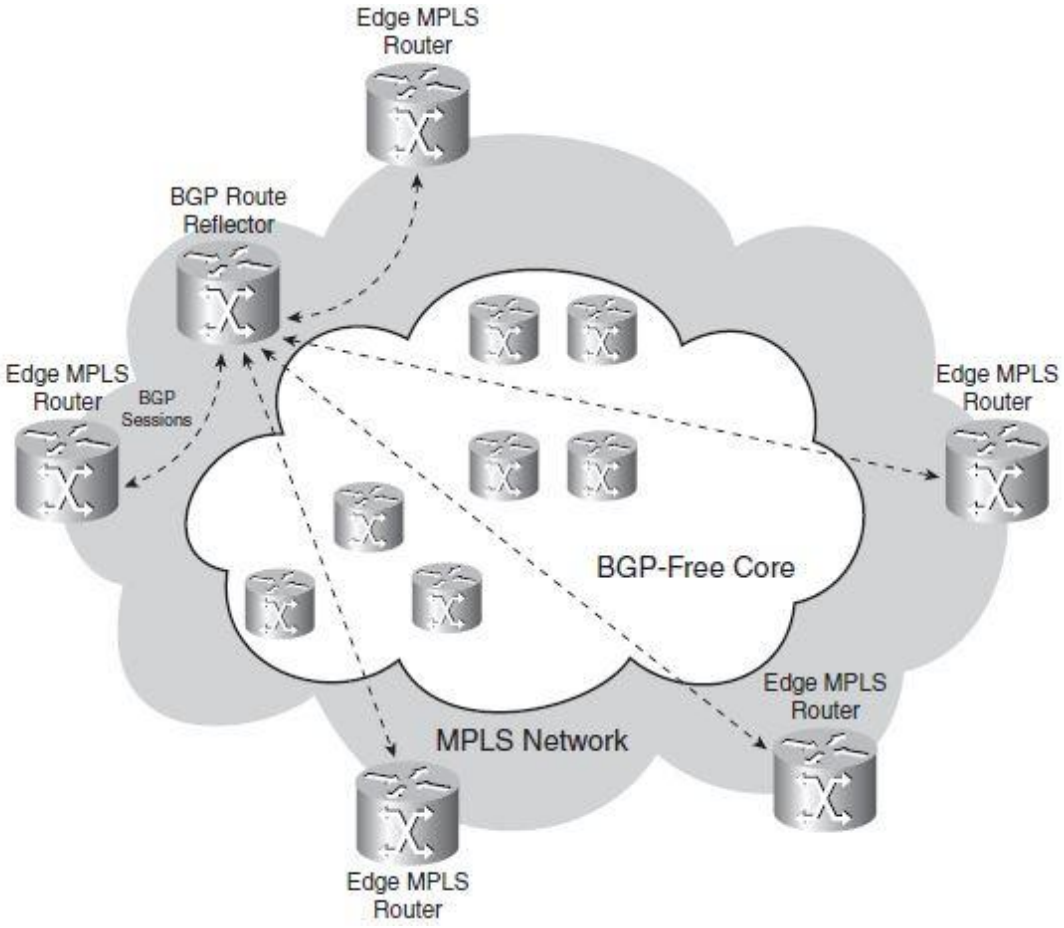
1.3.4 Μη χρήση BGP

Όταν πρέπει να σταλούν πακέτα σε προορισμούς εκτός του δικτύου του παρόχου, τα εξωτερικά IP προθέματα πρέπει να βρίσκονται στον πίνακα δρομολόγησης κάθε δρομολογητή. Το BGP μεταφέρει τα εξωτερικά προθέματα, όπως τα προθέματα των πελατών ή τα προθέματα του Internet, συνεπώς όλοι οι δρομολογητές στο δίκτυο του παρόχου πρέπει να τρέχουν το BGP.

Το MPLS ωστόσο επιτρέπει την προώθηση πακέτων βάσει αναζήτησης ετικέτας και όχι IP διεύθυνσης. Η ετικέτα σχετίζεται με έναν εξωτερικό δρομολογητή και όχι με την IP διεύθυνση του πακέτου και φέρει την πληροφορία που λέει σε κάθε ενδιάμεσο δρομολογητή σε ποιον εξωτερικό δρομολογητή πρέπει να προωθηθεί. Έτσι, οι δρομολογητές στο δίκτυο κορμού δε χρειάζεται να τρέχουν BGP.

Ο δρομολογητής στα άκρα του MPLS δικτύου εξακολουθεί να χρειάζεται να κοιτάξει την IP διεύθυνση του πακέτου και άρα πρέπει να τρέχει BGP. Ένας πάροχος με 200 δρομολογητές στο δίκτυο κορμού, αν υλοποιήσει MPLS, θα χρειαστεί να τρέχει BGP μόνο στα άκρα του, δηλαδή σε 50 περίπου δρομολογητές. Επειδή ο πλήρης πίνακας

του Internet περιέχει περισσότερες από 150.000 διαδρομές, είναι πολύ σημαντικό να μη χρειάζεται να τρέχουν BGP όλοι οι δρομολογητές, καθώς απαιτείται πολύ λιγότερη μνήμη. [1]



Εικόνα 1. BGP στους δρομολογητές του MPLS δικτύου

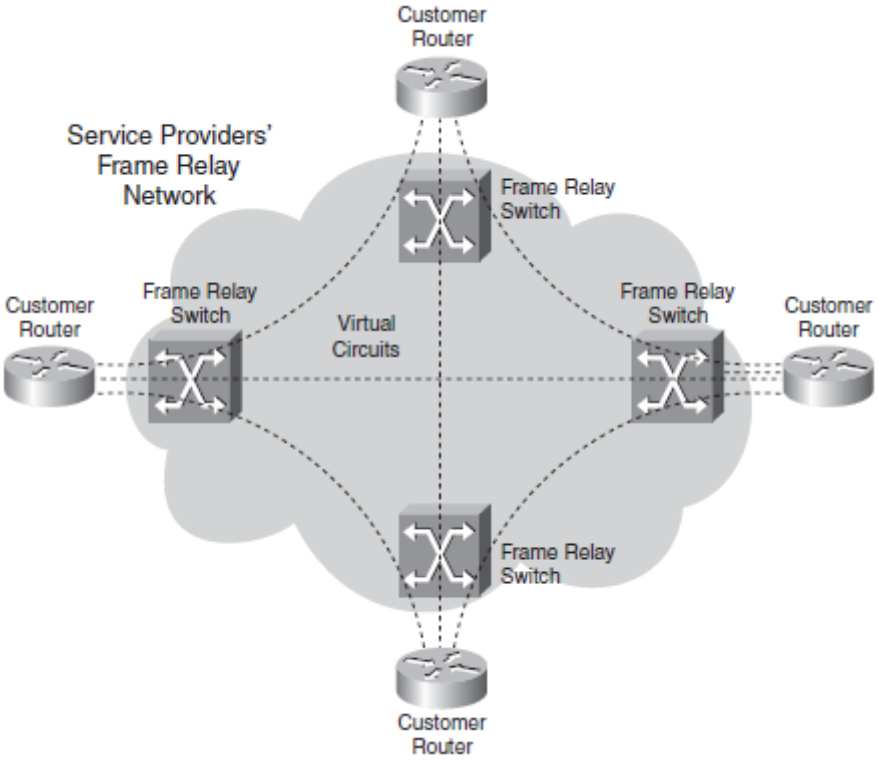
1.3.5 Μοντέλο Peer-to-Peer για το MPLS VPN

Ένα VPN (Virtual Private Network) είναι ένα δίκτυο που εξομοιώνει ένα ιδιωτικό δίκτυο πάνω από μία κοινή υποδομή, δηλαδή το Internet. Οι πάροχοι υλοποιούν δύο μοντέλα για να παρέχουν VPN υπηρεσίες στους πελάτες τους:

- Overlay VPN μοντέλο

Στο overlay μοντέλο, ο πάροχος προσφέρει μία υπηρεσία point-to-point συνδέσεων ή εικονικών κυκλωμάτων σε όλο το δίκτυό του ανάμεσα στους δρομολογητές του πελάτη. Οι δρομολογητές του πελάτη ανταλλάσσουν πληροφορίες δρομολόγησης μεταξύ τους, οι δρομολογητές και οι μεταγωγείς του παρόχου μεταφέρουν τα δεδομένα του πελάτη στο δίκτυο του παρόχου, αλλά μεταξύ ενός δρομολογητή του παρόχου και ενός αντίστοιχου του πελάτη δεν πραγματοποιείται καμία ανταλλαγή πληροφοριών δρομολόγησης. Κατά συνέπεια, οι δρομολογητές του παρόχου δε βλέπουν ποτέ τις διαδρομές του πελάτη.

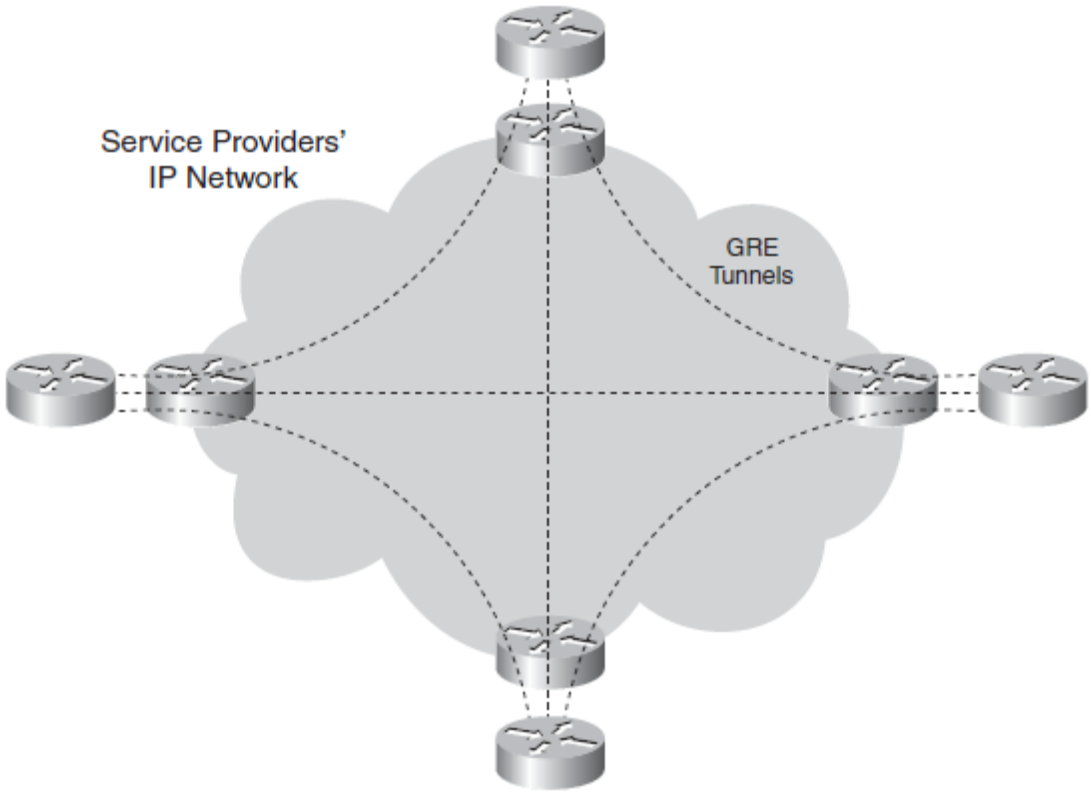
Αυτές οι point-to-point υπηρεσίες μπορεί να είναι L1, L2 ή L3. Παραδείγματα L1 είναι TDM (Time Division Multiplexing – πολυπλεξίας χρόνου), E1, E3, SONET και SDH συνδέσεις. Παραδείγματα L2 υπηρεσιών είναι εικονικά κυκλώματα δημιουργημένα από X.25, ATM ή Frame Relay.



Εικόνα 2. Overlay VPN μοντέλο πάνω από Frame Relay

Στην παραπάνω εικόνα, οι Frame Relay μεταγωγείς του δικτύου δημιουργούν τα εικονικά κυκλώματα μεταξύ των δρομολογητών του πελάτη στα άκρα του Frame Relay δικτύου.

Η overlay υπηρεσία μπορεί να παρασχεθεί και πάνω από το IP L3 πρωτόκολλο, με χρήση GRE (Generic Routing Encapsulation) τούνελ. Τα τούνελ αυτά ενθυλακώνουν τα πακέτα με μία GRE και μία IP κεφαλίδα. Η GRE κεφαλίδα, μεταξύ άλλων, υποδεικνύει το μεταφερόμενο πρωτόκολλο. Η IP κεφαλίδα χρησιμοποιείται για να δρομολογηθούν τα πακέτα διαμέσου του δικτύου του παρόχου. Πλεονέκτημα των GRE τούνελ είναι η δυνατότητα δρομολόγησης κίνησης πέραν του IP. Μπορούν να συνδυαστούν με IPsec τα GRE τούνελ για να παρέχουν ασφάλεια, καθώς τα δεδομένα κρυπτογραφούνται.

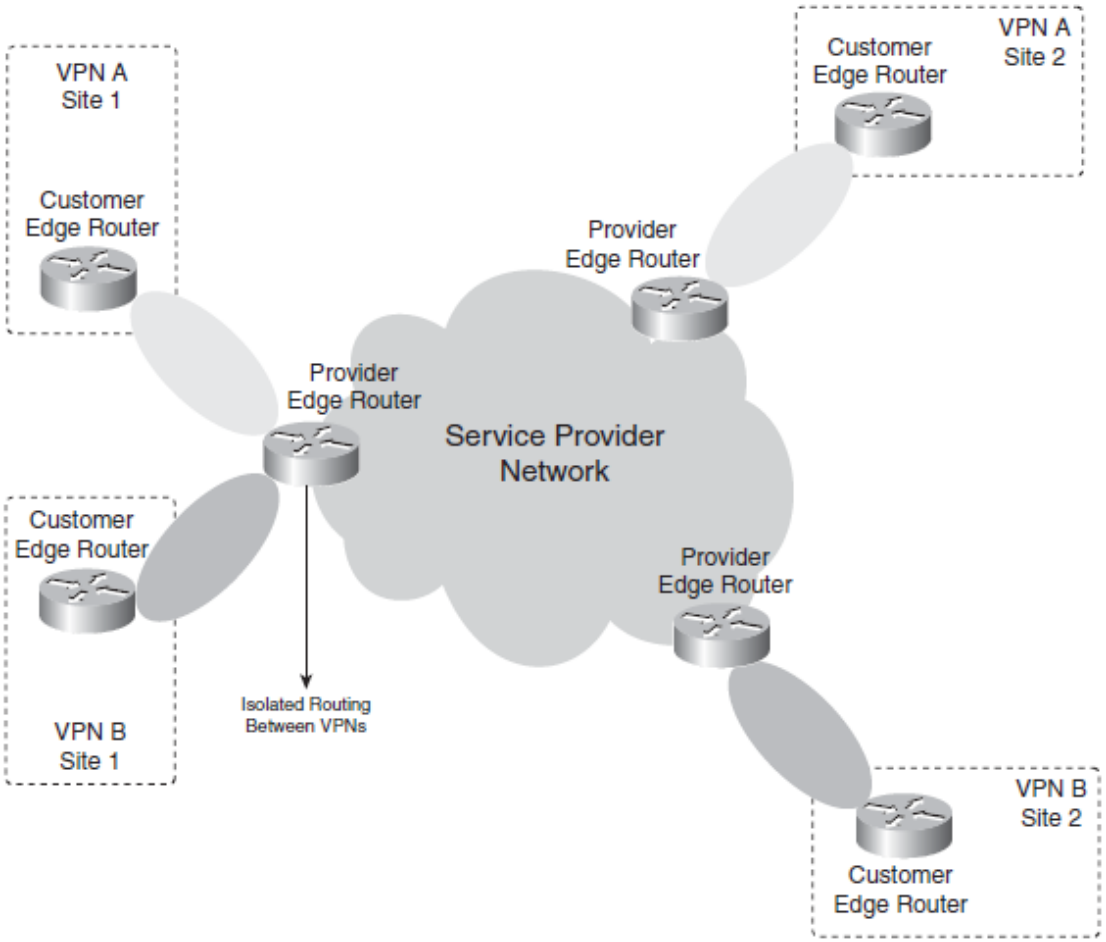


Εικόνα 3. Overlay VPN μοντέλο μέσω GRE τούνελ

- Peer-to-peer VPN μοντέλο

Στο peer-to-peer VPN μοντέλο, οι δρομολογητές του παρόχου μεταφέρουν δεδομένα στο δίκτυο, αλλά συμμετέχουν και στη δρομολόγηση στο δίκτυο του πελάτη. Το αποτέλεσμα είναι η ύπαρξη ενός πρωτοκόλλου δρομολόγησης γειτνίασης μεταξύ των δρομολογητών του πελάτη και του παρόχου.

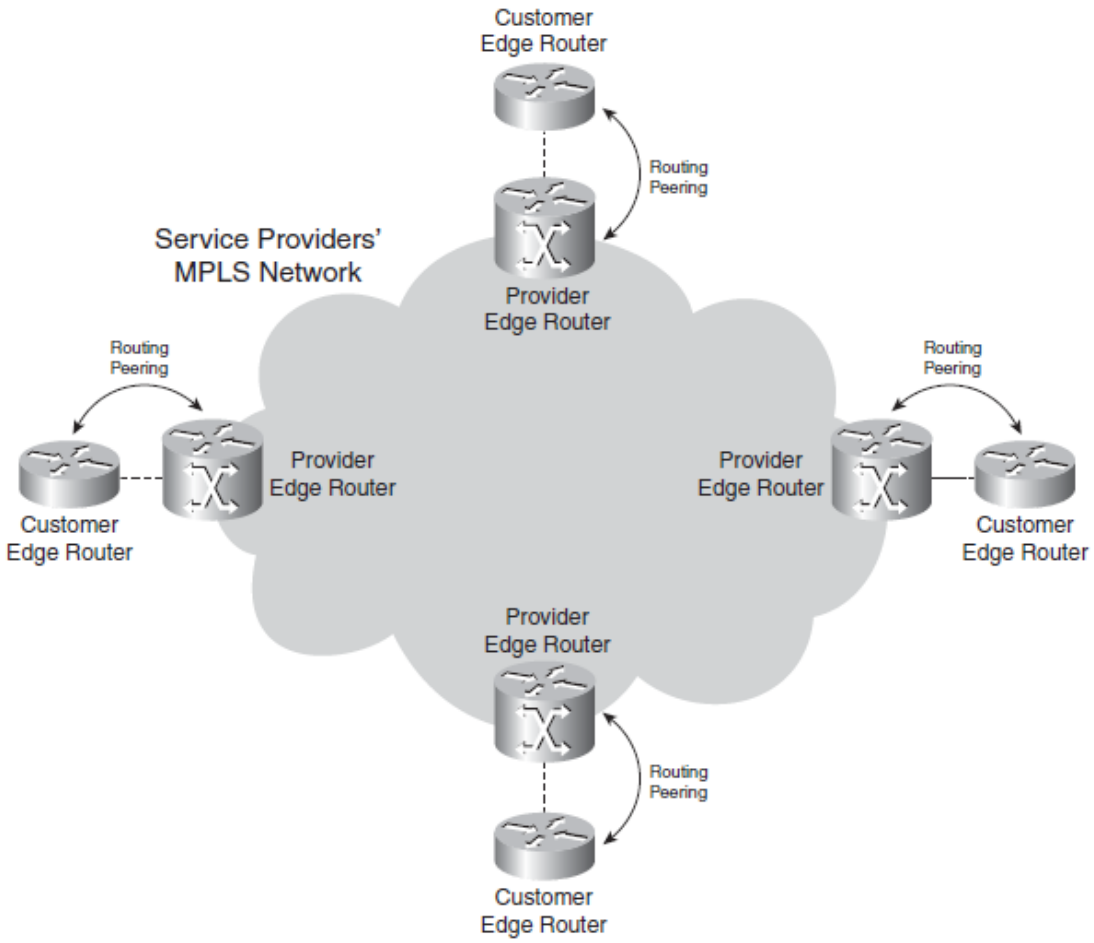
Πριν το MPLS, το peer-to-peer μοντέλο μπορούσε να επιτευχθεί μέσω της ανταλλαγής IP πληροφοριών δρομολόγησης μεταξύ των δρομολογητών του πελάτη και του παρόχου. Η ιδιωτικότητα, απαραίτητο συστατικό στα VPNs, γινόταν εφικτή είτε μέσω λιστών πρόσβασης (access lists – ACLs), είτε μέσω φίλτρων που επέτρεπαν ή σταματούσαν τη διαφήμιση διαδρομών στους δρομολογητές του πελάτη.



Εικόνα 4. Peer-to-peer VPN μοντέλο

Επίσης πριν την έλευση του MPLS, το overlay μοντέλο υλοποιούταν πιο συχνά από το peer-to-peer μοντέλο, καθώς το τελευταίο απαιτούσε πολλές αλλαγές σε περίπτωση επέκτασης του δικτύου του πελάτη. Το MPLS VPN έκανε την υλοποίηση του peer-to-peer μοντέλου πολύ πιο εύκολη. Η επέκταση του δικτύου ρυθμίζεται ευκολότερα, απαιτώντας λιγότερο χρόνο και προσπάθεια. Ένας δρομολογητής του πελάτη, αναφέρεται ως άκρο πελάτη (Customer Edge – CE), ανταλλάσσει πληροφορίες στο επίπεδο του IP με τουλάχιστον ένα δρομολογητή του παρόχου, που καλείται άκρο παρόχου (Provider Edge – PE).

Η ιδιωτικότητα στα MPLS VPN δίκτυα επιτυγχάνεται χρησιμοποιώντας την έννοια της εικονικής δρομολόγησης/προώθησης (VRF) και το γεγονός ότι τα δεδομένα προωθιύνται ως πακέτα με ετικέτες. Το VRF εξασφαλίζει ότι οι πληροφορίες δρομολόγησης από διαφορετικούς πελάτες διαχωρίζονται μεταξύ τους, και το MPLS στον κορμό του δικτύου εξασφαλίζει την προώθηση των πακέτων βάσει των πληροφοριών της ετικέτας και όχι της IP κεφαλίδας.



Εικόνα 5. Peer-to-peer μοντέλο εφαρμοσμένο στο MPLS VPN

Πλέον, η επέκταση του δικτύου του πελάτη σημαίνει απλά ανταλλαγή πληροφοριών μεταξύ του PE δρομολογητή και του νέου CE δρομολογητή. Δεν χρειάζεται να δημιουργηθούν επιπλέον εικονικά κυκλώματα, όπως στο overlay μοντέλο, ούτε να ορισθούν ACLs ή άλλα φίλτρα, όπως στο peer-to-peer μοντέλο. Αυτό αποτελεί ένα μεγάλο πλεονέκτημα για τον πάροχο, όπως και το γεγονός ότι πλέον χρειάζεται να φροντίζει μόνο για τη σύνδεση μεταξύ των PE και CE δρομολογητών και όχι όλων των εικονικών κυκλωμάτων, όπως συνέβαινε στο overlay μοντέλο.

Ωστόσο, υπάρχουν και μειονεκτήματα του peer-to-peer μοντέλου συγκριτικά με το overlay μοντέλο. Αρχικά, ο πελάτης δεν ελέγχει αποκλειστικά το δίκτυό του από άκρο σε άκρο, αλλά πρέπει να μοιραστεί την ευθύνη της δρομολόγησης με τον πάροχο. Επίσης, οι PE δρομολογητές επιβαρύνονται, καθώς πρέπει να μεταφέρουν τις διαδρομές πολλών πελατών και ταυτόχρονα να κρατούν ενημερωμένους τους πίνακες δρομολόγησης. [1]

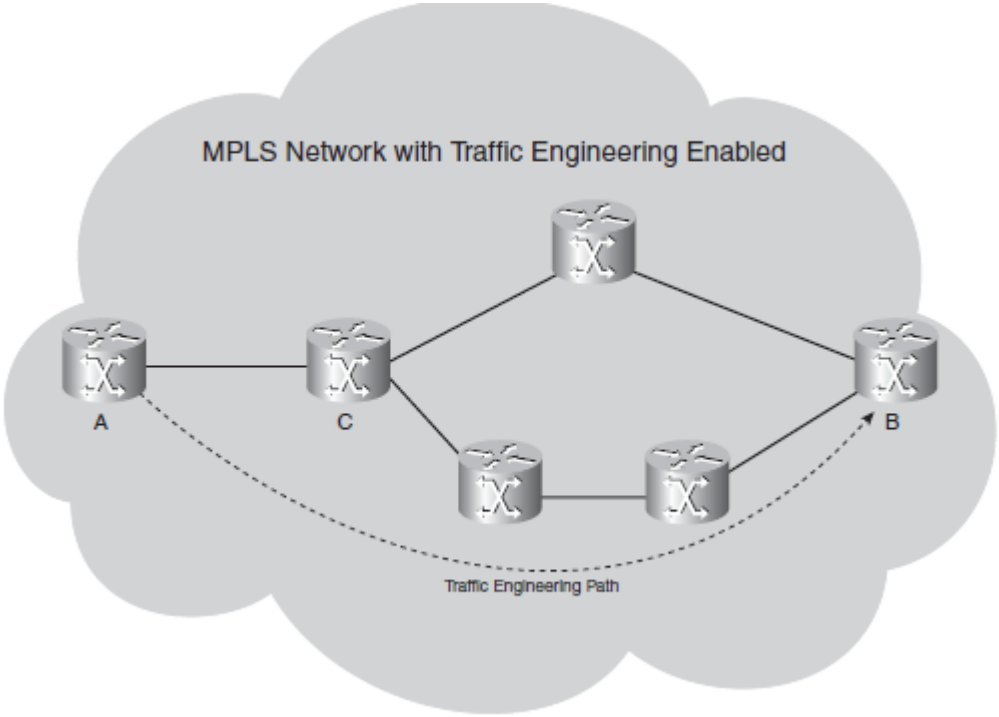
1.3.6 Βέλτιστη ροή της κίνησης

Επειδή οι ATM ή Frame Relay μεταγωγείς είναι L2 συσκευές, οι δρομολογητές διασυνδέονται μέσω αυτών με τη βοήθεια εικονικών κυκλωμάτων. Για να στείλει ένας δρομολογητής δεδομένα απευθείας σε κάποιο άλλο δρομολογητή στα άκρα του δικτύου, πρέπει να δημιουργηθεί ένα εικονικό κύκλωμα μεταξύ τους, διαδικασία που είναι εξαιρετικά χρονοβόρα και κοστοβόρα καθώς γίνεται χειροκίνητα. Αντίθετα, χρησιμοποιώντας το MPLS VPN όπως είδαμε και προηγουμένως, η ροή των

δεδομένων πηγαίνει απευθείας – άρα βέλτιστα – σε όλα τα άκρα του δικτύου του πελάτη. [1]

1.3.7 Traffic Engineering

Η βασική ιδέα πίσω από το traffic engineering είναι η βέλτιστη χρήση της δικτυακής υποδομής, συμπεριλαμβανομένων συνδέσεων που υποχρησιμοποιούνται γιατί δε βρίσκονται στο προτιμώμενο μονοπάτι. Αυτό σημαίνει πως το traffic engineering πρέπει να μπορεί να οδηγήσει την κίνηση των δεδομένων σε διαφορετικά μονοπάτια από το προτιμώμενο, το οποίο είναι το μονοπάτι ελαχίστου κόστους στην IP δρομολόγηση. Υλοποιώντας το traffic engineering στο MPLS δίκτυο, μια ροή δεδομένων που προορίζεται για ένα συγκεκριμένο πρόθεμα ή της έχει εφαρμοσθεί συγκεκριμένο Quality of Service (QoS), μπορεί να πάει από το σημείο A στο σημείο B μέσω ενός μονοπατιού διαφορετικού από το μονοπάτι ελάχιστου κόστους. Το αποτέλεσμα είναι ότι η κίνηση κατανέμεται πιο ομοιόμορφα στις διαθέσιμες συνδέσεις και χρησιμοποιούνται περισσότερες από τις υποχρησιμοποιούμενες συνδέσεις του δικτύου.



Εικόνα 6. Παράδειγμα Traffic Engineering

Ο χειριστής του MPLS δικτύου μπορεί να οδηγήσει τη ροή από το A στο B μέσω του κάτω μονοπατιού, που δεν είναι το συντομότερο (4 αναπηδήσεις έναντι 3 του πάνω μονοπατιού). Έτσι, μπορεί να οδηγηθεί η ροή σε συνδέσεις οι οποίες σε διαφορετική περίπτωση δεν θα χρησιμοποιούνταν πολύ.

Επιπλέον, αν πρόκειται για IP δίκτυο, ο δρομολογητής C δεν μπορεί να στείλει τη ροή από το κάτω μπνοπάτι απλά ρυθμίζοντας κάτι στον δρομολογητή A. Η απόφαση να στείλει τη ροή από το πάνω ή το κάτω μονοπάτι ανήκει αποκλειστικά στο δρομολογητή C. Μέσω του traffic engineering στο MPLS, ο δρομολογητής A μπορεί να επιλέξει τη διαδρομή που θα ακολουθήσουν τα δεδομένα μέχρι τον προορισμό B. Αυτό συμβαίνει λόγω του μηχανισμού προώθησης ετικετών. Ο αρχικός δρομολογητής ενός μονοπατιού όπου έχει επιβληθεί traffic engineering (στην περίπτωσή μας ο δρομολογητής A), είναι ο δρομολογητής που καθορίζει ολόκληρο το μονοπάτι το οποίο θα ακολουθήσουν τα

δεδομένα διαμέσου του MPLS δικτύου. Η ετικέτα που επισυνάπτει σε ένα πακέτο ο Α καθορίζει τη διαδρομή του και κανένας ενδιάμεσος δρομολογητής δεν μπορεί να προωθήσει το πακέτο από κάποιο άλλο μονοπάτι.

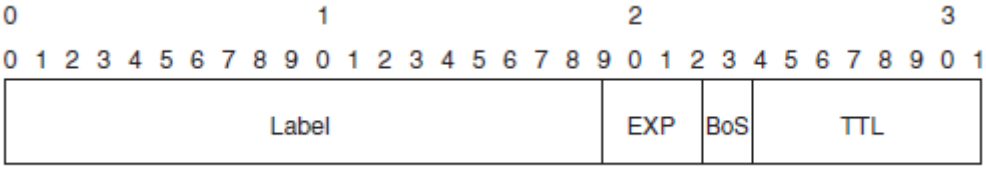
Ένα επιπλέον πλεονέκτημα του traffic engineering είναι η δυνατότητα γρήγορης αναδρομολόγησης, η οποία επιτρέπει την αναδρομολόγηση μίας ροής με ετικέτα γύρω από μία σύνδεση ή ένα δρομολογητή που δε λειτουργεί. Η αναδρομολόγηση γίνεται μέσα σε λιγότερο από 50ms, που θεωρείται γρήγορο ακόμα και για τα σημερινά δεδομένα. [1]

2. Αρχιτεκτονική του MPLS

2.1 Δομή MPLS ετικέτας

Το πρωτόκολλο MPLS αποτελεί μία βελτιωμένη μέθοδο προώθησης πακέτων σε ένα δίκτυο, κάνοντας χρήση των πληροφοριών που περιέχονται στις ετικέτες που επισυνάπτονται στα IP πακέτα. Οι ετικέτες μπαίνουν μεταξύ των επικεφαλίδων των επιπέδων 2 και 3 για τεχνολογίες που βασίζονται σε L2 πλαίσια ή περιέχονται στα πεδία VPI (Virtual Path Identifier) και VCI (Virtual Channel Identifier) για τεχνολογίες όπως το ATM. Πρωταρχικός στόχος του MPLS είναι η δημιουργία ενός ευέλικτου δικτυακού ιστού που θα παρέχει αυξημένη απόδοση και σταθερότητα. [2]

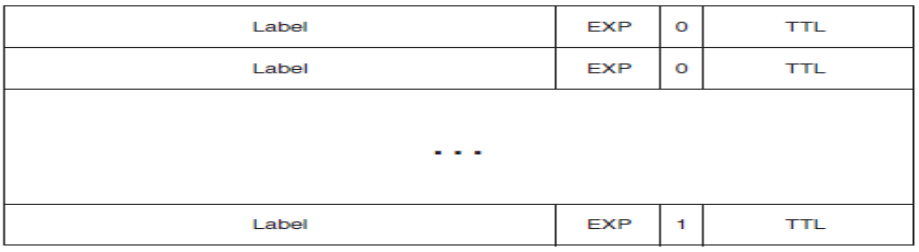
Μία MPLS ετικέτα είναι ένα πεδίο 32 bits με συγκεκριμένη δομή. Τα πρώτα 20 bits αποτελούν την τιμή της ετικέτας, η οποία κυμαίνεται από 0 έως 2^{20}-1, δηλαδή 1.048.575. Ωστόσο, οι 16 πρώτες τιμές εξαιρούνται από την κανονική χρήση, ενώ τα bits 20 ως 22 είναι τα πειραματικά (EXP) bits. Τα bits αυτά χρησιμοποιούνται αποκλειστικά για QoS.



Εικόνα 7. Δομή MPLS ετικέτας

Το bit 23 καλείται Bottom of Stack (BoS) bit. Εϊναι 0, εκτός και αν η ετικέτα βρίσκεται στη βάση της στοίβας, οπότε και παίρνει τιμή 1. Τα bits 24 έως 31 χρησιμοποιούνται στο πεδίο Time To Live (TTL). Το πεδίο TTL έχει ίδια λειτουργικότητα με το αντίστοιχο της IP κεφαλίδας. Μειώνεται κατά 1 σε κάθε αναπήδηση, καθώς κύρια λειτουργία του είναι η αποφυγή βρόχου δρομολόγησης (routing loop). Όταν φτάσει στο 0, το πακέτο απορρίπτεται.

Η στοίβα είναι η συλλογή των ετικετών που βρίσκονται πάνω στο πακέτο. Η πρώτη ετικέτα στη στοίβα αποκαλείται άνω ετικέτα και η τελευταία, κάτω ετικέτα. Μπορεί να περιέχει μία ή περισσότερες ετικέτες, κι ενώ δεν υπάρχει περιορισμός στον αριθμό των ετικετών, στην πράξη σπάνια υπάρχουν πάνω από 4 ετικέτες σε μία στοίβα. Όπως φαίνεται στην παρακάτω εικόνα, το BoS bit είναι 0 για όλες τις ετικέτες πλην της κάτω ετικέτας, για την οποία η τιμή του BoS bit είναι1.

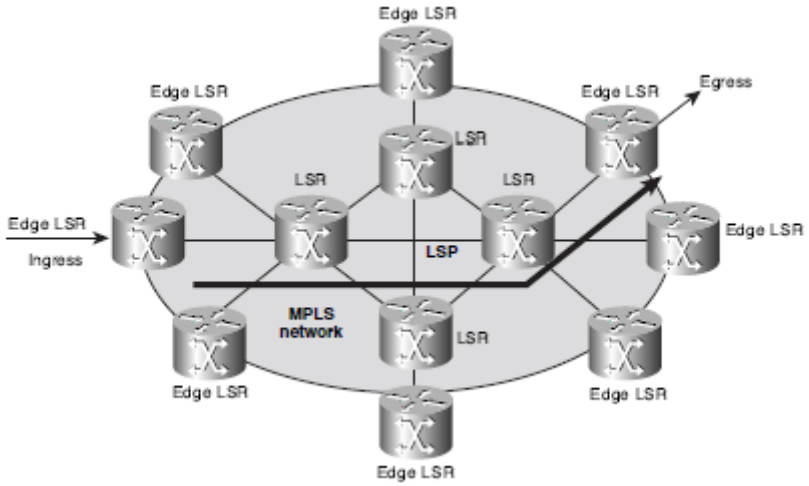


Εικόνα 8. Στοίβα ετικετών

Υπάρχουν MPLS εφαρμογές που απαιτούν περισσότερες από μία ετικέτες για να προωθήσουν τα πακέτα. Παραδείγματα τέτοιων εφαρμογών είναι το MPLS VPN και το AToM. Και οι δύο αυτές εφαρμογές βάζουν 2 ετικέτες στη στοίβα. [1]

2.2 Στοιχεία του MPLS

Σε ένα MPLS δίκτυο, η ετικέτα ανατίθεται στα εισερχόμενα πακέτα από έναν ingress LSR. Τα πακέτα προωθούνται σε ένα LSP όπου κάθε LSR παίρνει αποφάσεις προώθησης βασιζόμενος αποκλειστικά στα περιεχόμενα της ετικέτας. Σε κάθε αναπήδηση, ο LSR βγάζει την υπάρχουσα ετικέτα και βάζει μία καινούρια, που αναφέρει πως πρέπει να δρομολογηθεί το πακέτο στην επόμενη αναπήδηση. [2]



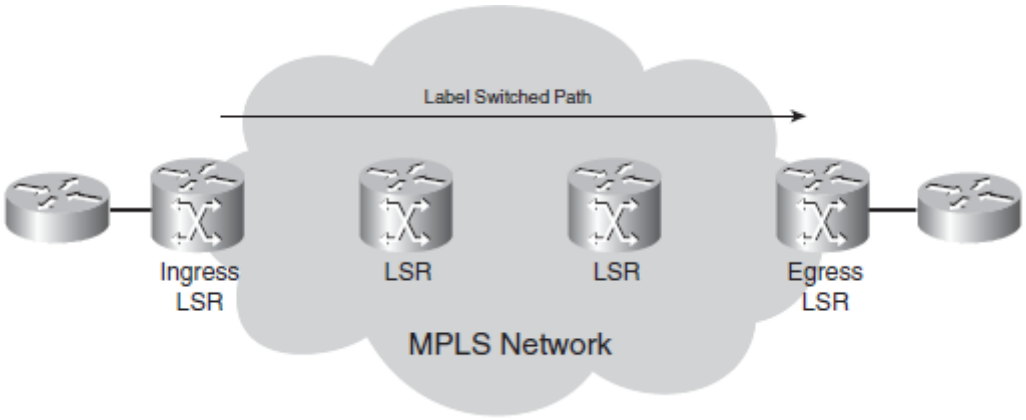
Εικόνα 9. Τοπολογία MPLS δικτύου

Ένας Label-Switched Router (LSR) είναι ένας δρομολογητής που υποστηρίζει MPLS. Υπάρχουν 3 είδη LSRs σε ένα MPLS δίκτυο:

- Ingress LSR – λαμβάνει ένα πακέτο χωρίς ετικέτα που έρχεται έξω από το MPLS δίκτυο, του αναθέτει μία ετικέτα και το προωθεί στη ζεύξη, εντός του MPLS δικτύου.
- Egress LSR – λαμβάνει πακέτα με ετικέτες από το MPLS δίκτυο, τις αφαιρεί και τις προωθεί στη ζεύξη, έξω από το MPLS δίκτυο. Τόσο ο ingress, όσο και ο egress LSR βρίσκονται στα άκρα του δικτύου.
- Ενδιάμεσος LSR – λαμβάνει ένα εισερχόμενο πακέτο με ετικέτα, το μετάγει και το στέλνει στην κατάλληλη ζεύξη.

Ένας LSR μπορεί να επιτελέσει 3 λειτουργίες: εξαγωγή, εισαγωγή ή εναλλαγή ετικέτας. Πριν προωθήσει ένα πακέτο, ο LSR πρέπει είτε να εξάγει είτε να εισάγει μία ή περισσότερες ετικέτες. Αν το πακέτο έχει ήδη ετικέτα, ο LSR εισάγει μία ετικέτα στη στοίβα. Αν δεν έχει ετικέτα, δημιουργεί πρώτα τη στοίβα και στη συνέχεια εισάγει την ετικέτα. Επίσης, αν παραληφθεί ένα πακέτο με ετικέτα, η ετικέτα στην κορυφή της στοίβας εναλλάσεται με μία καινούρια και στη συνέχεια το πακέτο προωθείται.

Ένα Label-Switched Path (LSP) είναι μία ακολουθία από LSR στους οποίους γίνεται η μεταγωγή του πακέτου στο δίκτυο. Ουσιαστικά, το LSP είναι το μονοπάτι το οποίο ακολουθούν τα πακέτα μέσα στο MPLS δίκτυο. Ο πρώτος LSR ενός LSP θεωρείται ως ingress LSR για το συγκεκριμένο LSP, ενώ ο τελευταίος ως egress LSR. Οι υπόλοιποι είναι ενδιάμεσοι LSRs.



Εικόνα 10. Label-Switched Path

Ένα LSP είναι μονής κατεύθυνσης. Συνεπώς, μία ροή πακέτων αντίθετης κατεύθυνσης από αυτή που επισημαίνεται με το βέλος στην παραπάνω εικόνα αποτελεί διαφορετικό LSP. [1]

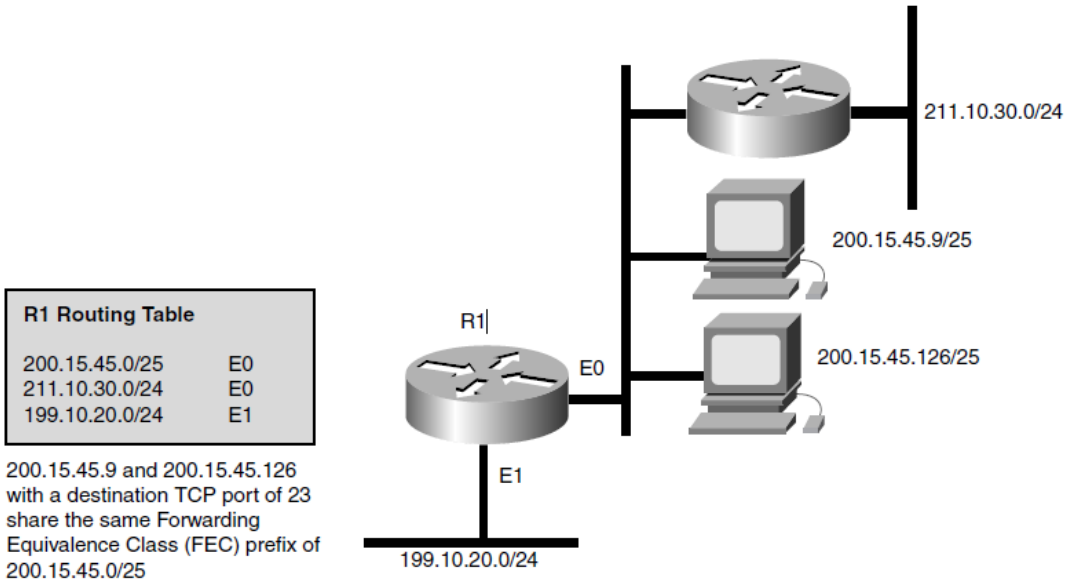
2.3 Forwarding Equivalence Class

Ως Forwarding Equivalence Class (FEC) ορίζεται μία ομάδα ή ροή πακέτων που προωθούνται στο ίδιο μονοπάτι και αντιμετωπίζονται με τον ίδιο τρόπο όσον αφορά την προώθηση. 'Ολα τα πακέτα που ανήκουν στο ίδιο FEC έχουν την ίδια ετικέτα. Ωστόσο, δεν είναι απαραίτητο όλα τα πακέτα που έχουν ίδια ετικέτα να ανήκουν στο ίδιο FEC, καθώς μπορεί να διαφέρουν οι τιμές EXP, άρα να υπόκεινται σε διαφορετική κατηγορία QoS.

