



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΚΑΤΕΥΘΥΝΣΗ ΔΙΚΤΥΟΚΕΝΤΡΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Μελέτη και Ανάπτυξη Αλγορίθμων Διαχείρισης Πόρων Δικτύων Κορμού σε Περιβάλλοντα Future Internet

Μεταπτυχιακή Διπλωματική Εργασία

Σαρσεμπάγιεβα Αικατερίνη

ME/08101

Επιβλέπων:

Παναγιώτης Δεμέστιχας

Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιά

Αθήνα, 2012

*Στη μαμά, στο Γιώργο
και στο Μελισσάκι.*

ΠΕΡΙΛΗΨΗ

Στη σύγχρονη εποχή στην οποία οι εφαρμογές, οι υπηρεσίες και τα δεδομένα του Διαδικτύου αυξάνονται εκθετικά, μαζί με την παράλληλη αύξηση τόσο των χρηστών όσο και των απαιτήσεών τους σε παροχή υψηλής Ποιότητας Υπηρεσίας, οι τρέχουσες υποδομές των δικτύων γρήγορα έφεραν τη στιγμή στην οποία η ζήτηση ξεπέρασε την προσφορά. Με αλγορίθμους δρομολόγησης όπως ο OSFP κ.ά., καθώς και με την παροχή υπηρεσιών με «Βέλτιστη Προσπάθεια», άρχισε να παρατηρείται όχι μόνο μεγάλη υποβάθμιση της ποιότητας των συνδέσεων αλλά και η κακώς κατανεμημένη χρησιμοποίηση των πόρων των δικτύων.

Αρκετές προσπάθειες και έρευνες έφεραν στο προσκήνιο αλγορίθμους και συστήματα τα οποία προσπαθούν να επιλύσουν σε τοπικό ή καθολικό επίπεδο τα σημαντικά αυτά ζητήματα, χρησιμοποιώντας τεχνικές ρύθμισης της κίνησης, αστυνόμευσης των χαρακτηριστικών που πρέπει να πληρούνται, δέσμευσης ζεύξεων ή ολόκληρων μονοπατιών και άλλα. Ωστόσο, ενόσω ένα τμήμα της επιστημονικής κοινότητας ερευνούσε και εφεύρισκε τέτοιες πρακτικές δημιουργώντας τεχνολογίες όπως οι IntServ και DiffServ, ένα άλλο σχεδίασε και ανέπτυξε μία νέα μορφή δικτύων η οποία συνδυάζει τα καλύτερα χαρακτηριστικά των ATM και του FrameRelay : Το MPLS.

Όντας υβριδικό δίκτυο Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων και προσφέροντας πρόσφορο έδαφος για τεχνολογίες από - άκρο - εις - άκρο διαχείρισης κίνησης και πόρων, το MPLS γρήγορα συγκέντρωσε το επιστημονικό ενδιαφέρον και συνέβαλλε στην προτυποποίηση και την έρευνα καινοτόμων και κλιμακούμενων μεθόδων στο πεδίο του Traffic Engineering, όπως ο ρητός προσδιορισμός μονοπατιών, τα επίπεδα προτεραιότητας, η τοπική και καθολική προστασία ζεύξεων και άλλα. Παράλληλα, όντας ακόμη ανοιχτό σε προτάσεις και προτυποποιήσεις, έφερε στο προσκήνιο αρκετούς νέους αλγορίθμους Κατανομής Κίνησης και Πόρων, τόσο δυναμικούς όσο και offline πληρώντας μία ή περισσότερες προδιαγραφές του MPLS - TE.

Συνεχίζοντας το έργο των ερευνητών, στην προκείμενη μεταπτυχιακή διπλωματική διατριβή σχεδιάζεται και αναπτύσσεται ένας νέος αλγόριθμος κατανομής Κίνησης και Πόρων σε MPLS δίκτυα του Future Internet, ο οποίος λαμβάνει υπ' όψιν όλες τις απαιτήσεις του MPLS - TE και κατανέμει σε offline λειτουργία τα σύνολα των ροών εντός του δικτύου με τρόπο τέτοιο ώστε να επιτυγχάνεται η ισορροπημένη δρομολόγηση των τελευταίων και η μεγιστοποίηση της ροής.

Πιο συγκεκριμένα, μετά την ενδελεχή έρευνα του MPLS - TE πεδίου, προηγούμενων μελετών στον ίδιο τομέα καθώς και προταθέντων αλγορίθμων στο πεδίο του Flow Maximization, σχεδιάζεται και αναπτύσσεται ο TiReD : ένας Traffic - Resource Distribution αλγόριθμος, εμπνευσμένος από τον Max Flow αλγόριθμο του Dinic, ο οποίος εισάγοντας νέες έννοιες και παραδοχές κατανέμει το φορτίο ισορροπημένα και σε ελάχιστο απαιτούμενο χρόνο εκτέλεσης τόσο σε μικρά, όσο και σε μεγάλα/ πολύπλοκα δίκτυα, δίνοντας βάση όχι μόνο στις απαιτήσεις του MPLS - TE και των SLAs, αλλά και την παράλληλη κατανομή του φορτίου γενικότερα και των ροών ειδικότερα.

ABSTRACT

During the last years, due to the exponential growth of the applications, services and data from the Internet, along with the simultaneous increase of both users and their demands for high quality services provision, the current network infrastructure rapidly came to the point in which the demand exceeded the supply. Using routing algorithms such as the OSPF and providing services in a “Best Effort” way, a serious downgrading of connections’ quality as well as an improper resource distribution has been observed in the current Internet architecture.

Numerous efforts have brought into focus algorithms and systems that try to solve, in a local or global manner, these important issues using techniques to regulate traffic, police the requirements that must be met, bind links or routes and more. However, while some part of the scientific community was investigating and inventing such practices by creating technologies such as the IntServ and the DiffServ, another part designed and developed a new type of networks that combines the best features of the ATM and Frame Relay: The MPLS.

Being a multiprotocol label switching hybrid network and giving the means to facilitate technologies for end – to – end traffic and resources management, MPLS quickly attracted the scientific interest and contributed to the research and standardization of innovative and scalable methods in the field of Traffic Engineering, such as the explicit routing, the priority levels, the local or global links protection and more. Moreover, being still open to suggestions and prototyping, MPLS has brought to light several new algorithms for Traffic and Resource Distribution that work not only in a dynamic but also in an offline way, while meeting one or more of the MPLS – TE requirements.

Continuing the effort of the previous researchers, this postgraduate diploma thesis designs and develops a new Traffic and Resource Distribution algorithm for MPLS networks in the Future Internet, which takes into account all the requirements of the MPLS –TE and distributes in an offline mode the

sets of flows within a network in such a way as to achieve both the balanced routing and the flow maximization.

More specifically, after a thorough research in the MPLS – TE field, the previous studies in the same area and the proposed algorithms in the field of Flow Maximization, TiReD was designed and developed: a Traffic – Resource Distribution Algorithm, inspired by the Dinic' Max Flow algorithm, that introduces new concepts and assumptions so as to distribute and balance of the load within a network in a minimum execution time, for both small and large/ complex networks, taking into consideration not only the requirements of the MPLS – TE and the SLAs, as well as the balanced distribution of the network load in general and the flows in particular.

ΕΥΧΑΡΙΣΤΙΕΣ

Πρώτο και κύριο θα ήθελα να ευχαριστήσω μέσα από την καρδιά μου τον κύριο Δεμέστιχα Παναγιώτη: Καθηγητή, εμπνευστή, σύμβουλο, (πρόεδρο!!!!) και πάνω από όλα Άνθρωπο. Τον ευχαριστώ θερμά που μου εμπιστεύτηκε το συγκεκριμένο θέμα εργασίας, που με τη διδασκαλία και την καθοδήγησή του μου δημιούργησε και μου αύξησε την αγάπη μου για το πεδίο των Δικτύων και του οποίου οι επιλογές, χωρίς καν να το γνωρίζει, κατήυθναν τη ζωή μου στο μονοπάτι που πάντα ονειρευόμουν.

Δεύτερο, θα ήθελα να ευχαριστήσω τον κύριο Τσαγκάρη Κώστα, έναν άνθρωπο πάντα πρόθυμο να βοηθήσει, ο οποίος όταν τίποτα στην διπλωματική δεν ήθελε να συμμορφωθεί και ο εγκέφαλός μου καιγόταν, με στήριζε, με συμβούλευε, με ηρεμούσε και μου προσέφερε πολύτιμες συμβουλές και αρωγή.

Εν συνεχεία, οφείλω ένα μέγιστο ευχαριστώ στη μητέρα μου για τη βοήθεια και τη στήριξή της σε όλο το διάστημα της εκπόνησης της διατριβής μου. Την ευχαριστώ για τις πολυάριθμες φορές που με ελάφρυνε από τις υποχρεώσεις μου προκειμένου να έχω χρόνο για διάβασμα και για τις άπειρες ώρες υπομονής της όταν της γκρίνιαζα για την εκάστοτε μορφή του αλγορίθμου μου που αρνούταν να συνεργαστεί.

Επίσης, χρωστάω ένα τεράστιο ευχαριστώ στους φίλους μου, που σε όλο αυτό το διάστημα έκαναν υπομονή, με συμβούλευαν και με στήριζαν. Ευχαριστώ τον Γιώργο Τριανταφυλλίδη για την κατανόηση και τις θυσίες που έκανε όλο αυτό τον καιρό για μένα. Τον Αλέξανδρο Σταθόπουλο – Λαμπρέλλη, για την πολύτιμη στήριξη, βοήθεια και υπομονή του. Τον Χρήστο Κυριακόπουλο για την δικτατορία του όλα εκείνα τα απογεύματα που διαβάζαμε μαζί ώστε να με επιβλέπει και να μου απαγορεύει να κάνω διαλείμματα (κι ας είχα να κοιμηθώ μέρες!!!), καθώς και για την ιδέα του “Network Randomizer” η οποία μείωσε το χρόνο παραγωγής αποτελεσμάτων από αρκετά χρόνια στους τρεις μήνες. Την Χρυσούλα Πουλοπούλου η οποία μου έδινε θάρρος, δύναμη και η οποία με ανάγκασε να σταματήσω να «σκαλίζω» το «σχέδιο του καλύτερου αλγορίθμου του σύμπαντος» και να παραδώσω τη διατριβή στην προκείμενη μορφή της. Το Μελισάκι μου, (το καλύτερο πλάσμα του χωροχρονικού συνεχούς), που μασούλησε το διάγραμμα ενός παλαιότερου αλγορίθμου που είχα σχεδιάσει, αναγκάζοντάς με να το επανασχεδιάσω, να ανακαλύψω ότι είναι λάθος, να τον διαγράψω και να σχεδιάσω τελικά έναν αλγόριθμο πιο κοντά σε αυτόν που παρουσιάζεται στην εργασία. Τέλος, ένα μεγάλο ευχαριστώ σε όλους τους φίλους μου, γενικά, για την κατανόησή τους όταν η φράση «Δεν μπορώ, έχω διπλωματική» έγινε πλέον σήμα κατατεθέν.

Τέλος, επειδή κατά πάσα πιθανότητα αυτή θα είναι η τελευταία μου εργασία στα Ψηφιακά Συστήματα, θα ήθελα να εκφράσω ένα σχετικά μαζικό, αλλά ειλικρινές ευχαριστώ μέσα από τα βάθη της καρδιάς μου για τους (νυν και τέως) καθηγητές του τμήματος, των οποίων οι γνώσεις, το background, ο χαρακτήρας, η διδασκαλία και το χιούμορ έκαναν τα χρόνια των σπουδών μου ευτυχισμένα, ενδιαφέροντα και, πάνω από όλα, πωρωμένα : (με αλφαβητική σειρά) Δεμέστιχας Παναγιώτης, Δημητρακόπουλος Γιώργος, Κανάτας Αθανάσιος, Ρούσκακας Άγγελος, Σγούρος Νικήτας – Μαρίνος, Σταυρουλάκη Βέρα, Τσαγκάρης Κώστας. You rock...

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ	3
ABSTRACT	5
ΕΥΧΑΡΙΣΤΙΕΣ	7
ΛΙΣΤΑ ΕΙΚΟΝΩΝ	12
ΛΙΣΤΑ ΔΙΑΓΡΑΜΜΑΤΩΝ	13
ΛΙΣΤΑ ΠΙΝΑΚΩΝ	13
ΕΙΣΑΓΩΓΗ	14
ΜΟΝΤΕΛΑ ΠΑΡΟΧΗΣ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΑΣ	20
Εισαγωγή.....	21
Μοντέλο Ενοποιημένων Υπηρεσιών.....	23
Κλάση Εγγυημένης Υπηρεσίας (Guaranteed Services).....	23
Κλάση Ελεγχόμενου Φορτίου (Controlled Load Class).....	24
Μηχανισμοί Παροχής Ποιότητας Υπηρεσίας.....	24
Αξιολόγηση του IntServ (και RSVP).....	28
Μοντέλο Διαφοροποιημένων Υπηρεσιών.....	30
Κλάση Εξασφαλισμένης Προώθησης.....	32
Κλάση Εσπευσμένης Προώθησης.....	34
Αξιολόγηση του DiffServ.....	35
Μεταγωγή Ετικετών Πολλαπλών Πρωτοκόλλων.....	37
Αρχιτεκτονική του MPLS.....	39
Λειτουργία του MPLS.....	45
Σύνοψη.....	47
ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΗΣΗΣ ΚΑΙ ΠΟΡΩΝ ΣΤΟ MPLS	50
Εισαγωγή.....	51
Διαχείριση Κίνησης και Πόρων.....	52
Διανομή Πληροφορίας Ζεύξεων.....	54
Υπολογισμός Διαδρομής.....	56

LSP Σηματοδοσία	57
Θέματα Σχεδιασμού Διαχείρισης Κίνησης και Πόρων	58
Παράμετροι Κίνησης και Περιορισμοί.....	59
Ρητός Προσδιορισμός Διαδρομών	59
Επαναβελτιστοποίηση Διαδρομών	60
Κατανομή Διαδρομών	61
TE Κλάσεις και Επίπεδα Προτεραιότητας.....	61
Προστασία και Ευελιξία.....	63
Αστυνόμευση	64
Online / Offline Δρομολόγηση.....	64
Αλγόριθμοι Δρομολόγησης και Κατανομής Φορτίου	65
Συμπεράσματα.....	79
MAX – FLOW ΠΡΟΒΛΗΜΑ	82
Εισαγωγή.....	83
Ορισμός Προβλήματος (Γενική Προσέγγιση).....	84
Βασικό Πλαίσιο	84
Περιορισμός Χωρητικότητας	85
Διατήρηση Φορτίου.....	86
Περιορισμός Μονοπατιού	87
Συμφόρηση Δικτύου	88
Πρόβλημα Βελτιστοποίησης.....	90
Επέκταση Βασικής Προσέγγισης.....	91
Προσδιορισμός Διαδρομών.....	91
Προτεραιότητα	93
Από Άκρο – εις – Άκρο Προστασία	94
Πρόβλημα Μέγιστης Ροής	96
Βασικό Πλαίσιο	96
Γραμμικό Πρόβλημα Μέγιστης Ροής.....	99
Γραμμικό Πρόβλημα Ελάχιστης Τομής.....	101