Ο δρομολογητής που αποφασίζει ποια πακέτα ανήκουν σε ποιο FEC είναι ο ingress LSR. Αυτό είναι λογικό, καθώς ο ingress LSR είναι υπεύθυνος για την ταξινόμηση και ανάθεση ετικετών στα πακέτα. Ακολουθούν ορισμένα παραδείγματα FECs:

- Πακέτα με L3 IP διευθύνσεις προορισμού που ταιριάζουν με ένα συγκεκριμένο πρόθεμα.
- Ένα σύνολο unicast πακέτων των οποίων οι διευθύνσεις προορισμού ταιριάζουν με ένα συγκεκριμένο πρόθεμα IP διευθύνσεων και έχουν είτε ίδια Type of Service (ToS) bits είτε ίδια TCP θύρα προορισμού.
- Multicast πακέτα που ανήκουν σε συγκεκριμένη ομάδα.
- Ένα σύνολο multicast πακέτων με ίδιες L3 διευθύνσεις πηγής και προορισμού.
- L2 πλαίσια σε ένα MPLS δίκτυο που λαμβάνονται σε ένα εικονικό κύκλωμα ή μία διεπαφή του ingress LSR και μεταδίδονται από ένα εικονικό κύκλωμα ή μία διεπαφή του egress LSR.
- Πακέτα με L3 IP διευθύνσεις προορισμού που ανήκουν σε ένα σύνολο BGP προθεμάτων.

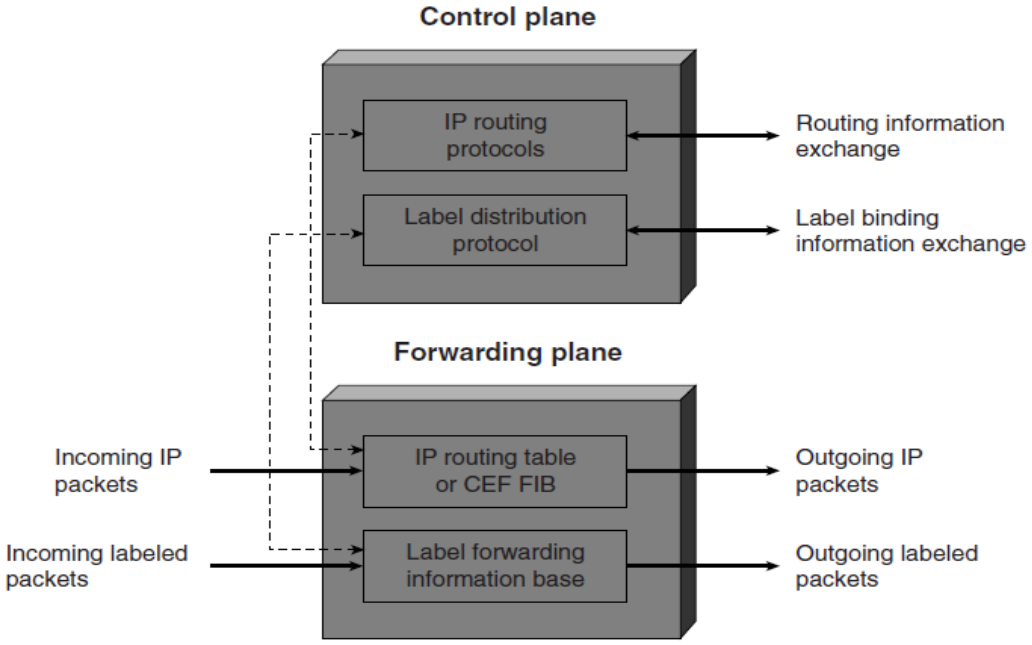
Για παράδειγμα, οι διευθύνσεις 200.15.45.9 και 200.15.45.126 ανήκουν στο ίδιο FEC με πρόθεμα διεύθυνσης 200.15.45.0/25 και TCP θύρα προορισμού 23. [1,2]



Εικόνα 11. Παράδειγμα FEC

2.4 Αρχιτεκτονική Κόμβου

Οι MPLS κόμβοι διαθέτουν 2 επίπεδα από πλευράς αρχιτεκτονικής: το MPLS επίπεδο προώθησης και το MPLS επίπεδο ελέγχου. Πέρα από μεταγωγή πακέτων που φέρουν ετικέτα, οι MPLS κόμβοι μπορούν να πραγματοποιήσουν L3 δρομολόγηση ή L2 μεταγωγή.

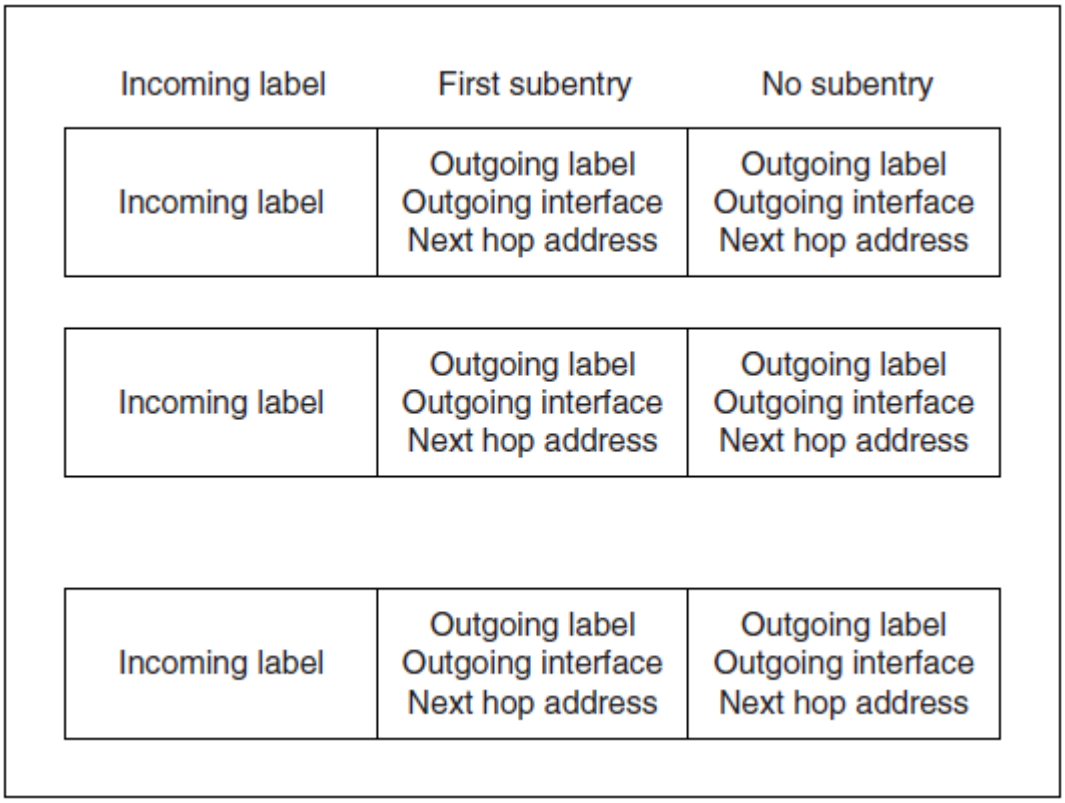


Εικόνα 12. Αρχιτεκτονική MPLS κόμβου

Το MPLS επίπεδο προώθησης είναι υπεύθυνο για την προώθηση των πακέτων βάσει των τιμών που περιέχονται στις ετικέτες τους. Χρησιμοποιεί μία Label Forwarding Information Base (LFIB) για την προώθηση των πακέτων. Η LFIB διατηρείται από έναν MPLS κόμβο και αποτελείται από μία ακολουθία εγγραφών. Κάθε εγγραφή αποτελείται από μία ετικέτα εισόδου και μία ή περισσότερες υποεγγραφές. Η LFIB δεικτοδοτείται βάσει της τιμής της εισερχόμενης ετικέτας.

Κάθε υποεγγραφή αποτελείται από μία ετικέτα εξόδου, μία διεπαφή εξόδου και τη next-hop διεύθυνση. Οι υποεγγραφές που ανήκουν σε μία συγκεκριμένη εγγραφή μπορεί να

έχουν ίδιες ή διαφορετικές ετικέτες εξόδου. Η προώθηση multicast πακέτων απαιτεί υποεγγραφές με πολλαπλές ετικέτες εξόδου, καθώς ένα εισερχόμενο πακέτο που φθάνει σε μία διεπαφή πρέπει να σταλεί σε πολλαπλές διεπαφές εξόδου.



Εικόνα 13. Δομή LFIB

Κάθε κόμβος διατηρεί 2 πίνακες σχετικούς με την MPLS προώθηση: τη Label Information Base (LIB) και την LFIB. Η LIB περιέχει όλες τις ετικέτες που έχει αναθέσει ο MPLS κόμβος και όλες τις αντιστοιχίσεις αυτών των ετικετών σε ετικέτες που έλαβε ο κόμβος από τους MPLS γείτονές του. Για την προώθηση των πακέτων, η LFIB χρησιμοποιεί ένα υποσύνολο των ετικετών που περιέχονται στη LIB.

Κόμβοι που διατηρούν μία LFIB εξάγουν τις τιμές των ετικετών από το αντίστοιχο πεδίο των εισερχόμενων πακέτων και τις χρησιμοποιούν ως δείκτη στην LFIB. Αφού βρεθεί η αντίστοιχη ετικέτα εισόδου στην LFIB, ο κόμβος αντικαθιστά την ετικέτα του πακέτου με την ετικέτα εξόδου από την υποεγγραφή και στέλνει το πακέτο από την καθορισμένη διεπαφή εξόδου προς τη next-hop διεύθυνση που ορίζεται από την υποεγγραφή. Αν ο κόμβος διατηρεί πολλαπλές LFIBs για κάθε μία από τις διεπαφές του, χρησιμοποιεί τη φυσική διεπαφήστην οποία έφθασε το πακέτο για να επιλέξει μία συγκεκριμένη LFIB, η οποία χρησιμοποιείται για την προώθηση του πακέτου.

Το επίπεδο ελέγχου είναι υπεύθυνο για τη συμπλήρωση και διατήρηση της LFIB. Όλοι οι κόμβοι πρέπει να τρέχουν ένα IP πρωτόκολλο δρομολόγησης για να ανταλλάξουν IP πληροφορίες δρομολόγησης με τους άλλους κόμβους του MPLS δικτύου. Επιλέγονται πρωτόκολλα όπως το OSPF και το IS-IS, καθώς δίνουν σε κάθε κόμβο μία άποψη ολόκληρου του δικτύου. Οι ετικέτες που ανταλλάσσονται με γειτονικούς κόμβους χρησιμοποιούνται για να δημιουργηθεί η LFIB. Το πρότυπο προώθησης που χρησιμοποιεί το MPLS μπορεί να συνδυαστεί με μία σειρά διαφορετικών μονάδων ελέγχου. Κάθε μονάδα ελέγχου είναι υπεύθυνη για την ανάθεση και διανομή ενός συνόλου ετικετών, καθώς και τη διατήρηση άλλων σχετικών πληροφοριών ελέγχου. Οι MPLS μονάδες ελέγχου περιλαμβάνουν:

- Μονάδα unicast δρομολόγησης: Συνθέτει τον FEC πίνακα χρησιμοποιώντας πρωτόκολλα δρομολόγησης όπως το OSPF, IS-IS κ.ο.κ.
- Μονάδα multicast δρομολόγησης: Συνθέτει τον FEC πίνακα χρησιμοποιώντας multicast πρωτόκολλα δρομολόγησης όπως το Protocol-Independent Multicast (PIM).
- Μονάδα Traffic Engineering: Επιτρέπει να δημιοργηθούν ρητά καθορισμένα LSPs για σκοπούς traffic engineering.
- Μονάδα VPN: Χρησιμοποιεί ανά-VPN πίνακες δρομολόγησης για τους FEC πίνακες, οι οποίοι δημιουργούνται με χρήση πρωτοκόλλων δρομολόγησης που τρέχουν μεταξύ των δρομολογητών του πελάτη και τους MPLS κόμβους του παρόχου.
- Μονάδα QoS: Συνθέτει τον FEC πίνακα χρησιμοποιώντας πρωτόκολλα δρομολόγησης όπως το OSPF, IS-IS κ.ο.κ. [2]

2.5 Label Distribution Protocol

2.5.1 Επισκόπηση LDP

Η αρχιτεκτονική του MPLS ορίζει το Label Distribution Protocol (LDP) ως πρωτόκολλο διανομής ετικετών. Είναι ένα σύνολο διαδικασιών και μηνυμάτων μέσω του οποίου οι LSRs εγκαθιδρύουν LSPs διαμέσου του δικτύου αντιστοιχίζοντας L3 πληροφορίες δρομολόγησης απευθείας σε L2 LSPs. Το LDP συνδέει ένα FEC με κάθε LSP που δημιουργεί. Το FEC είναι αυτό που προσδιορίζει ποια πακέτα αντιστοιχίζονται με το συγκεκριμένο LSP.

Δύο LSRs που χρησιμοποιούν το LDP για να ανταλλάξουν πληροφορίες αντιστοίχισης ετικέτας/FEC ονομάζονται LDP ομότιμοι και μεταξύ τους δημιουργείται μία LDP σύνοδος. Μία LDP σύνοδος είναι αμφίδρομη και επιτρέπει σε κάθε ομότιμο να μάθει τις αντιστοιχίσεις ετικετών του άλλου.

Υπάρχουν 4 κατηγορίες LDP μηνυμάτων που επιτρέπουν σε 2 ομότιμους να επικοινωνήσουν:

- DISCOVERY: Στέλνονται multicast HELLO μηνύματα μέσω UDP ώστε να μάθει ένας LSR τους άλλους LSRs με τους οποίους το LDP έχει απευθείας σύνδεση και στη συνέχεια δημιουργείται η LDP σύνοδος. Ο LSR σε οποιοδήποτε από τα 2 άκρα μπορεί να διαφημίσει ή να ζητήσει τις αντιστοιχίσεις του ομότιμού του.
- SESSION: Στέλνονται μέσω TCP και αρχικοποιούν τη σύνοδο χρησιμοποιώντας το μήνυμα INITIALIZATION στην αρχή της διαπραγμάτευσης της συνόδου. Περιέχει πληροφορίες σχετικά με τη διάρκεια των keepalive, αλλά και το εύρος τιμών ετικέτας που μπορεί να χρησιμοποιηθεί μεταξύ 2 LSRs. Τα keepalive μηνύματα στέλνονται περιοδικά μεταξύ των ομότιμων, ενώ αν δεν παραληφθούν εντός συγκεκριμένου χρονικού ορίου το αποτέλεσμα είναι ο τερματισμός της συνόδου.
- ADVERTISEMENT: Διαφημίζουν τις αντιστοιχίσεις μεταξύ των FECs και των ετικετών. Μηνύματα LABEL WITHDRAWAL χρησιμοποιούνται για να αντιστρέψουν τη διαδικασία αντιστοίχισης, ενώ μηνύματα LABEL RELEASE χρησιμοποιούνται από LSRs που έχουν λάβει τις πληροφορίες αντιστοίχισης και θέλουν να αποδεσμεύσουν την ετικέτα γιατί δεν τη χρειάζονται πλέον.
- NOTIFICATION: Παρέχουν συμβουλευτικές πληροφορίες, καθώς και πληροφορίες σφαλμάτων σήματος μεταξύ ομότιμων. [3]

2.5.2 Label Space και LDP Αναγνωριστικά

Η έννοια των label spaces είναι χρήσιμη στην ανάθεση και διανομή των ετικετών. Υπάρχουν 2 είδη:

- Ανά διεπαφή: Για διεπαφές που χρησιμοποιούν δικούς τους πόρους για ετικέτες. Για παράδειγμα, μία ATM διεπαφή χρησιμοποιεί VCIs (Virtual Channel Identifiers) ως ετικέτες ή μία Frame Relay διεπαφή που χρησιμοποιεί DLCIs (Data Link Connection Identifiers) ως ετικέτες. Η χρήση ενός label space ανά διεπαφή έχει νόημα μόνο όταν οι LDP ομότιμοι είναι άμεσα συνδεδεμένοι μέσω μίας διεπαφής και η ετικέτα πρόκειται να χρησιμοποιηθεί για αποστολή πακέτων μόνο μέσω της συγκεκριμένης διεπαφής.
- Ανά πλατφόρμα: Για διεπαφές που μπορούν να μοιραστούν τις ίδιες ετικέτες.

Ένα LDP αναγνωριστικό αποτελείται από 6 οκτάδες και χρησιμοποιείται για την αναγνώριση του label space ενός LSR. Οι 4 πρώτες οκτάδες αφορούν το LSR και πρέπει να είναι μία μοναδική τιμή (π.χ. ένα 32-bit router ID). Οι 2 τελευταίες οκτάδες αφορούν ένα συγκεκριμένο label space εντός του LSR. Για label spaces ανά πλατφόρμα και οι δύο οκτάδες έχουν τιμή μηδέν. [3]

2.5.3 Λειτουργία LDP

Είναι αναγκαίο να προσδιοριστεί ακριβώς ποια πακέτα αντιστοιχούν σε κάθε LSP. Αυτό γίνεται αναθέτοντας ένα FEC στοιχείο σε κάθε LSP. Κάθε FEC στοιχείο προσδιορίζει ένα σύνολο πακέτων που θα αντιστοιχιστεί με κάποιο LSP. Ορίζεται ένας βασικός τύπος στοιχείου, το FEC στοιχείο “Πρόθεμα Διεύθυνσης”. Το στοιχείο αυτό είναι ένα πρόθεμα διεύθυνσης οποιουδήποτε μήκους, από 0 έως και μία πλήρη διεύθυνση (π.χ. 192.168.0.0/16).

Μία διεύθυνση ταιριάζει με ένα συγκεκριμένο Πρόθεμα Διεύθυνσης αν και μόνο αν η διεύθυνση ξεκινάει με αυτό το πρόθεμα. Επίσης, ένα πακέτο αντιστοιχεί σε ένα LSP αν και μόνο αν το FEC στοιχείο Πρόθεμα Διεύθυνσης του LSP ταιριάζει με τη διεύθυνση προορισμού του πακέτου. Κάθε Πρόθεμα Διεύθυνσης που αντιστοιχεί σε ένα πακέτο αναφέρεται ως “πρόθεμα αντιστοίχισης”.

Η διαδικασία αντιστοίχισης ενός πακέτου σε ένα LSP ακολουθεί τους κάτωθι κανόνες. Κάθε κανόνας εφαρμόζεται με τη σειρά μέχρι να αντιστοιχιστεί το πακέτο σε ένα LSP.

- Αν το πακέτο ταιριάζει σε ένα μόνο LSP, τότε αντιστοιχίζεται στο συγκεκριμένο LSP.
- Αν το πακέτο ταιριάζει σε πολλαπλά LSPs, αντιστοιχίζεται στο LSP με το οποίο έχει το μεγαλύτερο πρόθεμα αντιστοίχισης. Αν δεν υπάρχει ένα LSP με μεγαλύτερο πρόθεμα αντιστοίχισης, το πακέτο θα αντιστοιχιστεί σε ένα LSP από το σύνολο των LSPs με το μεγαλύτερο πρόθεμα αντιστοίχισης από τα υπόλοιπα.
- Αν είναι γνωστό πως το πακέτο πρέπει να διασχίσει ένα συγκεκριμένο egress LSR και υπάρχει ένα LSP με Πρόθεμα Διεύθυνσης που να ταυτίζεται με τη διεύθυνση του δρομολογητή, τότε το πακέτο αντιστοιχίζεται σε αυτό το LSP. [3]

2.5.4 LDP Ανακάλυψη

Η LDP Ανακάλυψη είναι ένας μηχανισμός που επιτρέπει σε ένα LSR να ανακαλύψει πιθανούς ομότιμους, καθιστώντας αχρείαστο το ρητό καθορισμό τους. Υπάρχουν 2 είδη

μηχανισμών: ένας βασικός μηχανισμός ανακάλυψης που χρησιμοποιείται για να βρει τους γείτονες του LSR που είναι άμεσα συνδεδεμένοι στο επίπεδο ζεύξης, και ένας εκτεταμένος μηχανισμός για να εντοπίζει LSRs που δεν είναι άμεσα συνδεδεμένοι στο επίπεδο ζεύξης.

Για να μπορέσει να συμμετάσχει ένας LSR με μία διεπαφή του στο βασικό μηχανισμό ανακάλυψης, στέλνει περιοδικά LDP Link Hellos από τη διεπαφή. Τα LDP Link Hellos στέλνονται ως UDP πακέτα με προορισμό τη multicast διεύθυνση του γκρουπ “όλοι οι δρομολογητές του subnet” και μεταφέρουν το LDP αναγνωριστικό για το label space που ο LSR σκοπεύει να χρησιμοποιήσει. Λήψη ενός LDP Link Hello σε μία διεπαφή σηματοδοτεί μία γειτνίαση με έναν πιθανό LDP ομότιμο άμεσα προσβάσιμο στο επίπεδο ζεύξης.

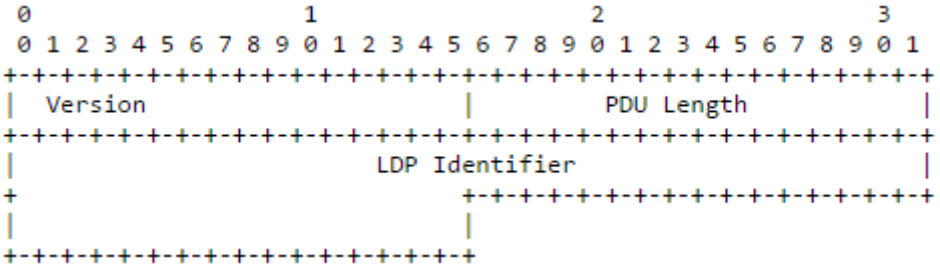
Οι LDP σύνοδοι μεταξύ έμμεσα συνδεδεμένων LSR υλοποιούνται μέσω του εκτεταμένου μηχανισμού ανακάλυψης. Ένας LSR στέλνει περιοδικά στοχευμένα LDP Hellos σε μία συγκεκριμένη διεύθυνση. Τα LDP Hellos στέλνονται ως UDP πακέτα με προορισμό τη συγκεκριμένη διεύθυνση και μεταφέρουν το LDP αναγνωριστικό για το label space που ο LSR σκοπεύει να χρησιμοποιήσει. Λήψη ενός στοχευμένου LDP Hello σηματοδοτεί μία γειτνίαση με έναν πιθανό LDP ομότιμο προσβάσιμο στο επίπεδο δικτύου.

Ο εκτεταμένος μηχανισμός διαφέρει από τον βασικό ως προς τα παρακάτω:

- Στέλνεται ένα στοχευμένο Hello πακέτο προς μία συγκεκριμένη διεύθυνση και όχι προς μία multicast διεύθυνση που περιλαμβάνει όλους τους δρομολογητές του subnet.
- Ο βασικός μηχανισμός είναι συμμετρικός, ενώ ο εκτεταμένος ασύμμετρος. Πρακτικά, αυτό σημαίνει πως όταν ένας LSR εκκινεί τον εκτεταμένο μηχανισμό για να βρεί έναν μη άμεσα συνδεδεμένο LSR, ο παραλήπτης LSR του απεσταλμένου Hello πακέτου θα αποφασίσει αν θα απαντήσει σε αυτό ή αν θα το αγνοήσει. Αν απαντήσει, θα στέλνει περιοδικά στοχευμένα Hellos προς τον LSR που εκκίνησε τον μηχανισμό. [3]

2.5.5 Προδιαγραφές LDP

Οι ανταλλαγές LDP μηνυμάτων επιτυγχάνονται στέλνοντας LDP PDUs (Protocol Data Unit) μέσω των LDP συνόδων. Κάθε LDP PDU είναι μία LDP κεφαλίδα που ακολουθείται από ένα ή περισσότερα LDP μηνύματα.



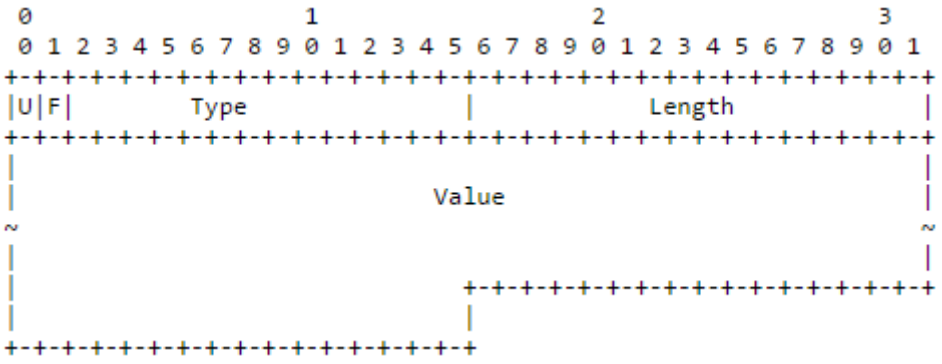
Εικόνα 14. Δομή LDP κεφαλίδας

Η LDP κεφαλίδα έχει τα εξής πεδία:

- Έκδοση (Version): Περιέχει τον αριθμό της έκδοσης του πρωτοκόλλου.
- Μήκος PDU (PDU Length): Περιέχει το συνολικό μήκος του PDU χωρίς τα πεδία Έκδοση και Μήκος PDU. Το μέγιστο επιτρεπόμενο μήκος καθορίζεται κατά την αρχικοποίηση μίας LDP συνόδου.

- LDP Αναγνωριστικό (LDP Identifier): Πεδίο 6 οκτάδων που αναγνωρίζει μοναδικά το label space του LSR αποστολέα του τρέχοντος PDU. Οι 4 πρώτες οκτάδες χρησιμοποιούνται για την αναγνώριση του LSR και πρέπει να περιέχεται μία καθολικά μοναδική τιμή. Οι 2 τελευταίες οκτάδες προσδιορίζουν ένα label space του LSR.

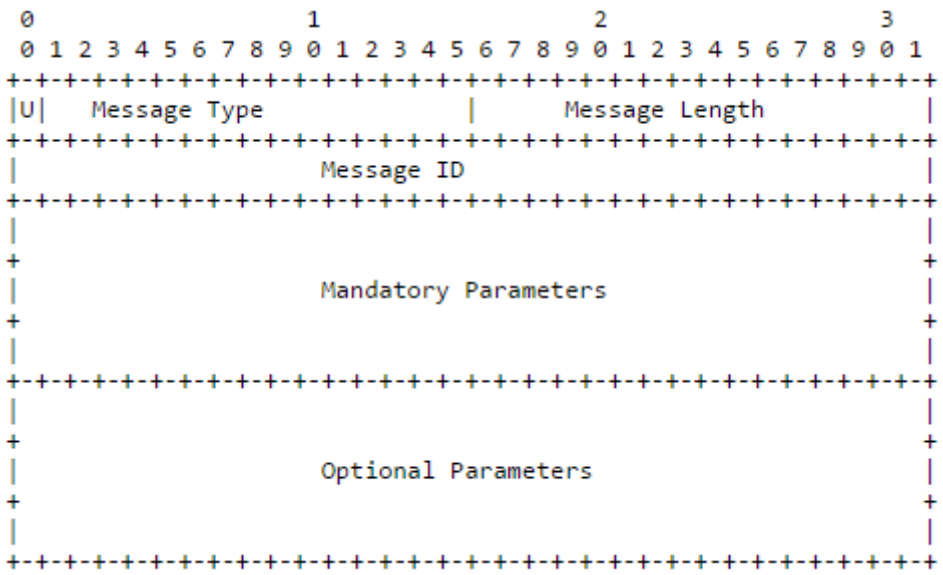
Το LDP χρησιμοποιεί ένα Τύπος-Μήκος-Τιμή (Type-Length-Value – TLV) σχήμα κωδικοποίησης για να κωδικοποιήσει την πληροφορία που μεταφέρουν τα LDP μηνύματα. Ένα LDP TLV κωδικοποιείται ως ένα πεδίο 2 οκτάδων που χρησιμοποιεί 14 bits για να προσδιορίσει τον Τύπο και 2 bits για να καθορίσει τη συμπεριφορά όταν ένας LSR δεν αναγνωρίζει τον Τύπο. Τα πεδία αυτά ακολουθούν ένα πεδίο 2 οκτάδων για το Μήκος και ένα μεταβλητού μήκους για την Τιμή.



Εικόνα 15. Πεδίο TLV

- U-bit: Bit άγνωστου TLV. Μετά τη λήψη ενός άγνωστου TLV, αν το U είναι 0, μία ειδοποίηση πρέπει να επιστρεαφεί στην προέλευση του μηνύματος και ολόκληρο το μήνυμα πρέπει να αγνοηθεί. Αν το U είναι 1, το άγνωστο TLV πρέπει να αγνοηθεί σιωπηρά και το υπόλοιπο μήνυμα να επεξεργαστεί σαν να μην υπήρχε το άγνωστο TLV.
- F-bit: Bit προώθησης άγνωστου TLV. Αυτό το bit ισχύει μόνο όταν το U bit είναι 1 και το LDP μήνυμα που περιέχει το άγνωστο TLV προορίζεται για προώθηση. Αν το F είναι 0, το άγνωστο TLV δεν προωθείται με το υπόλοιπο μήνυμα, ενώ αν είναι 1 προωθείται. Αν τόσο το U, όσο και το F bit είναι 1, ένα TLV μπορεί να διαδοθεί σαν αδιαφανές στοιχείο στους κόμβους που δεν το αναγνωρίζουν.
- Τύπος: Κωδικοποιεί το πώς θα ερμηνευθεί το πεδίο Τιμή.
- Μήκος: Υποδεικνύει το μήκος του πεδίου Τιμή σε οκτάδες.
- Τιμή: Κωδικοποιεί πληροφορία που θα ερμηνευθεί με τον τρόπο που υποδεικνύει το πεδίο Τύπος.

Τα LDP μηνύματα έχουν την ακόλουθη μορφή:



Εικόνα 16. Μορφή LDP μηνυμάτων

- U-bit: Bit άγνωστου μηνύματος. Μετά τη λήψη ενός άγνωστου μηνύματος, αν το U είναι 0 επιστρέφεται μία ειδοποίηση στο δημιουργό του μηνύματος, ενώ αν είναι 1 αγνοείται σιωπηρά.
- Τύπος Μηνύματος: Ταυτοποιεί τον τύπο του μηνύματος.
- Μήκος Μηνύματος: Προσδιορίζει το αθροιστικό μέγεθος των πεδίων ID Μηνύματος, Υποχρεωτικές Παράμετροι και Προαιρετικές Παράμετροι σε οκτάδες.
- ID Μηνύματος: Τιμή 32 bits για την αναγνώριση του μηνύματος. Χρησιμοποιείται από τον LSR αποστολέα για τη διευκόλυνση τυχόν μηνυμάτων Ειδοποίησης που μπορεί να αφορούν το συγκεκριμένο μήνυμα. Ένας LSR που στέλνει ένα μήνυμα Ειδοποίησης σε απόκριση αυτού του μηνύματος πρέπει να περιλαμβάνει το ID Μηνύματος στο αντίστοιχο TLV πεδίο του.
- Υποχρεωτικές Παράμετροι: Σύνολο απαιτούμενων παραμέτρων μεταβλητού μεγέθους.
- Προαιρετικές Παράμετροι: Σύνολο προαιρετικών παραμέτρων μεταβλητού μεγέθους.

Οι πιο συνήθεις τύποι μηνυμάτων είναι οι εξής:

- Ειδοποίηση: Ένας LSR στέλνει ένα μήνυμα Ειδοποίησης για να πληροοφορήσει έναν ομότιμο για ένα σημαντικό γεγονός. Σηματοδοτεί ένα κρίσιμο σφάλμα ή παρέχει συμβουλευτικές πληροφορίες (π.χ. το αποτέλεσμα της επεξεργασίας ενός LDP μηνύματος ή την κατάσταση μίας συνόδου).
- Hello: Ανταλλάσσονται ως μέρος του LDP Μηχανισμού Ανακάλυψης.
- Αρχικοποίηση: Ανταλλάσσεται ως μέρος της διαδικασίας LDP εγκαθίδρυσης συνόδου.
- KeepAlive: Ένας LSR στέλνει KeepAlive μηνύματα ως τμήμα ενός μηχανισμού που ελέγχει την αξιοπιστία της LDP συνόδου.
- Διεύθυνση: Ένας LSR στέλνει το μήνυμα Διεύθυνσης σε έναν ομότιμο για να διαφημίσει τις διευθύνσεις των διεπαφών του.
- Απόσυρση Διεύθυνσης: Ένας LSR στέλνει το μήνυμα Απόσυρσης Διεύθυνσης σε έναν ομότιμο για να αποσύρει τις διευθύνσεις των διεπαφών του που διαφήμισε προηγουμένως.
- Αντιστοίχιση Ετικέτας: Ένας LSR στέλνει το μήνυμα Αντιστοίχισης Ετικέτας σε έναν ομότιμο για να διαφημίσει τη συσχέτιση FEC–ετικέτας.

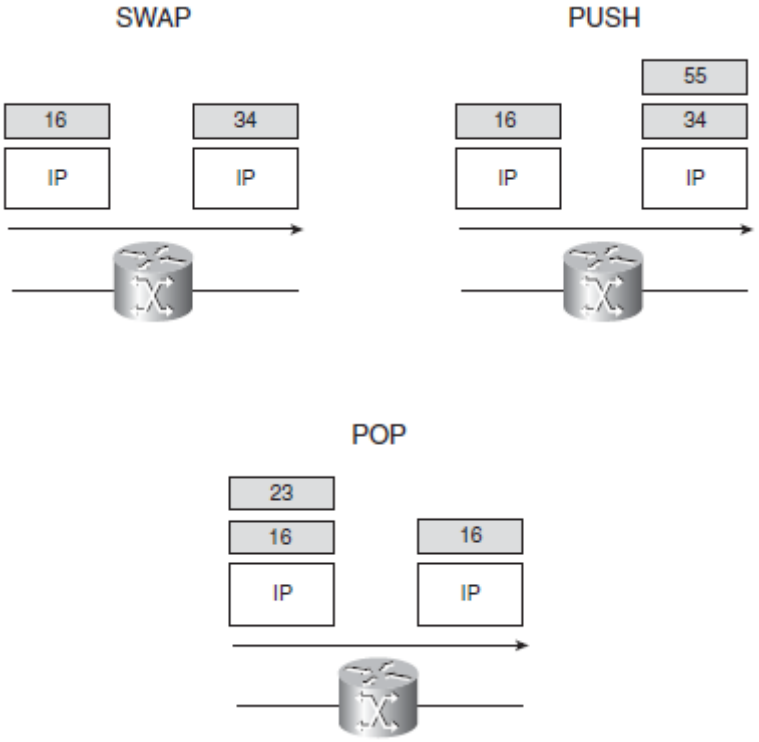
- Αίτηση Ετικέτας: Ένας LSR στέλνει το μήνυμα Αίτησης Ετικέτας σε έναν ομότιμο για να ζητήσει μία αντιστοίχιση για ένα FEC.
- Ακύρωση Αίτησης Ετικέτας: Χρησιμοποιείται για την ακύρωση ενός μηνύματος Αίτησης Ετικέτας που εκκρεμεί.
- Απόσυρση Ετικέτας: Ένας LSR στέλνει ένα μήνυμα Απόσυρσης Ετικέτας σε έναν ομότιμο για να τον ειδοποιήσει ότι ενδεχομένως να μη συνεχίσει να χρησιμοποιεί συγκεκριμένες αντιστοιχίσεις FEC-ετικετών που ο LSR είχε προηγουμένως διαφημίσει. Αυτό το μήνυμα σπάει και τη συσχέτιση μεταξύ FECs και ετικετών.
- Αποδέσμευση Ετικέτας: Ένας LSR στέλνει ένα μήνυμα Αποδέσμευσης Ετικέτας σε έναν ομότιμο για να τον ειδοποιήσει ότι δε χρειάζεται άλλο συγκεκριμένες αντιστοιχίσεις FEC-ετικετών που είχε προηγουμένως αιτηθεί ή/και διαφημίσει προς τον ομότιμο.

Κάποια από τα παραπάνω μηνύματα σχετίζονται μεταξύ τους, για παράδειγμα τα μηνύματα Αντιστοίχιση Ετικέτας, Αίτηση Ετικέτας, Απόσυρση Ετικέτας και Αποδέσμευση Ετικέτας. [3]

3. Προώθηση MPLS πακέτων

3.1 Λειτουργίες Ετικέτας

Υπάρχουν 3 πιθανές λειτουργίες για τις MPLS ετικέτες: εναλλαγή, εισαγωγή και εξαγωγή. Κοιτώντας την ετικέτα στην κορυφή της στοίβας και την αντίστοιχη εγγραφή στην LFIB, ο LSR μαθαίνει πως να προωθήσει το πακέτο. Συγκεκριμένα, καθορίζει ποια από τις λειτουργίες της ετικέτας να επιτελέσει (εναλλαγή, εισαγωγή ή εξαγωγή) και ποιο είναι το next hop στο οποίο θα προωθηθεί το πακέτο. Η λειτουργία εναλλαγής σημαίνει πως η ετικέτα στην κορυφή θα αντικατασταθεί από κάποια άλλη, ενώ η λειτουργία εισαγωγής ότι η ετικέτα στην κορυφή θα αντικατασταθεί από κάποια άλλη και έπειτα θα εισαχθούν στη στοίβα μία ή περισσότερες ετικέτες. Η λειτουργία εξαγωγής σημαίνει ότι θα αφαιρεθεί η ετικέτα στην κορυφή.



Εικόνα 17. Λειτουργίες ετικετών

Όταν ένας δρομολογητής δέχεται ένα πακέτο με ετικέτα ανατρέχει στην LFIB του για να καθορίσει πως θα το προωθήσει. Αφού ο δρομολογητής αναζητήσει την ετικέτα, το πακέτο μπορεί να προωθηθεί είτε με νέα ετικέτα είτε χωρίς ετικέτα, ως IP πακέτο. Η περίπτωση όπου ένας ingress LSR λαμβάνει ένα IP πακέτο και το προωθεί ως MPLS πακέτο (δηλαδή με ετικέτα) λέγεται IP-to-label. Αν ένας LSR λάβει ένα πακέτο με ετικέτα, μπορεί είτε να βγάλει όλες τις ετικέτες και να το προωθήσει ως IP πακέτο είτε να το προωθήσει με νέα ετικέτα. Η πρώτη περίπτωση καλείται label-to-IP και η δεύτερη label-to-label. Στην παρακάτω εικόνα φαίνεται η LFIB ενός δρομολογητή.

Local tag	Outgoing tag or VC	Prefix or Tunnel Id	Bytes tag switched	Outgoing interface	Next Hop
16	Untagged	10.1.1.0/24	0	Et0/0/0	10.200.200.2
17	16	10.200.202.0/24	0	Et0/0/0	10.200.200.2
18	Pop tag	10.200.203.0/24	0	Et0/0/0	10.200.200.2
19	Pop tag	10.200.201.0/24	0	Et0/0/0	10.200.200.2
20	18	10.200.254.4/32	0	Et0/0/0	10.200.200.2
21	Pop tag	10.200.254.2/32	0	Et0/0/0	10.200.200.2
22	17	10.200.254.3/32	0	Et0/0/0	10.200.200.2

Εικόνα 18. Η LFIB ενός LSR

Η τοπική ετικέτα (local tag) είναι η ετικέτα την οποία ο LSR περιμένει ένα εισερχόμενο πακέτο να έχει στην κορυφή της στοίβας του. Για παράδειγμα, αν ο LSR λάμβανε πακέτο με ετικέτα 22, θα την ενάλλασσε με την ετικέτα 17 και θα το προωθουσε μέσω της Ethernet0/0/0 διεπαφής του στη next-hop διεύθυνση 10.200.200.2 (περίπτωση label-to-label). Αντίστοιχα, αν λάβει πακέτο με ετικέτα 16, θα το προωθήσει χωρίς ετικέτα, δηλαδή ως IP πακέτο (περίπτωση label-to-IP). Τέλος, αν λάβει ένα πακέτο με ετικέτα 18, θα εξάγει την ετικέτα και θα το προωθήσει είτε ως MPLS είτε ως IP πακέτο (εξαρτάται από το πλήθος των ετικετών στη στοίβα του πακέτου).

Σε κανονική λειτουργία, ένας LSR πρέπει να λαμβάνει πακέτα με ετικέτες στην κορυφή τις οποίες να γνωρίζει (τις μαθαίνει μέσω του LDP). Ωστόσο, μπορεί να συμβεί κάποιο σφάλμα στο δίκτυο και ένας LSR Να λάβει πακέτο με ετικέτα που δεν μπορεί να βρει στην LFIB του. Τότε, έχει δύο επιλογές: να βγάλει τις ετικέτες και να προσπαθήσει να προωθήσει το πακέτο ή να το απορρίψει. Επειδή ο LSR δε γνωρίζει την ετικέτα, δε γνωρίζει ούτε τι είδους πακέτο είναι ούτε ποιος είναι ο προορισμός του. Οπότε προωθώντας το δεν εξασφαλίζεται ότι δεν θα απορριφθεί από κάποιο δρομολογητή στη συνέχεια. Οπότε, η ενδεδειγμένη λύση είναι ο LSR να απορρίψει ένα τέτοιο πακέτο. [1]

3.2 Στοιχεία MPLS Προώθησης

Το “Next Hop Label Forwarding Entry” (NHLFE) χρησιμοποιείται όταν προωθείται ένα πακέτο με ετικέτα. Περιέχει τις ακόλουθες πληροφορίες:

1. τη next-hop διεύθυνση του πακέτου.
2. πως να χειριστεί την ετικέτα στην κορυφή της στοίβας (εναλλαγή, εισαγωγή ή εξαγωγή).
3. τη L2 ενθυλάκωση που θα χρησιμοποιηθεί κατά τη μετάδοση του πακέτου.
4. τον τρόπο κωδικοποίησης της στοίβας ετικετών κατά τη μετάδοση του πακέτου.
5. άλλες χρήσιμες πληροφορίες για την κατάλληλη προώθηση των πακέτων.

Ενδέχεται σε κάποιον LSR η next-hop διεύθυνση ενός πακέτου να είναι ο ίδιος ο LSR. Στην περίπτωση αυτή, ο LSR θα έπρεπε να εξάγει την κορυφαία ετικέτα και να “προωθήσει” το πακέτο στον εαυτό του. Στη συνέχεια, αν στη στοίβα υπάρχουν κι άλλες ετικέτες, ο LSR το προωθεί ως MPLS πακέτο. Διαφορετικά, θα το προωθήσει ως IP πακέτο.