Αλγόριθμοι Εύρεσης Μέγιστης Ροής	104
Ford - Fulkerson	104
Edmonds – Karp	110
Dinic' Blocking Flow	111
ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΚΑΙ ΚΙΝΗΣΗΣ	117
Εισαγωγή.....	118
Δεδομένα Εισόδου - Εξόδου	121
Μοντέλο Οντοτήτων – Συσχετίσεων	122
Σχεσιακό Μοντέλο	124
Οντότητες του Αλγορίθμου.....	133
Network	133
Demands	137
Network Memo.....	140
Αλγόριθμος Διαχείρισης Πόρων και Κίνησης	141
Είσοδος	141
Έξοδος.....	141
Αρχικοποίηση	141
Φάση 1 – Έλεγχος Λίστας Demands List	142
Φάση 2 – Λήψη (pop) της Πρώτης Ροής στην Κορυφή της Demands List.....	142
Φάση 3 – Κατασκευή Layered Γράφου G_L από το Δίκτυο	142
Φάση 4 – Αναζήτηση Βέλτιστου Μονοπατιού	143
Φάση 5 – Επεξεργασία Βέλτιστου Μονοπατιού	144
Φάση 6 – Εξυπηρέτηση Ροής.....	145
Φάση 7 – Απόρριψη Ροής.....	145
ΑΠΟΤΕΛΕΣΜΑΤΑ.....	148
Χρόνοι εκτέλεσης.....	149
Χρόνοι συναρτήσεων Link Capacities.....	152
Χρόνοι συναρτήσεων Απαιτούμενου Bandwidth	153
Χρόνοι Συναρτήσεων Τοπολογίας και Πλήθους Demands	154

Κατανομή Φορτίου	157
Απλό δίκτυο μικρού μεγέθους	157
Συμμετρικό δίκτυο μεγάλου μεγέθους	162
Κατανομή Κίνησης (Σε Σχέση με Παρεμβολή)	166
Απλό δίκτυο μικρού μεγέθους	166
Συμμετρικό δίκτυο μεγάλου μεγέθους	167
ΕΠΙΣΚΟΠΗΣΗ.....	169
Συμπεράσματα.....	170
Περαιτέρω Μελέτη	175
ΒΙΒΛΙΟΓΡΑΦΙΑ	178
ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ	181
ΠΑΡΑΡΤΗΜΑ Ι.....	184
Network.h	184
Demands.h	186
RoutedDemands.h	187
Memo.h.....	187
TrafficResourceDistributionAlgorithm() //Network.c.....	188
CreateDemands() //Demands.c	191
ΠΑΡΑΡΤΗΜΑ ΙΙ.....	193
Οδηγίες Εγκατάστασης	193
Εγκατάσταση MySQL Server	193
Εγκατάσταση Βάσης Δεδομένων.....	193
Οδηγίες Εκτέλεσης.....	195

ΛΙΣΤΑ ΕΙΚΟΝΩΝ

Εικόνα 1. RSVP - Ανταλλαγή PATH και RESV μηνυμάτων.....	15
Εικόνα 2. Επεξεργασία Πακέτων στο DiffServ.....	20
Εικόνα 3. Αρχιτεκτονική MPLS.....	27
Εικόνα 4. Λειτουργία του MPLS.....	31
Εικόνα 5. Κρίσιμες Ζεύξεις βάσει MIRA.....	49
Εικόνα 6. Διατήρηση Φορτίου.....	64
Εικόνα 7. Δίκτυο μεταφοράς.....	74
Εικόνα 8. Παράδειγμα δικτύου μεταφοράς.....	83
Εικόνα 9. Σενάριο χείριστης περίπτωσης για τον Ford - Fulkerson αλγόριθμο.....	86
Εικόνα 10. Υπολειπόμενος γράφος μηδενικής ροής.....	89
Εικόνα 11. Γράφος Επιπέδων Μηδενικής Ροής.....	90
Εικόνα 12. Προσθήκη ροής προς τα πίσω.....	90
Εικόνα 13. Αφαίρεση διαδρομών προς τα εμπρός.....	90
Εικόνα 14. Προσθήκη νέας ροής προς τα πίσω.....	91
Εικόνα 15. Αφαίρεση νέων διαδρομών προς τα εμπρός.....	91
Εικόνα 16. Μοντέλο Οντοτήτων - Συσχετίσεων.....	96
Εικόνα 17. Σχεσιακό Μοντέλο.....	98
Εικόνα 18. Activity Diagram Συνολικού Αλγορίθμου.....	116
Εικόνα 19. Activity Diagram Κατασκευής Υπολειπόμενου Γράφου.....	116
Εικόνα 20. Χείριστη Περίπτωση Τοπολογίας.....	119
Εικόνα 21. Απλό Δίκτυο Μικρού Μεγέθους.....	126
Εικόνα 22. Κατανομή Κίνησης στο Μικρό Δίκτυο.....	128
Εικόνα 23. Δίκτυο Μεγάλου Μεγέθους.....	130

ΛΙΣΤΑ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 1. Χρόνοι Εκτέλεσης Βάσει Capacity	152
Διάγραμμα 2. Χρόνοι Εκτέλεσης Βάσει Bandwidth.....	153
Διάγραμμα 3. Χρόνοι Εκτέλεσης Βάσει Τοπολογίας	154
Διάγραμμα 4. Κατανομή Φορτίου στο Μικρό Δίκτυο	159
Διάγραμμα 5. Κατανομή Φορτίου στο Μεγάλο Δίκτυο	163
Διάγραμμα 6. Κατανομή Κίνησης στο Μικρό Δίκτυο	166
Διάγραμμα 7. Κατανομή Κίνησης στο Μεγάλο Δίκτυο	168

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

Πίνακας 1. Συγκεντρωτικός Πίνακας Τοπολογιών, Απαιτήσεων και Χρόνων	150
Πίνακας 2. Ιδιότητες και Χαρακτηριστικά των Ροών	159
Πίνακας 3. Εξυπηρέτηση Ροών στο Μικρό Δίκτυο	161
Πίνακας 4. Ιδιότητες και Χαρακτηριστικά των Ροών	163
Πίνακας 5. Εξυπηρέτηση Ροών στο Μεγάλο Δίκτυο	165

ΕΙΣΑΓΩΓΗ

Το Διαδίκτυο, χάρη στην ταχύτατη διάδοσή του καθώς και στην εκθετική αύξηση που παρουσιάζει τα τελευταία χρόνια, παίζει πλέον έναν από τους βασικότερους ρόλους σε κάθε τομέα της καθημερινής ζωής. Θέματα όπως η ανταλλαγή της πληροφορίας, η επικοινωνία, η μόρφωση, η διαχείριση και λειτουργία μίας επιχείρησης, η διασκέδαση και η κοινωνική ζωή, υποβοηθούνται συνεχώς από το Internet το οποίο έχει γίνει, πλέον, ένα αναπόσπαστο κομμάτι της καθημερινότητας σε κάθε τομέα της ανθρώπινης ζωής.

Τα δεδομένα που διακινούνται εν μέσω του Διαδικτύου διογκώνονται συνεχώς με ταχύτατους ρυθμούς: Από τη μία παρατηρείται εκθετική αύξηση του υλικού που διακινείται στο Διαδίκτυο τόσο από οργανισμούς (ταινίες, πληροφορίες, βιβλία, λογισμικό κ.ά.) όσο και από τους τελικούς χρήστες (βίντεο, εικόνες, παιχνίδια και ούτω καθ' εξής). Από την άλλη, αυξάνεται και ο όγκος της μεταδιδόμενης επικοινωνιακής πληροφορίας, η οποία εκτός από παραδοσιακά μέσα όπως το chat, το e-mail και το instant messaging, χρησιμοποιεί πλέον audiovisual τρόπους μετάδοσης όπως η IP τηλεφωνία και το video/ audioconferencing.