Το “Incoming Label Map” (ILM) αντιστοιχεί κάθε εισερχόμενη ετικέτα σε κάποιο NHLFE και χρησιμοποιείται για την προώθηση πακέτων με ετικέτα. Το “FEC-to-NHLFE” (FTN) αντιστοιχεί κάθε FEC σε κάποιο NHLFE και χρησιμοποιείται για την προώθηση IP πακέτων που πρέπει να μετατραπούν σε MPLS πακέτα πριν προωθηθούν.

Για να προωθήσει συνεπώς, ένας LSR ένα πακέτο με ετικέτα θα εξετάσει την ετικέτα στην κορυφή της στοίβας. Χρησιμοποιώντας το ILM θα αντιστοιχίσει την ετικέτα σε κάποιο NHLFE και, βάσει των πληροφοριών που περιέχει, θα καθορίσει που θα προωθήσει το πακέτο και πως θα διαχειριστεί τη στοίβα ετικετών. Για να προωθήσει ένα IP πακέτο, θα αναλύσει την κεφαλίδα του για να καθορίσει το FEC του πακέτου. Έπειτα, θα χρησιμοποιήσει το FTN για να αντιστοιχίσει το FEC σε κάποιο NHLFE και στη συνέχεια θα το προωθήσει (στην περίπτωση αυτή, δεν μπορεί φυσικά να κάνει εξαγωγή ετικέτας). [4]

3.3 Δεσμευμένες Ετικέτες

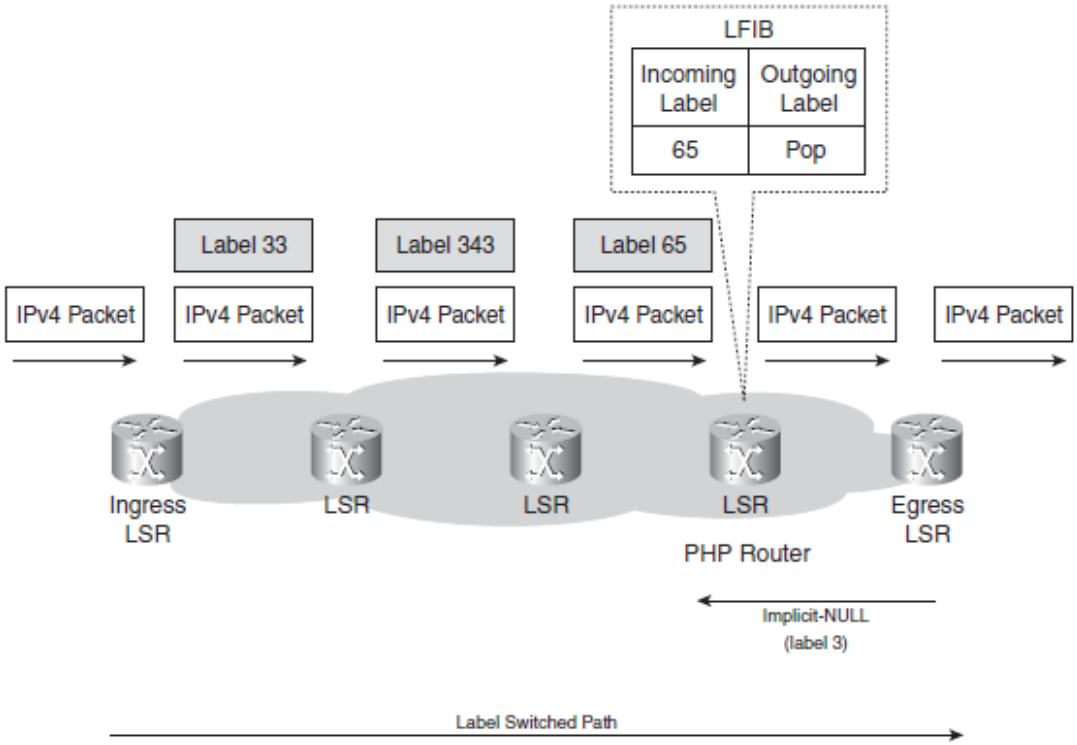
Οι ετικέτες 0 – 15 ονομάζονται δεσμευμένες ετικέτες. Ένας LSR δεν μπορεί να τις χρησιμοποιήσει υπό φυσιολογικές περιπτώσεις για την προώθηση πακέτων, αλλά θα

τους αναθέσει μία ιδιαίτερη λειτουργία. Η ετικέτα 0 είναι η explicit NULL, ενώ η ετικέτα 3 είναι η implicit NULL. Η ετικέτα 1 είναι η router alert και η ετικέτα 14 η OAM alert. Στις υπόλοιπες ετικέτες από 0 – 15 δεν έχει ανατεθεί κάποια συγκεκριμένη λειτουργία ακόμα.

3.3.1 Implicit NULL Ετικέτα

Η implicit NULL ετικέτα έχει τιμή 3. Ένας egress LSR αναθέτει την ετικέτα αυτή σε κάποιο FEC στο οποίο δεν θέλει να ανατεθεί κάποια επιπλέον ετικέτα. Με αυτό τον τρόπο ζητάει από τον προηγούμενο LSR να κάνει pop την ετικέτα του πακέτου. Ειδικότερα, ο egress LSR στέλνοντας την implicit NULL ετικέτα, θα ειδοποιήσει τον προτελευταίο LSR του LSP να προωθήσει τα πακέτα χωρίς να υπάρχει κάποια ετικέτα στη στοίβα. Δηλαδή, ο egress LSR θα λάβει ένα IP και όχι MPLS πακέτο. Με τον τρόπο αυτό, ο egress LSR βελτιώνει την απόδοσή του, καθώς θα κάνει μόνο μία IP αναζήτηση για να βρει τον next-hop δρομολογητή, αντι για 2 αναζητήσεις: μία στην LFIB, όπου θα καταλάβαινε ότι πρέπει να κάνει pop την MPLS ετικέτα, και μία IP αναζήτηση.

Η χρήση της implicit NULL ετικέτας στο τέλος ενός LSP λέγεται Penultimate Hop Popping (PHP). Ο PHP δρομολογητής θα συμβουλευθεί την LFIB του και θα πραγματοποιήσει εξαγωγή ετικέτας, προωθώντας το πακέτο όπως υποδείκνυε η ετικέτα που βρισκόταν προηγουμένως στην κορυφή της στοίβας.



Εικόνα 19. Penultimate Hop Popping

Η χρήση της implicit NULL ετικέτας θα μπορούσε να επεκταθεί και σε άλλες περιπτώσεις, για παράδειγμα σε πακέτα με 2 ή περισσότερες ετικέτες στην στοίβα. Τότε, η implicit NULL ετικέτα θα ειδοποιούσε τον PHP δρομολογητή να εξάγει μία ετικέτα και να στείλει το πακέτο στον egress LSR με μία ετικέτα λιγότερη. Έτσι, ο egress LSR θα κάνει μία ανζήτηση λιγότερη. Η χρήση της ετικέτας δεν σημαίνει πως πρέπει να αφαιρεθούν όλες οι ετικέτες της στοίβας, αλλά μόνο μία. Σε κάθε περίπτωση, επιτρέπει στον egress LSR να πραγματοποιήσει μία αναζήτηση λιγότερη. Παρότι η τιμή 3 σηματοδοτεί την implicit NULL ετικέτα, η ετικέτα 3 δεν θα φανεί σε καμία περίπτωση ως ετικέτα στη στοίβα ενός MPLS πακέτου. [1,4]

3.3.2 Explicit NULL Ετικέτα

Η χρήση της implicit NULL ετικέτας προσθέτει αποτελεσματικότητα κατά την προώθηση των πακέτων. Ωστόσο, το πακέτο προωθείται από τον PHP δρομολογητή είτε με μία ετικέτα λιγότερη είτε ως IP πακέτο. Εκτός από την τιμή, στην ετικέτα υπάρχουν και τα Experimental (EXP) bits. Όταν μια ετικέτα αφαιρείται, χάνονται και τα EXP bits. Τα bits αυτά χρησιμοποιούνται αποκλειστικά για σκοπούς QoS, γι'αυτό και αν χαθούν τα bits μαζί με την ετικέτα, χάνεται και το QoS κομμάτι του πακέτου.

Τη λύση στο πρόβλημα αυτό δίνει η explicit NULL ετικέτα, καθώς ο egress LSR ειδοποιεί με την explicit NULL ετικέτα (τιμή 0) τον PHP δρομολογητή. Τότε, ο egress LSR λαμβάνει πακέτα με ετικέτα με τιμή 0 στην κορυφή της στοίβας. Ο LSR δεν μπορεί να προωθήσει το πακέτο ψάχνοντας την τιμή 0 στην LFIB, καθώς μπορεί να έχει ανατεθεί σε πολλαπλά FECs. Ο LSR απλώς θα εξάγει την explicit NULL ετικέτα και θα κάνει μία νέα αναζήτηση για να προωθήσει το πακέτο. Το πλεονέκτημα όμως είναι ότι ο egress LSR μπορεί να ανακτήσει την QoS πληροφορία του πακέτου από τα EXP bits της explicit NULL ετικέτας.

3.3.3 Router Alert Ετικέτα

Η Router Alert ετικέτα έχει τιμή 1 και μπορεί να βρίσκεται οπουδήποτε στη στοίβα, εκτός από το τέλος της. Όταν είναι στην κορυφή της στοίβας ειδοποιεί τον LSR ότι το πακέτο απαιτεί μία πιο προσεκτική προσέγγιση. Συνεπώς, το πακέτο δεν προωθείται από το hardware, αλλά από μία διεργασία λογισμικού. Όταν προωθείται το πακέτο αφαιρείται η ετικέτα με τιμή 1. Έπειτα γίνεται μία αναζήτηση στην LFIB για την επόμενη ετικέτα στη στοίβα και αποφασίζεται που θα προωθηθεί το πακέτο και πως θα χειριστεί την ετικέτα στην κορυφή της στοίβας (εξαγωγή, εισαγωγή, εναλλαγή). Τέλος, θα εισάγει και πάλι τη Router Alert ετικέτα και θα προωθήσει το πακέτο στο next-hop δρομολογητή.

3.3.4 OAM Alert Ετικέτα

Η ετικέτα με τιμή 14 λέγεται Operation and Maintenance (OAM) ετικέτα και χρησιμοποιείται κυρίως για ανίχνευση βλαβών, εντοπισμό και παρακολούθηση απόδοσης. Με την ετικέτα αυτή τα OAM πακέτα διαφοροποιούνται από τα κανονικά MPLS πακέτα δεδομένων.

3.3.5 Μη Δεσμευμένες Ετικέτες

Πέραν των δεσμευμένων ετικετών 0 – 15, όλες οι υπόλοιπες μπορούν να χρησιμοποιηθούν για προώθηση πακέτων. Καθώς το πεδίο της τιμής ετικέτας αποτελείται από 20 bits, οι ετικέτες από 16 έως 1.048.574 (2^{20}-1) χρησιμοποιούνται για κανονική προώθηση πακέτων. [1]

3.4 Επιλογή Διαδρομής

Η επιλογή διαδρομής αναφέρεται στη μέθοδο που χρησιμοποιείται για την επιλογή LSP για ένα συγκεκριμένο FEC. Η MPLS αρχιτεκτονική υποστηρίζει δύο επιλογές: hop by hop δρομολόγηση και ρητή δρομολόγηση.

Η hop by hop δρομολόγηση επιτρέπει σε κάθε κόμβο να επιλέγει ανεξάρτητα το επόμενο βήμα για κάθε FEC. Αυτός είναι και ο συνήθης τρόπος στα υπάρχοντα IP δίκτυα. Σε ένα ρητά δρομολογημένο LSP, κάθε LSR δεν επιλέγει ανεξάρτητα το επόμενο βήμα, αλλά ένας LSR (συνήθως ο ingress ή ο egress) καθορίζει αρκετούς ή και όλους τους LSRs του LSP. Αν ένας LSR επιλέξει καθορίσει ολόκληρο το LSP, τότε το LSP θεωρείται “αυστηρά” ρητά δρομολογημένο, ενώ αν καθορίζει τμήμα μόνο του LSP, θεωρείται “γενικευμένα” ρητά δρομολογημένο.

Η αλληλουχία από LSRs που ακολουθείται από ένα ρητά δρομολογημένο LSP μπορεί να επιλεχθεί είτε μέσω ρύθμισης παραμέτρων είτε δυναμικά από έναν κόμβο. Για παράδειγμα, ο egress LSR μπορεί να κάνει χρήση των πληροφοριών τοπολογίας που έχει ώστε να υπολογίσει ολόκληρο το μονοπάτι που καταλήγει σε αυτόν. Η ρητή δρομολόγηση μπορεί να φανεί χρήσιμη σε αρκετές περιπτώσεις, όπως στις πολιτικές δρομολόγησης ή το traffic engineering. [4]

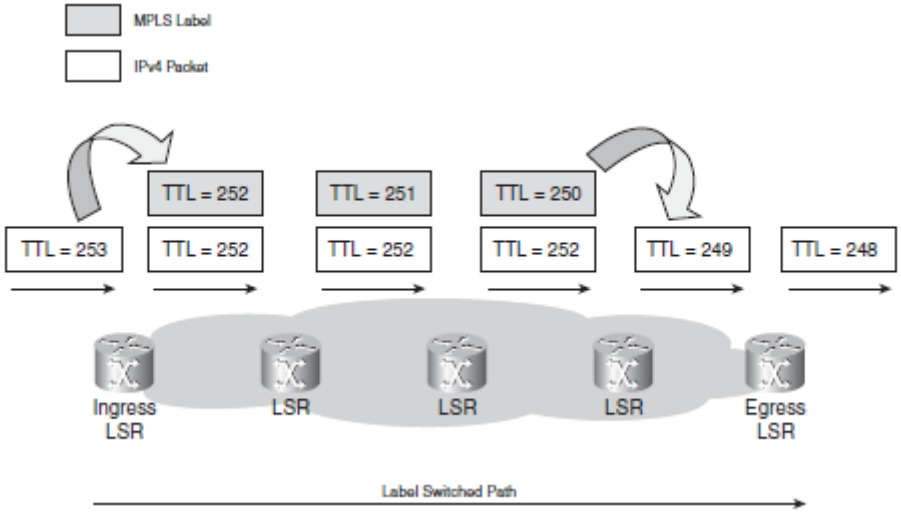
3.5 Μηχανισμός TTL σε MPLS πακέτα

Το Time To Live (TTL) είναι ένας γνωστός μηχανισμός του IP. Στην IP κεφαλίδα, ένα πεδίο 8 bits δηλώνει πόσο χρόνο έχει ακόμα ένα πακέτο πριν λήξει και απορριφθεί. Όταν στέλνεται ένα IP πακέτο, η TTL τιμή είναι συνήθως 255 και μειώνεται κατά 1 με κάθε hop. Αν το TTL φτάσει στο 0 το πακέτο απορρίπτεται και ο δρομολογητής που το απέρριψε στέλνει ένα ICMP μήνυμα τύπου 11 και κωδικού 0 (Time Exceeded) στην πηγή του IP πακέτου.

Με το MPLS, στα IP πακέτα προστίθενται ετικέτες. Για το λόγο αυτό χρειάζεται ένας μηχανισμός για τη μετάδοση του TTL από την IP κεφαλίδα στη στοίβα και αντίστροφα. Με τον τρόπο αυτό εξασφαλίζεται ότι τα πακέτα κάποια στιγμή θα απορριφθούν από κάποιον LSR, σε περίπτωση που υπάρχει κάποιος βρόχος δρομολόγησης.

3.5.1 Περίπτωση IP-to-Label ή Label-to-IP

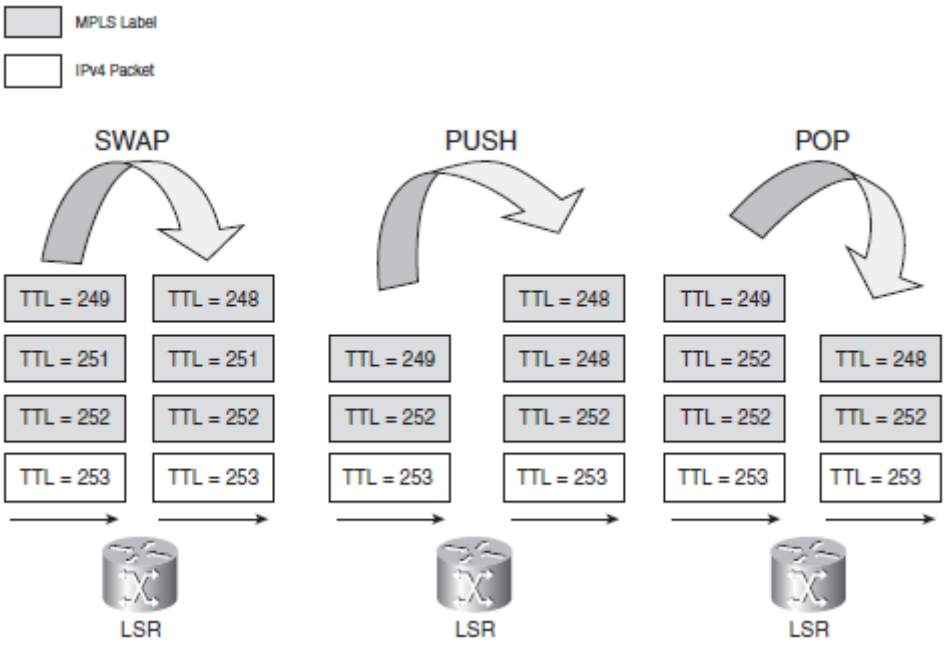
Στο MPLS, η χρήση του TTL πεδίου στην ετικέτα είναι ίδια με αυτή στην IP κεφαλίδα. Όταν ένα IP πακέτο εισέρχεται στον ingress LSR, η τιμή του IP TTL αντιγράφεται (αφού μειωθεί κατά 1) στο πεδίο MPLS TTL της ετικέτας (ή ετικετών) που εισάγεται στη στοίβα. Στον egress LSR, η ετικέτα απομακρύνεται και μένει η IP κεφαλίδα. Η MPLS TTL τιμή αντιγράφεται τώρα στο αντίστοιχο πεδίο της IP κεφαλίδας.



Εικόνα 20. Συμπεριφορά TTL μεταξύ IP κεφαλίδας και MPLS ετικέτας

3.5.2 Περίπτωση Label-to-Label

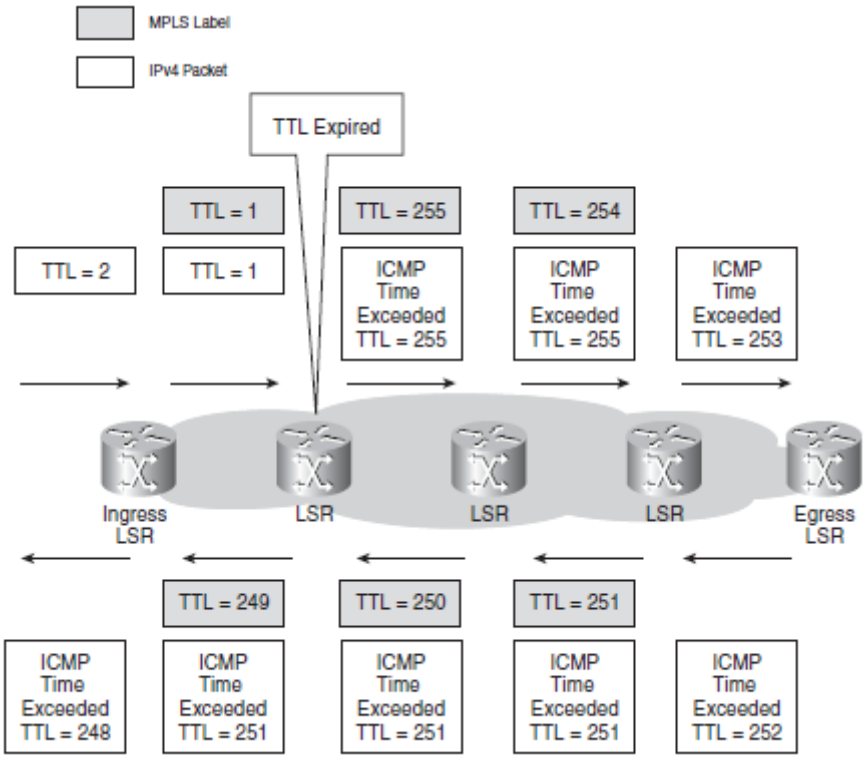
Αν γίνεται εναλλαγή ετικέτας, το TTL της αρχικής ετικέτας μειωμένο κατά 1 αντιγράφεται στη νέα ετικέτα. Αν γίνεται εισαγωγή μίας ή περισσότερων ετικετών, η τιμή της αρχικής ετικέτας (μειωμένη κατά 1) αντιγράφεται στις νέες ετικέτες που εισάγονται στη στοίβα. Τέλος, αν γίνεται εξαγωγή ετικέτας το TTL – 1 της ετικέτας που εξάγεται αντιγράφεται στην ετικέτα που βρισκόταν ακριβώς από κάτω και πλέον είναι στην κορυφή της στοίβας. Ωστόσο, σε περίπτωση που η ετικέτα που εξάγεται έχει μεγαλύτερο TTL από την ετικέτα που θα βρίσκεται πλέον στην κορυφή της στοίβας, δεν γίνεται η αντιγραφή της τιμής για να αποφευχθεί ένας βρόχος δρομολόγησης. Ο ενδιάμεσος LSR που εναλλάσει, εισάγει ή εξάγει τις ετικέτες δεν αλλάζει το TTL των υπόλοιπων ετικετών της στοίβας ή της IP κεφαλίδας, αλλά ασχολείται αποκλειστικά με την ετικέτα που βρίσκεται στην κορυφή.



Εικόνα 21. TTL συμπεριφορά για εναλλαγή, εισαγωγή και εξαγωγή ετικέτας

3.5.3 Λήξη TTL

Όταν ένας LSR λαμβάνει ένα MPLS πακέτο με TTL τιμή 1, το απορρίπτει και στέλνει ένα ICMP μήνυμα “time exceeded” στην πηγή του IP πακέτου. Ωστόσο, το ICMP μήνυμα δε στέλνεται αμέσως πίσω στην πηγή καθώς ένας ενδιάμεσος LSR μπορεί να μη γνωρίζει ένα IP μονοπάτι που να οδηγεί στην πηγή του πακέτου. Το ICMP μήνυμα προωθείται κατά μήκος του LSP που ακολουθούσε το γνήσιο πακέτο. Ο ενδιάμεσος LSR προωθεί το μήνυμα κατά μήκος του LSP ελπίζοντας πως θα φθάσει σε κάποιο δρομολογητή (τον egress LSR για παράδειγμα) ο οποίος θα γνωρίζει μία διαδρομή που να καταλήγει στην πηγή. [1]



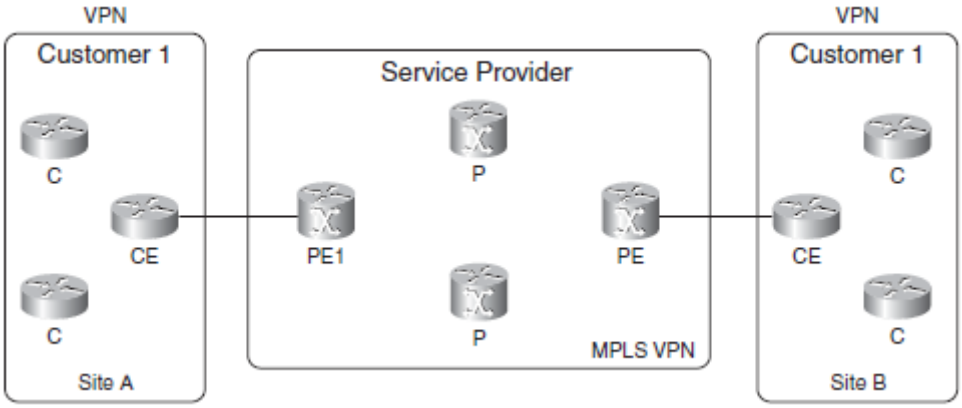
Εικόνα 22. Αποστολή ICMP "Time Exceeded" σε MPLS δίκτυο

4. MPLS VPN

4.1 Εισαγωγή στο MPLS VPN

Το Virtual Private Network (VPN) είναι ένα δίκτυο που προσομοιώνει ένα ιδιωτικό δίκτυο πάνω από μία κοινή υποδομή. Μπορεί να παρέχει επικοινωνία σε L2 ή L3 επίπεδο. Το VPN συνήθως ανήκει σε μία εταιρία που επεκτείνεται σε διάφορες τοποθεσίες, τις οποίες διασυνδέει μεταξύ τους μέσω της κοινής υποδομής του τηλεπικοινωνιακού παρόχου. Η ελάχιστη απαίτηση για να επιτευχθεί η συνδεσιμότητα είναι πως όλες οι περιοχές του δικτύου του πελάτη μπορούν να διασυνδεθούν και διαχωρίζονται εντελώς από άλλα VPNs. Ωστόσο, τα VPN μοντέλα στο επίπεδο του IP ενδεχομένως να απαιτούν περισσότερα. Μπορούν να παρέχουν συνδεσιμότητα μεταξύ διαφορετικών VPNs όταν αυτό απαιτείται, καθώς και να παρέχουν σύνδεση στο Ίντερνετ. Το MPLS VPN γίνεται εφικτό καθώς ο πάροχος τρέχει MPLS στο δίκτυο κορμού, το οποίο παρέχει μία αποσύνδεση των επιπέδων προώθησης και ελέγχου, την οποία δεν υποστηρίζει το IP.

Το VPN προϋπήρχε του MPLS. Πιο διαδεδομένες ήταν οι τεχνολογίες του ATM και του Frame Relay που παρείχαν VPN υπηρεσίες στο L2. Ο πάροχος είχε ένα Frame Relay ή ATM δίκτυο κορμού και παρείχε L3 συνδεσιμότητα στους δρομολογητές του πελάτη. Το μοντέλο αυτό αναφερόταν ως overlay μοντέλο. Ο πάροχος είχε την ιδιοκτησία ή τη διαχείριση των περιφερειακών δρομολογητών που συνδέονταν (και βρίσκονταν μέσα) στο δίκτυο του πελάτη. Τα peer-to-peer VPN δίκτυα επίσης υπήρχαν αλλά δεν ήταν δημοφιλή, κυρίως λόγω της δυσκολίας υλοποίησης και διατήρησης επειδή χρειάζονταν λίστες διανομής, φιλτράρισμα πακέτων και GRE τούνελ. Το MPLS VPN είναι ένα παράδειγμα επεκτάσιμου peer-to-peer VPN μοντέλου.



Εικόνα 23. Σχηματική επισκόπηση MPLS VPN

Ο τηλεπικοινωνιακός πάροχος προσφέρει την κοινή δημόσια υποδομή που θα χρησιμοποιήσουν οι πελάτες. Ως PE (ορισμοί στο 1^{ο} Κεφάλαιο) δρομολογητής αναφέρεται ο δρομολογητής που βρίσκεται στα άκρα του δικτύου του παρόχου και συνδέεται με τον CE δρομολογητή στο δίκτυο του πελάτη. Οι P δρομολογητές βρίσκονται εντός του δικτύου του παρόχου και δεν έχουν απευθείας σύνδεση με τους δρομολογητές του πελάτη. Τόσο οι P όσο και οι PE δρομολογητές υποστηρίζουν MPLS. Ένας CE δρομολογητής έχει απευθείας L3 σύνδεση με τον PE δρομολογητή και δε χρειάζεται να τρέχει MPLS. Ο C δρομολογητής βρίσκεται στο δίκτυο του πελάτη και δεν έχει απευθείας σύνδεση με τον PE.

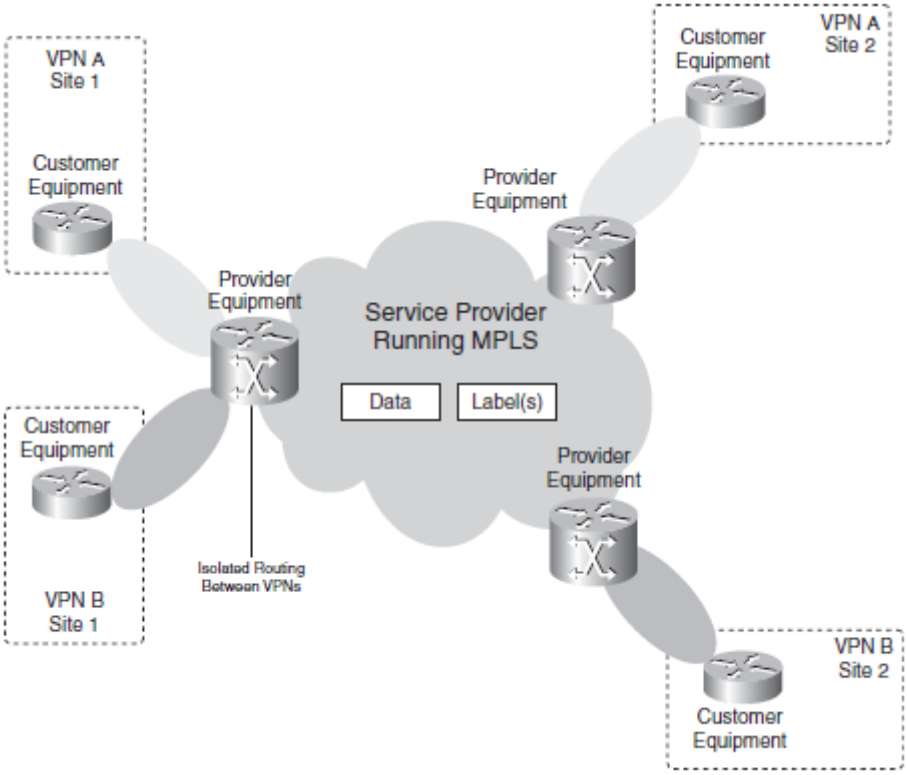
Επειδή οι CE και PE δρομολογητές αλληλεπιδρούν στο L3, πρέπει μεταξύ τους να τρέχει ένα πρωτόκολλο δρομολόγησης ή να υπάρχει στατική δρομολόγηση. Ο μοναδικός ομότιμος που έχει ο CE δρομολογητής εκτός της περιοχής του, είναι ο PE δρομολογητής. Δεν σχηματίζει σχέσεις ομοτιμίας με κάποιον από τους άλλους CE

δρομολογητές στις άλλες περιοχές κατά μήκος του δικτύου του παρόχου, όπως συμβαίνει στο overlay μοντέλο. Εξάλλου, το όνομα του peer-to-peer μοντέλου προήλθε ακριβώς από το γεγονός ότι οι CE και PE είναι ομότιμοι στο L3.

Το P στο VPN αναφέρεται στην ιδιωτικότητα. Οι πελάτες του παρόχου επιτρέπεται να έχουν το δικό τους σχήμα IP διευθυνσιοδότησης. Αυτό σημαίνει πως μπορούν να χρησιμοποιήσουν ιδιωτικές διευθύνσεις (RFC 1918) ή ακόμα και IP διευθύνσεις που χρησιμοποιούν άλλοι πελάτες του παρόχου. Αν τα πακέτα επρόκειτο να προωθηθούν στο δίκτυο του παρόχου ως IP πακέτα θα προκαλούταν σύγχυση στους P δρομολογητές. Αν οι πελάτες δεν μπορούν να ορίσουν ανεξάρτητοι το δικό τους σχήμα διευθυνσιοδότησης, τότε θα πρέπει να χρησιμοποιούν ένα μοναδικό εύρος διευθύνσεων και τα πακέτα θα προωθούνται μέσα στο δίκτυο του παρόχου με βάση την IP διεύθυνση προορισμού τους. Αυό σημαίνει πως οι P και οι PE δρομολογητές πρέπει να έχουν ολόκληρους τους πίνακες δρομολόγησης κάθε πελάτη, κάτι που θα οδηγούσε σε πολύ μεγάλους πίνακες δρομολόγησης. Το μόνο πρωτόκολλο δρομολόγησης που θα μπορούσε να μεταφέρει τόσο μεγάλους πίνακες είναι το BGP, και συνεπώς όλοι οι P και PE δρομολογητές θα έπρεπε να τρέχουν iBGP (internal BGP) μεταξύ τους. Ωστόσο, αυτό δεν αποτελεί VPN σχήμα, καθώς δεν είναι ιδιωτικό στους πελάτες.

Μία άλλη λύση είναι κάθε P και PE δρομολογητής να έχει έναν ιδιωτικό πίνακα δρομολόγησης για κάθε πελάτη. Σε κάθε δρομολογητή μπορούν να τρέχουν πολλαπλές διεργασίες ενός αλγόριθμου δρομολόγησης (μία διεργασία ανά VPN) ώστε να διανεμηθούν οι VPN διαδρομές. Όμως, το να τρέχει μία διεργασία δρομολόγησης ανά VPN σε κάθε P δρομολογητή δεν είναι πολύ ευέλικτη λύση. Επιπλέον, θα υπήρχε το πρόβλημα ότι ένας P δρομολογητής δεν θα μπορεί να καθορίσει σε ποιο VPN ανήκει ένα IP πακέτο, και άρα ποιον ιδιωτικό πίνακα δρομολόγησης πρέπει να συμβουλευθεί.

Η πιο ευέλικτη και επικρατέστερη λύση είναι οι P δρομολογητές να μη γνωρίζουν για τα VPNs. Έτσι, δεν επιβαρύνονται με επιπλέον πληροφορίες δρομολόγησης για τις VPN διαδρομές. Τα IP πακέτα του πελάτη μετατρέπονται σε MPLS πακέτα μέσα στο δίκτυο του παρόχου. Οι P δρομολογητές δε χρειάζεται πλέον να έχουν τους πίνακες δρομολόγησης των πελατών, απλά χρησιμοποιούν δύο MPLS ετικέτες: μία (εσωτερική) που βοηθάει τον PE ώστε να μπορεί να αναγνωρίσει το VPN και μία (εξωτερική) που χρησιμοποιείται από όλους τους δρομολογητές του παρόχου για να προωθήσει το πακέτο μέσα στο MPLS δίκτυο του παρόχου. Συνεπώς, τη γνώση των VPN την έχουν μόνο οι PE δρομολογητές, κάτι που κάνει ευέλικτη τη λύση του MPLS VPN. [1]



Εικόνα 24. MPLS VPN

4.2 Πλεονέκτηματα MPLS VPN

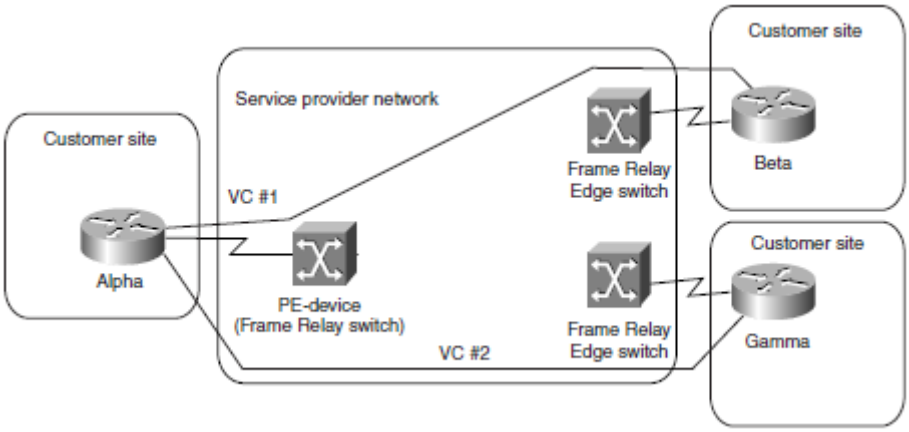
- Επεκτασιμότητα: Το MPLS σχεδιάστηκε ειδικά για να προσφέρει επεκτάσιμες λύσεις, επιτρέποντας δεκάδες χιλιάδες VPN στο ίδιο δίκτυο. Η connectionless αρχιτεκτονική επιτρέπει τη δημιουργία L3 VPNs, εξαλείφοντας την ανάγκη για τούνελ ή VCs.
- Ασφάλεια: Πακέτα από ένα VPN δεν περνούν σε κάποιο άλλο VPN (εκτός κι αν το επιτρέψει ο πάροχος). Η ασφάλεια παρέχεται στα άκρα του δικτύου του παρόχου, εξασφαλίζοντας πως τα πακέτα που παραλαμβάνει ο PE δρομολογητής από έναν πελάτη θα τοποθετηθούν στο σωστό VPN. Τα IP πακέτα που στέλνει ο CE δρομολογητής πρέπει να παραληφθούν σε μία συγκεκριμένη διεπαφή ώστε να ανανγνωριστούν μοναδικά με μία VPN ετικέτα.
- Ευκολία δημιουργίας: Όλη η MPLS λειτουργικότητα βρίσκεται στο δίκτυο του παρόχου, απαιτώντας ελάχιστες ή και καθόλου ρυθμίσεις στις εγκαταστάσεις του πελάτη. Επί της ουσίας, ο CE δρομολογητής ούτε γνωρίζει ούτε χρειάζεται να υποστηρίζει το MPLS.
- Ευέλικτη διευθυνσιοδότηση: Οι πελάτες σχεδιάζουν το δικό τους σχήμα IP διευθυνσιοδότησης, το οποίο είναι ανεξάρτητο από τα σχήματα που θα επιλέξουν άλλοι πελάτες του παρόχου. Μπορούν να χρησιμοποιήσουν ιδιωτικές διευθύνσεις (RFC 1918), χωρίς να χρειαστεί να τις μετατρέψουν σε δημόσιες και να επικοινωνούν ελεύθερα με ένα δημόσιο IP δίκτυο.
- Διαλειτουργικότητα: Το MPLS εξασφαλίζει τη διαλειτουργικότητα με όλες τις δικτυακές πλατφόρμες.
- Ενοποίηση: Οι δυνατότητες ενοποίησης δεδομένων, φωνής και video δίνει στους παρόχους τη δυνατότητα να μειώσουν τα λειτουργικά κόστη.
- Traffic engineering: Το MPLS επιτρέπει στους παρόχους να μεγιστοποιήσουν τη χρήση των δικτυακών πόρων και να λειτουργούν τα δίκτυά τους όσο πιο αποτελεσματικά γίνεται. Μπορεί να εφαρμοστεί ρητή δρομολόγηση, κάτι που

παρακάμπτει τις παραδοσιακές IP τεχνικές προώθησης και παρέχει μηχανισμούς γρήγορης αποκατάστασης και προστασίας.

- Υποστήριξη ολοκληρωμένης Κλάσης Υπηρεσίας (Class of Service – CoS): Το CoS είναι μία σημαντική προϋπόθεση για πολλούς VPN πελάτες, καθώς διευθετεί δύο θεμελιώδεις VPN απαιτήσεις: προβλέψιμη απόδοση και υλοποίηση πολιτικών και υποστήριξη για πολλαπλά επίπεδα υπηρεσιών σε ένα MPLS VPN. Η ταξινόμηση της δικτυακής κίνησης και η ανάθεση ετικετών γίνεται στα άκρα του δικτύου προτού η κίνηση συναθροιστεί σύμφωνα με πολιτικές που ορίζουν οι συνδρομητές και εφαρμόζει ο πάροχος και πριν μεταδοθεί κατά μήκος του δικτύου του παρόχου. Συνεπώς, η κίνηση στα άκρα και στον πυρήνα του δικτύου μπορεί να διαχωριστεί σε διαφορετικές κλάσεις ανάλογα την πιθανότητα απόρριψης ή την καθυστέρηση. [2]

4.3 Overlay VPN Μοντέλο

Το overlay VPN μοντέλο προσφέρει ξεκάθαρο διαχωρισμό μεταξύ των αρμοδιοτήτων του παρόχου και του πελάτη. Ο πάροχος προσφέρει στον πελάτη ένα σύνολο προσομοιωμένων μισθωμένων γραμμών, οι οποίες αναφέρονται ως VC (Virtual Circuits – εικονικά κυκλώματα) και μπορούν να είναι είτε μόνιμα διαθέσιμα (PVC) είτε να δημιουργούνται κατ'απαίτηση (SVC). Ο πελάτης δημιουργεί επικοινωνία μεταξύ των CE δρομολογητών του μέσω των VCs που του προσφέρει ο πάροχος.



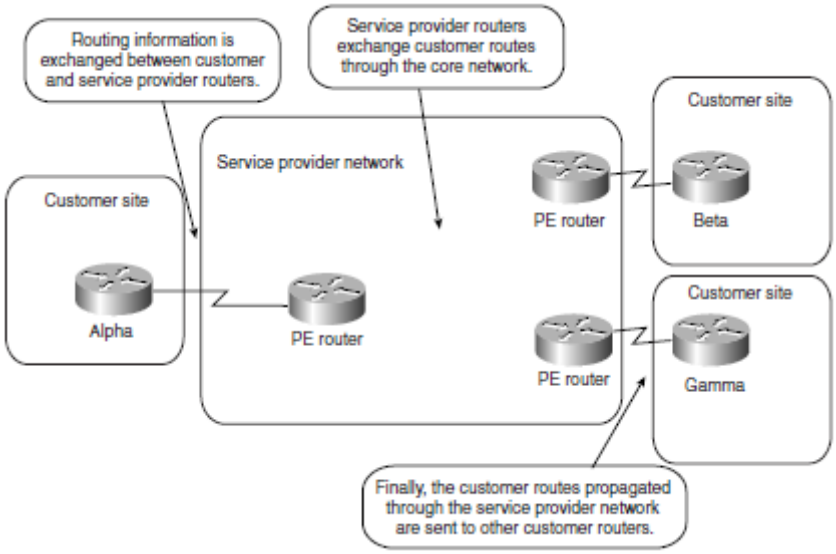
Εικόνα 25. Τοπολογία Overlay VPN μοντέλου

Οι QoS εγγυήσεις στο overlay VPN μοντέλο συνήθως εκφράζονται σε όρους εγγυημένου εύρους ζώνης σε ένα συγκεκριμένο VC (Committed Information Rate – CIR) και μέγιστου διαθέσιμου εύρους ζώνης σε ένα συγκεκριμένο VC (Peak Information Rate – PIR).

Το overlay VPN μοντέλο συνήθως υλοποιείται με WAN L2 τεχνολογίες όπως το ATM ή το Frame Relay, και όχι το MPLS. Παρότι είναι εύκολο στην κατανόηση και σχεδίαση, έχει αρκετά μειονεκτήματα. Ταιριάζει καλύτερα σε μη πλεονάζουσες τοπολογίες με λίγες κεντρικές τοποθεσίες και πολλές απομακρυσμένες, αλλά γίνεται υπερβολικά δύσκολο στη διαχείρισή του σε full mesh τοπολογίες. Επίσης, η σωστή πρόβλεψη της χωρητικότητας των VCs απαιτεί λεπτομερή γνώση της ροής κίνησης, η οποία συνήθως δεν είναι εκ των προτέρων γνωστή, ενώ το κόστος υλοποίησης μεγαλώνει γραμμικά με το πλήθος των point-to-point συνδέσεων. Τέλος, όταν συνδυάζεται με L2 τεχνολογίες εισάγει ένα επιπλέον αχρείαστο επίπεδο πολυπλοκότητας στο δίκτυο του παρόχου, αυξάνοντας τα λειτουργικά κόστη του δικτύου. [5]

4.4 Peer-to-Peer VPN Μοντέλο

Το peer-to-peer μοντέλο δημιουργήθηκε ώστε να εξαλείψει τα μειονεκτήματα του overlay μοντέλου. Ο PE δρομολογητής ανταλλάσσει πληροφορίες δρομολόγησης με τον CE δρομολογητή και οι P δρομολογητές μεταφέρουν τα πακέτα του πελάτη διαμέσου του δικτύου του παρόχου. Τέλος, ένας PE θα παραδώσει τα πακέτα σε έναν CE σε κάποια άλλη περιοχή του δικτύου του πελάτη.



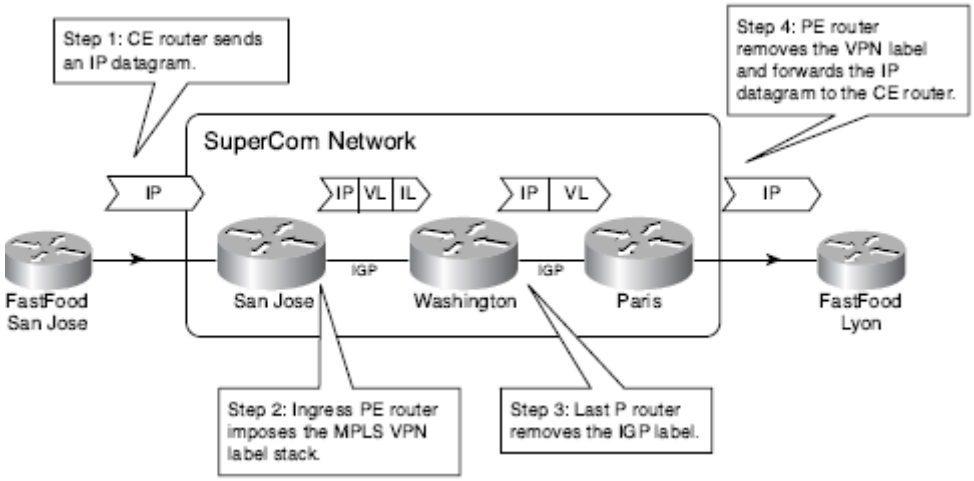
Εικόνα 26. Peer-to-peer VPN μοντέλο

Η διεπαφή μεταξύ των CE και PE δρομολογητών είναι connectionless, κάτι που σημαίνει πώς οι PE δρομολογητές συμμετέχουν στη μεταφορά δεδομένων μεταξύ των CE δρομολογητών και έτσι δεν υπάρχει ανάγκη δημιουργίας VCs στο MPLS VPN. Οι PE δρομολογητές χρησιμοποιούν ένα τροποποιημένο πρότυπο IP προώθησης: έναν ξεχωριστό πίνακα IP δρομολόγησης και προώθησης που λέγεται VRF (Virtual Routing and Forwarding table) και δημιουργείται για κάθε πελάτη. Οι CE και PE δρομολογητές ανταλλάσσουν τις διαδρομές του πελάτη χρησιμοποιώντας ένα πρωτόκολλο δρομολόγησης. Οι διαδρομές στη συνέχεια εισέρχονται στα VRFs των PE δρομολογητών, κάτι που εξασφαλίζει την πλήρη απομόνωση μεταξύ των πελατών.

Η προώθηση των VPN πακέτων κατά μήκος του MPLS δικτύου γίνεται με βάση τη στοίβα ετικετών που προσαρτά στο IP πακέτο ο ingress PE δρομολογητής. Οι 32bit IP διευθύνσεις του πελάτη επεκτείνονται κατά 64 bits για να γίνουν μοναδικά αναγνωρίσιμες στο MPLS δίκτυο. Τα 64 bits αποτελούνται από τις 2 MPLS ετικέτες που προσθέτει ο PE δρομολογητής, μία για την ανανγνώριση του συγκεκριμένου VPN (δεύτερη στη στοίβα – λέγεται και VPN ετικέτα) και μία για την προώθηση του πακέτου εντός του δίκτύου του παρόχου προς τον egress PE (πρώτη στη στοίβα). Οι προκύπτουσες 96bit διευθύνσεις αποκαλούνται VPNv4 διευθύνσεις.

Όταν ο ingress PE δρομολογητής λάβει ένα VPN πακέτο, εξετάζει το αντίστοιχο VRF και η ετικέτα που αντιστοιχίζεται με τη διεύθυνση προορισμού (την αντιστοίχιση έχει κάνει ο egress PE) τίθεται ως VPN ετικέτα. Την ετικέτα που υποδεικνύει πως θα προωθηθεί το πακέτο προς τον egress PE, την έχει αναθέσει το LDP και τη λαμβάνει εξετάζοντας τον πίνακα προώθησης. Όλοι οι P δρομολογητές του δικτύου μετάγουν το VPN πακέτο εξετάζοντας μόνο την πρώτη ετικέτα, η οποία υποδεικνύει τον egress PE δρομολογητή. Οι P δρομολογητές δε χρειάζεται να κοιτάξουν πέρα από την πρώτη ετικέτα και συνεπώς δε γνωρίζουν την ύπαρξη της VPN ετικέτας. Συνήθως, η πρώτη ετικέτα εξάγεται από τον τελευταίο P δρομολογητή, μέσω του Penultimate Hop Popping (Κεφ. 3) και ο egress PE εξάγει τη VPN ετικέτα και προωθεί το IP πακέτο στον CE

δρομολογητή. Ωστόσο, είναι πιθανό ο τελευταίος P δρομολογητής να μην εξάγει καμία ετικέτα και ο egress PE να πραγματοποιήσει δύο εξαγωγές, στέλνοντας κατόπιν το IP πακέτο στον CE δρομολογητή.



Εικόνα 27. Προώθηση πακέτων στο MPLS VPN

Η διαδικασία αναλύεται και στο παράδειγμα της Εικόνας 27. Ένα IP πακέτο στέλνεται από το San Jose στη Lyon και προωθείται μέσω του SuperCom MPLS δικτύου, ακολουθώντας τα εξής βήματα:

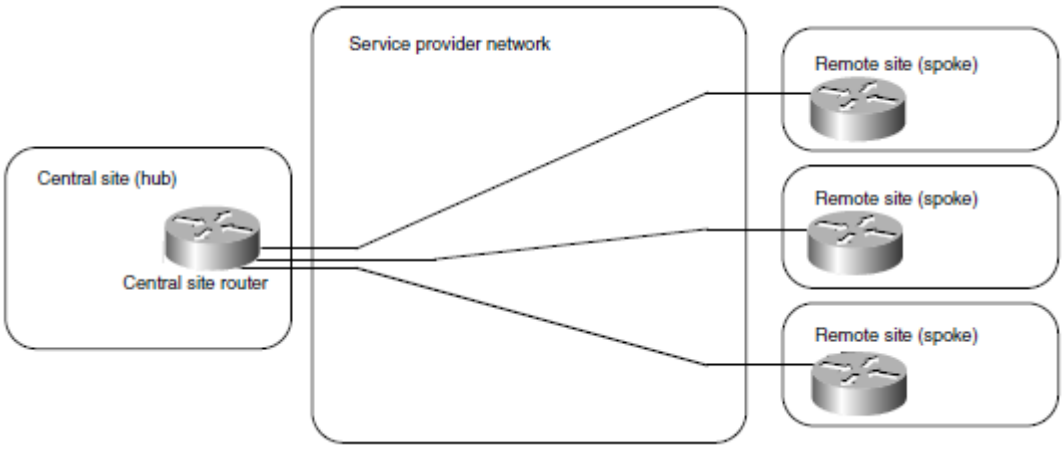
1. Το IP πακέτο στέλνεται από τον CE δρομολογητή (FastFood San Jose) στον PE δρομολογητή (SuperCom San Jose).
2. Ο PE δρομολογητής δημιουργεί το MPLS πακέτο προσθέτοντας την MPLS κεφαλίδα με τις δύο ετικέτες: Την πρώτη (IL στο παράδειγμα) που προσδιορίζει το μονοπάτι προς τον egress PE δρομολογητή (SuperCom Paris) και τη δεύτερη (VL στο παράδειγμα), τη VPN ετικέτα.
3. Ο τελευταίος P δρομολογητής εξάγει την IL ετικέτα, αφήνοντας μόνο τη VPN ετικέτα στην MPLS κεφαλίδα (θα μπορούσε να αφήνει την IL ετικέτα και να κάνει δύο εξαγωγές ο egress PE δρομολογητής).
4. Ο egress PE δρομολογητής εξάγει τη VPN ετικέτα (ή και τις δύο ετικέτες), αφαιρεί την MPLS κεφαλίδα και προωθεί το πακέτο στον προορισμό του, τον CE δρομολογητή (FastFood Lyon). [5,6]

4.5 Συνήθεις Τοπολογίες VPN Δικτύων

Η τοπολογία ενός VPN δικτύου καθορίζεται κάθε φορά από τις ανάγκες της επιχείρησης. Ωστόσο, κάποιες γνωστές τοπολογίες συναντώνται τόσο συχνά που αξίζει να αναλυθούν σε μεγαλύτερο βαθμό.

- **Τοπολογία Hub-and-Spoke**

Στη hub-and-spoke τοπολογία ένας αριθμός απομακρυσμένων γραφείων (spokes) συνδέονται με μία κεντρική τοποθεσία (hub). Τα απομακρυσμένα γραφεία μπορούν να ανταλλάξουν δεδομένα, ωστόσο το πλήθος των δεδομένων αυτών συνήθως είναι αμελητέα. Η τοπολογία αυτή χρησιμοποιείται κατά κύριο λόγο σε οργανισμούς με αυστηρές ιεραρχικές δομές, για παράδειγμα, τράπεζες, κυβερνητικές υπηρεσίες, διεθνείς οργανισμούς κλπ.

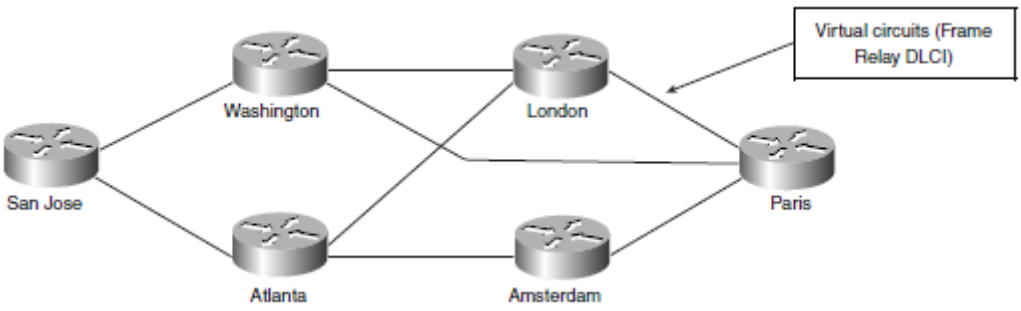


Εικόνα 28. Τοπολογία Hub-and-Spoke

- **Τοπολογία μερικού ή πλήρους πλέγματος**

Σε τοπολογίες όπου οι τοποθεσίες στο VPN συνδέονται μέσω VCs που υπαγορεύονται από τις απαιτήσεις της κίνησης, η ενδεδειγμένη λύση είναι η τοπολογία πλέγματος. Αν όλες οι τοποθεσίες συνδέονται απευθείας μεταξύ τους, η τοπολογία λέγεται πλήρους πλέγματος, ενώ αν δεν συνδέονται όλες οι τοποθεσίες μεταξύ τους λέγεται μερικού πλέγματος. Ενώ η πρόβλεψη για πλήρες πλέγμα είναι αρκετά απλή, η πρόβλεψη για μερικό πλέγμα έχει αρκετές προκλήσεις και προϋποθέτει τα εξής:

- Υπολογισμό της κίνησης.
- Σχεδιασμό βάση των αναγκών κίνησης και πλεονασμού.
- Καθορισμό μέσω ποιων ακριβώς VC θα περνάει η κίνηση μεταξύ δύο τοποθεσιών.

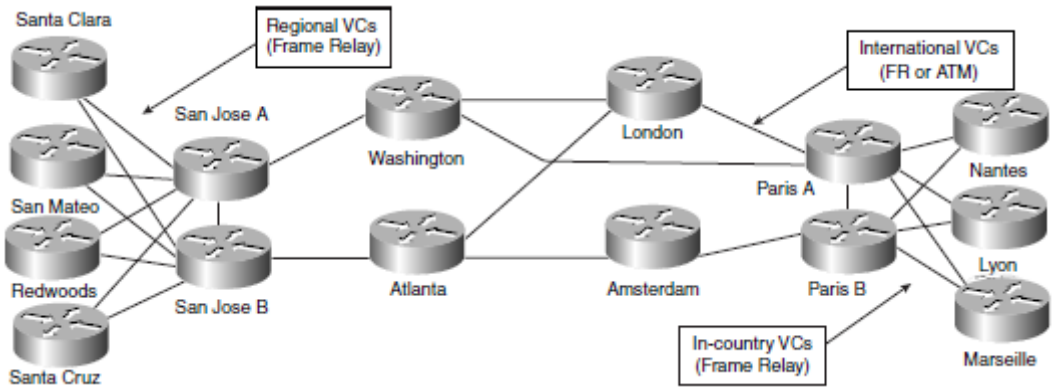


Εικόνα 29. Τοπολογία μερικού πλέγματος

- **Υβριδική Τοπολογία**

Σε μεγάλα VPN δίκτυα προτείνεται ο συνδυασμός των τοπολογιών hub-and-spoke και μερικού πλέγματος. Η καλύτερη προσέγγιση για το σχεδιασμό της υβριδικής τοπολογίας είναι:

- Ο διαχωρισμός του συνολικού δικτύου σε δίκτυα κορμού, διανομής και πρόσβασης.
- Ο σχεδιασμός των δικτύων κορμού και πρόσβασης να γίνει μεμονωμένα (π.χ. hub-and-spoke στο δίκτυο πρόσβασης, μερικό πλέγμα στο δίκτυο κορμού).
- Σύνδεση των δικτύων κορμού και πρόσβασης μέσω του επιπέδου διανομής με έναν τρόπο που τα απομονώνει όσο το δυνατόν περισσότερο. [5]

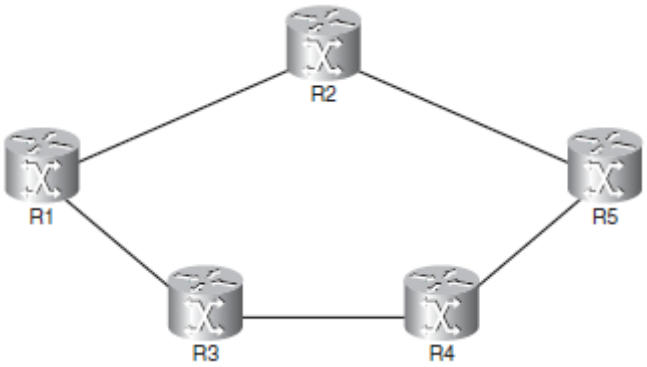


Εικόνα 30. Υβριδική τοπολογία

5. MPLS Traffic Engineering

5.1 Τι προσφέρει το Traffic Engineering

Η προώθηση πακέτων στο IP στηρίζεται στα μονοπάτια ελαχίστου κόστους. Τα πακέτα προωθούνται σε ένα δρομολογητή βασιζόμενα αποκλειστικά στην IP διεύθυνση προορισμού και ανεξάρτητα από το πως προωθούνταν τα πακέτα πριν ή μετά από τον συγκεκριμένο δρομολογητή. Επίσης, η προώθηση των IP πακέτων δε λαμβάνει υπόψιν τη διαθέσιμη χωρητικότητα της ζεύξης με συνέπεια ένας δρομολογητής να στέλνει δεδομένα στη ζεύξη, παρότι αυτή απορρίπτει πακέτα λόγω έλλειψης χωρητικότητας. Το αποτέλεσμα αυτής της συμπεριφοράς είναι κάποιες ζεύξεις να οδηγούνται σε κορεσμό ενώ άλλες να υποχρησιμοποιούνται. Το Traffic Engineering – TE μπορεί να δώσει λύση, οδηγώντας την κίνηση ή ένα μέρος της μακριά από τις υπερφορτωμένες ζεύξεις.



Εικόνα 31. Παράδειγμα προώθησης IP πακέτων

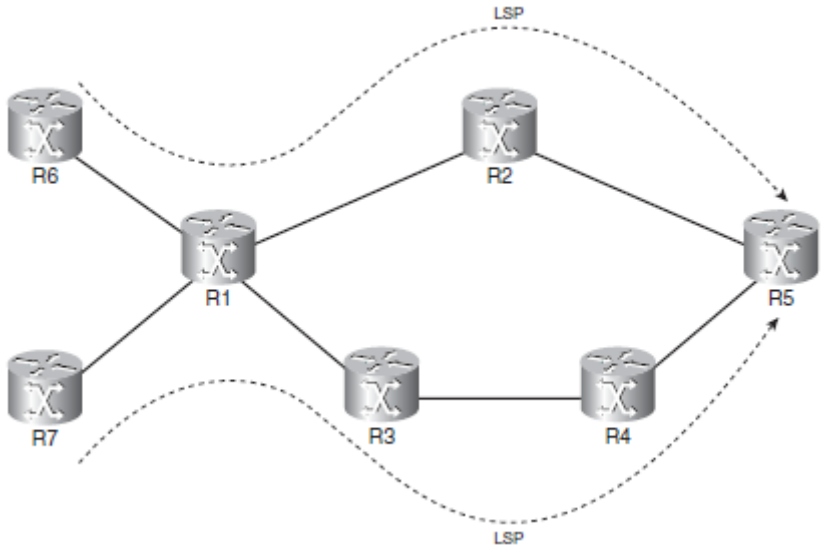
Αν κάθε ζεύξη στο παραπάνω δίκτυο έχει το ίδιο κόστος, το μονοπάτι ελάχιστου κόστους από το δρομολογητή R1 στο δρομολογητή R5 είναι το μονοπάτι R1-R2-R5, με συνέπεια όλα τα πακέτα από τον R1 στον R5 να διαλέγουν αυτό το μονοπάτι και το μονοπάτι R1-R3-R4-R5 να μη χρησιμοποιείται καθόλου. Το φορτίο θα μπορούσε να διανεμηθεί πιο δίκαια αν μεταβαλλόταν το κόστος των ζεύξεων για το εκάστοτε πρωτόκολλο δρομολόγησης. Στο συγκεκριμένο δίκτυο, μεταβάλλοντας κατάλληλα τα κόστη, μπορεί τα δύο μονοπάτια να μοιραστούν εξίσου τον φόρτο. Ωστόσο, αυτό θα λειτουργήσει άριστα για τη διαδρομή R1 – R5 αλλά όχι για τις άλλες διαδρομές (π.χ. από R2 σε R3 ή R4). Συνεπώς, το πρόβλημα του ισομερώς κατανεμημένου φόρτου δεν μπορεί να λυθεί σε ένα πραγματικό δίκτυο απλά μεταβάλλοντας το κόστος κάθε ζεύξης.

Το MPLS TE αποτελεί μία λύση για τους εξής λόγους:

- Παρέχει αποτελεσματική διάδοση του φόρτου στο δίκτυο, αποφεύγοντας τις υποχρησιμοποιούμενες και υπερχρησιμοποιούμενες ζεύξεις.
- Λαμβάνει υπόψιν τη δεδομένη χωρητικότητα των ζεύξεων.
- Λαμβάνει υπόψιν χαρακτηριστικά των ζεύξεων, όπως η καθυστέρηση και το jitter.
- Προσαρμόζεται αυτόματα στις αλλαγές χωρητικότητας και χαρακτηριστικών των ζεύξεων.
- Ο φόρτος δρομολογείται με βάση την προέλευση και όχι τον προορισμό.

Το MPLS TE επιτρέπει στον ingress LSR ενός LSP να υπολογίσει την πιο αποτελεσματική διαδρομή μέσω του δικτύου προς τον egress LSR του LSP.Ο ingress LSR πρέπει να γνωρίζει την τοπολογία του δικτύου και την υπολειπόμενη χωρητικότητα όλων των ζεύξεων. Επίσης, το MPLS με τη μεταγωγή ετικετών επιτρέπει τη δρομολόγηση με βάση την προέλευση, σε αντίθεση με το IP που δρομολογεί με βάση τον προορισμό. Ο ingress LSR θα δρομολογήσει με βάση την πρώτη ετικέτα στη

στοίβα, αφότου οι LSRs θα συμφωνήσουν ποιες ετικέτες να χρησιμοποιήσουν για κάθε LSP. Αυτά τα LSPs λέγονται MPLS TE τούνελ.



Εικόνα 32. Πρόβλημα ψαριού

Έστω ότι οι δρομολογητές R6 και R7 επιχειρούν να στείλουν δεδομένα στον R5. Αν το δίκτυο χρησιμοποιεί IP, θα ακολουθηθεί η διαδρομή R1-R2-R5. Όμως, μπορεί οι R6 και R7 να έχουν διαφορετικές πολιτικές δρομολόγησης και έτσι ο R6 θέλει να δρομολογήσει μέσω του μονοπατιού R6-R1-R2-R5 και ο R7 μέσω του R7-R1-R3-R4-R5. Σε ένα IP δίκτυο αυτό δεν είναι δυνατό να επιτευχθεί. Αν το δίκτυο όμως τρέχει MPLS, μπορούν να δημιουργηθούν δύο διαφορετικά MPLS TE τούνελ και να χρησιμοποιηθούν διαφορετικές ετικέτες. Στον R1, η εισερχόμενη ετικέτα θα υποδείξει σε ποιο τούνελ ανήκει το κάθε πακέτο και ανάλογα με την εγγραφή που έχει στην LFIB του θα το δρομολογήσει κατάλληλα. [1]

5.2 Επισκόπηση του MPLS TE

Οι θεμέλιοι λίθοι του MPLS TE είναι:

- Περιορισμοί στις ζεύξεις, δηλαδή πόσο φόρτο μπορεί να υποστηρίζει κάθε ζεύξη και ποιο TE τούνελ μπορεί να χρησιμοποιήσει.
- Διανομή της TE πληροφορίας.
- Ένας αλγόριθμος (CSPF) για να υπολογιστεί η καλύτερη διαδρομή σπό τον ingress LSR στον egress LSR.
- Ένα πρωτόκολλο σηματοδοσίας (RSVP) για να σηματοδοτήσει το TE τούνελ κατά μήκος του δικτύου.
- Ένας τρόπος να προωθηθούν τα δεδομένα στο TE τούνελ.

Ένας σημαντικός λόγος για την ύπαρξη του MPLS TE είναι η δρομολόγηση του φόρτου σύμφωνα με τους διαθέσιμους πόρους και τους περιορισμούς αυτών. Οι πόροι αυτοί είναι το εύρος ζώνης των ζεύξεων και κάποια χαρακτηριστικά τους που καθορίζει ο διαχειριστής του δικτύου. Με βάση τις διαθέσιμες TE πληροφορίες, δημιουργείται μία βάση δεδομένων που περιέχει όλες τις ζεύξεις που υποστηρίζουν το MPLS TE και τα χαρακτηριστικά τους. Χρησιμοποιώντας τις πληροφορίες αυτές, ο Constrained SPF (CSPF) υπολογίζει τη συντομότερη διαδρομή που υπακούει στους περιορισμούς των ζεύξεων (κυρίως το εύρος ζώνης). Ο CSPF είναι ένας Shortest Path First αλγόριθμος τροποποιημένος ώστε να λαμβάνει υπόψιν τους περιορισμούς των ζεύξεων.

Οι ενδιάμεσοι LSRs του LSP πρέπει να γνωρίζουν ποιες είναι οι εισερχόμενες και εξερχόμενες ετικέτες για το συγκεκριμένο τούνελ. Οι ενδιάμεσοι LSRs θα μάθουν τις ετικέτες μόνο αν συνεννοηθούν για τις ετικέτες με τον ingress LSR μέσω κάποιου πρωτοκόλλου σηματοδοσίας, όπως το RSVP. Το RSVP επίσης επιβεβαιώνει ότι όλοι οι LSRs του τούνελ μπορούν να υποστηρίξουν τη δρομολόγηση με βάση τους περιορισμούς των ζεύξεων.

Απαιτείται η χρήση ενός link state πρωτοκόλλου δρομολόγησης (όπως το OSPF) για να διαδοθούν οι περιορισμοί των ζεύξεων σε όλους τους LSRs. Οι περιορισμοί είναι μία συλλογή πληροφοριών για τους πόρους που διαθέτουν οι ζεύξεις. Οι πόροι μίας ζεύξης είναι:

- Κόστος TE
- Μέγιστο εύρρος ζώνης
- Μέγιστο εύρος ζώνης που μπορεί να δεσμευτεί
- Εύρος ζώνης που δεν έχει δεσμευτεί

Το κόστος του TE είναι μία παράμετρος που χρησιμοποιείται για να δημιουργηθεί μία TE τοπολογία διαφορετική από την IP τοπολογία. Μπορεί να είναι διαφορετικό από το κόστος του OSPF στη ζεύξη. Οι δρομολογητές θα στείλουν πληροφορίες για αυτούς τους πόρους στις εξής περιπτώσεις:

- Όταν κάποιος πόρος αλλάξει.
- Σε προκαθορισμένα χρονικά διαστήματα
- Όταν αλλάξει η κατάσταση της ζεύξης.
- Μετά από αποτυχία δημιουργίας ενός τούνελ. [1]

5.3 Constrained Shortest Path First (CSPF)

Η διαδικασία εύρεσης βέλτιστου μονοπατιού για ένα TE τούνελ διαφέρει σε ορισμένα σημεία από τη συνήθη διαδικασία ενός SPF αλγορίθμου. Πρώτον, η διαδικασία καθορισμού μονοπατιού δεν είναι σχεδιασμένη να βρίσκει την καλύτερη διαδρομή προς όλους τους δρομολογητές, αλλά μόνο προς το τερματικό σημείο του τούνελ. Επιπλέον, υπάρχουν περισσότερες παράμετροι σε κάθε κόμβο για τον υπολογισμό του κόστους που περιλαμβάνουν:

- Το εύρος ζώνης
- Τα χαρακτηριστικά της ζεύξης
- Το κόστος της ζεύξης (μπορεί να περιέχει και την παράμετρο της καθυστέρησης)

Κατά την εκτέλεση του αλγορίθμου, κάθε δρομολογητής έχει δύο λίστες: η πρώτη λίστα περιέχει τους κόμβους που είναι γνωστό πως βρίσκονται στο συντομότερο μονοπάτι προς έναν προορισμό και λέγεται λίστα PATH, και η δεύτερη λίστα περιέχει next-hop κόμβους που δεν είναι βέβαιο ότι βρίσκονται στο συντομότερο μονοπάτι προς έναν προορισμό. Η λίστα αυτή λέγεται δοκομαστική ή TENTative (για συντομία TENT).

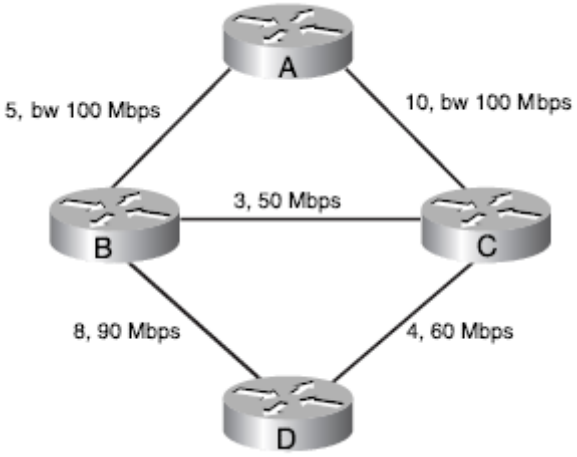
Κάθε λίστα αποτελείται από την τετράδα {κόμβος, κόστος, next hop, διαθέσιμο εύρος ζώνης}, από την οπτική του κάθε δρομολογητή. Ο κάθε δρομολογητής τρέχει τον παρακάτω αλγόριθμο:

Βήμα 1 – Στη λίστα PATH θέσε “self” με κόστος 0 και next hop self. Επίσης, θέσε ως εύρος ζώνης το N/A.

Βήμα 2 – Ονόμασε τον κόμβο που μπήκε στη λίστα PATH ως κόμβο PATH. Δες τη λίστα γειτόνων του κόμβου. Πρόσθεσε κάθε γείτονα του κόμβου στη λίστα TENT με next hop τον κόμβο PATH, εκτός και αν ο γείτονας βρίσκεται ήδη στη λίστα TENT ή στη

λίστα PATH με χαμηλότερο κόστος. Μην προσθέσεις τη ζεύξη στη λίστα TENT αν δεν πληροί τους περιορισμούς για το επιθυμητό τούνελ.

Βήμα 3 – Βρες τον γείτονα στη λίστα TENT με το μικρότερο κόστος, πρόσθεσε τον στη λίστα PATH και επανέλαβε το Βήμα 2. Αν η λίστα TENT είναι άδεια ή ο τερματικός κόμβος του τούνελ είναι στη λίστα PATH, σταμάτησε.



Εικόνα 33. Ενδεικτική τοπολογία

Στην παραπάνω τοπολογία, ο δρομολογητής A θέλει να δημιουργήσει ένα TE τούνελ προς το δρομολογητή D με ελάχιστο εύρος ζώνης στα 60 Mbps. Σε κάθε ζεύξη φαίνεται το κόστος και το διαθέσιμο εύρος ζώνης. Υπενθυμίζεται ότι κάθε λίστα αποτελείται από την τετράδα {κόμβος, κόστος, next hop, διαθέσιμο εύρος ζώνης}. Η εκτέλεση του αλγορίθμου στην πράξη γίνεται ως εξής:

Βήμα 1 – Στη λίστα PATH θέσε “A” με κόστος 0 και next hop self. Επίσης, θέσε ως εύρος ζώνης το N/A.

PATH List	TENT List
{A,0,self,N/A}	(empty)

Εικόνα 34. Βήμα 1

Βήμα 2 – Πρόσθεσε τους γείτονες του δρομολογητή A στη λίστα TENT.

PATH List	TENT List
{A,0,self,N/A}	{B,5,B,100}
	{C,10,C,100}

Εικόνα 35. Βήμα 2

Βήμα 3 – Μετακίνησε τον B από τη λίστα TENT στη λίστα PATH και βάλε τους γείτονες του B στη λίστα TENT.

PATH List	TENT List
{A,0,self,N/A}	{C,10,C,100}
{B,5,B,100}	{D,13,B,90}

Εικόνα 36. Βήμα 3

Η εγγραφή {C,8,B,50} δεν προστέθηκε στη λίστα TENT γιατί δεν πληροί το κριτήριο για ελάχιστο εύρος ζώνης.

Βήμα 4 – Μετακίνησε τον C από τη λίστα TENT στη λίστα PATH και βάλε τους γείτονες του C στη λίστα TENT. Η εγγραφή {D,14,C,60} δεν θα μπει στη λίστα TENT γιατί το κόστος προς τον D διαμέσου του B είναι χαμηλότερο απ'ότι διαμέσου του C.

PATH List	TENT List
{A,0,self,N/A}	{D,13,B,90}
{B,5,B,100}	
{C,10,C,100}	

Εικόνα 37. Βήμα 4

Βήμα 5 – Μετακίνησε τον D από τη λίστα TENT στη λίστα PATH. Πλέον, η λίστα PATH περιέχει το βέλτιστο μονοπάτι από τον A στον D και αυτό είναι το A – B – D. Σημειώνεται πως με χρήση κάποιου άλλου SPF αλγόριθμου (π.χ. του OSPF) που δε λαμβάνει υπόψιν το ελάχιστο εύρος ζώνης, το βέλτιστο μονοπάτι θα ήταν A – B – C – D.

PATH List	TENT List
{A,0,self,N/A}	
{B,5,B,100}	
{C,10,C,100}	
{D,13,B,90}	

Εικόνα 38. Βήμα 5

Υπάρχει η περίπτωση ο CSPF να θέλει να τοποθετήσει έναν κόμβο στη λίστα TENT, αλλά ο κόμβος αυτός ήδη να βρίσκεται εκεί με ίδιο κόστος. Ανά πάσα στιγμή, ένας κόμβος μπορεί να βρίσκεται μόνο μία φορά στη λίστα TENT. Αν και στους συνήθεις SPF αλγόριθμους είναι επιθυμητό να υπάρχουν πολλαπλά μονοπάτια προς τον προορισμό, ο CSPF θέλει μόνο ένα μονοπάτι προς τον προορισμό. Για να διαλέξει συνεπώς ο CSPF μεταξύ των δύο διαδρομών, θα χρησιμοποιήσει τα εξής κριτήρια:

- Διάλεξε το μονοπάτι με το μεγαλύτερο ελάχιστο διαθέσιμο εύρος ζώνης.
- Αν είναι και πάλι ίδιο, διάλεξε το μονοπάτι με το μικρότερο πλήθος κόμβων.
- Αν είναι και πάλι ίδιο, διάλεξε ένα τυχαία. [7]

5.4 Resource Reservation Protocol (RSVP)

Αφού ο CSPF υπολογίσει το βέλτιστο μονοπάτι, το μονοπάτι αυτό πρέπει να σηματοδοτηθεί σε όλο το δίκτυο για δύο λόγους: να δημιουργηθεί μία hop-by-hop αλυσίδα ετικετών που αντιπροσωπεύουν το μονοπάτι και να καταναλωθεί κάθε αναλώσιμος πόρος στη διαδρομή. Η διαδικασία πραγματοποιείται με τη χρήση του RSVP με τις επεκτάσεις του για το MPLS TE.

Το RSVP είναι ένας μηχανισμός σηματοδοσίας που χρησιμοποιείται για να δεσμεύσει πόρους σε ένα δίκτυο. Δεν είναι πρωτόκολλο δρομολόγησης, αλλά σηματοδοτεί και διατηρεί τις δεσμεύσεις πόρων στο δίκτυο. Στο MPLS TE δεσμεύει το εύρος ζώνης της ζεύξης. Έχει 3 λειτουργίες:

- Δημιουργία και διατήρηση μονοπατιού
- Αποδέσμευση μονοπατιού
- Ανίχνευση σφαλμάτων

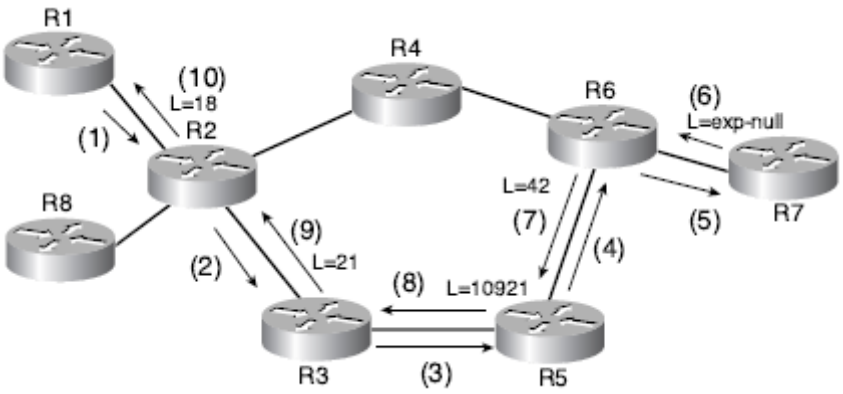
Το RSVP πρέπει ανά τακτά διαστήματα να ανανεώνει τις δεσμεύσεις πόρων στο δίκτυο. Μία αίτηση χάνεται είτε αν απορριφθεί ρητά από το δίκτυο είτε αν λήξει ο χρόνος δέσμευσης. Διαθέτει 8 διαφορετικούς τύπους μηνυμάτων:

- Path – Για τη δημιουργία και διατήρηση των δεσμεύσεων
- Resv – Αποτελεί την απάντηση στα μηνύματα Path
- PathTear – Όπως τα Path μηνύματα, αλλά για την αποδέσμευση των δεσμεύσεων από το δίκτυο
- ResvTear – Η απάντηση στα PathTear μηνύματα
- PathErr – Στέλνονται από έναν αποδέκτη ενός Path μηνύματος που εντοπίζει κάποιο σφάλμα στο μήνυμα
- ResvErr – Στέλνονται από έναν αποδέκτη ενός Resv μηνύματος που εντοπίζει κάποιο σφάλμα στο μήνυμα
- ResvConf – Προαιρετικά στέλνεται στον αποστολέα ενός Resv μηνύματος για να επιβεβαιώσει πως μία συγκεκριμένη δέσμευση εγκαταστάθηκε στο δίκτυο
- Hello – Μία επέκταση του RFC 3209 που επιτρέπει keepalives μεταξύ δύο άμεσα συνδεδεμένων RSVP γειτόνων

5.4.1 Δημιουργία μονοπατιού

Όταν ο ingress LSR υπολογίσει τη διαδρομή για ένα τούνελ, πρέπει πρώτα να ενημερώσει το δίκτυο. Για να γίνει αυτό, θα στείλει ένα Path μήνυμα στον next hop δρομολογητή, όπως προκύπτει από την εκτέλεση του CSPF. Ο δρομολογητής που στέλνει το μήνυμα Path λέγεται upstream και ο δρομολογητής που το λαμβάνει downstream. Αφού ο downstream δρομολογητής λάβει το Path μήνυμα, θα κάνει τις εξής κινήσεις: θα ελέγξει τη μορφή του μηνύματος για να εξασφαλίσει πως όλα είναι σωστά και στη συνέχεια θα ελέγξει πόσο εύρος ζώνης αιτείται ο upstream δρομολογητής μέσω του μηνύματος. Η διαδικασία αυτή είναι γνωστή ως έλεγχος αποδοχής.

Αν ο έλεγχος αποδοχής είναι επιτυχής και επιτραπεί στο Path μήνυμα να δεσμεύσει το εύρος ζώνης που θέλει, ο downstream δρομολογητής δημιουργεί ένα νέο Path μήνυμα και το στέλνει στον επόμενο δρομολογητή του τούνελ. Τα μηνύματα Path ακολουθούν αυτή την αλυσίδα μέχρι να φτάσουν στον egress LSR του τούνελ. Ο egress LSR, όπως και κάθε downstream δρομολογητής, θα κάνει έλεγχο αποδοχής στο Path μήνυμα. Μόλις αντιληφθεί πως είναι ο προορισμός του Path μηνύματος, θα απαντήσει με ένα Resv μήνυμα. Το μήνυμα Resv λειτουργεί ως επιβεβαίωση για τον upstream δρομολογητή, αλλά επίσης περιέχει την εισερχόμενη ετικέτα την οποία πρέπει να χρησιμοποιήσει ο upstream δρομολογητής για να στείλει πακέτα.



Εικόνα 39. Ανταλλαγή Path και Resv μηνυμάτων

Έστω ότι ο R1 έχει εκτελέσει τον CSPF και θέλει να δεσμεύσει εύρος ζώνης στο μονοπάτι R1→R2→R3→R5→R6→R7. Η διαδικασία που θα ακολουθήσει είναι η εξής:

1. Ο R1 στέλνει ένα Path μήνυμα στον R2, ο οποίος ελέγχει αν το μήνυμα είναι συντακτικά σωστό και έπειτα ελέγχει αν το εύρος ζώνης που αιτήθηκε ο R1 είναι όντως διαθέσιμο. Αν κάποιος έλεγχος αποτύχει, ο R2 θα στείλει μήνυμα σφάλματος στον R1.
2. Αν οι έλεγχοι είναι σωστοί, ο R2 στέλνει ένα Path μήνυμα στον R3 και ο R3 εκτελεί την ίδια διαδικασία με τον R2.
3. Ο R3 στέλνει Path μήνυμα στον R5, κάνοντας τους ίδιους ελέγχους.
4. Ο R5 στέλνει Path μήνυμα στον R6, κάνοντας τους ίδιους ελέγχους.
5. Ο R6 στέλνει Path μήνυμα στον R7, κάνοντας τους ίδιους ελέγχους.
6. Ο R7 ως egress LSR στέλνει ένα μήνυμα Resv στον R6. Το μήνυμα αυτό υποδεικνύει την ετικέτα που ο R7 θέλει να έχουν τα εισερχόμενα πακέτα. Επειδή είναι egress LSR, θα ζητήσει την implicit-null ετικέτα.
7. Ο R6 στέλνει ένα Resv μήνυμα στον R5 και υποδεικνύει πως, για το συγκεκριμένο τούνελ, θέλει τα εισερχόμενα πακέτα να έχουν την ετικέτα 42. Αυτό σημαίνει πως όταν ο R6 θα λάβει ένα πακέτο με ετικέτα 42, θα αφαιρέσει την ετικέτα και θα στείλει το πακέτο ως IP πακέτο στον R7 (λόγω της implicit null ετικέτας).
8. Ο R5 στέλνει με τη σειρά του Resv μήνυμα στον R3 ζητώντας την ετικέτα 10921. Όταν ο R5 λάβει πακέτο με αυτή την ετικέτα, θα την κάνει εναλλαγή με την ετικέτα 42 και θα στείλει το πακέτο στον R6.
9. Ο R3 θα στείλει το Resv μήνυμα στον R2 ζητώντας την ετικέτα 21.
10. Ο R2 θα στείλει το Resv μήνυμα στον R1 ζητώντας την ετικέτα 18. Στο σημείο αυτό, ο R1 έχει λάβει ένα μήνυμα Resv για το τούνελ προς τον R7 που δημιούργησε και γνωρίζει και ποια εξερχόμενη ετικέτα να χρησιμοποιήσει.

5.4.2 Διατήρηση μονοπατιού

Με μία πρώτη ματιά, η διατήρηση του μονοπατιού μοιάζει με τη διαδικασία της δημιουργίας του. Κάθε 30 δευτερόλεπτα ο ingress LSR στέλνει ένα Path μήνυμα στους downstream γείτονές του. Αν στείλει 4 Path μηνύματα στη σειρά και δεν λάβει ένα Resv σε αυτό το διάστημα, θεωρεί ότι έχει χαθεί η δέσμευση του τούνελ. Μία σημαντική παράμετρος είναι πως τα μηνύματα Path και Resv στέλνονται ανεξάρτητα και ασύγχρονα από τον ένα γείτονα στον άλλο. Ένα Resv μήνυμα που ανανεώνει μία υπάρχουσα δέσμευση ενός τούνελ δεν στέλνεται σε απόκριση ενός Path μηνύματος, όπως ένα ICMP Echo Reply θα στελνόταν σε απόκριση ενός ICMP Echo Request.