Μαζί με την παράλληλη αύξηση και του πλήθους των χρηστών οι οποίοι χρησιμοποιούν το υπάρχον υλικό και ανεβάζουν νέο, ταχέως έφτασε η στιγμή στην οποία η ζήτηση ξεπέρασε την προσφορά. Στο ρόλο της «ζήτησης» υπάγονται δύο θέσεις: στη μία βρίσκονται οι χρήστες οι οποίοι αιτούνται βέλτιστης ποιότητας, γρήγορων συνδέσεων και αστυνόμευσης των συμφωνητικών τους. Στην άλλη βρίσκονται οι πάροχοι οι οποίοι επιθυμούν την καλύτερη διαχείριση και τη μέγιστη χρησιμοποίηση των δικτύων. Στο ρόλο της προσφοράς, δε, υπάγεται η IPυποδομή η οποία προσφέρει υπηρεσίες με «Βέλτιστη Προσπάθεια» χωρίς αστυνόμευση και διαχείριση της κίνησης και του φορτίου. Το τελευταίο, μαζί με τους Shortest Path αλγορίθμους δρομολόγησης συνεπάγεται ότι όλες οι κλήσεις γίνονται δεκτές, χωρίς από – άκρο – εις – άκρο αστυνόμευση και διαχείριση των συνδέσεων, με αποτέλεσμα αφ' ενός κάθε νέα σύνδεση να γίνεται δεκτή μειώνοντας έτσι την ποιότητα των ήδη εγκατεστημένων και αφ' ετέρου δρομολογώντας την κίνηση στα μονοπάτια μικρότερου μήκους αυξάνοντας έτσι τη συμφόρηση σε κάποια

σημεία του δικτύου ενώ παράλληλα αφήνοντας άλλα εντελώς αχρησιμοποίητα.

Η ανάγκη για τη λήψη μέτρων τα οποία να επιλύουν τα ουσιώδη παραπάνω ζητήματα έγινε γρήγορα εμφανής. Πληθώρα ερευνητικών εργασιών και προσπαθειών έφεραν στο προσκήνιο νέους αλγορίθμους για την αστυνόμευση της κίνησης, την αύξηση της ταχύτητας της διακινούμενης πληροφορίας, την μείωση καθυστερήσεων και των διακυμάνσεων της, την μείωση των απωλειών και ούτω καθ' εξής. Βέβαια, ακόμη και λειτουργώντας συμπληρωματικά και όχι αυτόνομα, οι αλγόριθμοι δεν κατάφεραν να επιλύσουν ουσιώδη ζητήματα παροχής ποιότητας υπηρεσίας και διαχείρισης φορτίου και κίνησης από άκρο σε άκρο. Έτσι, η ερευνητική κοινότητα στράφηκε προς την έρευνα πιο ολοκληρωμένων συστημάτων τα οποία να εγγυώνται τη βέλτιστη απόδοση και χρήση των δικτύων τόσο από την πλευρά των χρηστών όσο και από την πλευρά των παρόχων.

Οι ποικίλες, πολύ σημαντικές προσπάθειες, επισκιάστηκαν τελικά από μία νέα αρχιτεκτονική δικτύων υπό το όνομα «Δίκτυα Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων» τα οποία αποτελούν, ουσιαστικά, ένα υβριδικό σύστημα ενσωματώνοντας τα βέλτιστα χαρακτηριστικά των προηγούμενων υποδομών. Τα δίκτυα αυτά, τα οποία μπορούν να χαρακτηριστούν σαν κλιμακούμενοι, ανεξάρτητοι από πρωτόκολλα μηχανισμοί μεταφοράς δεδομένων, επιλύουν τα πολλαπλά προβλήματα σχετικά με την κίνηση, τη διαχείριση εύρους ζώνης καθώς και το IP management τα οποία χαρακτηρίζουν άλλες τεχνολογίες.

Έχοντας, λοιπόν, εστιάσει στα δίκτυα Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων, η επιστημονική κοινότητα στράφηκε στο συνεχή έλεγχο και τη ρύθμιση της κίνησης με το βέλτιστο τρόπο εντός των τοπολογιών αυτού του τύπου. Γνωστή και ως Διαχείριση Κίνησης, η τεχνολογία αυτή έχει σαν γενικότερο στόχο τη βελτιστοποίηση απόδοσης των δικτύων προκειμένου τα τελευταία να λειτουργούν αξιόπιστα με την ταυτόχρονη βελτιστοποίηση κατανομής πόρων και κυκλοφοριακής επίδοσης.

Στο πεδίο της Διαχείρισης Κίνησης, το οποίο είναι ακόμη ανοιχτό σε προτάσεις και έρευνα, αρκετές αξιολογές προσπάθειες και μελέτες έχουν

πραγματοποιηθεί με στόχο την καλύτερη κατανομή της Κίνησης και του Φορτίου εντός των δικτύων Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων με τρόπο τέτοιο ώστε να επιτυγχάνεται άλλοτε το link και άλλοτε το path optimization αλλά και ώστε να πληρούνται οι απαιτήσεις του πρωτοκόλλου είτε μερικώς είτε στο σύνολό τους.

Συνεχίζοντας την προηγούμενη προσπάθεια της ερευνητικής κοινότητας, σκοπός της προκείμενης διπλωματικής εργασίας είναι ο σχεδιασμός και η ανάπτυξη ενός αλγορίθμου διαχείρισης κίνησης και πόρων, ο οποίος να κατανέμει με βέλτιστο τρόπο το φορτίο εντός του δικτύου όντας σε πλήρη συμφωνία με τις απαιτήσεις του MPLS και να επιτυγχάνει τη μέγιστη χρησιμοποίηση της εκάστοτε τοπολογίας.

Στα κεφάλαια που ακολουθούν, τα οποία συγγράφηκαν σε συμφωνία με την μεθοδολογία που ακολουθήθηκε κατά την εκπόνηση της προκείμενης διατριβής, παρουσιάζεται και αναλύεται σε λογικά βήματα η έρευνα που πραγματοποιήθηκε, ο ορισμός του βασικού προβλήματος, ο σχεδιασμός του αλγορίθμου καθώς και τα τελικά αποτελέσματά του σε πλαίσια ποικίλων τοπολογιών. Πιο συγκεκριμένα, στο δεύτερο κεφάλαιο παρουσιάζονται οι πιο σημαντικές προσεγγίσεις στην κατεύθυνση της Διαχείρισης Κίνησης και Πόρων με στόχο την αύξηση της Ποιότητας Υπηρεσίας. Τεχνολογίες και αρχιτεκτονικές όπως το Μοντέλο Ενοποιημένων Υπηρεσιών, το Μοντέλο Διαφοροποιημένων Υπηρεσιών καθώς και η Μεταγωγή Ετικετών Πολλαπλών Πρωτοκόλλων, αναλύονται, συγκρίνονται και αξιολογούνται ως προς τον βασικότερο στόχο που τέθηκε κατά την εκπόνηση της εργασίας.

Στο τρίτο κεφάλαιο, παρουσιάζεται σε μεγαλύτερη ανάλυση η Διαχείριση Κίνησης και Πόρων στο MPLS. Αναλύονται ενδελεχώς όλες οι απαιτήσεις του MPLS Traffic Engineering οι οποίες έχουν προδιαγραφεί στα αντίστοιχα RFCs, ενώ παράλληλα δομείται η εννοιολογική και θεωρητική υποδομή πάνω στην οποία βασίστηκε ο τελικός αλγόριθμος. Στο τέλος, έχοντας αναλύσει τόσο την αρχιτεκτονική της MPLS τεχνολογίας όσο και τις προδιαγραφές που εισήχθησαν στον τομέα του MPLS - TE, παρουσιάζονται και αξιολογούνται προηγούμενες έρευνες και προτάσεις στο θέμα της δρομολόγησης και της κατανομής του φορτίου στο MPLS.