5.4.3 Αποδέσμευση μονοπατιού

Αν ένας δρομολογητής (συνήθως ο ingress) αποφασίσει πως η δέσμευση ενός μονοπατιού δεν είναι πλέον απαραίτητη για το δίκτυο, στέλνει ένα PathTear μήνυμα κατά μήκος της διαδρομής που ακολούθησε το Path μήνυμα και ένα ResvTear μήνυμα κατά μήκος της διαδρομής που ακολούθησε το Resv μήνυμα. Τα PathTear μηνύματα είναι πιο συνηθισμένα όταν ο ingress LSR αποφασίζει να αποδεσμεύσει ένα TE τούνελ. Τα ResvTear μηνύματα στέλνονται ως απόκριση στα PathTear μηνύματα ενημερώνοντας πως ο egress LSR αποδέσμευσε το τούνελ. Τα PathTear μηνύματα έχουν άμεση ισχύ. Στο παράδειγμα της εικόνας 39, αν ο R1 στείλει PathTear στον R2, ο R2 απαντάει άμεσα με ένα ResvTear και στη συνλεχεια στέλνει το δικό του downstream PathTear.

5.4.4 Ανίχνευση σφάλματος

Περιστασιακά, μπορεί να υπάρξουν λάθη στην εκτέλεση του RSVP. Τα σφάλματα αυτά επισημαίνονται από PathErr ή ResvErr μηνύματα. Ένα σφάλμα σε ένα Path μήνυμα αντιστοιχίζεται σε ένα PathErr μήνυμα, ενώ ένα σφάλμα σε ένα Resv μήνυμα αντιστοιχίζεται σε ένα ResvErr μήνυμα. Τα μηνύματα σφάλματος στέλνονται upstream προς την προέλευση του σφάλματος. [7]

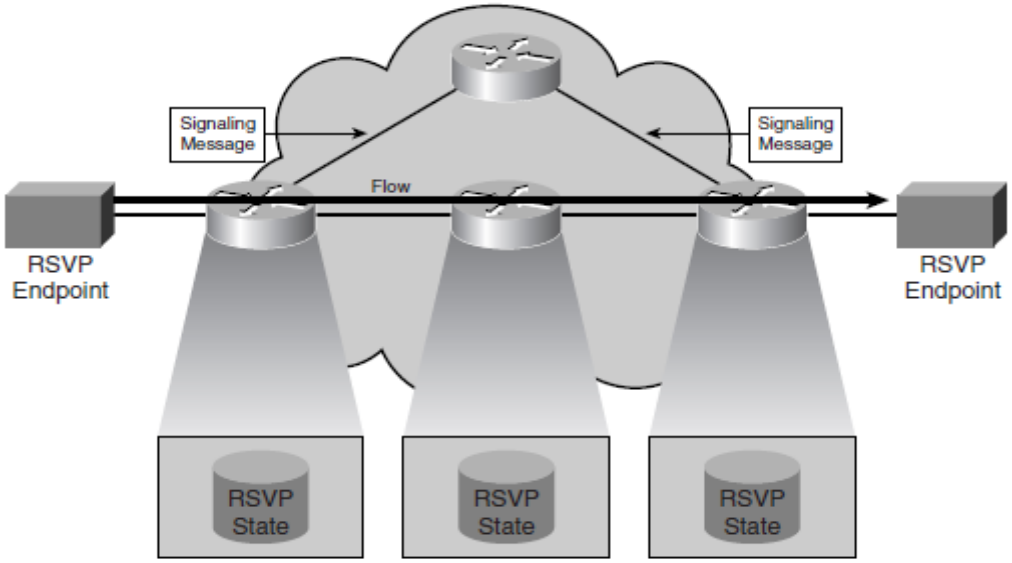
6. MPLS Quality of Service

6.1 Ενοποιημένες Υπηρεσίες (Integrated Services – IntServ)

Οι ενοποιημένες υπηρεσίες (Integrated Services – IntServ) παρέχουν μία από άκρο σε άκρο QoS λύση μέσω της άκρο προς άκρο σηματοδότησης, της διατήρησης κατάστασης (για κάθε RSVP ροή και δέσμευση) και του ελέγχου αποδοχής για κάθε δικτυακό στοιχείο. Ο όρος IntServ αναφέρεται στη συνολική QoS αρχιτεκτονική που ανέπτυξε η IETF και καθορίζει έναν αριθμό κλάσεων υπηρεσιών, σχεδιασμένων να καλύψουν τις ανάγκες διαφορετικών τύπων εφαρμογών.

Μία βασική αρχή της IntServ αρχιτεκτονικής είναι η απαίτηση για δέσμευση πόρων. Η απαίτηση αυτή περιλαμβάνει έλεγχο αποδοχής για τη διαχείριση πεπερασμένων πόρων. Οι IntServ κόμβοι πρέπει να αποφεύγουν να αποδέχονται μη εξουσιοδοτημένες αιτήσεις ή αιτήσεις που μπορεί να επηρεάσουν υπάρχουσες δεσμεύσεις. Διαφορετικοί τύποι χρηστών θα έχουν διαφορετικά δικαιώματα στη δέσμευση δικτυακών πόρων. Επιπλέον, ο φόρτος του δικτύου πρέπει να ελέγχεται ώστε να ανταποκρίνεται στις ποσοτικές προδιαγραφές των υποχρεώσεων παροχής υπηρεσιών ποιότητας των υφιστάμενων ροών. Το IntServ αφήνει την επιλογή του QoS στην εφαρμογή και όχι στο δίκτυο.

Η αρχιτεκτονική ορίζει μία ροή ως τη βασική μονάδα εξυπηρέτησης. Αυτή η γενίκευση αντιπροσωπεύει μία διακριτή ροή πακέτων που απαιτεί το ίδιο QoS. Οι ροές είναι μονόδρομες. Έχουν μία πηγή και έναν ή πολλούς προορισμούς. Το IntServ απαιτεί τη χρήση της ανά ροής κατάστασης των δικτυακών κόμβων. Αυτή η ανάγκη είναι αποτέλεσμα της διακριτότητας της ροής και της δέσμευσης πόρων με έλεγχο αποδοχής. Η δυνατότητα ύπαρξης δικτυακών κόμβων που διατηρούν κατάσταση ανά ροή αποτελεί μία σημαντική αλλαγή σε σχέση με την IP αρχιτεκτονική που το άφηνε στα τερματικά συστήματα. Η αρχιτεκτονική προτείνει τη χρήση ενός πρωτοκόλλου σηματοδοσίας για την εγκαθίδρυση και ανανέωση της κατάστασης ώστε να διατηρήσει την ευρωστία του IP πρωτοκόλλου.



Εικόνα 40. Δίκτυο με υλοποίηση IntServ

Το IntServ ορίζει μία προδιαγραφή κίνησης που λέγεται Traffic Specification (Tspec) και καθορίζει το είδος κίνησης εφαρμογής που θα εισέλθει στο δίκτυο. Το IntServ απαιτεί από δικτυακά στοιχεία όπως δρομολογητές και μεταγωγείς την εκτέλεση καθηκόντων όπως αστυνόμευση και επαλήθευση ότι η κίνηση είναι εναρμονισμένη με το Tspec. Αν η

κίνηση δεν εναρμονίζεται με τις παραμέτρους του Tspec, τα μη συμμορφούμενα πακέτα απορρίπτονται. Το IntServ ορίζει επίσης μία προδιαγραφή δέσμευσης που λέγεται Service Request Specification (Rspec), η οποία απαιτεί συγκεκριμένα επίπεδα QoS και τη δέσμευση δικτυακών πόρων. Απαιτεί από τα δικτυακά στοιχεία να εκτελέσουν καθήκοντα όπως έλεγχο αποδοχής, το οποίο ελέγχει να δει αν υπάρχουν αρκετοί πόροι για να καλύψουν μία QoS αίτηση. Το IntServ επιπλέον απαιτεί την ταξινόμηση των πακέτων, τα οποία χρειάζονται συγκεκριμένα επίπεδα QoS, καθώς και μηχανισμούς αναμονής και χρονοπρογραμματισμού. Ο συνδυασμός των Tspec και Rspec δίνει το flowspec (Flow Specification), το οποίο χρησιμοποιούν οι κόμβοι του δικτύου ως είσοδο για αποφάσεις ελέγχου-αποδοχής. Οι παράμετροι του Tspec είναι:

- Ένα token bucket (r,b) – Περιλαμβάνει το ρυθμό r του token και το μέγεθος b του token bucket.
- Ένα μέγιστο ρυθμό (p) – Η κίνηση της ροής δεν μπορεί να φθάσει με ρυθμό μεγαλύτερο του p.
- Μία ελάχιστη μονάδα αστυνόμευσης (m) – Οι κόμβοι του δικτύου συμπεριφέρονται σε πακέτα μεγέθους μικρότερου της ελάχιστης μονάδας αστυνόμευσης ως πακέτα μεγέθους m. Με τον τρόπο αυτό διευκολύνεται ο υπολογισμός του πραγματικού εύρους ζώνης που απαιτεί μία ροή.
- Ένα μέγιστο μέγεθος πακέτου (M) – Ένας κόμβος θεωρεί τα πακέτα μεγέθους μεγαλύτερου του M ως πακέτα που δε συμμορφώνονται με τις προδιαγραφές κίνησης. Τα πακέτα αυτά ενδεχομένως να μη λάβουν τις ίδιες υπηρεσίες με τα πακέτα που συμμορφώνονται στις προδιαγραφές.

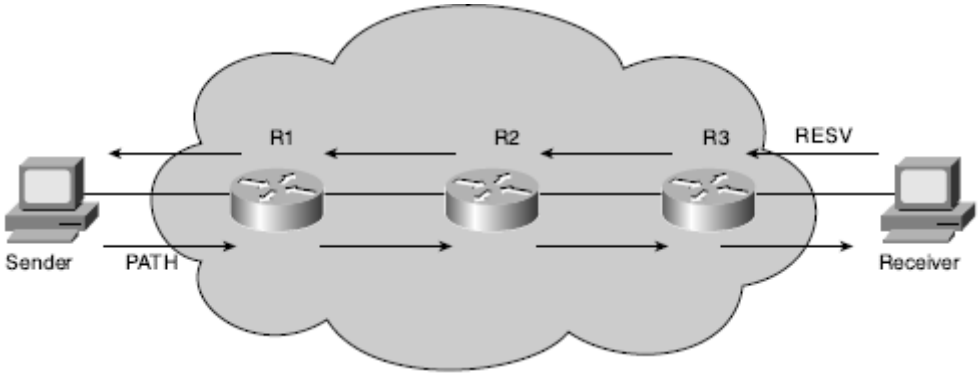
Η αρχιτεκτονική ορίζει δύο υπηρεσίες: Εγγυημένη Υπηρεσία (Guaranteed Sevice – GS) και Υπηρεσία Ελεγχόμενου Φόρτου (Controlled Load Service – CLS). Είναι συμπληρωματικές της υπηρεσίας βέλτιστης προσπάθειας (best effort) που είναι μέρος του πρωτοκόλλου IP. Το IntServ δεν εισάγει αλλαγές στη λειτουργία της υπηρεσίας βέλτιστης προσπάθειας και δεν καθιστά υποχρεωτικό ένα συγκεκριμένο σχεδιασμό για τους μηχανισμούς διαχείρισης κίνησης που υλοποιούν μία υπηρεσία.

Η εγγυημένη υπηρεσία παρέχει ροές με εγγυημένο εύρος ζώνης και καθυστέρηση. Διασφαλίζει μία ισχυρή δέσμευση στη μέγιστη καθυστέρηση αναμονής από άκρο σε άκρο για πακέτα που υπακούουν στο flowspec. Μία ροή θα λάβει εγγυημένη υπηρεσία αν όλοι οι κόμβοι κατά μήκος του μονοπατιού υποστηρίζουν την υπηρεσία. Ο παραλήπτης θα παράσχει τα Tspec και Rspec όταν ζητήσει τη GS. Το Rspec περιλαμβάνει το service rate (R) και το time slack (S). Το time slack καθορίζει την αυξητική καθυστέρηση από άκρο σε άκρο που μπορεί να γίνει ανεκτή από τον αποστολέα αν ένας κόμβος μεταβάλλει την κατανομή πόρων της ροής. Οι δικτυακοί κόμβοι πρέπει να προσεγγίσουν την υπηρεσία που μία αποκλειστική γραμμή σε αυτό το ρυθμό θα παρέχει στη ροή και οι εφαρμογές χρησιμοποιούν αυτή την πληροφορία ώστε να υπολογίσουν τη μέγιστη καθυστέρηση από άκρο σε άκρο που θα αντιμετωπίσει η ροή.

Η υπηρεσία ελεγχόμενου φόρτου προσεγγίζει τη συμπεριφορά της υπηρεσίας βέλτιστης προσπάθειας σε συνθήκες που δεν υπάρχει φόρτος. Οι δικτυακοί κόμβοι ικανοποιούν αυτή τη συμπεριφορά ακόμα και με ύπαρξη συμφόρησης. Οι εφαρμογές μπορούν να υποθέσουν ότι το δίκτυο θα παραδώσει ένα μεγάλο ποσοστό των πακέτων στον προορισμό τους και ότι η πλειονότητα των πακέτων που θα παραδοθούν θα αντιμετωπίσουν καθυστέρηση που δεν θα ξεπεράσει την ελάχιστη καθυστέρηση οποιοιδήποτε πακέτου. Η υπηρεσία αυτή υποστηρίζει τις εφαρμογές που λειτουργούν ικανοποιητικά με μία υπηρεσία βέλτιστης προσπάθειας, αλλά είναι πολύ ευαίσθητες σε συνθήκες υψηλού φόρτου.

Το RSVP είναι το IntServ πρωτόκολλο σηματοδότησης που επιτρέπει στις εφαρμογές να σηματοδοτήσουν τις QoS απαιτήσεις τους στο δίκτυο. Το δίκτυο έπειτα επιβεβαιώνει τη QoS αίτηση με απόκριση επιτυχίας ή αποτυχίας. Το RSVP μεταφέρει πληροφορίες ταξινόμησης, όπως διευθύνσεις πηγής και προορισμού και UDP αριθμούς θυρών, ώστε οι ροές που έχουν συγκεκριμένα QoS χαρακτηριστικά να μπορούν να αναγνωριστούν μέσα στο δίκτυο. Επίσης, μεταφέρει τα Tspecs, Rspecs και πληροφορίες για την επιθυμητή κλάση υπηρεσίας. Το RSVP μεταφέρει όλες αυτές τις πληροφορίες από την εφαρμογή σε κάθε δικτυακό στοιχείο κατά μήκος της διαδρομής από τον αποστολέα στον παραλήπτη.

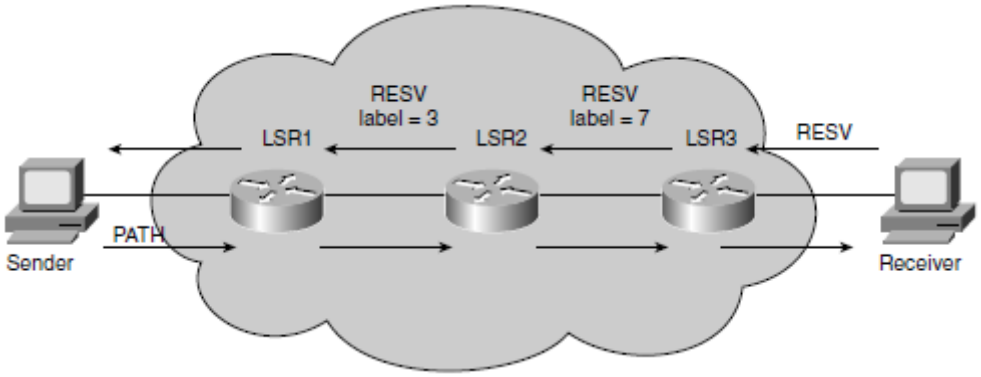
Το RSVP μεταφέρει τις πληροφορίες του χρησιμοποιώντας δύο τύπους μηνυμάτων: τα PATH και RESV μηνύματα. Τα PATH μηνύματα ταξιδεύουν από τον αποστολέα σε έναν ή περισσότερους παραλήπτες και περιλαμβάνουν τα Tspecs και τις πληροφορίες ταξινόμησης που παρέχει ο αποστολέας. Όταν ο παραλήπτης λάβει το PATH μήνυμα απαντάει στον αποστολέα με ένα RESV μήνυμα, προσδιορίζοντας τη σύνοδο για την οποία θα γίνει η δέσμευση. Περιλαμβάνει ένα Rspec που υποδεικνύει το QoS επίπεδο που απαιτείται από τον παραλήπτη. [2,8]



Εικόνα 41. Ροή PATH και RESV μηνυμάτων

6.2 Υλοποίηση του IntServ στο MPLS

Το MPLS μπορεί να ενεργοποιηθεί στους LSRs συνδυάζοντας ετικέτες με ροές που έχουν RSVP δεσμεύσεις. Πακέτα για τα οποία έχει γίνει μία RSVP δέσμευση μπορούν να θεωρηθούν ως FECs. Μία ετικέτα μπορεί να αναγνωρίσει κάθε FEC. Αντιστοιχίσεις που δημιουργούνται μεταξύ ετικετών και των RSVP ροών πρέπει να διανεμηθούν μεταξύ των LSRs.



Εικόνα 42. Ανταλλαγή RSVP μηνυμάτων

Όπως φαίνεται στην παραπάνω εικόνα, με τη λήψη ενός PATH μηνύματος, το τερματικό αποκρίνεται με ένα RESV μήνυμα. Ο LSR3 το λαμβάνει και αναθέτει μία ετικέτα (στην προκειμένη περίπτωση την ετικέτα 7). Έπειτα, στέλνει ένα νέο RESV μήνυμα, που περιέχει την ετικέτα 7, στον LSR2. Ο LSR3 θα δημιουργήσει και μία εγγραφή στην LFIB

του με την ετικέτα 7 ως ετικέτα εισόδου, ενώ ο LSR2 θα την καταγράψει στη δική του LFIB ως ετικέτα εξόδου. Ο LSR2 θα αναθέσει μία νέα ετικέτα (την 3), που θα χρησιμοποιεί ως ετικέτα εισόδου, και θα στείλει το RESV μήνυμα (που θα περιέχει τη νέα ετικέτα) στον LSR1. Καθώς τα RESV μηνύματα μαζί με τις αναθέσεις ετικετών στέλνονται upstream, δημιουργείται ένα LSP και κάθε LSR μπορεί να συσχετίσει QoS πόρους με το LSP. Σε λειτουργικό επίπεδο, όταν ο LSR2 λάβει ένα πακέτο από τον LSR1 με ετικέτα 3, θα αναζητήσει την ετικέτα στην LFIB του και θα αναγνωρίσει όλους τους QoS μηχανισμούς που σχετίζονται με το πακέτο, όπως αυτοί της αστυνόμευσης και της αναμονής. Οι L3 και L4 κεφαλίδες δε χρειάζεται να εξεταστούν.

Ο LSR1 μπορεί να συσχετίσει όλα τα πακέτα με ένα FEC και να τα αναθέσει σε ένα συγκεκριμένο LSP. Με τον τρόπο αυτό, ένα μόνο LSP μπορεί να παρέχει μία QoS εγγύηση για ένα μεγάλο σύνολο ροών κίνησης. Το RSVP μπορεί να χρησιμοποιηθεί για να μοιράσει ετικέτες ως κομμάτι της διαδικασίας δέσμευσης πόρων και να δημιουργήσει ένα LSP με δεσμευμένους πόρους. Ένα τέτοιο LSP λέγεται LSP εγγυημένου εύρους ζώνης. Αν επρόκειτο να γίνει μία δέσμευση κατά μήκος του μονοπατιού από τον LSR1 στον LSR3, ο LSR1 θα επέλεγε (συμβουλευόμενος το link state πρωτόκολλο δρομολόγησης) ένα μονοπάτι προς τον LSR3 πριν στείλει ένα PATH μήνυμα προς τον κόμβο LSR3. Το μονοπάτι αυτό θα έπρεπε να πληροί τις απαιτήσεις περιορισμού του εύρους ζώνης σε όλες τις ζεύξεις του για να μπορεί να υποστηρίξει τη δέσμευση.

Η παραπάνω ανά ροή προσέγγιση δεν είναι ιδιαίτερα επεκτάσιμη και οδηγεί σε πολύπλοκες υλοποιήσεις. Για το λόγο αυτό, η IETF προτείνει την προσέγγιση του IP Precedence, υιοθετώντας ένα αθροιστικό μοντέλο ροών ταξινομώντας ποικίλες ροές σε αθροιστικές κλάσεις και παρέχοντάς τους το κατάλληλο QoS. Τα πακέτα ταξινομούνται στις παρυφές του δικτύου σε 1 από 8 διαφορετικές κλάσεις. Αυτό επιτυγχάνεται θέτοντας 3 bits προτεραιότητας στο πεδίο TOS της IP κεφαλίδας. Πακέτα χαμηλότερης προτεραιότητας απορρίπτονται για χάρη πακέτων υψηλότερης προτεραιότητας, όταν υπάρχει συμφόρηση στο δίκτυο. Μόλις τα πακέτα επισημανθούν με τα κατάλληλα bits προτεραιότητας, κάθε κόμβος του δικτύου κατά μήκος της διαδρομής γνωρίζει το επίπεδο προτεραιότητας του πακέτου και μπορεί να εφαρμόσει προνομιακή προώθηση για τα πακέτα υψηλότερης προτεραιότητας. Σημειώνεται, πως η προσέγγιση του IP Precedence επιτρέπει μόνο την ταξιμόμηση των πακέτων σε επίπεδα προτεραιότητας και δεν μπορεί να καθορίσει διαφορετική προτεραιότητα απόρριψης για πακέτα ίδιου επιπέδου προτεραιότητας. Για παράδειγμα, αν το SMTP και το Telnet έχουν ανατεθεί στην ίδια κλάση, δεν υπάρχει τρόπος για το IP Precedence να απορρίψει τα Telnet πακέτα σε περίπτωση συμφόρησης προς όφελος των SMTP πακέτων. [2]

Number	Name
0	Routine
1	Priority
2	Immediate
3	Flash
4	Flash override
5	Critical
6	Internet control
7	Network control

Εικόνα 43. Τιμές IP Precedence

6.3 Διαφοροποιημένες Υπηρεσίες (Differentiated Services – DiffServ)

Το μοντέλο Διαφοροποιημένων Υπηρεσιών (Differentiated Services – DiffServ) χωρίζει την κίνηση σε ένα μικρό αριθμό κλάσεων και αναθέτει πόρους ανά κλάση. Το μοντέλο αυτό είναι παρόμοιο με την προσέγγιση του IP Precedence. Μία ομάδα 6 bits που λέγεται DSCP επισημαίνει την κλάση του πακέτου. Τα DSCP bits ανήκουν στο ToS πεδίο της IP κεφαλίδας. Οι κόμβοι του δικτύου επιθεωρούν τα DSCP bits ώστε να αναγνωρίσουν την κλάση του πακέτου και να αναθέσουν τους πόρους σύμφωνα τις τοπικά ορισμένες πολιτικές. [2]

IP Precedence	DSCP
IP precedence 0	DSCP 0
IP precedence 1	DSCP 8
IP precedence 2	DSCP 16
IP precedence 3	DSCP 24
IP precedence 4	DSCP 32
IP precedence 5	DSCP 40
IP precedence 6	DSCP 48
IP precedence 7	DSCP 56

Εικόνα 44. DSCP bits

Η DiffServ αρχιτεκτονική ορίζει μία ιεραρχία που πηγαίνει από μία συσκευή, σε ένα δίκτυο, σε μία ομάδα δικτύων. Μία ομάδα κόμβων με κοινή DiffServ υλοποίηση σχηματίζει έναν τομέα. Οι κόμβοι μέσα σε έναν τομέα εκτελούν παρόμοιες πολιτικές και ορισμούς υπηρεσιών. Ένας τομέας συνήθως βρίσκεται κάτω από έναν μοναδικό διαχειριστικό έλεγχο. Ένα σύνολο συνεχών τομέων σχηματίζει μία DiffServ περιοχή. Οι τομείς μέσα στην περιοχή πρέπει να είναι σε θέση να παρέχουν DiffServ σε κίνηση που διασχίζει τους διαφόρους τομείς στην περιοχή.

Οι δύο τύποι κόμβων που βρίσκονται μέσα σε ένα DiffServ τομέα είναι οι συνοριακοί και οι εσωτερικοί κόμβοι. Οι συνοριακοί κόμβοι αλληλεπιδρούν με το εξωτερικό του τομέα και διασφαλίζουν την κατάλληλη ταξινόμηση της κίνησης. Οι συνοριακοί και οι εσωτερικοί κόμβοι υλοποιούν τις τοπικές πολιτικές εξυπηρέτησης σύμφωνα με την επισήμανση του πακέτου. Η κίνηση εισέρχεται σε έναν τομέα σε έναν ingress συνοριακό κόμβο και εξέρχεται από αυτόν μέσω ενός egress συνοριακού κόμβου. Ο ingress κόμβος συνήθως εκτελεί την ταξινόμηση της κίνησης, ενώ οι συνοριακοί κόμβοι λειτουργούν τόσο ως ingress όσο και ως egress κόμβοι, καθώς η διαφοροποίηση της κίνησης είναι επιθυμητή για ροές και προς τις δύο κατευθύνσεις.

Η διαδικασία ταξινόμησης και οροθέτησης της κίνησης αναγνωρίζει την κίνηση που θα λάβει διαφοροποιημένη υπηρεσία και διασφαλίζει ότι ανταποκρίνεται σε μία σύμβαση παροχής υπηρεσιών. Οι εξωτερικοί κόμβοι που συνδέονται στον DiffServ τομέα έχουν συμφωνήσει σε κάποιους όρους παροχής υπηρεσιών που η αρχιτεκτονική ορίζει ως SLA. Ο συνοριακός κόμβος εφαρμόζει την SLA χρησιμοποιώντας τη διαδικασία ταξινόμησης και οροθέτησης της κίνησης. Χρησιμοποιεί ένα συνδυασμό ταξινόμησης, επισήμανσης, μέτρησης, διαμόρφωσης και αστυνόμευσης των πακέτων ώστε να διασφαλιστεί πως τα πακέτα είναι συμβατά με τους όρους της SLA.

Η ταξινόμηση της κίνησης είναι η πρώτη ενέργεια που εκτελεί ο συνοριακός κόμβος στην κίνηση που εισέρχεται στον τομέα. Ο συνοριακός κόμβος εξετάζει το πακέτο και, σύμφωνα με τους όρους της SLA, θα υλοποιήσει την κατάλληλη πράξη (π.χ. μέτρηση ή αστυνόμευση). Το τελικό αποτέλεσμα της ταξινόμησης του πακέτου είναι ο τοπικός συσχετισμός κάθε πακέτου με μία κλάση. Ο συνοριακός κόμβος θέτει όρους στα πακέτα

σύμφωνα με την SLA μετά την ταξινόμησή τους. Περιλαμβάνει ένα συνδυασμό των παρακάτω μηχανισμών: μέτρηση, επισήμανση, αστυνόμευση ή διαμόρφωση. Η διαδικασία χρησιμοποιεί το αποτέλεσμα της ταξινόμησης του πακέτου ως είσοδο. Η διαδικασία οροθέτησης μπορεί να διαφέρει από κλάση σε κλάση. Οι SLAs συνήθως καθορίζουν ένα όριο στην ποσότητα της κίνησης που κάθε συνοριακός κόμβος θα πάρει για κάθε κλάση. Στις περιπτώσεις αυτές, ο συνοριακός κόμβος μετράει την κίνηση και παίρνει μία απόφαση επισήμανσης, απόρριψης ή ενταμίευσης για το πακέτο.

Η διαδικασία ταξινόμησης και οροθέτησης της κίνησης μπορεί να συμβεί σε διαφορετικά σημεία της διαδρομής ενός πακέτου. Σε γενικές γραμμές, οι συνοριακοί κόμβοι εκτελούν αυτές τις ενέργειες. Ωστόσο, η αρχιτεκτονική δεν αποκλείει εσωτερικούς κόμβους από τη διαδικασία, αν αυτό είναι αναγκαίο. [8]

Τα δικτυακά στοιχεία (ή hops) κατά μήκος της διαδρομής εξετάζουν την τιμή του πεδίου DSCP και προσδιορίζουν το QoS που απαιτείται από το πακέτο. Αυτό είναι γνωστό ως συμπεριφορά ανά hop (per-hop behavior – PHB). Κάθε κόμβος του δικτύου έχει έναν πίνακα που αντιστοιχίζει το DSCP πεδίο του πακέτου με το PHB που καθορίζει πώς να αντιμετωπιστεί το πακέτο. Τα PHBs είναι καλά ορισμένες συμπεριφορές που εφαρμόζονται στα πακέτα. Μία συλλογή πακέτων που έχουν την ίδια τιμή στο DSCP πεδίο και διέρχονται από έναν κόμβο του δικτύου σε μία συγκεκριμένη κατεύθυνση, λέγεται Αθροιστική Συμπεριφορά (Behavior Aggregate – BA). Το PHB αναφέρεται στη συμπεριφορά χρονοπρογραμματισμού, αναμονής, αστυνόμευσης ή διαμόρφωσης ενός κόμβου πάνω σε οποιοδήποτε πακέτο που ανήκει σε μία BA. Υπάρχουν τέσσερες προτυποποιημένες PHB υλοποιήσεις:

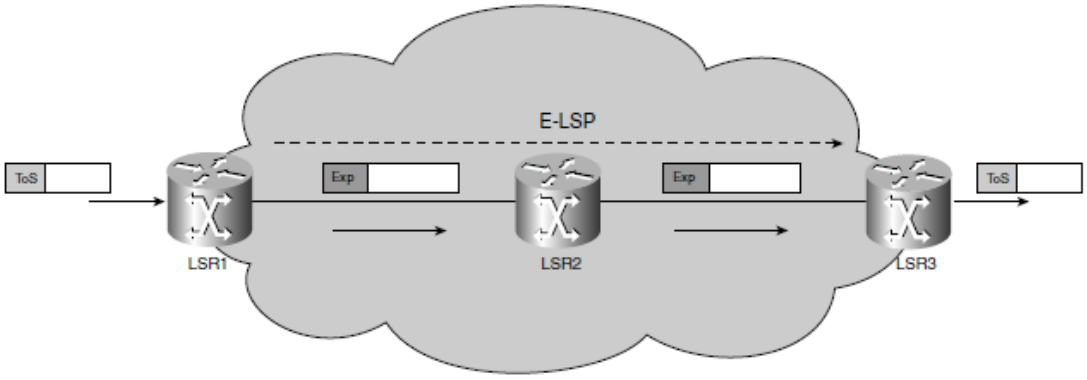
- Default PHB: Τα πακέτα με DSCP τιμή 000000 λαμβάνει την υπηρεσία βέλτιστης προσπάθειας από έναν κόμβο συμβατό με το DiffServ.
- Class-Selector PHB: Για να διατηρηθεί η συμβατότητα με το IP Precedence, ορίστηκαν οι DSCP τιμές της μορφής *xxx*000 (όπου x είναι 0 ή 1). Τα PHBs που συσχετίζονται με αυτές τις τιμές διατηρούν την ίδια συμπεριφορά προώθησης με τους κόμβους που χρησιμοποιούν το IP Precedence. Για παράδειγμα, πακέτα με DSCP τιμή 101000 (IP Precedence 101) έχουν υψηλότερη προτεραιότητα σε σχέση με τα πακέτα με τιμή 011000 (IP Precedence 011).
- Expedited Forwarding (EF) PHB: Έχει ως αποτέλεσμα ταχεία προώθηση με ελάχιστη καθυστέρηση και χαμηλές απώλειες. Τα πακέτα αυτά έχουν υψηλότερη προτεραιότητα από τα υπόλοιπα. Το EF PHB στο DiffServ μοντέλο παρέχει χαμηλές απώλειες, χαμηλή καθυστέρηση και jitter και υπηρεσίες εγγυημένου εύρους ζώνης. Εφαρμογές όπως το VoIP απαιτούν τέτοιες εγγυήσεις. Η DSCP τιμή είναι 101110.
- Assured Forwarding (AF) PHB: Η DSCP τιμή των AF πακέτων καθορίζει την AF κλάση και την πιθανότητα απόρριψης των πακέτων. Πακέτα με διαφορετικές πιθανότητες απόρριψης μέσα στην ίδια AF κλάση απορρίπτονται βάσει των τιμών προτεραιότητας απόρριψης. [2]

6.4 Υλοποίηση του DiffServ στο MPLS

Οι LSRs του MPLS δεν εξετάζουν τα περιεχόμενα της IP κεφαλίδας και την τιμή του DSCP πεδίου όπως απαιτείται από το DiffServ. Η MPLS κεφαλίδα έχει ένα πεδίο 3 bits που λέγεται Exp και είχε δημιουργηθεί αρχικά για πειραματική χρήση. Το πεδίο αυτό υποστηρίζει 8 διαφορετικές τιμές και χρησιμοποιείται για να υποστηρίζει το MPLS ως και 8 DiffServ κλάσεις.

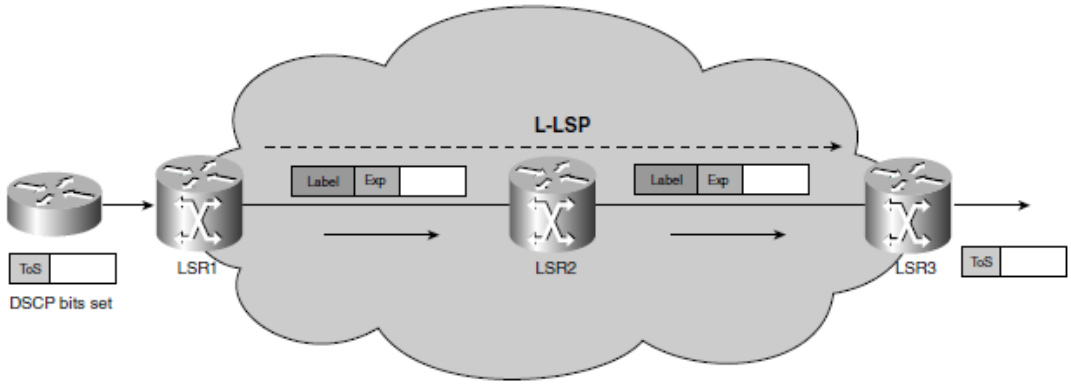
Τα IP Precedence bits ή τα τρία πρώτα bits του DSCP πεδίου αντιγράφονται στο Exp πεδίο της MPLS κεφαλίδας στα άκρα του δικτύου. Κάθε LSR του LSP αντιστοιχεί τα Exp

bits σε ένα PHB. Ο πάροχος μπορεί επίσης να θέσει το Class of Service (CoS) ενός MPLS πακέτου σε μία διαφορετική τιμή. Αυτή η δυνατότητα επιτρέπει στον πάροχο να θέτει το MPLS Exp πεδίο αντί να αντιγράφει την τιμή του IP Precedence πεδίου. Έτσι η IP κεφαλίδα μένει άθικτη και διαθέσιμη για χρήση από τον πελάτη. Το CoS δεν αλλάζει καθώς το πακέτο διέρχεται από το MPLS δίκτυο. Τα LSPs που δημιουργούνται με τον τρόπο αυτό είναι γνωστά ως E-LSPs και μπορούν να υποστηρίξουν έως και οκτώ PHBs ανά LSP.



Εικόνα 45. MPLS E-LSP

Αν χρειάζονται περισσότερα από οκτώ PHBs στο MPLS δίκτυο, χρησιμοποιούνται τα L-LSPs (Label LSPs), στην περίπτωση των οποίων το PHB του LSR συνάγεται από την ετικέτα. Μόνο ένα PHB ανά L-LSP είναι δυνατό. Στην περίπτωση του DiffServ AF, πακέτα που μοιράζονται ένα κοινό PHB μπορούν να ομαδοποιηθούν σε ένα FEC το οποίο μπορεί να ανατεθεί σε ένα LSP. Αυτό ονομάζεται PHB κλάση προγραμματισμού. Οι πιθανότητες απόρριψης κωδικοποιούνται στα Exp bits της κεφαλίδας.

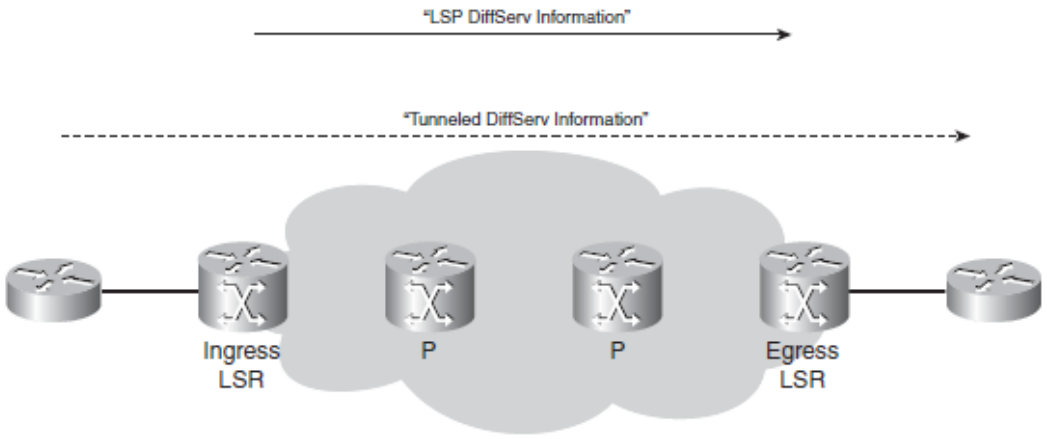


Εικόνα 46. MPLS L-LSP

Τα E-LSPs είναι πιο αποτελεσματικά από τα L-LSPs, καθώς είναι παρόμοια με το σύνηθες DiffServ μοντέλο. Ένα E-LSP μπορεί να υποστηρίξει πολλαπλά PHBs, περιορίζοντας το συνολικό αριθμό LSPs που δημιουργούνται και άρα εξοικονομούνται και ετικέτες. Η MPLS DiffServ υλοποίηση ασχολείται αποκλειστικά με την υποστήριξη των PHBs που ικανοποιούν τις QoS απαιτήσεις της κίνησης που εξυπηρετεί. Επιπλέον, όσο και αν αυξηθεί το πλήθος των LSPs, το δίκτυο μπορεί να μεγαλώσει χωρίς να χρειαστεί να επιφέρει σημαντικές αλλαγές στο DiffServ σχεδιασμό του. [2,8]

Το MPLS δίνει τη δυνατότητα να μην αντιγράφεται η τιμή του Exp πεδίου στο IP Precedence ή στο DSCP πεδίο, όσες φορές κι αν αλλάξει η τιμή του πεδίου Exp. Στην πράξη, αυτό επιτρέπει στον διαχειριστή του MPLS δικτύου να μεταφέρει χωρίς αλλαγές την QoS τιμή του IP πακέτου διαμέσου του δικτύου. Όσες φορές κι αν αλλάξουν τα Exp bits, τα IP Precedence ή DSCP bits του IP πακέτου θα διατηρηθούν. Έτσι, το MPLS

δίκτυο χρησιμοποιείται σαν τούνελ για τη DiffServ τιμή του IP πακέτου και το προφανές πλεονέκτημα αυτής της προσέγγισης είναι ότι το MPLS δίκτυο μπορεί να έχει διαφορετική QoS υλοποίηση από το δίκτυο των πελατών που συνδέονται σε αυτό. Ορίζονται τρία μοντέλα τούνελ για τη διατήρηση της DiffServ πληροφορίας, των οποίων οι διαφορές εντοπίζονται μόνο στους ingress και egress LSRs. Στην παρακάτω εικόνα, “Tunneled DiffServ Innformation” θεωρείται η QoS πληροφορία που μεταφέρεται μέσω του MPLS δικτύου χωρίς να υπόκειται σε αλλαγές, ενώ “LSP DiffServ Information” είναι η QoS πληροφορία που χρησιμοποιούν οι LSRs στο MPLS δίκτυο, δηλαδή τα Exp bits.



Εικόνα 47. DiffServ τούνελ

Το πρώτο μοντέλο λέγεται pipe model. Σε αυτό ισχύουν οι εξής κανόνες:

- Τα Exp bits μπορεί να αντιγραφούν από το IP Precedence ή να ρυθμιστούν στον ingress LSR.
- Στους P δρομολογητές, τα Exp bits μεταφέρονται από την εισερχόμενη στη εξερχόμενη ετικέτα.
- Στον egress LSR, η προώθηση του πακέτου βασίζεται στο MPLS PHB (δηλαδή τα Exp bits) και τα Exp bits δεν μεταφέρονται στο IP Precedence.

Το δεύτερο μοντέλο είναι παρόμοιο με το pipe model, λέγεται short pipe model και έχει μία διαφορά όσον αφορά τον τρόπο προώθησης. Το τρίτο bullet αλλάζει ως εξής:

- Στον egress LSR, η προώθηση του πακέτου βασίζεται στο IP PHB (δηλαδή το IP Precedence) και τα Exp bits δεν μεταφέρονται στο IP Precedence.

Το τρίτο μοντέλο, το uniform model, διαφέρει αρκετά από τα δύο προηγούμενα:

- Τα Exp bits προέρχονται αναγκαστικά από το IP Precedence στον ingress LSR.
- Στους P δρομολογητές, τα Exp bits μεταφέρονται από την εισερχόμενη στη εξερχόμενη ετικέτα.
- Στον egress LSR τα Exp bits πρέπει να μεταφερθούν στο IP Precedence.

Συμπερασματικά, προκύπτει πως στο μοντέλο αυτό ένα πακέτο ανήκει πάντα στην ίδια QoS κλάση. [1]

6.5 Μηχανισμοί διαχείρισης κίνησης

Η υλοποίηση του QoS στηρίζεται σε ένα σύνολο μηχανισμών διαχείρισης κίνησης. Οι μηχανισμοί αυτοί επιτρέπουν στους δικτυακούς κόμβους να αποφεύγουν και να διαχειρίζονται τη συμφόρηση. Οι μηχανισμοί αυτοί έχουν εφαρμογή τόσο σε DiffServ όσο και σε IntServ αρχιτεκτονικές.

- Ταξινόμηση Κίνησης

Οι δικτυακοί κόμβοι συνήθως ταξινομούν την κίνηση πριν εφαρμόσουν τους μηχανισμούς διαχείρισης της κίνησης. Η συνολική κίνηση που διέρχεται από έναν κόμβο συνδυάζει κίνηση με διαφορετικές QoS απαιτήσεις. Στις περιπτώσεις αυτές, οι κόμβοι πρέπει να ταξινομήσουν την κίνηση για να παρέχουν το αναμενόμενο επίπεδο διαφοροποίησης. Συνήθως, η ταξινόμηση της κίνησης αφορά stateless επιθεώρηση των κεφαλίδων των πακέτων.

- Σήμανση Κίνησης

Η σήμανση της κίνησης περιλαμβάνει την ανάθεση μίας νέας τιμής σε ένα πεδίο σχετικό με το QoS στην κεφαλίδα του πακέτου. Η σήμανση αυτή συσχετίζει το πακέτο με μία κλάση ή μία πιθανότητα απόρριψης. Το DiffServ στηρίζεται στις σημάνσεις των πακέτων για να υποδείξει το PHB κάθε πακέτου.

- Αστυνόμευση Κίνησης

Η αστυνόμευση της κίνησης είναι μία κοινά χρησιμοποιημένη προσέγγιση για έλεγχο του ρυθμού. Σε διάφορες περιπτώσεις, ένας κόμβος μπορεί να χρειαστεί να ελέγξει την ποσότητα μίας συγκεκριμένης ροής. Μετριέται η κίνηση και συγκρίνεται με ένα προκαθορισμένο προφίλ κίνησης. Το αποτέλεσμα της σύγκρισης καθορίζει το τι θα συμβεί στο πακέτο. Οι βασικές ενέργειες είναι μετάδοση, επισήμανση ή απόρριψη του πακέτου. Ο μηχανισμός χρησιμοποιεί ένα token bucket με δύο παραμέτρους: το ρυθμό των tokens και το μέγεθος του κουβά. Ο ρυθμός των tokens καθορίζει το ρυθμό με τον οποίο φθάνουν νέα tokens. Το μέγεθος του κουβά καθορίζει το μέγιστο αριθμό tokens που μπορεί να χωρέσει. Ο αλγόριθμος είναι αρκετά απλός: Προστίθενται διαρκώς tokens στον κουβά με τον δεδομένο ρυθμό. Ελέγχει αν ο κουβάς έχει B tokens όταν φθάνει ένα πακέτο μεγέθους B. Ένα θετικό ή αρνητικό αποτέλεσμα προκαλεί δύο διαφορετικές ενέργειες. Αν το αποτέλεσμα είναι θετικό (ο κουβάς έχει περισσότερα από B tokens), τότε απομακρύνονται B tokens και εκτελείται μία προαποφασισμένη ενέργεια. Αλλιώς, εκτελεί μία εναλλακτικλη ενέργεια. Ο αλγόριθμος έχει μία μικρή διαφορά για το DiffServ. Χρησιμοποιούνται δύο token buckets, C και E. Τα tokens φθάνουν στον κουβά C με ρυθμό CIR. Τα tokens υπερχειλίζουν στον κουβά E όταν γεμίζει ο C.

- Διαμόρφωση Κίνησης

Η διαμόρφωση κίνηησς είναι ένας ακόμα μηχανισμός ελέγχου του ρυθμού. Μετράει την κίνηση και συγκρίνει τη μέτρηση με ένα προφίλ. Το αποτέλεσμα της σύγκρισης καθορίζει κατά πόσον ο διαμορφωτής θα καθυστερήσει ένα πακέτο ή αν θα επιτρέψει περαιτέρω επεξεργασία. Συνεπώς, η διαμόεφωση απαιτεί την ενταμίευση των πακέτων που υπερβαίνουν το καθορισμένο προφίλ. Η διαμόρφωση επιτρέπει σε έναν κόμβο να απορροφήσει ριπές πακέτων σε μία ροή, αλλά μπορεί να οδηγήσει σε απώλεια πακέτων αν η ροή ξεπεράσει δραστικά το προκαθορισμένο προφίλ. Μπορεί να χρησιμοποιηθεί ένα token bucket ως διαμορφωτής, με τον αλγόριθμο να παραμένει ο ίδιος.

- Διαχείριση Συμφόρησης

Η διανομή ενταμιευτή και ο προγραμματισμός της κίνησης είναι δύο δημοφιλείς μηχανισμοί διαχείρισης της κίνησης. Για παάδειγμα, έστω ένας κόμβος που μετάγει κίνηση σε μία διεπαφή σε ρυθμό που υπερβαίνει τη χωρητικότητα της διεπαφής τη δεδομένη στιγμή. Όταν συμβεί η συμφόρηση, η διεπαφή θα διαχειριστεί την πλεονάζουσα κίνηση ενταμιεύοντας τη και ένας προγραμματιστής θα αποφασίσει πώς να εξυπηρετήσει την ουρά. Ένας κόμβος μπορεί να δημιουργήσει πολλαπλές ουρές σε ένα σημείο συμφόρησης. Κάθε ουρά μπορεί να λάβει διαφορετική διανομή ενταμιευτών και εύρους ζώνης. Αυτή η διανομή πόρων, μαζί με την πειθαρχία στον προγραμματισμό

μεταξύ των ουρών, παρέχει διαφορετικά χαρακτηριστικά καθυστέρησης, jitter και απώλειας για την κίνηση στις διάφορες ουρές. Συνήθεις τρόποι προγραμματισμού της ουράς αποτελούν οι FIFO, Weighted Fair Queuing (WFQ), Round Robin. [8]

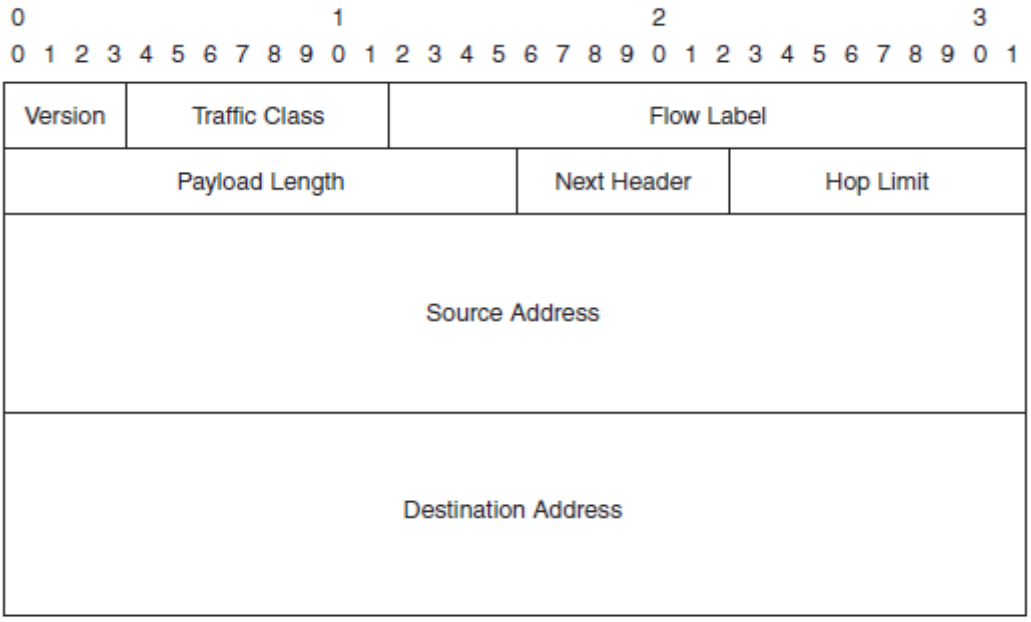
7. IPv6 και MPLS

7.1 Εισαγωγή στο IPv6

Τα IP δίκτυα είχαν τεράστια επιτυχία τις τελευταίες δύο δεκαετίες, με προεξέχον παράδειγμα το Internet. Παρότι το τρέχον IP πρωτόκολλο, το IPv4, έχει ανταπεξέλθει στο χρόνο, μπορεί να βελτιωθεί. Ο βασικός λόγος για τη βελτίωση αυτή είναι η εκρηκτική αύξηση των IP δικτύων στον κόσμο. Οι IPv4 διευθύνσεις είναι μήκους 32 bits. Σύντομα, έγινε αντιληπτό πως με τη ραγδαία αύξηση του Internet οι RIRs (Regional Internet Registries) θα στέρευαν από διαθέσιμα IP δίκτυα. Για το επόμενο IP πρωτόκολλο, το IPv6, η IETF δημιούργησε μία IP διεύθυνση μήκους 128 bits. Το IPv4 παρείχε περίπου 4.3 δις διευθύνσεις, ενώ το IPv6 περίπου 3.4×10^{38} διευθύνσεις, αριθμός ικανός να καλύψει τις ανάγκες για πολλά χρόνια.

Η ανάγκη για ένα νέο IP πρωτόκολλο για τις μεγαλύτερες διευθύνσεις έχει μειωθεί λόγω των τεχνικών που έχουν χρησιμοποιηθεί, όπως οι ιδιωτικές IP διευθύνσεις και το NAT (Network Address Translation) για σύνδεση στο Internet. Εναλλακτικά, μπορεί να χρησιμοποιηθεί το DHCP για να αναθέτει διευθύνσεις δυναμικά όταν χρειάζεται και να τις ανακτά όταν αποσυνδέονται οι χρήστες.

Η μεγαλύτερη IP διεύθυνση είναι η πιο προφανής βελτίωση από το IPv4 στο IPv6. Ωστόσο, το IPv6 έχει φέρει και άλλες αλλαγές, λιγότερο προφανείς. Πρώτον, η IPv6 κεφαλίδα είναι πιο απλή. Κάποια από τα IPv4 πεδία έχουν παραλειφθεί, με βασικότερο το πεδίο του checksum, κάτι που έχει μειώσει το κόστος της προώθησης ενός πακέτου μέσω ενός δρομολογητή. Δεύτερον, το IPv6 έχει σταθερό μέγεθος κεφαλίδας. Τρίτον, η IETF έχει κάνει την ασφάλεια στο IPv6 βασικό θέμα, Επιπλέον, έχει προστεθεί η δυνατότητα της ετικέτας ροής, που επιτρέπει σε ένα δρομολογητή να αναγνωρίσει τη ροή στην οποία ανήκει το πακέτο εξετάζοντας μόνο την IP κεφαλίδα.



Εικόνα 48. IPv6 κεφαλίδα

Υπάρχουν τρεις κατηγορίες IPv6 διευθύνσεων: unicast, anycast και multicast. Η unicast διεύθυνση είναι μία IP διεύθυνση σε μία διεπαφή. Η anycast διεύθυνση είναι μία διεύθυνση που έχει ανατεθεί σε πολλαπλές διεπαφές σε πολλαπλούς κόμβους. Ένα πακέτο που προορίζεται για αυτή την IPv6 διεύθυνση στέλνεται στην πλησιέστερη διεπαφή με αυτή την anycast διεύθυνση. Ως πλησιέστερη θεωρείται η συντομότερη διαδρομή σύμφωνα με το πρωτόκολλο δρομολόγησης. Μία multicast IP διεύθυνση είναι

μία διεύθυνση που ανατίθεται σε πολλαπλές διεπαφές σε πολλούς κόμβους. Ένα πακέτο που αποστέλλεται σε αυτή την IPv6 διεύθυνση στέλνεται σε όλες τις διεπαφές με αυτή τη multicast διεύθυνση. Η broadcast διεύθυνση δε διασώθηκε στη μετάβαση από το IPv4 στο IPv6 και αντικαταστάθηκε από τις multicast διευθύνσεις.

Η 128 bits διεύθυνση γράφεται ως εξής: x:x:x:x:x:x:x:x, με κάθε x να αναπαριστά μία δεκαεξαδική τιμή 16 bits για ένα σύνολο 128 bits IPv6 διεύθυνσης. Ένα παράδειγμα μίας τέτοιας IPv6 διεύθυνσης είναι 2001:DB08:7654:3210:FEDC:BA98:7654:3210. Η γραφική αναπαράσταση των IPv6 διευθύνσεων είναι αρκετά μεγάλη συγκρινόμενη με την αντίστοιχη του IPv4. Οι IPv6 διευθύνσεις μπορεί να μικρύνουν αντικαθιστώντας πολλαπλές ομάδες των 16 bits από μηδενικά με το ::. Αυτό ωστόσο μπορεί να γίνει μόνο μία φορά στη διεύθυνση. Επίσης, μπορεί να παραλειφθούν τα αρχικά μηδενικά σε κάθε πεδίο 16 bits.

Ο τύπος της IPv6 διεύθυνσης (παρόμοια με τις IPv4 διευθύνσεις) υποδεικνύεται από τα αρχικά bits της διεύθυνσης. Αν τα τρία πρώτα bits είναι 001, η IPv6 διεύθυνση είναι global unicast (παρόμοια με τις κανονικές IPv4 διευθύνσεις). Η unicast διεύθυνση είναι 0:0:0:0:0:0:0:1 λέγεται loopback διεύθυνση και ένας κόμβος μπορεί να τη χρησιμοποιήσει για να στείλει ένα IPv6 πακέτο στον εαυτό του. Η διεύθυνση 0:0:0:0:0:0:0:0 είναι η απροσδιόριστη διεύθυνση και δεν πρέπει ποτέ να ανατίθεται σε έναν κόμβο. Μία IPv6 διεύθυνση με τα πρώτα 10 bits να είναι 1111111010 είναι μία link-local unicast διεύθυνση. Οι διευθύνσεις αυτές περιορίζονται σε μία ζεύξη και δεν δρομολογούνται ποτέ στο Internet. Χρησιμοποιούνται μόνο για να ανακαλύψουν τους γείτονες και για την ανταλλαγή των updates των πρωτοκόλλων δρομολόγησης. Μία link-local διεύθυνση ανήκει στο εύρος FE80::/64. Μία multicast διεύθυνση ξεκινάει με 11111111 και άρα ανήκει στο εύρος FF::/8. Τέλος, οι anycast διευθύνσεις δεν έχουν κάποια συγκεκριμένη μορφή σύνταξης.

Άλλες διαφορές που εισάγει το IPv6 είναι:

- ICMPv6
- Ανακαλυψη γείτονα
- Ανακάλυψη δρομολογητή
- Stateless autoconfiguration
- DHCPv6
- Path MTU ανακάλυψη
- Νέες DNS λειτουργίες

Το ICMP, γνωστό από το IPv4, προσφέρει λειτουργίες όπως το ping και τα redirects. Τα ίδια υποστηρίζει και το ICMPv6 μαζί με την ανακάλυψη γείτονα και δρομολογητή. Η ανακάλυψη δρομολογητή κάνει τους δρομολογητές να στέλνουν διαφημίσεις των δρομολογητών ώστε ένας IPv6 κόμβος να μπορεί να ανακαλύψει αυτόματα έναν δρομολογητή στην τοπική ζεύξη. Η ανακάλυψη γείτονα στο IPv6 είναι ένας τρόπος ώστε οι IPv6 κόμβοι να ανακαλύπτουν άλλους IPv6 κόμβους στη ζεύξη. Το stateless autoconfiguration είναι μία δυνατότητα όπου οι κόμβοι μπορούν να λαμβάνουν IPv6 διευθύνσεις από το δρομολογητή. Το DHCPv6 είναι το αντίστοιχο του DHCP για το IPv4 και επιτρέπει σε έναν κόμβο να λάβει IP διεύθυνση από έναν εξυπηρετητή. Το DHCP επιτρέπει περισσότερο έλεγχο από το stateless autoconfiguration, αλλά είναι πιο πολύπλοκο. Η ανακάλυψη path MTU επιτρέπει την αποφυγή του κατακερματισμού από τους δρομολογητές. Τέλος, το DNS αλλάζει για να λειτουργεί στο IPv6, προσθέτοντας νέες εγγραφές. [1]

7.2 Το IPv6 στο MPLS

Λόγω της τεράστιας επιτυχίας του MPLS VPN, οι περισσότεροι πάροχοι τρέχουν MPLS στο δίκτυό τους. Αν ο πάροχος έχει πελάτες συνδεδεμένους στο δίκτυο που θέλουν να τρέξουν IPv6 και πρέπει να μεταφέρει IPv6 διαμέσου του δικτύου του, η προφανής λύση είναι να τρέξει IPv6 στους LSRs του. Ωστόσο, αυτή η προσέγγιση έχει δύο μειονεκτήματα. Πρώτον, ο πάροχος πρέπει να ενεργοποιήσει ένα νέο πρωτόκολλο στους δρομολογητές του, μετατρέποντάς τους σε dual-stack. Δεύτερον, οι υπόλοιποι πελάτες δεν θα μεταβούν άμεσα στο IPv6 και συνεπώς το IPv4 και το IPv6 θα πρέπει να τρέχουν παράλληλα για αρκετό χρόνο.

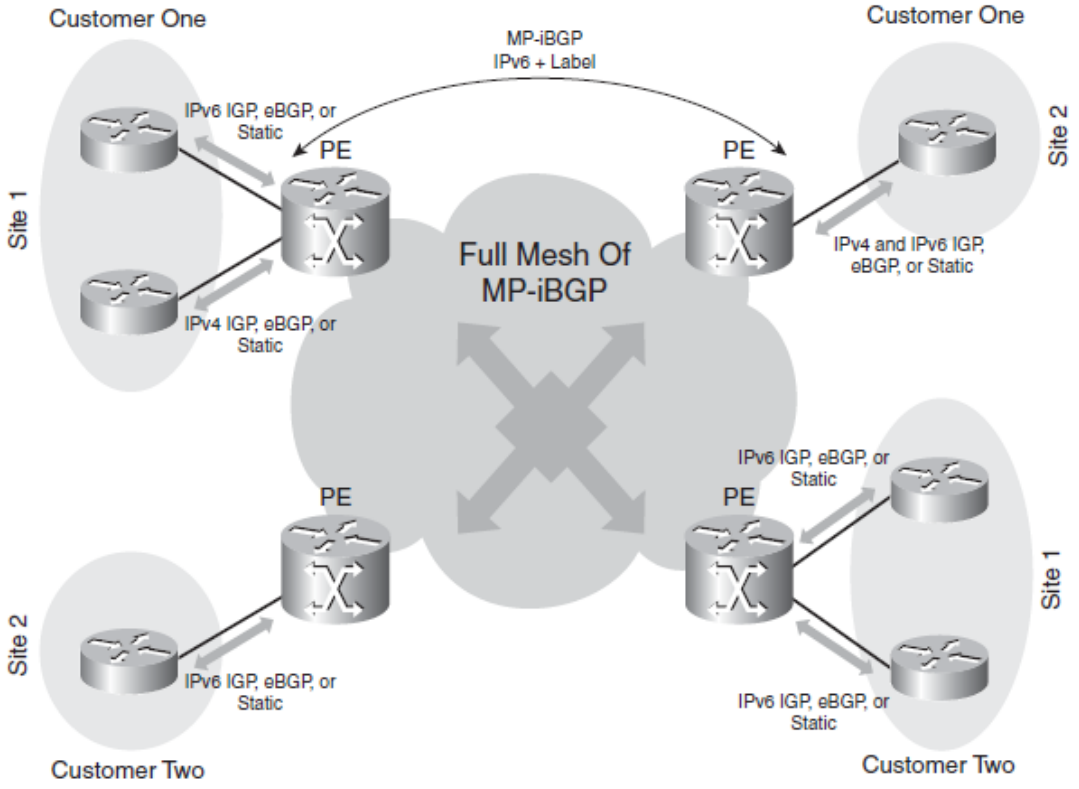
Αν ο πάροχος θέλει να τρέξει MPLS για IPv6, χρειάζεται την υποστήριξη του LDP το οποίο δεν έχει υλοποιηθεί ακόμα για το IPv6. Ωστόσο, στα MPLS δίκτυα τα MPLS πακέτα μπορεί να μεταφέρουν IPv6 πακέτα χωρίς να χρειάζεται οι P δρομολογητές να τρέχουν IPv6. Οι λύσεις 6PE και 6VPE βασίζονται σε αυτό. Μία ακόμα μέθοδος για τη μεταφορά IPv6 μέσω του MPLS κορμού είναι το AToM (Any Transport over MPLS). Με τη λύση αυτή, το φορτίο του MPLS είναι ένα L2 πλαίσιο. Στους ingress LSRs, ανατίθενται ετικέτες στα πλαίσια και μεταφέρονται έπειτα στο MPLS δικτύο μέσω ενός εικονικού κυκλώματος.

Και οι τρεις λύσεις έχουν το πλεονέκτημα ότι οι P δρομολογητές δε χρειάζεται να τρέχουν IPv6 καθώς μετάγουν μόνο πακέτα με ετικέτα. Συνεπώς, οι λύσεις αυτές είναι πιο δημοφιλείς από το να τρέχει IPv6 το δίκτυο κορμού. Η λύση του AToM έχει δύο μειονεκτήματα συγκριτικά με τα 6PE και 6VPE. Το πρώτο είναι πως το MPLS φορτίο απαρτίζεται από πλαίσια και όχι IPv6 πακέτα, με συνέπεια να πρέπει να μεταφέρεται μία επιπρόσθετη L2 κεφαλίδα. Το δεύτερο είναι πως τα εικονικά κυκλώματα είναι point-to-point, ενώ τα 6PE και 6VPE είναι any-to-any.

Μία τελευταία μέθοδος για τη μεταφορά IPv6 μέσω του MPLS δικτύου είναι η λύση του MPLS VPN. Στην περίπτωση του MPLS VPN, το IPv4 μεταφέρεται μέσα σε VPNs από το MPLS δίκτυο. Για να μεταφερθεί IPv6 κίνηση πάνω από το IPv4, οι CE δρομολογητές χρειάζονται τούνελ ανάμεσα τους, άρα πρέπει να είναι dual-stack. Οι δρομολογητές αυτοί θα είναι οι μόνοι που θα τρέχουν IPv6, καθώς οι PE δρομολογητές βλέπουν μόνο IPv4 πακέτα να έρχονται από τους CE δρομολογητές. Εν ολίγοις, το πλεονέκτημα είναι ότι το MPLS VPN είναι ήδη υλοποιημένο στα περισσότερα δίκτυα παρόχων και οι PE και P δρομολογητές δε χρειάζεται να τρέχπυν IPv6. Το μειονέκτημα είναι ότι οι CE δρομολογητές πρέπει να έχουν τούνελ μεταξύ τους και να προσθέσουν μία επιπλέον IPv4 κεφαλίδα. [1]

7.3 Μεταφορά IPv6 μέσω MPLS

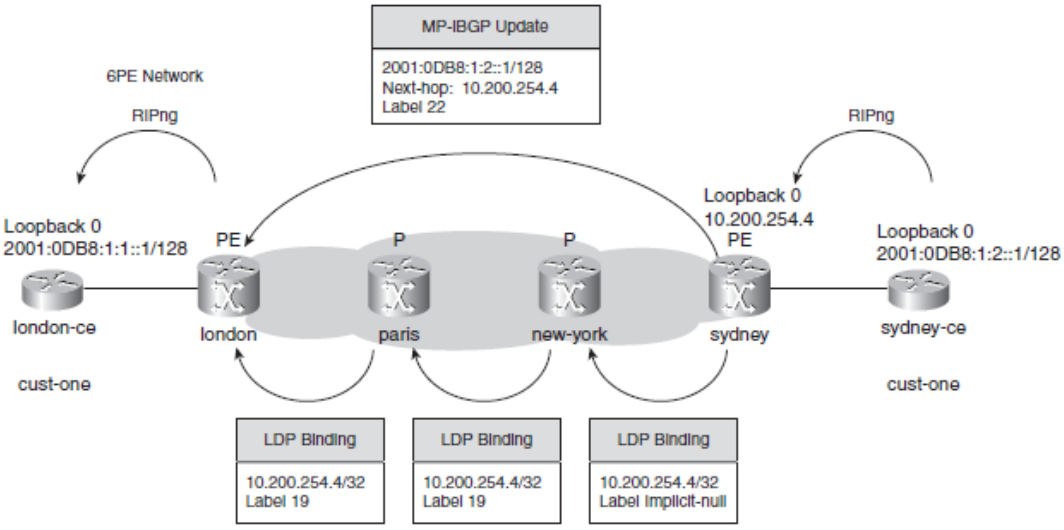
Η δυνατότητα απευθείας μεταφοράς IPv6 πακέτων πάνω από το MPLS δίκτυο ονομάζεται 6PE. Οι PE δρομολογητές είναι dual-stack, δηλαδή τρέχουν και IPv4 και IPv6. Οι CE δρομολογητές που τρέχουν IPv6 συνδέονται στον PE δρομολογητή μέσω μίας κανονικής διεπαφής. Η διανομή των IPv6 πληροφοριών δρομολόγησης γίνεται μέσω του MP-iBGP. Ταυτόχρονα, το MP-iBGP διανέμει την ετικέτα που θα χρησιμοποιηθεί για το συγκεκριμένο IPv6 πρόθεμα. Η ετικέτα αυτή βοηθάει να αναγνωριστεί το IPv6 πακέτο στον egress PE. Ο egress PE θα ψάξει την ετικέτα στην LFIB και θα τη χρησιμοποιήσει για να προωθήσει το πακέτο προς τον egress CE.



Εικόνα 49. Δίκτυο 6PE

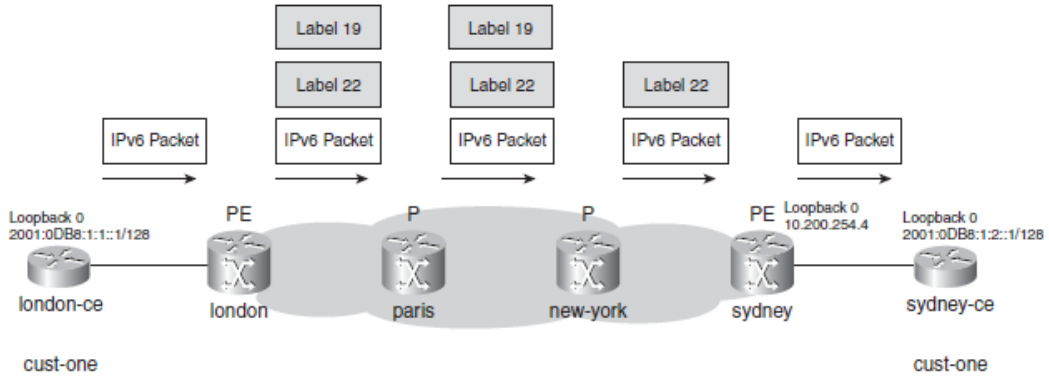
Στην παραπάνω εικόνα απεικονίζεται ένα MPLS δίκτυο που τρέχει 6PE. Οι PE δρομολογητές συνδέονται με τους CE δρομολογητές, κάποιοι εκ των οποίων τρέχουν IPv6, ενώ κάποιοι άλλοι IPv4. Οι PE δρομολογητές τρέχουν ένα full mesh MP-iBGP για IPv6. Οι iBGP σύνοδοι διανέμουν τα IPv6 προθέματα και τις αντίστοιχες MPLS ετικέτες.

Στην παρακάτω εικόνα βλέπουμε ένα δίκτυο που έχει μόνο δύο PEs να τρέχουν 6PE. Ο sydney PE δρομολογητής στέλνει το IPv6 πρόθεμα 2001:DB8:1:2::1/128 με την ετικέτα 22 στον London PE με χρήση του MP-iBGP. Όλοι οι PE και P δρομολογητές τρέχουν ένα IGP και LDP. Για να προωθηθούν τα πακέτα στο CE δρομολογητή που συνδέεται στον sydney PE πρέπει να διανεμηθεί με ετικέτα στο MPLS δίκτυο το BGP next hop 10.200.254.4/32.



Εικόνα 50. 6PE και διανομή ετικετών

Στην επόμενη εικόνα φαίνεται πως λειτουργεί η προώθηση πακέτων. Στη στοίβα των ετικετών κάθε πακέτου υπάρχουν δύο ετικέτες: η LDP ετικέτα στην κορυφή και η BGP ετικέτα από κάτω. Η LDP ετικέτα αφορά το BGP next hop του egress PE. Αυτό το next hop κωδικοποιείται ως μία IPv6 διεύθυνση που περιέχει μία IPv4 διεύθυνση του egress PE δρομολογητή. Έτσι, στον ingress PE δρομολογητή, η LDP ετικέτα σχετίζεται με την IPv4 διεύθυνση, ενώ η BGP ετικέτα είναι η ετικέτα που ο απομακρυσμένος PE έστειλε για το IPv6 πρόθεμα.



Εικόνα 51. Προώθηση πακέτων στο 6PE

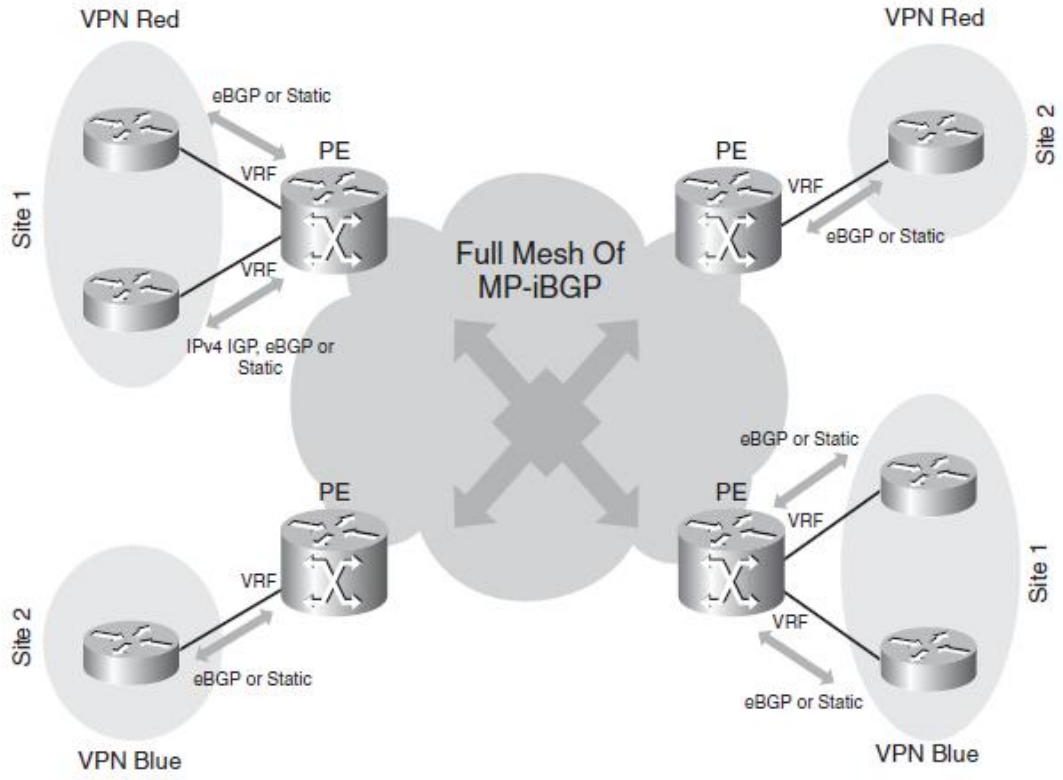
Ένα πλεονέκτημα του 6PE είναι πως οι P δρομολογητές δε χρειάζεται να τρέχουν IPv6, άρα η λύση του 6PE είναι εύκολα υλοποιήσιμη πάνω από ένα υπάρχον MPLS δίκτυο. Ένα δεύτερο πλεονέκτημα είναι πως τα IPv6 πακέτα παίρνουν απευθείας ετικέτα χωρίς επιπλέον κεφαλίδα. Ένας λόγος της υλοποίησης του 6PE είναι πως πολλοί πάροχοι έχουν ήδη MPLS δίκτυο κορμού λόγω της δημοφιλίας του MPLS VPN, και η λειτουργία του 6PE είναι παρόμοια με του MPLS VPN. Ορισμένες ομοιότητες είναι:

- Απαιτείται full mesh MP-iBGP.
- Πρέπει να υπάρχει ένα IGP για IPv6 ή eBGP ή στατική δρομολόγηση μεταξύ του PE και του CE.
- Τα IPv6 πακέτα έχουν δύο ετικέτες στη στοίβα τους. [1]

7.4 Μεταφορά IPv6 μέσω MPLS VPN

Η MPLS VPN λύση για το IPv6 (γνωστή ως 6VPE) είναι παρόμοια με τη λειτουργία του MPLS VPN για το IPv4. Η εμφανής διαφορά μεταξύ του 6PE και του 6VPE είναι ότι στο 6VPE, τα IPv6 προθέματα των πελατών ανήκουν σε ένα VPN και διαχωρίζονται πλήρως από τα προθέματα άλλων πελατών που συνδέονται στο MPLS VPN δίκτυο. Το 6VPE έχει τα εξής χαρακτηριστικά:

- Έχει ένα MPLS δίκτυο κορμού που τρέχει ένα IPv4 πρωτόκολλο δρομολόγησης και το LDP.
- Οι PE δρομολογητές τρέχουν IPv6.
- Ένα full mesh από MP-iBGP συνόδους υπάρχει μεταξύ των PE δρομολογητών και χρησιμοποιείται για να διανείμει τα IPv6 προθέματα και τις ετικέτες που τους αντιστοιχούν.
- Τα IPv6 πακέτα μεταφέρονται κατά μήκος του MPLS δικτύου με δύο ετικέτες: μία LDP ετικέτα στην κορυφή και μία VPN ετικέτα από κάτω.
- Οι PE και CE δρομολογητές έχουν ένα IPv6 ορωτόκολλο δρομολόγησης μεταξύ τους.



Εικόνα 52. Δίκτυο 6VPE

Σημειώνεται πως οι PE δρομολογητές πρέπει να τρέχουν IPv4 και IPv6, ενώ οι P δρομολογητές δε χρειάζεται να τρέχουν IPv6. [1]

8. Εξέλιξη του MPLS

8.1 GMPLS

Το Generalized MPLS βασίζεται στο MPLS TE αλλά οι επιπρόσθετες επεκτάσεις του το καθοστούν ικανό να δουλεύει με νεότερες πλατφόρμες. Οι πλατφόρμες αυτές δεν είναι απλά δρομολογητές ή ATM μεταγωγείς που έχουν ενεργοποιημένο το MPLS, αλλά τρέχουν MPLS στο επίπεδο ελέγχου, ενώ απουσιάζει από το επίπεδο δεδομένων. Αυτό γίνεται γιατί οι καινούριες αυτές πλατφόρμες δεν μετάγουν πακέτα με ετικέτες ή ATM κελιά, αλλά μετάγουν μήκη κύματος, κανάλια διαίρεσης χρόνου και φυσικές πόρτες ή ίνες.

Οι θεμέλιοι λίθοι του GMPLS είναο οι ίδιοι με του MPLS TE στο επίπεδο ελέγχου: το IPv4, ένα link state πρωτόκολλο δρομολόγησης και το RSVP. Ένα νέο πρωτόκολλο που χρειάζεται το GMPLS είναι το Link Management Protocol (LMP) που αναπτύχθηκε για ευκολότερη διαχείριση των ζεύξεων. Το GMPLS χρειάζεται το LMP γιατί οι νέες αυτές πλατφόρμες μπορεί να έχουν έναν τεράστιο αριθμό μηκών κύματος μεταξύ τους, κάτι που κάνει τη διαχείριση των ζεύξεων περίπλοκη. Το LMP αναλαμβάνει τη διαχείριση και επαλήθευση της συνδεσιμότητας των ζεύξεων. Το GMPLS διανέμει τους δικτυακούς περιορισμούς των φυσικών μέσων σε όλες τις πλατφόρμες που συμμετέχουν στο GMPLS. Οι περιορισμοί αυτοί μπορούν να χρησιμοποιηθούν για τη δημιουργία LSPs στο δίκτυο που ενδεχομένως να αποκλίνουν από το συντομότερο μονοπάτι. Οι περιορισμοί διαφέρουν από αυτούς του MPLS TE, καθώς διαφέρουν και τα φυσικά μέσα. Περιλαμβάνονται διαφορετικοί περιορισμοί χωρητικότητας ζεύξης, προστασίας και αποκατάστασης. [1]

8.2 Multicast με μεταγωγή ετικέτας

Τα IP και VPN multicast δε χρησιμοποιούν MPLS LSPs και αυνεπώς δεν μπορούν να χρησιμοποιηθούν οι δυνατότητες του MPLS στο multicast, όπως η ταχεία αναδρομολόγηση για προστασία ζεύξεων, κόμβων και εύρους ζώνης. Ωστόσο, αν multicast κίνηση μεταφερθεί μέσω MPLS LSPs, το MPLS TE και η ταχεία αναδρομολόγηση μπορούν να χρησιμοποιηθούν για να προστατεύσουν ταυτόχρονα τη multicast και unicast κίνηση.

Ολοένα και περισσότεροι πάροχοι επιλέγουν MPLS και GMPLS για τα next-gen δίκτυα τους. Έχουν την απαίτηση να μεταφέρεται όλη η κίνηση – unicast και multicast – μέσω LSPs. Έχοντας κοινό επίπεδο δεδομένων κερδίζουν εξοικονόμηση λειτουργικών εξόδων και αύξηση στην αποτελεσματικότητα του δικτύου όσον αφορά τη χρησιμοποίηση πόρων και εύρους ζώνης. Τα LSPs μπορεί να είναι πέρα από point-to-point και point-to-multipoint ή multipoint-to-multipoint. Τόσο το RSVP όσο και το LDP μπορούν να δημιουργήσουν τα point-to-multipoint ή multipoint-to-multipoint LSPs για αποτελεσματική multicast μετάδοση. [9]

8.3 Δυναμικά κρυπτογραφημένα VPNs

Τα MPLS VPNs παρέχουν full mesh συνδεσιμότητα, καθώς και διαχωρισμό της κίνησης από τον πελάτη A στον πελάτη B. Ωστόσο, δεν παρέχουν κρυπτογράφηση της κίνησης. Τα δυναμικά κρυπτογραφημένα VPNs δίνουν τη δυνατότητα να κρυπτογραφηθεί οποιαδήποτε κίνηση μεταξύ των CEs. Οι πληροφορίες ασφαλείας είτε τροφοδοτούνται στατικά είτε ανταλλάσσονται μέσω του BGP. Αφού μαθευτεί η γειτνίαση ασφαλείας, δημιουργείται η κρυπτογράφηση για κίνηση που προορίζεται προς κάποιο συγκεκριμένο πρόθεμα. Με τον τρόπο αυτό δημιουργείται μία ευέλικτη μέθοδος για την αντιμετώπιση των απαιτήσεων κρυπτογράφησης σε ένα MPLS VPN δίκτυο. [9]

8.4 Βελτιώσεις ασφαλείας

Μελλοντικές βελτιώσεις όσον αφορά την ασφάλεια αφορούν τον καλύτερο χειρισμό των ετικετών, την ανίχνευση, απόκριση και αποφυγή DoS επιθέσεων, την πιστοποίηση συνόδων και ομότιμων και την αποφυγή λανθασμένων προωθήσεων που οφείλονται σε αλλαγές κατάστασης. Η ασφάλεια αποτελεί την πλέον σημαντική πτυχή στη δημιουργία αξιόπιστων VPN υπηρεσιών. [9]

8.5 Προσομοίωση κυκλώματος

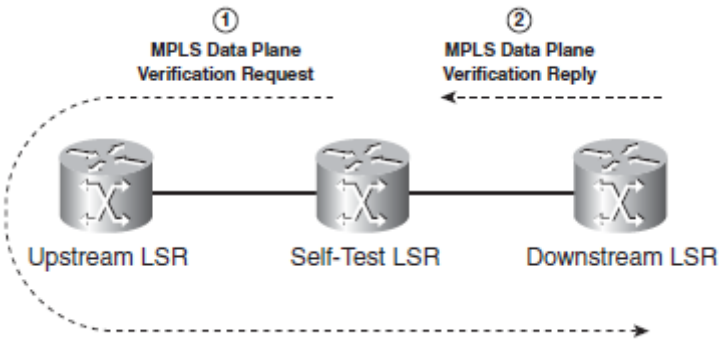
Υπάρχει ακόμα ένας μεγάλος αριθμός ιδιωτικών γραμμών και παλαιού τύπου συσκευών που χρησιμοποιούν υπηρεσίες πολυπλεξίας διαίρεσης χρόνου (TDM). Συνεπώς, είναι λογική η μεταφορά TDM μέσω MPLS. Το πλεονέκτημα της μεταφοράς τέτοιων υπηρεσιών στο MPLS είναι πως ένα κοινό δίκτυο – το MPLS δίκτυο – μεταφέρει και τα IP και τα TDM δεδομένα. Με την προσομοίωση TDM κυκλώματος, η TDM ροή bit μεταφέρεται στο MPLS δίκτυο μέσω ενός MPLS pseudowire. Το δύσκολο κομμάτι αφορά την προσομοίωση του κυκλώματος, όπως την ανάκτηση του ρολογιού και τις διαδικασίες για σηματοδότηση συναγερμού.

Μία άλλη περιοχή ανάπτυξης είναι η προσομοίωση κυκλωμάτων SDH/SONET (Synchronous Digital Hierarchy/Synchronous Optical Network) στο MPLS. Το SONET και το SDH είναι πρότυπα που περιγράφουν μία ψηφιακή ιεραρχία για τη μεταφορά συγχρονισμένων δεδομένων σε δίκτυα ινών. Το SONET είναι ευρέως διαδεδομένο στις ΗΠΑ, ενώ το SDH στην Ευρώπη. [1]

8.6 Εντοπισμός σφαλμάτων

Το Bidirectional Forwarding Detection – BFD είναι ένα νέο, ελαφρύ πρωτόκολλο που δεν εξαρτάται από το φυσικό μέσο και ανιχνεύει σφάλματα στο επίπεδο δεδομένων μεταξύ δύο συσκευών. Έχει σχεδιαστεί ώστε να είναι ανεξάρτητο τόσο από το φυσικό μέσο όσο και από το πρωτόκολλο δρομολόγησης και να εντοπίζει γρήγορα τα σφάλματα δεδομένων. Το BFD εντοπίζει γρήγορα όλα τα σφάλματα μεταξύ δρομολογητών αντί να βασίζεται στον μηχανισμό hello των πρωτοκόλλων δρομολόγησης. Το BFD εντοπίζει επίσης σφάλματα στο επίπεδο δεδομένων για MPLS LSPs. Μία BFD σύνοδος δημιουργείται μεταξύ του ingress και του egress LSR και στέλνονται BFD πακέτα ελέγχου κατά μήκος. Έτσι, το BFD παρακολουθεί την κατάσταση του LSP και εντοπίζει τα σφάλματα στα δεδομένα.

Ένα πρόβλημα με το MPLS είναι ότι συχνά το επίπεδο ελέγχου λειτουργεί σωστά ενώ το επίπεδο δεδομένων όχι. Το LDP, RSVP ή BGP μπορεί να υποδεικνύουν τις σωστές εισερχόμενες και εξερχόμενες ετικέτες, αλλά το επίπεδο προώθησης (δηλαδή η LFIB) μπορεί να κάνει λάθος προώθηση, με αποτέλεσμα το πακέτο να δρομολογείται λανθασμένα ή να απορρίπτεται. Μία λύση για το πρόβλημα αυτό είναι ένας LSR να ελέγχει τις δικές του πληροφορίες επιπέδου δεδομένων. Η λειτουργία αυτή λέγεται LSR Self-Test.