Το τέταρτο κεφάλαιο αφιερώνεται στον ορισμό του μαθηματικού προβλήματος της διατριβής. Αρχικά δίδεται η γενική βάση της μαθηματικής προσέγγισης, ενώ στη συνέχεια παρουσιάζεται η επέκταση αυτής σύμφωνα με την θεωρία του MPLS - TE πεδίου. Κατόπιν, έχοντας θέσει τη μαθηματική θεμελίωση του προβλήματος προς επίλυση και ορμώμενο εκ της παραδοχής ότι στόχος είναι η μεγιστοποίηση της ροής, αναλύεται το γραμμικό πρόβλημα της μέγιστης ροής καθώς και της ελάχιστης τομής. Στο τέλος, παρουσιάζονται, συγκρίνονται και αξιολογούνται οι πιο γνωστοί αλγόριθμοι στο πεδίο της μεγιστοποίησης της ροής πάνω στους οποίους βασίστηκε η κύρια έρευνα κατά το σχεδιασμό του τελικού αλγορίθμου.

Στο πέμπτο κεφάλαιο, έχοντας δει τόσο τις προηγούμενες έρευνες όσο και τη μαθηματική θεμελίωση του προβλήματος, σχεδιάζεται και αναλύεται ο αλγόριθμος TiReD. Παρουσιάζονται τα δεδομένα εισόδου και εξόδου του ενώ παράλληλα αναλύονται ενδελεχώς οι οντότητες και οι παραδοχές που λαμβάνουν χώρα κατά την εκτέλεσή του. Εν συνεχεία, μετά τη θεωρητική ανάλυση του αλγορίθμου και των εννοιών που αυτός εισάγει, παρουσιάζονται με ψευδογλώσσα οι φάσεις και τα βήματα που αυτός ακολουθεί. Στο τέλος του κεφαλαίου δίδεται και μια σχηματική απεικόνιση μέσω Activity Diagrams της ροής του αλγορίθμου.

Το έκτο κεφάλαιο επιδεικνύει τα αποτελέσματα από ποικίλες εκτελέσεις του αλγορίθμου σε πλαίσια αναφοράς μικρών, απλών, μεγάλων και σύνθετων τοπολογιών. Αρχικά παρουσιάζονται οι χρόνοι εκτέλεσης του αλγορίθμου και εξετάζονται τα στοιχεία από τα οποία αυτοί εξαρτώνται. Κατόπιν, αναλύεται ο τρόπος με τον οποίο κατανέμεται το φορτίο σε δύο πλαίσια αναφοράς : σε ένα μικρό και απλό δίκτυο καθώς και σε ένα μεγάλο συμμετρικό. Στο τέλος εξετάζεται η κατανομή της κίνησης, περισσότερο από την σκοπιά της παρεμβολής, στα ίδια πλαίσια αναφοράς με τα προηγούμενα. Όλα τα αποτελέσματα παρουσιάζονται τόσο περιγραφικά όσο και διαγραμματικά, ενώ παράλληλα δίδεται και μία σχετική σύγκριση των αποτελεσμάτων του TiReD αλγορίθμου με τα αντίστοιχα προηγούμενων ερευνών.

Στο έβδομο και τελευταίο κεφάλαιο δίδεται μία σύντομη επισκόπηση της εργασίας και παρουσιάζονται τα συμπεράσματα από την έρευνα που διεξήχθη

καθώς και από τα αποτελέσματα που ελήφθησαν και παρουσιάστηκαν στο προηγούμενο κεφάλαιο. Κλείνοντας, αναφέρονται θέματα προς περαιτέρω μελέτη καθώς και κάποιες από τις διεργασίες που θα πρέπει να εκτελεστούν σε επόμενη, ενδεχομένως, εργασία.

ΜΟΝΤΕΛΑ ΠΑΡΟΧΗΣ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΑΣ

ΕΙΣΑΓΩΓΗ

Η ραγδαία εξάπλωση του Διαδικτύου και της επιστήμης της πληροφορικής γενικότερα, έφερε στο προσκήνιο ποικίλους τύπους εφαρμογών, δημιουργώντας, ουσιαστικά, τρεις τομείς υπηρεσιών: παραδοσιακές (όπως το FTP, ο Παγκόσμιος Ιστός και το ηλεκτρονικό ταχυδρομείο), επιχειρηματικές (όπως τα ERP, SCM και CRM συστήματα) και πραγματικού χρόνου (για παράδειγμα βιντεοδιασκέψεις, IP τηλεφωνία, ηλεκτρονική μάθηση και τηλεσυνδέσεις) [22]. Το γεγονός ότι οι παραπάνω υπηρεσίες απαιτούν IP δίκτυα, κάνει προφανή την ανάγκη αφ' ενός για τη διάθεσή τους εντός πλαισίου υψηλής απόδοσης και ποιότητας και αφ' ετέρου την απαίτηση για την παροχή ασφάλειας, εμπιστοσύνης και τρόπων διαχείρισής τους.

Στοχεύοντας προς την κατεύθυνση αυτή, η δημιουργία της IP υποδομής μπορεί να χωριστεί σε τρεις λογικές (όχι απαραίτητα χρονικές) φάσεις. Η πρώτη φάση (2001 και πριν) χαρακτηρίστηκε κυρίως από μικρή ποικιλομορφία υπηρεσιών οι οποίες περιορίζονταν σε υπηρεσίες μεταφοράς δεδομένων υπό μορφή κειμένου ή εικόνας και παρέχονταν με «Βέλτιστη Προσπάθεια» (Best Effort) που σήμαινε : καμία πρόβλεψη για τον έλεγχο της από άκρο εις άκρο καθυστέρησης και ρυθμαπόδοσης, καμία οριοθέτηση διακυμάνσεων καθυστέρησης (jitter), απωλειών και ρυθμού μετάδοσης και κανένας έλεγχος αποδοχής. Τα προηγούμενα μπορούν εύκολα να σημαίνουν ότι σε ένα φορτωμένο δίκτυο οποιαδήποτε σύνδεση θα γίνει δεκτή, δε θα πλαισιώνεται από κανένα περιορισμό ποιότητας και ταυτόχρονα θα επιβαρύνει ανεξέλεγκτα την απόδοση των ήδη εγκατεστημένων συνδέσεων.

Στη δεύτερη φάση έγινε προσπάθεια επίλυσης των αδυναμιών της πρώτης μέσω της ενίσχυσης δυνατοτήτων ελέγχου πρόσβασης και διαχείρισης συνδέσεων. Συγκεκριμένα, άρχισαν να υλοποιούνται μηχανισμοί και τεχνολογίες για την επίλυση των παραπάνω προβλημάτων με τα πιο γνωστά παραδείγματα να αποτελούν: η προσωρινή αποθήκευση ως λύση για τις διακυμάνσεις καθυστέρησης, η μορφοποίηση κίνησης για την εξομάλυνση του μέσου ρυθμού μετάδοσης δεδομένων και των ριπών (π.χ. Αλγόριθμος Τρύπιου Κουβά, Turner 1986 και Αλγόριθμος Κουβά Κουπονιών), ο

χρονοπρογραμματισμός πακέτων για την πιο «δίκαιη» επεξεργασία τους στις ουρές δρομολογητών (π.χ. Fair Queuing, Nagle 1987 και Weighted Fair Queuing) και άλλα. Παράλληλα, μεγάλη προσοχή άρχισε να δίνεται στην φιλοσοφία της δέσμευσης πόρων στους κόμβους του δικτύου, όπως και στον έλεγχο αποδοχής.