Εικόνα 53. LSR Self-Test

Ο LSR που κάνει τον έλεγχο στέλνει στον upstream γείτονά του ένα ειδικό πακέτο που λέγεται MPLS Data Plane Verification Request. Το πακέτο αυτό έχει την εισερχόμενη στοίβα ετικετών την οποία ο LSR που πραγματοποιεί τον έλεγχο περιμένει να έχουν τα πακέτα που έρχονται από τον upstream γείτονα του. Ο upstream LSR τότε προωθεί το MPLS πακέτο στον downstream γείτονα του self-test LSR. Ο self-test LSR προωθεί κανονικά τις ετικέτες στο πακέτο και έτσι εξετάζει την ορθότητα του LSP στο επίπεδο δεδομένων. Με άλλα λόγια, ο self-test LSR πραγματοποιεί τις κανονικές ενέργειες (εισαγωγή, εξαγωγή, εναλλαγή) στο πακέτο και το προωθεί στον downstream γείτονα του. Ο downstream γείτονας σταματάει το πακέτο και στέλνει ένα MPLS Data Plane Verification Reply στον self-test LSR. Το MPLS Data Plane Verification Reply πακέτο υποδεικνύει τη διεπαφή του downstream γείτονα στην οποία παραλήφθηκε το πακέτο και τη στοίβα ετικετών. Ο self-test LSR μπορεί να επαληθεύσει τις πληροφορίες αυτές. [1]

8.7 Συνεργασία ATM – MPLS

Η μετεγκατάσταση των L2 VPNs από ATM δίκτυα σε νέα MPLS L2 VPNs έχει αρκετές προκλήσεις όσον αφορά την τροφοδοσία και διαχείριση. Κατά τη φάση της μετεγκατάστασης κάποιοι κόμβοι πρέπει να είναι στο παλιό και κάποιοι στο νέο δίκτυο. Η διαδικασία θα ήταν απλούστερη αν τα δύο δίκτυα είχαν κοινό επίπεδο ελέγχου. Τα δύο επίπεδα ελέγχου πρέπει να διασυνεργαστούν για να δημιουργηθούν L2 συνδέσεις και VCs στα ATM και MPLS δίκτυα. Αυτό απαιτεί μία λειτουργία διασυνεργασίας στα άκρα των δικτύων που να “μεταφράζει” το ATM στο MPLS. [9]

8.8 Προσαρμοζόμενα αυτοθεραπευόμενα δίκτυα

Ο εντοπισμός σφαλμάτων είναι εξαιρετικά σημαντικός για να παρθούν αποφάσεις επαναδρομολόγησης και σύγκλισης σε μία τοπολογία. Με τις ενσωματωμένες δυνατότητες διαχείρισης του MPLS, η ζωτικότητα των επιπέδων ελέγχου και δεδομένων μπορεί να ελέγχεται και να κάνει τα ακόλουθα:

- Αποφάσεις ταχείας αναδρομολόγησης.
- Αυτόματη επίκληση ίχνους για τον εντοπισμό του σφάλματος.
- Απομόνωση του σφάλματος επαναδρομολογώντας γύρω από αυτό.

Αυτό δημιουργεί ένα ισχυρό δίκτυο που “αυτοθεραπεύεται” όταν εντοπίζει σφάλματα. Ένας έλεγχος ζωτικότητας των LSPs και TE τούνελ βοηθά στη διάκριση των σφαλμάτων του επιπέδου ελέγχου από τα σφάλματα του επιπέδου δεδομένων. Η αυτόματη κλήση των ιχνών βοηθά στην εύρεση της τοποθεσίας του σφάλματος στο δίκτυο και αλλάζοντας τα κόστη της δρομολόγησης, μπορεί να επιτευχθεί η

επαναδρομολόγηση της κίνησης γύρω από το σφάλμα μέσα σε σύντομο χρονικό διάστημα. [9]

8.9 Διάδοση του MPLS

Το MPLS δε χρησιμοποιείται πλέον αποκλειστικά από παρόχους, αλλά όλο και περισσότερο από δίκτυα επιχειρήσεων που έχουν μεγαλύτερη διάμετρο δικτύου ή ιδιαίτερες ανάγκες. Επιπλέον, το MPLS έχει ήδη μετακινηθεί από ττον πυρήνα του δικτύου στα άκρα του. Ένα χαρακτηριστικό παράδειγμα είναι οι προεκτάσεις των LSPs στους CE δρομολογητές για ευκολότερη υλοποίηση του QoS σε MPLS VPN δίκτυα.

Παρότι τα MPS VPN αυτόνομα συστήματα ακόμη διασυνδέονται μέσω IP ως επί το πλείστον, στο εγγύς μέλλον, όλο και περισσότερα MPLS VPN δίκτυα θα διασυνδέονται μέσω του MPLS και τα πακέτα θα αποστέλλονται με ετικέτα προς το άλλο αυτόνομο σύστημα. Η διασύνδεση μεταξύ των MPLS δικτύων δεν θα περιοριστεί στα MPLS VPN δίκτυα, αλλά θα χρησιμοποιηθεί και στη μεταγωγή AToM ή IPv6 κίνησης από τον έναν πάροχο στον άλλο. [1]

9. MPLS Προσομοιώσεις στο OMNeT++

9.1 Τι είναι το OMNeT++

Το OMNeT++ είναι ένα αντικειμενοστρεφές framework προσομοίωσης δικτύου διακριτών γεγονότων. Έχει μία γενική αρχιτεκτονική, ώστε να μπορεί να χρησιμοποιηθεί σε ποικίλα προβλήματα όπως:

- Μοντελοποίηση ενσύρματων και ασύρματων δικτύων επικοινωνιών
- Μοντελοποίηση πρωτοκόλλων
- Μοντελοποίηση δικτύων αναμονής
- Μοντελοποίηση πολυεπεξεργαστών και άλλων κατανεμημένων συστημάτων υλικού
- Επικύρωση αρχιτεκτονικών υλικού
- Αξιολόγηση απόδοσης σύνθετων συστημάτων λογισμικού
- Σε γενικές γραμμές, μοντελοποίηση και προσομοίωση κάθε συστήματος όπου ταιριάζει η προσέγγιση των διακριτών γεγονότων και μπορεί να αντιστοιχιστεί σε οντότητες που επικοινωνούν ανταλλάσσοντας μηνύματα.

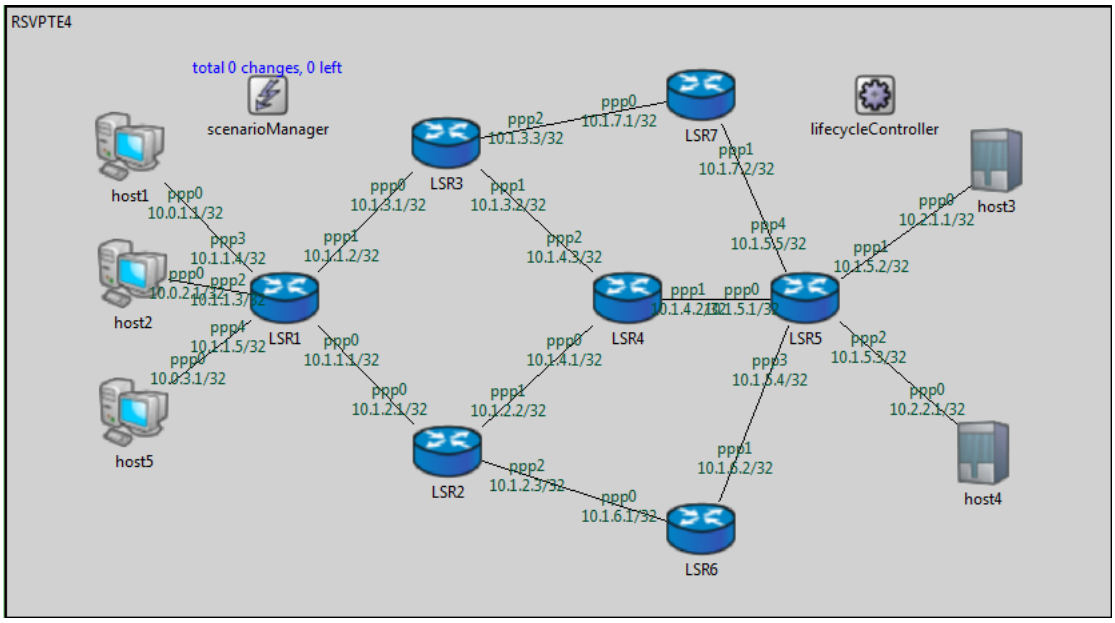
Το OMNeT++ δεν προσομοιώνει κάτι συγκεκριμένο, αλλά παρέχει τη δομή και τα εργαλεία για να γραφούν προσομοιώσεις. Ένα από τα θεμελιώδη συστατικά αυτής της δομής είναι μία σύνθετη αρχιτεκτονική για μοντέλα προσομοίωσης. Τα μοντέλα συντίθενται από επαναχρησιμοποιούμενα συστατικά που λέγονται ενότητες. Οι καλά γραμμένες ενότητες είναι πράγματι επαναχρησιμοποιούμενες και μπορούν να συνδυαστούν μεταξύ τους με πολλούς τρόπους.

Οι ενότητες μπορούν να ενωθούν μεταξύ τους μέσω πυλών (ή θυρών) και να συνδυαστούν για να σχηματίσουν σύνθετες ενότητες. Το βάθος της εμφώλευσης ενοτήτων είναι απεριόριστο. Οι ενότητες επικοινωνούν μέσω μηνυμάτων, τα οποία μπορεί να μεταφέρουν αφηρημένες δομές δεδομένων. Οι ενότητες μπορούν να στείλουν μηνύματα μέσω προκαθορισμένων μονοπατιών μέσω πυλών και συνδέσεων ή απευθείας στον προορισμό τους. Το τελευταίο είναι χρήσιμο για ασύρματες προσομοιώσεις. Οι ενότητες μπορεί να έχουν παραμέτρους που μπορεί να χρησιμοποιηθούν για να προσαρμόσουν τη συμπεριφορά της ενότητας και/ή να παραμετροποιήσουν την τοπολογία του μοντέλου. Οι ενότητες στο χαμηλότερο επίπεδο τηης ιεραρχίας καλούνται απλές ενότητες και ενθυλακώνουν τη συμπεριφορά του μοντέλου. Οι απλές ενότητες γράφονται σε C++ και χρησιμοποιούν τη βιβλιοθήκη της προσομοίωσης. Οι σύνθετες ενότητες και τα μοντέλα προσομοίωσης χρησιμοποιούν τη NED, μία υψηλού επιπέδου γλώσσα προγραμματισμού. [10,11]

Στην παρούσα διπλωματική, μελετάται μέσω του OMNeT++ η συμπεριφορά του πρωτοκόλλου MPLS, καθώς και οι πιο δημοφιλείς εφαρμογές του. Για το λόγο αυτό, χρησιμοποιούνται τόσο υπάρχουσες προσομοιώσεις όσο και νέες.

9.2 Case Study I – Προώθηση πακέτων σε MPLS δίκτυο

Στο πρώτο case study θα μελετήσουμε τον τρόπο με τον οποίο προωθούνται τα πακέτα μέσα στο MPLS δίκτυο και το ρόλο που παίζουν οι ετικέτες στην προώθηση. Για το σκοπό αυτό, θα μελετήσουμε μία υπάρχουσα προσομοίωση του OMNeT++ που έχει γραφεί από τον Vojta Janota. Είναι η προσομοίωση testte_tunnel, η οποία περιέχεται στην open source βιβλιοθήκη INET Framework. Το μονοπάτι είναι: inet→examples→mpls→testte_tunnel. Η τοπολογία του δικτύου που θα μελετήσουμε είναι η εξής:



Εικόνα 54. Τοπολογία δικτύου

Το δίκτυο αποτελείται από 5 hosts (αν και ο host5 δεν χρησιμοποιείται στην προσομοίωση) και 7 LSRs. Ο ingress LSR είναι ο LSR1 και ο egress LSR ο LSR5. Τα πακέτα που στέλνει ο host1 έχουν προορισμό τον host3, ενώ τα πακέτα που στέλνει ο host2 προορίζονται για τον host4. Αυτό φαίνεται και στο αρχείο omnetpp.ini:

- **.host1.udpApp[0].destAddresses = "host3"
- **.host2.udpApp[0].destAddresses = "host4"

Οι αποστολείς (host1 και host2) στέλνουν IP πακέτα, τα οποία μετατρέπει σε MPLS πακέτα ο ingress LSR. Αφού ακολουθήσουν τη διαδρομή τους εντός του MPLS δικτύου, ο egress LSR θα αφαιρέσει τις MPLS ετικέτες και θα στείλει στους παραλήπτες (host3 και host4) τα IP πακέτα. Στο αρχείο LSR1_fec.xml καθορίζεται πως τα πακέτα που εισέρχονται στον ingress LSR με ετικέτα 1 έχουν προορισμό τον host3:

- <fecentry>
- <id>1</id>
- <destination>host3</destination>
- <label>1</label>
- </fecentry>

ενώ τα πακέτα με ετικέτα 2 έχουν προορισμό τον host4:

- <fecentry>
- <id>2</id>
- <destination>host4</destination>
- <label>2</label>
- </fecentry>

Τα αρχεία LSR*_lib.xml λειτουργούν ως η LIB του εκάστοτε LSR. Η LIB αντιστοιχίζει τις εισερχόμενες ετικέτες των πακέτων που λαμβάνει ο LSR με τις εξέρχομενες ετικέτες και προσδιορίζει ποια ενέργεια θα επιτελέσει ο LSR στη στοίβα ετικετών (push,pop,swap). Ταυτόχρονα, καθορίζει και τις διεπαφές εισόδου και εξόδου για τα πακέτα. Από τα

αρχεία αυτά θα παρακολουθήσουμε και τα LSPs που θα ακολουθήσουν τα πακέτα στο MPLS δίκτυο, καθώς και τη διαμόρφωση της στοίβας ετικετών σε κάθε LSR.

Ξεκινώντας με τον LSR1 – που είναι ο ingress LSR – βλέπουμε πως εισέρχονται τα πακέτα που στέλνει ο host1, ο LSR κάνει push την ετικέτα 101 και το πακέτο εξέρχεται μέσω της διεπαφής ppp0.

- <inLabel>1</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="101"/>
- </outLabel>

Επίσης, εισέρχονται τα πακέτα που στέλνει ο host2, ο LSR κάνει push την ετικέτα 301 και το πακέτο εξέρχεται μέσω της διεπαφής ppp0.

- <inLabel>2</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="301"/>
- </outLabel>

Από την τοπολογία του δικτύου, παρατηρούμε πως τα πακέτα που εξέρχονται από τη διεπαφή ppp0 του LSR1 κατευθύνονται προς τη διεπαφή ppp0 του LSR2. Ο LSR2 θα κάνει push την ετικέτα 202 τόσο στα πακέτα με ετικέτα 101, όσο και στα πακέτα με ετικέτα 301 και κατόπιν θα τα προωθήσει προς τον LSR4 μέσω της διεπαφής ppp1. Η στοίβα ετικετών των πακέτων που στέλνει ο host1 περιέχει τις ετικέτες 202 101, ενώ η αντίστοιχη των πακέτων του host2 τις ετικέτες 202 301.

- <inLabel>101</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="202"/>
- </outLabel>
- <inLabel>301</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="202"/>
- </outLabel>

Ο LSR4 θα κάνει swap την ετικέτα 202 με τη 203 και θα προωθήσει τα πακέτα μέσω της διεπαφής ppp2 στον LSR3. Οι στοίβες ετικετών τώρα περιέχουν τις ετικέτες 203 101 και 203 301 αντίστοιχα.

- <inLabel>202</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>

- <op code="swap" value="203"/>
- </outLabel>

Ο LSR3 θα κάνει pop την ετικέτα στην κορυφή της στοίβας, οπότε πλέον οι στοίβες θα περιέχουν τις ετικέτες 101 και 301 αντίστοιχα. Στη συνέχεια, μέσω της διεπαφής ppp2 στέλνει τα πακέτα στον LSR7.

- <inLabel>203</inLabel>
- <inInterface>ppp1</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Ο LSR7 θα κάνει swap την ετικέτα 101 με την ετικέτα 102 και την ετικέτα 301 με την ετικέτα 302. Έπειτα, θα προωθήσει τα πακέτα προς τον LSR5 μέσω της διεπαφής ppp1. Οι στοίβες, στην παρούσα φάση, περιέχουν τις ετικέτες 102 και 302 αντίστοιχα.

- <inLabel>101</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="102"/>
- </outLabel>
- <inLabel>301</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="302"/>
- </outLabel>

Τέλος, ο LSR5, που είναι ο egress LSR, θα κάνει pop τις ετικέτες, θα αφαιρέσει τις MPLS κεφαλίδες και τα πακέτα (που δεν είναι πλέον MPLS πακέτα, αλλά IP πακέτα) θα προωθηθούν προς τους τελικούς προορισμούς τους, δηλαδή τον host3 (μέσω της διεπαφής ppp1) και τον host4 (μέσω της διεπαφής ppp2).

- <inLabel>102</inLabel>
- <inInterface>ppp4</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>
- <inLabel>302</inLabel>
- <inInterface>ppp4</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

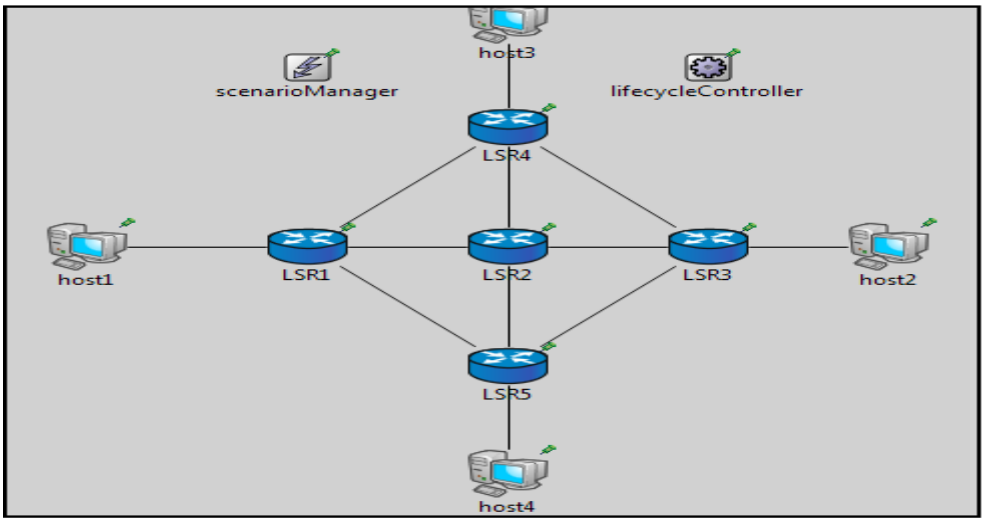
Συμπερασματικά, βλέπουμε πως τα πακέτα που στέλνει τόσο ο host1, όσο και ο host2 ακολουθούν το ίδιο LSP μέσα στο MPLS δίκτυο. Συγκεκριμένα, πρόκειται για το LSP: LSR1→LSR2→LSR4→LSR3→LSR7→LSR5. Παρότι λοιπόν, τα πακέτα περνάνε από το ίδιο LSP, ο εκάστοτε LSR βλέπει την ετικέτα στην κορυφή της στοίβας κάθε πακέτου, συμβουλεύεται την LIB του και γνωρίζει από πού προέρχεται και που πηγαίνει κάθε πακέτο, καθώς και πώς να αντιμετωπίσει τις ετικέτες. Ακολουθούν οι συνοπτικοί πίνακες με όλες τις λεπτομέρειες της διαδρομής για τα πακέτα που στέλνει ο host1 και ο host2.

Host1	**LSR1**	**LSR2**	**LSR4**	**LSR3**	**LSR7**	**LSR5**
inLabel	1	101	202	203	101	102
inInterface	ppp3	ppp0	ppp0	ppp1	ppp0	ppp4
outInterface	ppp0	ppp1	ppp2	ppp2	ppp1	ppp1
op_code(s)	push	push	swap	pop	swap	pop
Label Stack	101	202 101	203 101	101	102	-

Host2	**LSR1**	**LSR2**	**LSR4**	**LSR3**	**LSR7**	**LSR5**
inLabel	2	101	202	203	101	102
inInterface	ppp2	ppp0	ppp0	ppp1	ppp0	ppp4
outInterface	ppp0	ppp1	ppp2	ppp2	ppp1	ppp2
op_code(s)	push	push	swap	pop	swap	pop
Label Stack	301	202 301	203 301	301	302	-

9.3 Case Study II – Label Distribution Protocol (LDP)

Στο δεύτερο case study μελετάμε τη λειτουργία του LDP για την κατανομή των ετικετών. Για το σκοπό αυτό, θα μελετήσουμε μία υπάρχουσα προσομοίωση του OMNeT++ που έχει γραφεί από τον Vojta Janota. Είναι η προσομοίωση ldp, η οποία περιέχεται στην open source βιβλιοθήκη INET Framework. Το μονοπάτι είναι: inet→examples→mpls→ldp. Η τοπολογία του δικτύου που θα μελετήσουμε είναι η εξής:



Εικόνα 55. Τοπολογία LDP δικτύου

Το δίκτυο αποτελείται από 4 hosts και 5 LSRs. Τα πακέτα που στέλνει ο host1 προορίζονται για τον host2, ενώ τα πακέτα που στέλνει ο host3 για τον host4. Αυτό φαίνεται και από τις παρακάτω γραμμές κώδικα που υπάρχουν στο αρχείο omnetpp.ini:

- **.host1.udpApp[0].destAddresses = "host2"
- **.host3.udpApp[0].destAddresses = "host4"

Τρέχοντας την προσομοίωση παρατηρούμε πως τα πακέτα που αποστέλλει ο host1 ακολουθούν τη διαδρομή host1→LSR1→LSR2→LSR3→host2, ενώ τα πακέτα που στέλνει ο host3 τη διαδρομή host3→LSR4→LSR2→LSR5→host4. Επίσης, στα πρώτα πακέτα πακέτα που στέλνουν οι host1 και host3 λαμβάνουν, από τους LSR1 και LSR4 αντίστοιχα, μηνύματα ICMP Destination Unreachable. Αυτό συμβαίνει γιατί μόλις έχουν ξεκινήσει οι hosts να στέλνουν πακέτα, αλλά οι LSRs δεν έχουν προλάβει να ολοκληρωσουν τους πίνακες δρομολόγησης και δεν έχουν τις διαδρομές προς τους host2 και host4.

```
LSR1 --> host1          ICMP-error-#1-type13-code0 ICMP dest unreachable
LSR4 --> host3          ICMP-error-#2-type13-code0 ICMP dest unreachable
```

Εικόνα 56. Μήνυμα ICMP Destination Unreachable

Η προσομοίωση εξετάζει το εξής σενάριο: Τη χρονική στιγμή t=2sec ο κόμβος LSR2 πέφτει και ανακάμπτει τη χρονική στιγμή t=10sec. Αυτό φαίνεται και στο αρχείο scenario.xml:

- <at t="2s">
- <tell module="lifecycleController" target="LSR2" operation="NodeShutdownOperation"/>
- </at>
- <at t="10s">
- <tell module="lifecycleController" target="LSR2" operation="NodeStartOperation"/>
- </at>

Συνεπώς, από τα 2sec κι έπειτα οι κόμβοι LSR1 και LSR4 εξακολουθούν να στέλνουν πακέτα προς τον LSR2, αλλά αυτός δεν μπορεί να τα λάβει, οπότε στην ουσία η επιικοινωνία στο δίκτυο έχει διακοπεί. Για να αντιληφθεί το δίκτυο πως ένας κόμβος έχει πέσει χρησιμοποιεί τον μηχανισμό των hello μηνυμάτων. Ειδικότερα, κάθε LSR στέλνει

ένα hello μήνυμα κάθε 2sec προς κάθε γείτονά του. Το hello μήνυμα στέλνεται ως UDP πακέτο με προορισμό τη multicast διεύθυνση 224.0.0.2 στην οποία ακούνε όλοι δρομολογητές του υποδικτύου (all routers in the subnet).

Η λήψη ενός hello μηνύματος από έναν LSR σημαίνει τη δημιουργία γειτνίασης με τον αποστολέα. Αφού δημιουργηθεί η γειτνίαση, τα hello μηνύματα εξακολουθούν να στέλνονται με τον ίδιο ρυθμό. Αν όμως ένας LSR στείλει προς έναν γείτονά του 3 συνεχόμενα μηνύματα hello και δε λάβει απάντηση, θεωρεί πως ο συγκεκριμένος γείτονας έχει πέσει. Το χρονικό διάστημα στο οποίο ο LSR δε λαμβάνει hello μηνύματα από τον γείτονά του, αλλά εξακολουθεί να διατηρεί τη σχέση γειτνίασης λέγεται hold time. Τόσο το hold time όσο και το hello interval (το χρονικό διάστημα που μεσολαβεί μεταξύ της αποστολής δύο hello μηνυμάτων) καθορίζονται στο αρχείο omnetpp.ini:

- # LDP, MPLS settings
- **.LSR*.holdTime = 6s
- **.LSR*.helloInterval = 2s

Είναι συνήθης πρακτική το hold time να είναι τριπλάσιο του hello interval. Ωστόσο χρειάζεται ιδιαίτερη προσοχή στην επιλογή του hello interval. Αν επιλεγεί μεγάλο χρονικό διάστημα δημιουργείται ο κίνδυνος να αργήσει το δίκτυο να καταλάβει ότι έχει πέσει ένας κόμβος και να μην ψάξει έγκαιρα για εναλλακτική διαδρομή. Έτσι, χάνεται η συνδεσιμότητα για μεγάλο χρονικό διάστημα και καταναλώνονται άσκοπα οι πόροι του δικτύου. Από την άλλη πλευρά, αν το χρονικό διάστημα είναι πολύ μικρό, το δίκτυο θα υπερφορτωθεί από τις ριπές hello μηνυμάτων, ενώ ενδέχεται να δημιουργηθεί πρόβλημα και στους δρομολογητές από τα πολλά μηνύματα που καλούνται να επεξεργαστούν.

Στη συγκεκριμένη προσομοίωση, ισχύει η πρώτη περίπτωση. Οι LSRs του δικτύου έχουν καθορίσει το hello interval στα 2sec. Το διάστημα αυτό είναι αρκετά μεγάλο, ειδικά λαμβάνοντας υπόψιν πως οι hosts στέλνουν πακέτα κάθε 0.01sec. Αυτό ορίζεται στο αρχείο omnetpp.ini:

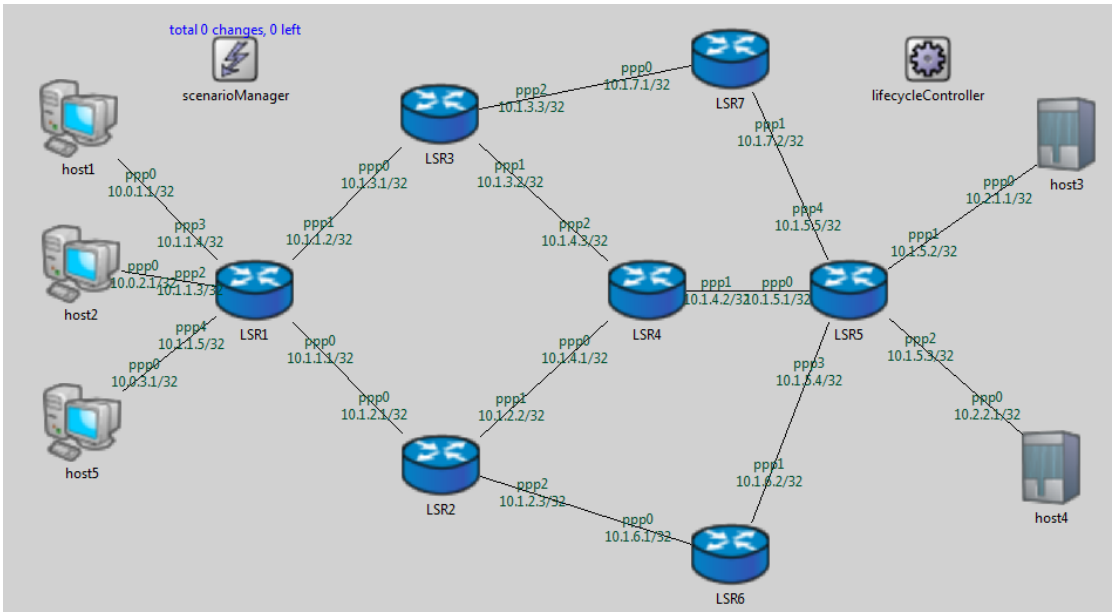
- **.host1.udpApp[0].sendInterval = 0.01s
- **.host3.udpApp[0].sendInterval = 0.01s

Έτσι, το δίκτυο αργεί χαρακτηριστικά να εντοπίσει ότι έχει πέσει ο LSR2 λόγω του υψηλού hold interval. Το αποτέλεσμα είναι οι hosts να στέλνουν τα πακέτα τους κανονικά και οι LSR1 και LSR4 να τα προωθούν προς τον LSR2 που έχει πέσει και δεν μπορεί να τα λάβει. Η επικοινωνία στο δίκτυο διακόπτεται, τα πακέτα δε φθάνουν στον προορισμό τους και οι LSRs δεν αναζητούν εναλλακτικές διαδρομές για να παραδώσουν τα πακέτα. Για παράδειγμα, τα πακέτα του host1 θα μπορούσαν να ακολουθήσουν τη διαδρομή host1→LSR1→LSR4→LSR3→host2 ή host1→LSR1→LSR5→ LSR3→host2 και να αποφύγουν τον LSR2.

Συμπερασματικά, μέσω της προσομοίωσης βλέπουμε αρχικά το LDP σε κανονική λειτουργία. Οι LSRs ανταλλάσσουν τα hello μηνύματα, δημιουργούν γειτνιάσεις και το δίκτυο έχει συνδεσιμότητα. Έπειτα, παρατηρούμε τη μεγάλη καθυστέρηση στον εντοπισμό της βλάβης στη λειτουργία του δικτύου που οφείλεται στο υψηλό hello interval και πως χάνεται η συνδεσιμότητα στο δίκτυο. Τέλος, βλέπουμε πως λειτουργεί ο μηχανισμός των hello μηνυμάτων αφενός για τη δημιουργία γειτνιάσεων μεταξύ των κόμβων και αφετέρου για τον εντοπισμό κόμβων που έχει διακοπεί η λειτουργία τους.

9.4 Case Study III – Λειτουργία MPLS VPN

Στο τρίτο case study δημιουργούμε μία προσομοίωση που θα αφορά τον τρόπο λειτουργίας του MPLS VPN. Για τη δημιουργία της προσομοίωσης βασιζόμαστε στο μοντέλο testte_tunnel που ανέπτυξε ο Vojta Janota και περιέχεται στην open source βιβλιοθήκη INET Framework. Ο κώδικας που περιέχεται στα αρχεία της προσομοίωσης παρατίθεται στο Παράρτημα Ι. Η τοπολογία του δικτύου είναι η εξής:



Εικόνα 57. Τοπολογία MPLS VPN δικτύου

Το δίκτυο αποτελείται από 5 hosts (αν και ο host5 δεν χρησιμοποιείται στην προσομοίωση) και 7 LSRs. Ο ingress LSR είναι ο LSR1 και ο egress LSR ο LSR5. Οι υπόλοιποι είναι οι P δρομολογητές. Τα πακέτα που στέλνει ο host1 έχουν προορισμό τον host3, ενώ τα πακέτα που στέλνει ο host2 προορίζονται για τον host4. Αυτό φαίνεται και στο αρχείο omnetpp.ini:

- **.host1.udpApp[0].destAddresses = "host3"
- **.host2.udpApp[0].destAddresses = "host4"

Τα αρχεία LSR*_lib.xml λειτουργούν ως η LIB του εκάστοτε LSR. Θα ξεκινήσουμε την ανάλυση της προσομοίωσης για τα πακέτα που στέλνει ο host1. Ο ingress LSR, δηλαδή ο LSR1 θα λάβει στη διεπαφή ppp3 τα IP πακέτα του host1 και θα τα μετατρέψει σε MPLS πακέτα προσθέτοντας την MPLS κεφαλίδα και εισάγοντας ετικέτες στη στοίβα. Συγκεκριμένα, ο ingress LSR θα προσθέσει δύο ετικέτες: την ετικέτα 130 και την ετικέτα 100. Η ετικέτα 130 βρίσκεται στην κορυφή της στοίβας και είναι η ετικέτα που χρησιμοποιείται από όλους τους δρομολογητές και προσδιορίζει το μονοπάτι προς τον egress δρομολογητή, δηλαδή τον LSR5. Η ετικέτα 100 είναι η VPN ετικέτα, δηλαδή η ετικέτα που βοηθάει τους PE δρομολογητές να αναγνωρίσουν το VPN. Έπειτα θα προωθήσει τα πακέτα μέσω της διεπαφής ppp1 προς τον επόμενο LSR, τον LSR3.

- <inLabel>1</inLabel>
- <inInterface>ppp3</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="100"/>
- <op code="push" value="130"/>
- </outLabel>

Ο LSR3 θα λάβει στη διεπαφή ppp0 πακέτα με εισερχόμενη ετικέτα 130. Θα την κάνει swap με την ετικέτα 370 και θα τα στείλει μέσω της ppp2 διεπαφής προς τον LSR7. Φυσικά, δεν θα πειράξει τη VPN ετικέτα, δηλαδή την ετικέτα 100 καθώς αυτή αφορά αποκλειστικά και μόνο τους PE δρομολογητές. Η στοιβα ετικετών περιέχει τις ετικέτες 370 100.

- <inLabel>130</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="swap" value="370"/>
- </outLabel>

Ο LSR7 θα λάβει τα πακέτα με ετικέτα 370 στη διεπαφή ppp0 και αφού εξάγει την ετικέτα στην κορυφή τη στοίβας, δηλαδή την ετικέτα 370, θα τα προωθήσει προς τον LSR5 μέσω της διεπαφής ppp1. Στη στοίβα ετικετών υπάρχει πλέον μονάχα η VPN ετικέτα.

- <inLabel>370</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Ο egress LSR θα λάβει τα πακέτα στη διεπαφή ppp4 με τη VPN ετικέτα. Εξετάζει την LIB του για να αναγνωρίσει σε ποιο VPN ανήκουν τα πακέτα με ετικέτα 100. Στη συνέχεια, θα εξάγει την ετικέτα, θα αφαιρέσει την MPLS κεφαλίδα και θα προωθήσει τα πακέτα, ως IP πακέτα πλέον, προς τον τελικό προορισμό τους, τον host3.

- <inLabel>100</inLabel>
- <inInterface>ppp4</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Το VPN τούνελ που ακολουθούν τα πακέτα από τον host1 ως τον host3 είναι: host1→LSR1→LSR3→LSR7→LSR5→host3.

Συνεχίζοντας με τα πακέτα που στέλνει ο host2, θα εξετάσουμε και πάλι την LIB του LSR1 που είναι ο ingress LSR. Ο LSR1 θα λάβει στη διεπαφή ppp2 τα IP πακέτα του host2 και θα εισάγει την MPLS κεφαλίδα και ετικέτες στη στοίβα. Συγκεκριμένα, ο ingress LSR θα προσθέσει την ετικέτα 120 και την ετικέτα 200. Η ετικέτα 120 βρίσκεται στην κορυφή της στοίβας και είναι η ετικέτα αυτή που χρησιμοποιείται από όλους τους δρομολογητές και προσδιορίζει το μονοπάτι προς τον egress δρομολογητή. Η ετικέτα 200 είναι η VPN ετικέτα. Έπειτα θα προωθήσει τα πακέτα μέσω της διεπαφής ppp0 προς τον επόμενο LSR, τον LSR2.

- <inLabel>2</inLabel>
- <inInterface>ppp2</inInterface>

- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="200"/>
- <op code="push" value="120"/>
- </outLabel>

Ο LSR2 θα λάβει στη διεπαφή ppp0 πακέτα με εισερχόμενη ετικέτα 120 και θα την κάνει swap με την ετικέτα 240. Έπειτα, θα προωθήσει τα πακέτα μέσω της διεπαφής ppp1 προς τον LSR4. Η στοίβα ετικετών περιέχει τις ετικέτες 240 και 200.

- <inLabel>120</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="240"/>
- </outLabel>

Ο LSR4 θα εκτελέσει παρόμοιες ενέργειες με τον LSR2. Θα λάβει τα πακέτα με ετικέτα 240, θα κάνει την ετικέτα swap με την ετικέτα 450 και θα στείλει τα πακέτα στον egress LSR. Η στοίβα ετικετών περιέχει τις ετικέτες 450 και 200.

- <inLabel>240</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="450"/>
- </outLabel>

Ο LSR5 θα λάβει αυτή τη φορά πακέτα που έχουν στη στοίβα τους δύο ετικέτες. Θα κάνει pop την ετικέτα 450 που είναι η ετικέτα που χρησιμοποιούν οι LSRs για να φθάσουν στον egress LSR και θα εξετάσει την ετικέτα 200 για να δει σε ποιο VPN ανήκει το πακέτο και στη συνέχεια θα την εξάγει και αυτή. Άρα, στην περίπτωση αυτή ο LSR5 θα χρειαστεί να κάνει δύο pop, θα αφαιρέσει την MPLS κεφαλίδα και θα στείλει τα IP πακέτα στον host4.

- <inLabel>450</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="pop"/>
- <op code="pop"/>
- </outLabel>

Το VPN τούνελ που ακολουθούν τα πακέτα από τον host2 ως τον host4 είναι: host2→LSR1→LSR2→LSR4→LSR5→host4.

Συμπερασματικά, στην προσομοίωση βλέπουμε πως λειτουργεί το MPLS VPN. Παρέχει ασφάλεια, καθώς τα πακέτα των δύο VPNs δεν μπερδεύονται μεταξύ τους. Έτσι, παρόλο που ο ingress LSR δέχεται τα πακέτα και των δύο VPNs, εξασφαλίζει πως τα πακέτα του host1 θα παραληφθούν στον host3 και τα πακέτα του host2 στον host4. Με βάση τη διεπαφή στην οποία λαμβάνει ο ingress LSR τα IP πακέτα γνωρίζει σε ποιο

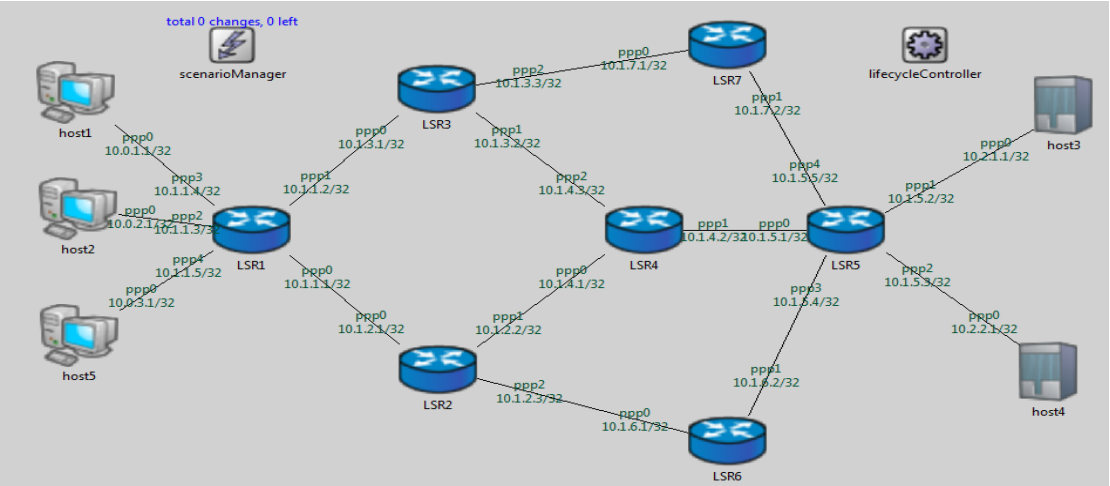
VPN πρέπει να τα τοποθετήσει και αντίστοιχα και ο egress LSR γνωρίζει σε ποιο VPN ανήκει το κάθε πακέτο που λαμβάνει και που να το προωθήσει. Αυτό συμβαίνει λόγω της VPN ετικέτας που φέρει το κάθε πακέτο και προσδιορίζει το VPN στο οποίο ανήκει. Η ετικέτα αυτή αφορά μόνο τους PE δρομολογητές, οι οποίοι είναι οι μόνοι στο δίκτυο που έχουν γνώση των VPNs, και όχι τους P. Οι P δρομολογητές δε γνωρίζουν ότι μεταφέρουν πακέτα που ανήκουν σε VPN καθώς δε γνωρίζουν καν την ύπαρξη της VPN ετικέτας. Η μόνη ετικέτα που γνωρίζουν είναι η εξωτερική, δηλαδή η ετικέτα που τους υποδεικνύει πώς να προωθήσουν το πακέτο στον egress LSR. Συνεπώς η εισαγωγή δύο ετικετών από τον ingress LSR, μίας που χρησιμοποιούν όλοι οι δρομολογητές για την προώθηση εντός του δικτύου και μίας που χρησιμοποιεί μόνο ο ίδιος και ο egress LSR, είναι αυτή που καθιστά τη λύση του MPLS VPN ιδιαίτερα ευέλικτη. Ακολουθούν οι συνοπτικοί πίνακες με όλες τις λεπτομέρειες της διαδρομής για τα πακέτα που στέλνει ο host1 και ο host2.

Host 1	**LSR1**	**LSR3**	**LSR7**	**LSR5**
inLabel	1	130	370	100
inInterface	ppp3	ppp0	ppp0	ppp4
outInterface	ppp1	ppp2	ppp1	ppp1
op_code(s)	push push	swap	pop	pop
Label Stack	130 100	370 100	100	-

Host 2	**LSR1**	**LSR2**	**LSR4**	**LSR5**
inLabel	2	120	240	450
inInterface	ppp2	ppp0	ppp0	ppp0
outInterface	ppp0	ppp1	ppp1	ppp2
op_code(s)	push push	swap	swap	pop pop
Label(s)	120 200	240 200	450 200	-

9.5 Case Study IV – Το Traffic Engineering στο MPLS

Στο τέταρτο case study δημιουργούμε μία προσομοίωση που θα αφορά το Traffic Engineering στο MPLS. Για τη δημιουργία της προσομοίωσης βασιζόμαστε στο μοντέλο testte_tunnel που ανέπτυξε ο Vojta Janota και περιέχεται στην open source βιβλιοθήκη INET Framework. Ο κώδικας που περιέχεται στα αρχεία της προσομοίωσης παρατίθεται στο Παράρτημα ΙΙ. Η τοπολογία του δικτύου είναι η εξής:



Εικόνα 58. Τοπολογία MPLS TE δικτύου

Το δίκτυο αποτελείται από 5 hosts (αν και ο host5 δεν χρησιμοποιείται στην προσομοίωση) και 7 LSRs. Ο ingress LSR είναι ο LSR1 και ο egress LSR ο LSR5. Τα πακέτα που στέλνει ο host1 έχουν προορισμό τον host3, ενώ τα πακέτα που στέλνει ο host2 προορίζονται για τον host4. Αυτό φαίνεται και στο αρχείο omnetpp.ini:

- **.host1.udpApp[0].destAddresses = "host3"
- **.host2.udpApp[0].destAddresses = "host4"

Στην προσομοίωση δημιουργούνται δύο TE τούνελ με απαίτηση για ελάχιστο εύρος ζώνης ζεύξης τα 500 kbps. Θα ξεκινήσουμε την ανάλυση της προσομοίωσης με το TE τούνελ που αφορά τα πακέτα που στέλνει ο host1. Ο LSR1, ο ingress LSR δηλαδή, θα λάβει το IP πακέτο που στέλνει ο host1, θα προσθέσει την MPLS κεφαλίδα για να το μετατρέψει σε MPLS πακέτο και θα του κάνει push την ετικέτα 103 στη στοίβα. Έπειτα, θα το στείλει στον LSR3.