Φυσικά, οι τεχνικές αυτές δεν παρέχουν από μόνες τους αποδοτική και αξιόπιστη ποιότητα υπηρεσιών με το βέλτιστο τρόπο, καθώς η κάθε μία καλύπτει ένα μικρό τομέα στο συνολικό πρόβλημα της παροχής Ποιότητας Υπηρεσιών. Είναι ουσιώδης, λοιπόν, η συνδυαστική και αλληλοσυμπληρούμενη εφαρμογή τους σε περιβάλλοντα δικτύων που να τις υποστηρίζουν ή και να τις βελτιστοποιούν. Σύμφωνα με το [8], ο βασικός τρόπος για την επίτευξη QoS είναι η παροχή διαφοροποιημένων υπηρεσιών σε επίπεδο πακέτων ή συνεδριών, με στόχο: πρώτον, την απόδοση υψηλότερης προτεραιότητας σε έναν χρήστη έναντι κάποιου άλλου και δεύτερον τον έλεγχο της χρήσης bandwidth έτσι ώστε η ποιότητα να μπορεί πλέον να εγγυηθεί. Έτσι, φτάνουμε στην Τρίτη φάση η οποία χαρακτηρίζεται από την διαχείριση κίνησης για την παροχή Ποιότητας Υπηρεσίας μέσω διαφοροποίησης των υπηρεσιών. Τα βασικά μοντέλα που κυριάρχησαν στη φάση αυτή, είναι τρία: Το πρώτο, είναι βασισμένο σε ροές και ακολουθεί τη φιλοσοφία της αναγνώρισης και δέσμευσης των πόρων πριν την εγκατάσταση της συνεδρίας. Το δεύτερο, που βασίζεται σε τάξεις προκειμένου να αποφύγει την δέσμευση πόρων ανά ροή, διαφοροποιεί την κίνηση σε ένα σύνολο κλάσεων, αναθέτοντας διαφορετικούς βαθμούς προτεραιότητας σε κάθε μία. Τέλος το τρίτο, εστιάζει στην ταχύτερη προώθηση πακέτων μέσα σε ένα πλαίσιο Ποιότητας Υπηρεσιών, πολλαπλών πρωτοκόλλων και εύκολης διαχειρισιμότητας. Οι παραπάνω προσεγγίσεις ακολουθούνται από τα μοντέλα Ενοποιημένων Υπηρεσιών (IntServ), Διαφοροποιημένων Υπηρεσιών (DiffServ) και Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων (MPLS), αντίστοιχα. Στις υποενότητες που ακολουθούν, θα εξεταστούν τα βασικά τους χαρακτηριστικά, εστιάζοντας κυρίως στις δυνατότητες παροχής Ποιότητας Υπηρεσιών που αυτά προσφέρουν.

ΜΟΝΤΕΛΟ ΕΝΟΠΟΙΗΜΕΝΩΝ ΥΠΗΡΕΣΙΩΝ

Ανάμεσα στο 1995 και το 1997, καταβλήθηκαν μεγάλες προσπάθειες για την επινόηση μίας αρχιτεκτονικής η οποία να μπορεί να υποστηρίξει την συνεχή ροή πολυμέσων [1]. Το γενικό όνομα της προσπάθειας αυτής είναι «αλγόριθμοι που βασίζονται σε ροές» ή «ολοκληρωμένες υπηρεσίες» (Integrated Services, IntServ) και έχει ως στόχο τόσο τις εφαρμογές αποκλειστικής διανομής όσο και τις εφαρμογές πολυδιανομής. Το IntServ ορίζει πολλαπλές κατηγορίες τηλεπικοινωνιακής κίνησης (traffic classes) οι οποίες έχουν διαφορετικές απαιτήσεις από το διαδίκτυο όσον αφορά την ποιότητα της υπηρεσίας και βασίζεται στην προκαταβολική δέσμευση πόρων ανά ροή, κατά μήκος της διαδρομής της τελευταίας. Έτσι, εκτός από την κλάση Βέλτιστης Προσπάθειας, η οποία χαρακτηρίζεται, όπως αναφέρθηκε πιο πάνω, από την απουσία οποιουδήποτε QoS περιορισμού αφήνοντας το δίκτυο να προσφέρει την καλύτερη δυνατή ποιότητα, ορίζονται άλλες δύο κλάσεις: η κλάση Εγγυημένης Υπηρεσίας και η κλάση Ελεγχόμενου Φορτίου.

ΚΛΑΣΗ ΕΓΓΥΗΜΕΝΗΣ ΥΠΗΡΕΣΙΑΣ (GUARANTEED SERVICES)

Η κλάση εγγυημένης υπηρεσίας παρέχει στους χρήστες εγγυημένη ποσότητα εύρους ζώνης, μηδενικές απώλειες πακέτων στις ουρές αναμονής και σταθερά όρια από άκρο – εις – άκρο καθυστέρησης. Οι παράμετροι αυτοί προσυμφωνούνται μεταξύ του υπολογιστή υπηρεσίας και του δικτύου και μπορούν να εγγυηθούν μόνο εάν τηρηθεί ορθά η συμφωνία (για παράδειγμα, εάν η πηγή αποστέλλει πακέτα με τον προσυμφωνημένο ρυθμό ή εντός καλά ορισμένων τιμών άνω και κάτω ορίου ρυθμού αποστολής).

Λόγω της απαίτησης σε μορφοποίηση της κίνησης, την σωστή τήρησή της και την ελαχιστοποίηση της καθυστέρησης, η κλάση αυτή αφορά περισσότερο πολυμεσικές υπηρεσίες που έχουν πολύ αυστηρές απαιτήσεις αναφορικά με

τον συνολικό χρόνο παράδοσης (π.χ. Ήχος, βίντεο) και η προσωρινή τους αποθήκευσή τους γίνεται σε ενταμιευτές μικρής, σταθερής χωρητικότητας.

ΚΛΑΣΗ ΕΛΕΓΧΟΜΕΝΟΥ ΦΟΡΤΙΟΥ (CONTROLLED LOAD CLASS)

Η κλάση Ελεγχόμενου Φορτίου εγγυάται στους χρήστες ότι θα λάβουν υπηρεσίες σε επίπεδο ποιότητας ίσο με αυτό της Best Effort κλάσης υπό συνθήκη χαμηλού δικτυακού φόρτου. Παρέχει την ίδια ποιότητα τόσο κάτω από συνθήκες υπερφόρτωσης δικτύου όσο και κάτω από ελαφριά κίνηση, εξαλείφοντας έτσι την επιδείνωση της ποιότητας υπηρεσιών λόγω αύξησης της τηλεπικοινωνιακής κίνησης. Επιπρόσθετα, κρίνεται καταλληλότερη για πολυμεσικές εφαρμογές που απαιτούν διατηρούμενο ρυθμό μετάδοσης, έχουν σχετική ανοχή σε απώλειες πακέτων και καθυστερήσεις και παρουσιάζουν προσαρμοστικότητα σε χαρακτηριστικά όπως η διακύμανση καθυστέρησης.

Παράλληλα, όπως η προηγούμενη, έτσι και αυτή η κλάση απαιτεί εκ των προτέρων συμφωνία με το δίκτυο αναφορικά με τα χαρακτηριστικά κίνησης και τη δέσμευση των απαιτούμενων πόρων από το τελευταίο. Εάν η αναφερθείσα διαδικασία ολοκληρωθεί με επιτυχία και διατηρηθεί κατά τη διάρκεια της συνεδρίας, η αιτούμενη υπηρεσία θα παρέχεται στην ίδια ποιότητα ανεξάρτητα από τα επίπεδα φορτίου εντός του δικτύου.

ΜΗΧΑΝΙΣΜΟΙ ΠΑΡΟΧΗΣ ΠΟΙΟΤΗΤΑΣ ΥΠΗΡΕΣΙΑΣ

Προκειμένου η εκάστοτε αρχιτεκτονική διαδικτύου να μπορεί να υποστηρίξει τις δυνατότητες του IntServ, πρέπει να ισχύουν τρεις βασικές προϋποθέσεις: Πρώτον, πρέπει να υπάρχει ένας μηχανισμός μέσω του οποίου οι εφαρμογές θα γνωστοποιούν στα στοιχεία του δικτύου τους περιορισμούς τους και τις απαιτήσεις τους σε ποιότητα υπηρεσιών. Δεύτερον, τα στοιχεία του δικτύου θα πρέπει να «απαντάνε» στις απαιτήσεις αυτές, συγκρίνοντας τους πόρους

που μπορούν να διαθέσουν με τους πόρους που ζητούνται και να αποδέχονται ή να απορρίπτουν τις κλήσεις. Τρίτον, εάν οι κλήσεις τύχουν αποδοχής και δεσμευτούν οι πόροι, οι κόμβοι θα πρέπει να μπορούν να εκπληρώσουν τις υποσχέσεις τους, μέσω μηχανισμών ελέγχου ποιότητας.

Ο έλεγχος της ποιότητας πραγματοποιείται κυρίως στους δρομολογητές, οι οποίοι ανάλογα με τις απαιτήσεις κάθε ροής, εφαρμόζουν τους αντίστοιχους μηχανισμούς χρονοπρογραμματισμού, ταξινόμησης, μορφοποίησης κ.ο.κ. Οι μηχανισμοί αυτοί, οι οποίοι εκτελούνται σε κάθε δρομολογητή και μπορούν να διαφέρουν από κόμβο σε κόμβο, είναι γνωστοί με τον όρο «Per Hop Behaviors» (PHBs) και στόχος τους είναι η παροχή «δίκαιης» εξυπηρέτησης στις συνεδρίες που διασχίζουν το διαδίκτυο. Έτσι, παραδείγματος χάριν, όποιος και να είναι ο εκτελούμενος αλγόριθμος προγραμματισμού σε οποιοδήποτε δρομολογητή, τα πακέτα της ουράς αναμονής θα εξυπηρετηθούν σύμφωνα με σειρά προτεραιότητας η οποία εξαρτάται από την κλάση στην οποία αυτά ανήκουν ή το βαθμό της απαιτούμενης ποιότητας υπηρεσίας.

Ο έλεγχος της αποδοχής, δε, όπως αναφέρθηκε πιο πάνω, έγκειται στην απόφαση της αποδοχής ή της απόρριψης μίας κλήσης η οποία βασίζεται στην σύγκριση των στοιχείων, που ζητά η ροή, με τους πόρους που διαθέτει ο εκάστοτε δρομολογητής. Το συγκεκριμένο ζήτημα, αν και μοιάζει σαν μία απλή σύγκριση εύρους ζώνης, χωρητικότητας ενταμιευτών και κύκλων επεξεργαστή, είναι πολύ πιο περίπλοκο και δύσκολο. Πρέπει να τονιστεί ότι οι περισσότερες εφαρμογές δεν γνωρίζουν τις απαιτήσεις τους σε μνήμη ή κύκλους επεξεργαστή. Επιπρόσθετα, οι απαιτήσεις τους μπορούν να μην αφορούν μόνο στους αναφερθέντες πόρους, αλλά και σε χαρακτηριστικά όπως ο επιθυμητός ρυθμός μετάδοσης, ο βαθμός ανοχής σε απώλειες και καθυστερήσεις και άλλα. Έτσι, το πρόβλημα σύγκρισης του «προφίλ» της ροής μπορεί τελικά να αναλυθεί σε παραμέτρους όπως ο ρυθμός και το μέγεθος κουβά κουπονιών, ο μέγιστος και ελάχιστος ρυθμός μετάδοσης, το μέγιστο και ελάχιστο μέγεθος των πακέτων, το ποσοστό ανοχής σε απώλειες και ούτω καθ' εξής.

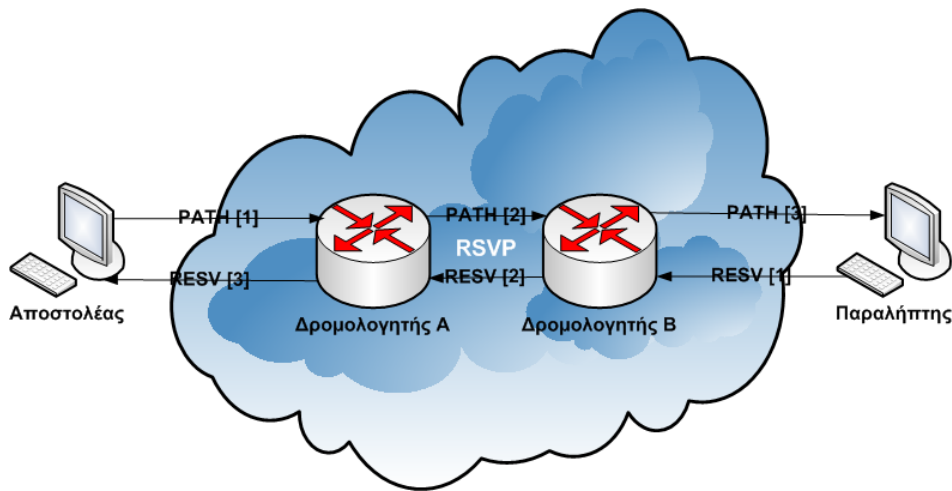
Τέλος, για να μπορεί να γίνει η παραπάνω σύγκριση των παραμέτρων και κατ' επέκταση η δέσμευση των πόρων ή η απόρριψη της κλήσης, πρέπει να υπάρχει ένας μηχανισμός ο οποίος θα τις μεταφέρει σε όλους τους κόμβους μεταξύ του αποστολέα και του παραλήπτη. Για την υποστήριξη αυτής της απαίτησης χρησιμοποιείται το Πρωτόκολλο Δέσμευσης Πόρων ή RSVP (Resource ReserVation Protocol) το οποίο είναι, ουσιαστικά, ένας μηχανισμός σηματοδότησης που δεσμεύει πόρους κατά μήκος μίας διαδρομής και υποστηρίζει την πολυδιανομή, την αλλαγή καναλιών από μεμονωμένους χρήστες καθώς και τη μεταβολή των παραμέτρων ποιότητας υπηρεσίας χωρίς την κατάργηση και επανεγκατάσταση ροών. Η λειτουργία του RSVP μπορεί να περιγραφεί ως ακολούθως:

Ο αποστολέας μίας εφαρμογής αποστέλλει PATH μηνύματα στους παραλήπτες, τα οποία περιέχουν τις προδιαγραφές κίνησης που θα χρησιμοποιηθούν για τη δέσμευση. Με τον τρόπο αυτόν, ενημερώνει τόσο το δέκτη όσο και όλους τους ενδιάμεσους δρομολογητές για τα χαρακτηριστικά της κίνησης που θα αρχίσει να εκπέμπει.

Κάθε δρομολογητής, που λαμβάνει το PATH μήνυμα, πραγματοποιεί δύο ελέγχους: Πρώτον, εξακριβώνει την ταυτότητα του δέκτη και τα δικαιώματά του για τη συγκεκριμένη δέσμευση πόρων από το δίκτυο (Policy Control) και δεύτερον, ελέγχει εάν υπάρχουν αρκετοί διαθέσιμοι εσωτερικοί πόροι για την εξυπηρέτηση της συγκεκριμένης δέσμευσης (Flow Admission Control). Εάν οποιοσδήποτε από τους δύο αυτούς ελέγχους αποτύχει, αποστέλλεται το κατάλληλο μήνυμα σφάλματος (RESV_ERR) στον αποστολέα της εφαρμογής.