- <inLabel>1</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="103"/>
- </outLabel>

Ο LSR3 θα λάβει στη διεπαφή ppp0 πακέτα με εισερχόμενη ετικέτα 103, θα κάνει push στη στοίβα την ετικέτα 304 και θα το προωθήσει στον LSR4. Η στοίβα περιέχει τις ετικέτες 304 και 103.

- <inLabel>103</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="304"/>
- </outLabel>

Ο LSR4 λαμβάνει πακέτα με ετικέτα 304, την οποία κάνει pop πριν στείλει το πακέτο στον egress LSR. Η στοίβα τώρα περιέχει μόνο την ετικέτα 103.

- <inLabel>304</inLabel>

- <inInterface>ppp2</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Τέλος, ο LSR5 θα λάβει στη διεπαφή ppp0 τα πακέτα με εισερχόμενη ετικέτα 103. Ως egress LSR, θα κάνει pop την ετικέτα, θα αφαιρέσει την MPLS ετικέτα και θα στείλει το IP πακέτο στον προορισμό του, τον host3.

- <inLabel>103</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Συνεπώς, τα πακέτα από τον host1 στον host3 ακολουθούν το TE τούνελ που απαρτίζουν οι: LSR1→LSR3→LSR4→LSR5.

Συνεχίζουμε την ανάλυση της προσομοίωσης με το TE τούνελ που αφορά τα πακέτα που στέλνει ο host2. Ο LSR1 θα λάβει το IP πακέτο που στέλνει ο host1, θα το μετατρέψει σε MPLS πακέτο και θα του κάνει push την ετικέτα 102 στη στοίβα. Στη συνέχεια θα το προωθήσει στον LSR2.

- <inLabel>2</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="102"/>
- </outLabel>

Ο LSR2 λαμβάνει εισερχόμενα πακέτα με ετικέτα 102 και κάνει push την ετικέτα 204. Μετά, τα προωθεί στον LSR4. Η στοίβα περιέχει τις ετικέτες 204 και 102.

- <inLabel>102</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="204"/>
- </outLabel>

Ο LSR4 λαμβάνει στην ppp0 διεπαφή τα πακέτα με ετικέτα 204, την οποία κάνει swap με την ετικέτα 403. Μέσω της διεπαφής ppp2 προωθεί τα πακέτα στον LSR3. Στη στοίβα υπάρχουν οι ετικέτες 403 και 102.

- <inLabel>204</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="swap" value="403"/>
- </outLabel>

Ο LSR3 λαμβάνει στη διεπαφή ppp1 πακέτα με εισερχόμενη ετικέτα 403. Στα πακέτα αυτά κάνει swap την ετικέτα με την ετικέτα 307 και τα προωθεί στον LSR7 μέσω της ppp2 διεπαφής. Η στοίβα περιέχει τις ετικέτες 307 και 102.

- <inLabel>403</inLabel>
- <inInterface>ppp1</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="swap" value="307"/>
- </outLabel>

Ο LSR7 λαμβάνει τα πακέτα που φέρουν στην κορυφή της στοίβας την ετικέτα 307. Κάνει pop την ετικέτα και προωθεί τα πακέτα στον egress LSR, με τη στοίβα να περιέχει πλέον μόνο την ετικέτα 102.

- <inLabel>307</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Τέλος, ο LSR5 θα λάβει στη διεπαφή ppp4 τα πακέτα με εισερχόμενη ετικέτα 102. Ως egress LSR, θα κάνει pop την ετικέτα, θα αφαιρέσει την MPLS ετικέτα και θα στείλει το IP πακέτο στον προορισμό του, τον host4.

- <inLabel>102</inLabel>
- <inInterface>ppp4</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>

Συνεπώς, τα πακέτα από τον host2 στον host4 ακολουθούν το TE τούνελ που απαρτίζουν οι: LSR1→LSR2→LSR4→LSR3→LSR7→LSR5.

Όπως αναφέρθηκε νωρίτερα, υπάρχει απαίτηση για ελάχιστο εύρος ζώνης ζεύξης τα 500 kbps. Προφανώς, εφόσον τα τούνελ έχουν δημιουργηθεί πάνω από τις συγκεκριμένες ζεύξεις, σημαίνει πως ο αλγόριθμος CSPF δεν τις απορρίπτει, άρα έχουν το απαιτούμενο εύρος ζώνης. Αυτό μπορούμε να το ελέγξουμε και στο αρχείο package.ned όπου περιγράφονται τα connections με τα χαρακτηριστικά κάθε ζεύξης, όπως το delay και το datarate (δηλαδή το εύρος ζώνης):

```
LSR1.pppg[0] <--> { delay = 15ms; datarate = 1000kbps; } <--> LSR2.pppg[0];
LSR1.pppg[1] <--> { delay = 5ms; datarate = 1000kbps; } <--> LSR3.pppg[0];
host2.pppg++ <--> { delay = 10ms; datarate = 600kbps; } <--> LSR1.pppg[2];
host1.pppg++ <--> { delay = 10ms; datarate = 600kbps; } <--> LSR1.pppg[3];
LSR2.pppg[1] <--> { delay = 5ms; datarate = 800kbps; } <--> LSR4.pppg[0];
LSR3.pppg[1] <--> { delay = 5ms; datarate = 2400kbps; } <--> LSR4.pppg[2];
LSR4.pppg[1] <--> { delay = 5ms; datarate = 500kbps; } <--> LSR5.pppg[0];
LSR5.pppg[1] <--> { delay = 10ms; datarate = 600kbps; } <--> host3.pppg++;
```

LSR5.pppg[2] <--> { delay = 10ms; datarate = 600kbps; } <--> host4.pppg++;
LSR2.pppg[2] <--> { delay = 10ms; datarate = 300kbps; } <--> LSR6.pppg[0];
LSR5.pppg[3] <--> { delay = 10ms; datarate = 500kbps; } <--> LSR6.pppg[1];
LSR3.pppg[2] <--> { delay = 10ms; datarate = 1200kbps; } <--> LSR7.pppg[0];
LSR5.pppg[4] <--> { delay = 10ms; datarate = 1000kbps; } <--> LSR7.pppg[1];
host5.pppg++ <--> { delay = 10ms; datarate = 600kbps; } <--> LSR1.pppg[4];

Παρατηρούμε ότι η ζεύξη από τον LSR2 στον LSR6 έχει εύρος ζώνης 300 kbps, που είναι χαμηλότερο από το ελάχιστο απαιτούμενο. Άρα, αυτή η ζεύξη δεν μπορεί να χρησιμοποιηθεί στον CSPF και γι'αυτό ο LSR6 δε συμμετέχει σε κανένα από τα 2 TE τούνελ.

Ένα άλλο στοιχείο που παρατηρείται είναι πως το μεγαλύτερο εύρος ζώνης το έχει η ζεύξη μεταξύ των LSR3 και LSR4. Είναι λογικό αν αναλογιστούμε πως αναμένεται να έχει να εξυπηρετήσει τον μεγαλύτερο φόρτο από όλες τις υπόλοιπες, καθώς είναι η μόνη ζεύξη που χρησιμοποιείται και από τα 2 τούνελ.

Θα εστιάσουμε λίγο περισσότερο στον LSR3. Ο κόμβος αυτός συμμετέχει και στα δύο TE τούνελ και συνεπώς πρέπει να κάνει τις κατάλληλες αντιστοιχίσεις στην LIB του ώστε να αποφύγει να μπερδέψει τα πακέτα κάθε τούνελ. Αυτό ρυθμίζεται από το πρωτόκολλο RSVP. Κάθε δρομολογητής κατά τη δημιουργία κάθε TE τούνελ ενημερώνει με τα Resv μηνύματα την ετικέτα που περιμένει να δει για κάθε τούνελ. Αν ο LSR3 λάβει πακέτο με ετικέτα 103, αυτό το πακέτο ανήκει στο 1^{ο} τούνελ, θα το λάβει στη διεπαφή ppp0 και θα το προωθήσει στον LSR4 μέσω της διεπαφής ppp1, αφού πρώτα κάνει push την ετικέτα 304. Αντίστοιχα, αν λάβει πακέτο με ετικέτα 403, αυτό το πακέτο ανήκει στο 2^{ο} τούνελ, θα το λάβει στη διεπαφή ppp1 και θα το προωθήσει στον LSR7 μέσω της διεπαφής ppp2, αφού πρώτα κάνει swap την ετικέτα 307. Με τον ίδιο τρόπο διατηρούν ξεχωριστά μεταξύ τους τα TE τούνελ και οι δρομολογητές LSR1, LSR4 και LSR5.

Τέλος, από το αρχείο omnetpp.ini προκύπτει πως οι δύο hosts στέλνουν με διαφορετικό ρυθμό. Συγκεκριμένα, ο host1 στέλνει κάθε 0.01sec, ενώ ο host2 κάθε 0.03sec:

- **.host1.udpApp[0].sendInterval = 0.01s
- **.host2.udpApp[0].sendInterval = 0.03s

Αυτό το γεγονός μπορεί να δώσει και μία εξήγηση ως προς το γιατί το δεύτερο τούνελ αποτελείται από περισσότερα hops σε σχέση με το πρώτο. Ο host2 στέλνει λιγότερα πακέτα από τον host1 και επιβαρύνει πολύ λιγότερο τις ζεύξεις. Επιπλέον, αν το 2^{ο} TE τούνελ επέλεγε π.χ. τη συντομότερη (όσον αφορά τα hops) διαδρομή LSR1→LSR2→LSR4→LSR5, οι ζεύξεις μεταξύ του LSR3, LSR7 και LSR5 θα έμεναν αχρησιμοποίητες και θα επωμιζόταν όλο το φόρτο η ζεύξη μεταξύ των LSR4 και LSR5, η οποία δεν έχει ιδιαίτερα υψηλό εύρος ζώνης, δημιουργώντας καθυστερήσεις στο δίκτυο. Για τους λόγους αυτούς, θεωρήθηκε προτιμότερο να ακολουθήσει το 2^{ο} τούνελ, που μεταφέρει λιγότερα πακέτα από το 1^{ο}, μία μεγαλύτερη διαδρομή.

Ακολουθούν οι συνοπτικοί πίνακες με όλες τις λεπτομέρειες που αφορούν τα δύο TE τούνελ.

1ο τούνελ	**LSR1**	**LSR3**	**LSR4**	**LSR5**
inLabel	1	103	304	103
inInterface	ppp3	ppp0	ppp2	ppp0
outInterface	ppp1	ppp1	ppp1	ppp1
op_code(s)	push	push	pop	pop
Label Stack	103	304 103	103	-

2ο τούνελ	**LSR1**	**LSR2**	**LSR4**	**LSR3**	**LSR7**	**LSR5**
inLabel	2	102	204	403	307	102
inInterface	ppp2	ppp0	ppp0	ppp1	ppp0	ppp4
outInterface	ppp0	ppp1	ppp2	ppp2	ppp1	ppp2
op_code(s)	push	push	swap	swap	pop	pop
Label(s)	102	204 102	403 102	307 102	102	-

10. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η επιτυχία του MPLS οφείλεται δίχως αμφιβολία στο γεγονός ότι επιτρέπει στο δίκτυο να μεταφέρει κάθε είδος κίνησης. Αποτελεί το μέσο με το οποίο ένα IP δίκτυο μπορεί να συνενώσει πολλαπλά δίκτυα σε ένα. Διαφορετικές τεχνολογίες, όπως ATM, Frame Realy, VoIP και IP μπρούν να ενωθούν σε μία ενιαία δικτυακή υποδομή, δημιουργώντας οφέλη και σε οικονομικό επίπεδο. Ταυτόχρονα, βασικά κίνητρα για την ανάπτυξη του MPLS αποτέλεσαν η υψηλή επεκτασιμότητα, η γρήγορη προώθηση πακέτων, η ενσωμάτωση IP και ATM, καθώς και οι δύο βασικές εφαρμογές του, το MPLS VPN και το MPLS Traffic Engineering.

Το MPLS έχει καταφέρει να ωριμάσει κατά τη διάρκεια των χρόνων και να αποδείξει πψς πρόκειται για μία σταθερή, αλλά ευέλικτη τεχνολογία με νέα χαρακτηριστικά και προοπτικές ανάπτυξης. Αποτελεί μία λύση που επιλέγουν τόσο μεγάλης κλίμακας δίκτυα παροχής υπηρεσιών, όσο και μεγάλες επιχειρήσεις με δίκτυο που εκτείνεται σε διαφορετικές γεωγραφικές τοποθεσίες. Πολλοί τηλεπικοινωνιακοί πάροχοι πέτυχαν σημαντική εξοικονόμηση κόστους με την υιοθέτηση του MPLS VPN για την παροχή των υπηρεσιών τους.

Η επιτυχία της τεχνολογίας επιβεβαιώνεται και από τον ολοένα αυξανόμενο αριθμό όσων την υιοθετούν. Τη δεδομένη στιγμή υπάρχουν περισσότερα από 100 IETF προσχέδια και RFCs για το MPLS. Ταυτόχρονα, τόσο το MPLS όσο και η μετεξέλιξη του, το GMPLS, αποτελούν το κλειδί στη δημιουργία των αρχιτεκτονικών επόμενης γενιάς. Με βάση τα παραπάνω δεδομένα, αλλά και το γεγονός πως το MPLS, όπως και το Internet, βασίζεται στο IP, μοιάζει βέβαιο πως πρόκειται για μία τεχνολογία της οποίας το μέλλον είναι εξασφαλισμένο για μεγάλο διάστημα.

ΠΙΝΑΚΑΣ ΟΡΟΛΟΓΙΑΣ

Ξενόγλωσσος όρος	**Ελληνικός Όρος**
Layer	Επίπεδο
Hardware	Υλικό
CPU	Επεξεργαστής
Customer Edge – CE	Άκρο Πελάτη
Provider Edge – PE	Άκρο Παρόχου
Access List – ACL	Λίστα Πρόσβασης
Traffic Engineering	Διαχείριση Κίνησης
Routing Loop	Βρόχος Δρομολόγησης
Next Hop	Επόμενη Αναπήδηση
Virtual Private Network – VPN	Εικονικό Ιδιωτικό Δίκτυο
Scalability	Επεκτασιμότητα
Interoperability	Διαλειτουργικότητα
Upstream	Ανοδικός
Downstream	Καθοδικός
Ingress	Είσοδος
Egress	Έξοδος
Integrated Services – IntServ	Ενοποιημένες Υπηρεσίες
Differentiated Services – DiffServ	Διαφοροποιημένες Υπηρεσίες
Full Mesh	Πλήρες Πλέγμα

ΣΥΝΤΜΗΣΕΙΣ – ΑΡΚΤΙΚΟΛΕΞΑ – ΑΚΡΩΝΥΜΙΑ

MPLS	Multiprotocol Label Switching
ATM	Asynchronous Transfer Mode
IP	Internet Protocol
WAN	Wide Area Network
VPN	Virtual Private Network
ASIC	Application-Specific Integrated Circuit
AToM	Any Transport over MPLS
BGP	Border Gateway Protocol
TDM	Time Division Multiplexing
SONET	Synchronous Optical Network
SDH	Synchronous Digital Hierarchy
GRE	Generic Routing Encapsulation
ACL	Access List
CE	Customer Edge
PE	Provider Edge
VRF	Virtual Routing and Forwarding
QoS	Quality of Service
VPI	Virtual Path Identifier
VCI	Virtual Channel Identifier
BoS	Bottom of Stack
TTL	Time To Live
LSR	Label-Switched Router
LSP	Label-Switched Path
FEC	Forwarding Equivalence Class
ToS	Type of Service
TCP	Transmission Control Protocol
LFIB	Label Forwarding Information Base
LIB	Label Information Base
OSPF	Open Shortest Path First
IS-IS	Intermediate System to Intermediate System
PIM	Protocol Independent Multicast
DLCI	Data Link Connection Identifier
UDP	User Datagram Protocol

PDU	Protocol Data Unit
TLV	Type-Length-Value
NHLFE	Next Hop Label Forwarding Entry
ILM	Incoming Label Map
FTN	FEC-to-NHLFE
OAM	Operation and Maintenance
LDP	Label Distribution Protocol
PHP	Penultimate Hop Popping
ICMP	Internet Control Message Protocol
RFC	Request for Comment
CoS	Class of Service
VC	Virtual Circuit
CIR	Committed Information Rate
PIR	Peak Information Rate
VL	VPN Label
IL	Interior Label
TE	Traffic Engineering
CSPF	Constrained Shortest Path First
RSVP	Resource Reservation Protocol
IntServ	Integrated Services
IETF	Internet Engineering Task Force
Tspec	Traffic Specification
Rspec	Service Request Specification
flowspec	Flow Specification
GS	Guaranteed Sevice
CLS	Controlled Load Service
SMTP	Simple Mail Transfer Protocol
PHB	Per-Hop Behavior
BA	Behavior Aggregate
EF	Expedited Forwarding
AF	Assured Forwarding
L-LSP	Label LSP
DSCP	Differentiated Services Code Point
RIR	Regional Internet Registries

NAT	Network Address Translation
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
MTU	Maximum Transmission Unit
GMPLS	Generalized Multiprotocol Label Switching
LMP	Link Management Protocol
BFD	Bidirectional Forwarding Detection

ΠΑΡΑΡΤΗΜΑ Ι

Στο παρόν παράρτημα παρατίθεται ο κώδικας που περιέχεται στα αρχεία της προσομοίωσης που αφορά το MPLS VPN.

Αρχείο host1.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.1.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host2.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.2.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host3.rt

- ifconfig:
- name: ppp0 inet_addr: 10.2.1.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.5.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.5.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host4.rt

- ifconfig:
- name: ppp0 inet_addr: 10.2.2.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.5.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.5.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host5.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.3.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0

- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο LSR1_fec.xml

- <?xml version="1.0"?>
- <fectable>
- <fecentry>
- <id>1</id>
- <destination>host3</destination>
- <label>1</label>
- </fecentry>
- <fecentry>
- <id>2</id>
- <destination>host4</destination>
- <label>2</label>
- </fecentry>
- </fectable>

Αρχείο LSR1_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>1</inLabel>
- <inInterface>ppp3</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="100"/>
- <op code="push" value="130"/>
- </outLabel>
- <color>100</color>
- </libentry>
- <libentry>
- <inLabel>2</inLabel>
- <inInterface>ppp2</inInterface>
- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="200"/>
- <op code="push" value="120"/>
- </outLabel>
- <color>300</color>
- </libentry>
- </libtable>

Αρχείο LSR1.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.1.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.1.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.1.3 MTU: 1500 Metric: 1
- name: ppp3 inet_addr: 10.1.1.4 MTU: 1500 Metric: 1
- name: ppp4 inet_addr: 10.1.1.5 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.2.1 10.1.2.1 255.255.255.255 H 0 ppp0

- 10.1.3.1 10.1.3.1 255.255.255.255 H 0 ppp1
- 10.0.2.1 10.0.2.1 255.255.255.255 H 0 ppp2
- 10.0.1.1 10.0.1.1 255.255.255.255 H 0 ppp3
- 10.0.3.1 10.0.3.1 255.255.255.255 H 0 ppp4
- routeend.

Αρχείο LSR2_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>120</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="240"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR2.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.2.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.2.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.2.3 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 10.1.1.1 255.255.255.255 H 0 ppp0
- 10.1.4.1 10.1.4.1 255.255.255.255 H 0 ppp1
- 10.1.6.1 10.1.6.1 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR3_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>130</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="swap" value="370"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR3.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.3.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.3.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.3.3 MTU: 1500 Metric: 1

- ifconfigend.

- route:
- 10.1.1.1 10.1.1.2 255.255.255.255 H 0 ppp0
- 10.1.4.1 10.1.4.3 255.255.255.255 H 0 ppp1
- 10.1.7.1 10.1.7.1 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR4_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>240</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="swap" value="450"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR4.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.4.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.4.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.4.3 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.2.1 10.1.2.2 255.255.255.255 H 0 ppp0
- 10.1.5.1 10.1.5.1 255.255.255.255 H 0 ppp1
- 10.1.3.1 10.1.3.2 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR5_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>100</inLabel>
- <inInterface>ppp4</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>
- </libentry>
- <libentry>
- <inLabel>450</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="pop"/>
- <op code="pop"/>
- </outLabel>

- </libentry>
- </libtable>

Αρχείο LSR5.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.5.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.5.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.5.3 MTU: 1500 Metric: 1
- name: ppp3 inet_addr: 10.1.5.4 MTU: 1500 Metric: 1
- name: ppp4 inet_addr: 10.1.5.5 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.4.1 10.1.4.2 255.255.255.255 H 0 ppp0
- 10.2.1.1 10.2.1.1 255.255.255.255 H 0 ppp1
- 10.2.2.1 10.2.2.1 255.255.255.255 H 0 ppp2
- 10.1.6.1 10.1.6.2 255.255.255.255 H 0 ppp3
- 10.1.7.1 10.1.7.2 255.255.255.255 H 0 ppp4
- routeend.

Αρχείο LSR6.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.6.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.6.2 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.2.1 10.1.2.3 255.255.255.255 H 0 ppp0
- 10.1.5.1 10.1.5.4 255.255.255.255 H 0 ppp1
- routeend.

Αρχείο LSR7_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>370</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>
- <color>100</color>
- </libentry>
- </libtable>

Αρχείο LSR7.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.7.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.7.2 MTU: 1500 Metric: 1
- ifconfigend.

- route:

- 10.1.3.1 10.1.3.3 255.255.255.255 H 0 ppp0
- 10.1.5.1 10.1.5.5 255.255.255.255 H 0 ppp1
- routeend.

Αρχείο omnetpp.ini

- [General]
- network = RSVPTE4
- sim-time-limit = 5s
- #cpu-time-limit= 5800000s
- total-stack = 64MiB
- tkenv-plugin-path = ../../../etc/plugins

- **.host{1..2}.numUdpApps = 1
- **.host{1..2}.udpApp[*].typename = "UDPBasicApp"
- **.host{1..2}.udpApp[0].localPort = 100
- **.host{1..2}.udpApp[0].destPort = 100
- **.host{1..2}.udpApp[0].messageLength = 128 bytes
- **.host{1..2}.udpApp[0].sendInterval = 0.01s
- **.host1.udpApp[0].destAddresses = "host3"
- **.host2.udpApp[0].destAddresses = "host4"

- **.host{3..4}.numUdpApps = 1
- **.host{3..4}.udpApp[*].typename = "UDPSink"
- **.host{3..4}.udpApp[0].localPort = 100

- # ip config
- **.host1.routingFile = "host1.rt"
- **.host2.routingFile = "host2.rt"
- **.host3.routingFile = "host3.rt"
- **.host4.routingFile = "host4.rt"
- **.host5.routingFile = "host5.rt"

- # LSR configuration

- **.LSR1.classifier.config = xmldoc("LSR1_fec.xml")

- **.LSR1.libTable.config = xmldoc("LSR1_lib.xml")
- **.LSR2.libTable.config = xmldoc("LSR2_lib.xml")
- **.LSR4.libTable.config = xmldoc("LSR4_lib.xml")
- **.LSR3.libTable.config = xmldoc("LSR3_lib.xml")
- **.LSR7.libTable.config = xmldoc("LSR7_lib.xml")
- **.LSR5.libTable.config = xmldoc("LSR5_lib.xml")

- **.LSR*.rsvp.helloInterval = 0.2s
- **.LSR*.rsvp.helloTimeout = 0.5s

- **.LSR1.routingFile = "LSR1.rt"
- **.LSR2.routingFile = "LSR2.rt"
- **.LSR3.routingFile = "LSR3.rt"
- **.LSR4.routingFile = "LSR4.rt"
- **.LSR5.routingFile = "LSR5.rt"
- **.LSR6.routingFile = "LSR6.rt"
- **.LSR7.routingFile = "LSR7.rt"

- # NIC configuration
- **.ppp[*].queueType = "DropTailQueue" # in routers

- **.ppp[*].queue.frameCapacity = 10 # in routers

- # scenario
- **.scenarioManager.script = xml("<scenario/>")

Αρχείο package.ned

- package vpn;

- @license(LGPL);

- import inet.base.LifecycleController;
- import inet.base.UnimplementedModule;
- import inet.nodes.inet.StandardHost;
- import inet.nodes.mpls.RSVP_LSR;
- import inet.world.scenario.ScenarioManager;

- network RSVPTE4
- {
- parameters:
- **.networkLayer.configurator.networkConfiguratorModule = "";

- submodules:
- LSR1: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1";
- @display("p=160,167");
- gates:
- pppg[5];
- }
- LSR2: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1 ppp2";
- @display("p=254,255");
- gates:
- pppg[3];
- }
- LSR3: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1 ppp2";
- @display("p=253,78");
- gates:
- pppg[3];
- }
- LSR4: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1 ppp2";
- @display("p=358,167");
- gates:
- pppg[3];
- }
- LSR5: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp3 ppp4";
- @display("p=460,167");
- gates:
- pppg[5];

```
        }
        LSR6: RSVP_LSR {
            parameters:
                peers = "ppp0 ppp1";
                @display("p=400,300");
            gates:
                pppg[2];
        }
        LSR7: RSVP_LSR {
            parameters:
                peers = "ppp0 ppp1";
                @display("p=400,50");
            gates:
                pppg[2];
        }
        host1: StandardHost { // client
            parameters:
                @display("p=71,80");
        }
        host2: StandardHost { // client
            parameters:
                @display("p=72,153");
        }
        host3: StandardHost { // server
            parameters:
                @display("p=570,88;i=device/server");
        }
        host4: StandardHost { // server
            parameters:
                @display("p=562,256;i=device/server");
        }
        host5: StandardHost { // client
            parameters:
                @display("p=73,233");
        }
        scenarioManager: ScenarioManager {
            parameters:
                @display("p=150,50");
        }
        lifecycleController: LifecycleController {
            parameters:
                @display("p=500,50");
        }
    connections:
        LSR1.pppg[0] <--> {delay = 15ms; datarate = 600kbps;} <-->      LSR2.pppg[0];
        LSR1.pppg[1] <--> {delay = 5ms; datarate = 600kbps;} <-->       LSR3.pppg[0];
        host2.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR1.pppg[2];
        host1.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR1.pppg[3];
        LSR2.pppg[1] <--> {delay = 5ms; datarate = 600kbps;} <-->       LSR4.pppg[0];
        LSR3.pppg[1] <--> {delay = 5ms; datarate = 600kbps;} <-->       LSR4.pppg[2];
        LSR4.pppg[1] <--> {delay = 5ms; datarate = 600kbps;} <-->       LSR5.pppg[0];
        LSR5.pppg[1] <--> {delay = 10ms; datarate = 600kbps;} <-->      host3.pppg++;
        LSR5.pppg[2] <--> {delay = 10ms; datarate = 600kbps;} <-->      host4.pppg++;
        LSR2.pppg[2] <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR6.pppg[0];
        LSR5.pppg[3] <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR6.pppg[1];
        LSR3.pppg[2] <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR7.pppg[0];
        LSR5.pppg[4] <--> {delay = 10ms; datarate = 600kbps;} <-->      LSR7.pppg[1];
```

- host5.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <--> LSR1.pppg[4];
- }

ΠΑΡΑΡΤΗΜΑ ΙΙ

Στο παρόν παράρτημα παρατίθεται ο κώδικας που περιέχεται στα αρχεία της προσομοίωσης που αφορά το MPLS Traffic Engineering.

Αρχείο host1.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.1.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host2.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.2.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host3.rt

- ifconfig:
- name: ppp0 inet_addr: 10.2.1.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.5.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.5.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host4.rt

- ifconfig:
- name: ppp0 inet_addr: 10.2.2.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.5.1 * 255.255.255.255 H 0 ppp0
- default: 10.1.5.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο host5.rt

- ifconfig:
- name: ppp0 inet_addr: 10.0.3.1 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 * 255.255.255.255 H 0 ppp0

- default: 10.1.1.1 0.0.0.0 G 0 ppp0
- routeend.

Αρχείο LSR1_fec.xml

- <?xml version="1.0"?>
- <fectable>
- <fecentry>
- <id>1</id>
- <destination>host3</destination>
- <label>1</label>
- </fecentry>
- <fecentry>
- <id>2</id>
- <destination>host4</destination>
- <label>2</label>
- </fecentry>
- </fectable>

Αρχείο LSR1_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>1</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="push" value="103"/>
- </outLabel>
- <color>100</color>
- </libentry>
- <libentry>
- <inLabel>2</inLabel>
- <inInterface>any</inInterface>
- <outInterface>ppp0</outInterface>
- <outLabel>
- <op code="push" value="102"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR1.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.1.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.1.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.1.3 MTU: 1500 Metric: 1
- name: ppp3 inet_addr: 10.1.1.4 MTU: 1500 Metric: 1
- name: ppp4 inet_addr: 10.1.1.5 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.2.1 10.1.2.1 255.255.255.255 H 0 ppp0
- 10.1.3.1 10.1.3.1 255.255.255.255 H 0 ppp1

- 10.0.2.1 10.0.2.1 255.255.255.255 H 0 ppp2
- 10.0.1.1 10.0.1.1 255.255.255.255 H 0 ppp3
- 10.0.3.1 10.0.3.1 255.255.255.255 H 0 ppp4
- routeend.

Αρχείο LSR2_lib.xml

```
<?xml version="1.0"?>
<libtable>
    <libentry>
        <inLabel>102</inLabel>
        <inInterface>ppp0</inInterface>
        <outInterface>ppp1</outInterface>
        <outLabel>
            <op code="push" value="204"/>
        </outLabel>
        <color>200</color>
    </libentry>
</libtable>
```

Αρχείο LSR2.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.2.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.2.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.2.3 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 10.1.1.1 255.255.255.255 H 0 ppp0
- 10.1.4.1 10.1.4.1 255.255.255.255 H 0 ppp1
- 10.1.6.1 10.1.6.1 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR3_lib.xml

```
<?xml version="1.0"?>
<libtable>
    <libentry>
        <inLabel>103</inLabel>
        <inInterface>ppp0</inInterface>
        <outInterface>ppp1</outInterface>
        <outLabel>
            <op code="push" value="304"/>
        </outLabel>
        <color>100</color>
    </libentry>
    <libentry>
        <inLabel>403</inLabel>
        <inInterface>ppp1</inInterface>
        <outInterface>ppp2</outInterface>
        <outLabel>
            <op code="swap" value="307"/>
        </outLabel>
        <color>200</color>
    </libentry>
</libtable>
```

Αρχείο LSR3.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.3.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.3.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.3.3 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.1.1 10.1.1.2 255.255.255.255 H 0 ppp0
- 10.1.4.1 10.1.4.3 255.255.255.255 H 0 ppp1
- 10.1.7.1 10.1.7.1 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR4_lib.xml

- <?xml version="1.0"?>
- <libtable>
- <libentry>
- <inLabel>304</inLabel>
- <inInterface>ppp2</inInterface>
- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>
- <color>100</color>
- </libentry>
- <libentry>
- <inLabel>204</inLabel>
- <inInterface>ppp0</inInterface>
- <outInterface>ppp2</outInterface>
- <outLabel>
- <op code="swap" value="403"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR4.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.4.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.4.2 MTU: 1500 Metric: 1
- name: ppp2 inet_addr: 10.1.4.3 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.2.1 10.1.2.2 255.255.255.255 H 0 ppp0
- 10.1.5.1 10.1.5.1 255.255.255.255 H 0 ppp1
- 10.1.3.1 10.1.3.2 255.255.255.255 H 0 ppp2
- routeend.

Αρχείο LSR5_lib.xml

```
<?xml version="1.0"?>
<libtable>
    <libentry>
        <inLabel>103</inLabel>
        <inInterface>ppp0</inInterface>
        <outInterface>ppp1</outInterface>
        <outLabel>
            <op code="pop"/>
        </outLabel>
    </libentry>
    <libentry>
        <inLabel>102</inLabel>
        <inInterface>ppp4</inInterface>
        <outInterface>ppp2</outInterface>
        <outLabel>
            <op code="pop"/>
        </outLabel>
    </libentry>
</libtable>
```

Αρχείο LSR5.rt

```
ifconfig:
name: ppp0      inet_addr: 10.1.5.1   MTU: 1500   Metric: 1
name: ppp1      inet_addr: 10.1.5.2   MTU: 1500   Metric: 1
name: ppp2      inet_addr: 10.1.5.3   MTU: 1500   Metric: 1
name: ppp3      inet_addr: 10.1.5.4   MTU: 1500   Metric: 1
name: ppp4      inet_addr: 10.1.5.5   MTU: 1500   Metric: 1
ifconfigend.

route:
10.1.4.1   10.1.4.2      255.255.255.255 H   0      ppp0
10.2.1.1   10.2.1.1      255.255.255.255     H   0      ppp1
10.2.2.1   10.2.2.1      255.255.255.255     H   0      ppp2
10.1.6.1   10.1.6.2      255.255.255.255     H   0      ppp3
10.1.7.1   10.1.7.2      255.255.255.255     H   0      ppp4
routeend.
```

Αρχείο LSR6.rt

```
ifconfig:
name: ppp0      inet_addr: 10.1.6.1   MTU: 1500   Metric: 1
name: ppp1      inet_addr: 10.1.6.2   MTU: 1500   Metric: 1
ifconfigend.

route:
10.1.2.1   10.1.2.3      255.255.255.255 H   0      ppp0
10.1.5.1   10.1.5.4      255.255.255.255     H   0      ppp1
routeend.
```

Αρχείο LSR7_lib.xml

```
<?xml version="1.0"?>
<libtable>
    <libentry>
        <inLabel>307</inLabel>
        <inInterface>ppp0</inInterface>
```

- <outInterface>ppp1</outInterface>
- <outLabel>
- <op code="pop"/>
- </outLabel>
- <color>200</color>
- </libentry>
- </libtable>

Αρχείο LSR7.rt

- ifconfig:
- name: ppp0 inet_addr: 10.1.7.1 MTU: 1500 Metric: 1
- name: ppp1 inet_addr: 10.1.7.2 MTU: 1500 Metric: 1
- ifconfigend.

- route:
- 10.1.3.1 10.1.3.3 255.255.255.255 H 0 ppp0
- 10.1.5.1 10.1.5.5 255.255.255.255 H 0 ppp1
- routeend.

Αρχείο omnetpp.ini

- [General]
- network = RSVPTE4
- sim-time-limit = 5s
- #cpu-time-limit= 5800000s
- total-stack = 64MiB
- tkenv-plugin-path = ../../../etc/plugins

- **.host{1..2}.numUdpApps = 1
- **.host{1..2}.udpApp[*].typename = "UDPBasicApp"
- **.host{1..2}.udpApp[0].localPort = 100
- **.host{1..2}.udpApp[0].destPort = 100
- **.host{1..2}.udpApp[0].messageLength = 128 bytes
- **.host1.udpApp[0].sendInterval = 0.01s
- **.host2.udpApp[0].sendInterval = 0.03s
- **.host1.udpApp[0].destAddresses = "host3"
- **.host2.udpApp[0].destAddresses = "host4"

- **.host{3..4}.numUdpApps = 1
- **.host{3..4}.udpApp[*].typename = "UDPSink"
- **.host{3..4}.udpApp[0].localPort = 100

- # ip config
- **.host1.routingFile = "host1.rt"
- **.host2.routingFile = "host2.rt"
- **.host3.routingFile = "host3.rt"
- **.host4.routingFile = "host4.rt"
- **.host5.routingFile = "host5.rt"

- # LSR configuration

- **.LSR1.classifier.config = xmldoc("LSR1_fec.xml")

- **.LSR1.libTable.config = xmldoc("LSR1_lib.xml")
- **.LSR2.libTable.config = xmldoc("LSR2_lib.xml")
- **.LSR4.libTable.config = xmldoc("LSR4_lib.xml")

- **.LSR3.libTable.config = xmldoc("LSR3_lib.xml")
- **.LSR7.libTable.config = xmldoc("LSR7_lib.xml")
- **.LSR5.libTable.config = xmldoc("LSR5_lib.xml")

- **.LSR*.rsvp.helloInterval = 0.2s
- **.LSR*.rsvp.helloTimeout = 0.5s

- **.LSR1.routingFile = "LSR1.rt"
- **.LSR2.routingFile = "LSR2.rt"
- **.LSR3.routingFile = "LSR3.rt"
- **.LSR4.routingFile = "LSR4.rt"
- **.LSR5.routingFile = "LSR5.rt"
- **.LSR6.routingFile = "LSR6.rt"
- **.LSR7.routingFile = "LSR7.rt"

- # NIC configuration
- **.ppp[*].queueType = "DropTailQueue" # in routers
- **.ppp[*].queue.frameCapacity = 10 # in routers

- # scenario
- **.scenarioManager.script = xml("<scenario/>")

Αρχείο package.ned

- package mpls_te;

- import inet.base.LifecycleController;
- import inet.base.UnimplementedModule;
- import inet.nodes.inet.StandardHost;
- import inet.nodes.mpls.RSVP_LSR;
- import inet.world.scenario.ScenarioManager;

- network RSVPTE4
- {
- parameters:
- **.networkLayer.configurator.networkConfiguratorModule = "";

- submodules:
- LSR1: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1";
- @display("p=160,167");
- gates:
- pppg[5];
- }
- LSR2: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1 ppp2";
- @display("p=254,255");
- gates:
- pppg[3];
- }
- LSR3: RSVP_LSR {
- parameters:
- peers = "ppp0 ppp1 ppp2";
- @display("p=253,78");

```
        gates:
            pppg[3];
    }
    LSR4: RSVP_LSR {
        parameters:
            peers = "ppp0 ppp1 ppp2";
            @display("p=358,167");
        gates:
            pppg[3];
    }
    LSR5: RSVP_LSR {
        parameters:
            peers = "ppp0 ppp3 ppp4";
            @display("p=460,167");
        gates:
            pppg[5];
    }
    LSR6: RSVP_LSR {
        parameters:
            peers = "ppp0 ppp1";
            @display("p=400,300");
        gates:
            pppg[2];
    }
    LSR7: RSVP_LSR {
        parameters:
            peers = "ppp0 ppp1";
            @display("p=400,50");
        gates:
            pppg[2];
    }
    host1: StandardHost { // client
        parameters:
            @display("p=71,80");
    }
    host2: StandardHost { // client
        parameters:
            @display("p=72,153");
    }
    host3: StandardHost { // server
        parameters:
            @display("p=570,88;i=device/server");
    }
    host4: StandardHost { // server
        parameters:
            @display("p=562,256;i=device/server");
    }
    host5: StandardHost { // client
        parameters:
            @display("p=73,233");
    }
    scenarioManager: ScenarioManager {
        parameters:
            @display("p=150,50");
    }
    lifecycleController: LifecycleController {
        parameters:
```

```
                @display("p=500,50");
        }
connections:
    LSR1.pppg[0] <--> {delay = 15ms;datarate = 1000kbps;} <-->    LSR2.pppg[0];
    LSR1.pppg[1] <--> {delay = 5ms; datarate = 1000kbps;} <-->    LSR3.pppg[0];
    host2.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <-->    LSR1.pppg[2];
    host1.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <-->    LSR1.pppg[3];
    LSR2.pppg[1] <--> {delay = 5ms; datarate = 800kbps; } <-->    LSR4.pppg[0];
    LSR3.pppg[1] <--> {delay = 5ms; datarate = 2400kbps;} <-->    LSR4.pppg[2];
    LSR4.pppg[1] <--> {delay = 5ms; datarate = 500kbps; } <-->    LSR5.pppg[0];
    LSR5.pppg[1] <--> {delay = 10ms; datarate = 600kbps;} <-->    host3.pppg++;
    LSR5.pppg[2] <--> {delay = 10ms; datarate = 600kbps;} <-->    host4.pppg++;
    LSR2.pppg[2] <--> {delay = 10ms; datarate = 300kbps;} <-->    LSR6.pppg[0];
    LSR5.pppg[3] <--> {delay = 10ms; datarate = 500kbps;} <-->    LSR6.pppg[1];
    LSR3.pppg[2] <--> {delay = 10ms;datarate = 1200kbps;} <-->    LSR7.pppg[0];
    LSR5.pppg[4] <--> {delay = 10ms;datarate = 1000kbps;} <-->    LSR7.pppg[1];
    host5.pppg++ <--> {delay = 10ms; datarate = 600kbps;} <-->    LSR1.pppg[4];
}
@license(LGPL);
```

ΑΝΑΦΟΡΕΣ

[1] Luc De Ghein, "*MPLS Fundamentals*", 2007, Cisco Press, ISBN: 1-58705-197-4

[2] Vivek Alwayn, "*Advanced MPLS Design and Implementation"*, 2002, Cisco Press, ISBN: 1-58705-020-x

[3] L. Andersson, I. Minei and B. Thomas, "LDP Specification", RFC 5036, October 2007

[4] E. Rosen, A. Viswanathan and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001

[5] I. Pepelnjak, J. Guichard, *"MPLS and VPN Architectures CCIP Edition"*, 2002, Cisco Press, ISBN: 1-58705-081-1

[6] J. Guichard, I. Pepelnjak, J. Apcar "*MPLS and VPN Architectures, Volume II"*, 2003, Cisco Press, ISBN: 1-58705-112-5

[7] E. Osborne, A. Simha, *"Traffic Engineering with MPLS"*, 2003, Cisco Press, ISBN: 1-58705-031-5

[8] S. Alvarez, *"QoS for IP/MPLS Networks"*, 2006, Cisco Press, ISBN: 1-58705-233-4

[9] M. Morrow, A. Sayeed, *"MPLS and Next-Generation Networks"*, 2007, Cisco Press, ISBN: 1-58720-120-8

[10] OMNeT++. *OMNeT++ User Manual*. Ανακτήθηκε από: https://omnetpp.org/doc/omnetpp/manual/usman.html

[11] OMNeT++. *What is OMNeT++*. Ανακτήθηκε από: https://omnetpp.org/intro