Εάν οι έλεγχοι δώσουν θετικό αποτέλεσμα, τίθεται ο χρονιστής επανεκκίνησης για τη συγκεκριμένη ροή. Ο χρονιστής αυτός χρησιμοποιείται, ουσιαστικά, για τη διασφάλιση της σωστής λειτουργίας του δικτύου, έτσι ώστε ο δρομολογητής να είναι πάντα ενήμερος για την κατάσταση της συγκεκριμένης ροής. Για παράδειγμα, εάν σε χρονικό διάστημα ΔT ο δρομολογητής δε λάβει μήνυμα ανανέωσης από τον πομπό, μπορεί να θεωρήσει ότι ο τελευταίος σταμάτησε να εκπέμπει κίνηση και, αφού ανατρέξει στις αντίστοιχες εγγραφές της βάσης δεδομένων, να αποδεσμεύσει όλες τις πόρους που είχαν συσχετιστεί με αυτόν. Μετά την ενημέρωση της μονάδας χρονοδρομολόγησης

πακέτων για τη νέα ροή, ο δρομολογητής αποστέλλει το PATH μήνυμα στον επόμενο δρομολογητή.



Εικόνα 1. RSVP - Ανταλλαγή PATH και RESV μηνυμάτων

Όταν το μήνυμα PATH φτάσει στους δέκτες, ο κάθε ένας από αυτούς θα απαντήσει με ένα μήνυμα RESV στο οποίο προσδιορίζει τους πόρους που επιθυμεί να δεσμεύσει από το δίκτυο. Πρέπει να τονιστεί εδώ, ότι μία εφαρμογή ή υπηρεσία μπορεί να έχει άνω του ενός δέκτη, κάθε ένας από τους οποίους μπορεί να έχει διαφορετικές απαιτήσεις σε σχέση με τους υπολοίπους. Στην περίπτωση αυτή, οι ενδιαμέσοι δρομολογητές θα αποστείλουν στους επόμενους ένα μόνο μήνυμα RESV από το σύνολο των έγκυρων υπαρχόντων μηνυμάτων, του οποίου η τιμή για το εύρος ζώνης που απαιτείται είναι η μεγαλύτερη.

Το αποτέλεσμα της παραπάνω διαδικασίας είναι (σε περιπτώσεις δίχως σφάλματα) η δέσμευση όλων των απαραίτητων πόρων για τη δοθείσα ροή κατά μήκος της διαδρομής που αυτή θα ακολουθήσει. Αξίζει να τονιστεί ότι η ανταλλαγή των PATH και RESV μηνυμάτων δεν τελειώνει εδώ, καθώς ο αποστολέας και οι δέκτες συνεχίζουν να τα αποστέλλουν περιοδικά για να ανανεώνουν την κατάσταση των πόρων.

ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ INTSERV (ΚΑΙ RSVP)

Το RSVP, με τις τρεις κλάσεις υπηρεσίας που αναφέρθηκαν παραπάνω, μπορεί να παράσχει απόλυτη εγγύηση υπηρεσίας, παρακολουθώντας συνεχώς κάθε ροή ώστε να αποτρέπεται η κατανάλωση περισσοτέρων πόρων από αυτούς που έχουν δεσμευτεί αρχικά και να αντιδρά άμεσα σε αλλαγές τοπολογίας του δικτύου ή της συμπεριφοράς των αποστολέων/δεκτών. Παράλληλα, όπως παρουσιάστηκε, δίνει τη δυνατότητα λεπτομερούς προδιαγραφής της κατηγορίας υπηρεσίας που αιτούν οι RSVP clients, όχι μόνο από την άποψη εύρους ζώνης, μνήμης και κύκλων επεξεργαστή, αλλά και άλλων σημαντικών παραμέτρων όπως τα μήκη των πακέτων, τα ανώτατα όρια καθυστερήσεων, ο βαθμός ανοχής στο jitter και τις απώλειες πακέτων και άλλα.

Επιπρόσθετα, χρησιμοποιεί τα ήδη υπάρχοντα πρωτόκολλα δρομολόγησης για τον ορισμό του μονοπατιού της κάθε ροής, μειώνοντας την απαίτηση για σχεδιασμό και ανάπτυξη νέων αλγορίθμων ή μηχανισμών για το σκοπό αυτό, ενώ παράλληλα δεν απαιτεί την χρήση αποκλειστικά RSVP οντοτήτων δικτύου, επιτρέποντας και σε non – RSVP δρομολογητές να προωθούν τα μηνύματα PATH και RESV. Σχεδιασμένο να επιτρέπει τόσο unicast όσο και multicast ροές, επιτρέπει τα RESV μηνύματα που σχετίζονται με ένα PATH, να συνδυάζονται σε μία μόνο αίτηση στα σημεία συγχώνευσης (η οποία απεικονίζει τη μέγιστη απαίτηση του συνόλου των απαιτήσεων), μειώνοντας με αυτόν τον τρόπο το overhead του δικτύου που θα παρουσιαζόταν εάν τα μηνύματα αποστέλλονταν στο σύνολό τους, σε διαφορετικά κανάλια.

Παρ' όλο, εντούτοις, που τα πλεονεκτήματα του IntServ είναι σημαντικά, το μοντέλο αυτό εισάγει και εξαιρετικά μειονεκτήματα τα οποία δεν μπορούν να αγνοηθούν. Ένα από τα κυριότερα είναι ο μεγάλος όγκος πληροφορίας που απαιτείται για την κατάσταση κάθε ροής. Η πληροφορία αυτή αφορά τον προσδιορισμό και την ταυτότητα της ροής, τους πόρους που αυτή καταναλώνει, την πληροφορία της αστυνόμευσης της κίνησης και του προγραμματισμού της σύμφωνα με τις υποχρεώσεις που προκύπτουν από την προκαθορισμένη δέσμευση πόρων και άλλα. Σε, διόλου απίθανες,

περιπτώσεις που θα υπήρχαν εκατομμύρια τέτοιες συνδέσεις, το πρόβλημα του όγκου της αποθηκευμένης πληροφορίας μπορεί να γίνει προφανές. Επιπλέον, η κατάσταση γίνεται ακόμα χειρότερη εάν αναλογιστεί κανείς ότι το μεγαλύτερο μέρος της τηλεπικοινωνιακής κίνησης αποτελείται από συνεδρίες μικρής διάρκειας και λίγων απαιτήσεων σε ποιότητα.

Ένα άλλο πολύ σημαντικό πρόβλημα του RSVP signaling μηχανισμού είναι το γεγονός ότι λειτουργεί είτε αποκτώντας πόρους σε όλους τους δρομολογητές του μονοπατιού για μία ροή, είτε απεγκαθιστώντας τελείως τη σύνδεση ακόμα και εάν μόνο ένας δρομολογητής δεν έχει τους απαραίτητους πόρους τη στιγμή που πραγματοποιήθηκε η αίτηση. Ακόμα, δε, και αν οι πόροι γίνουν διαθέσιμοι αμέσως μετά, η δέσμευση δεν γίνεται αποδεκτή και ο χρήστης (ή το σύστημα του άκρου) δεν πληροφορείται ούτε για αυτό το γεγονός, ούτε και για την πιθανή μελλοντική διαθεσιμότητα των πόρων, ώστε να μπορεί να υπολογίσει τον χρόνο στον οποίο μπορεί να προσπαθήσει ξανά. Σε συνθήκες μεγάλου φόρτου, αυτό μπορεί να οδηγήσει σε αλληπάλληλες προσπάθειες δέσμευσης πόρων από την πλευρά του χρήστη, αυξάνοντας το overhead του δικτύου από άποψη πλήθους πακέτων και φόρτου επεξεργασίας αυτών. Το γεγονός, μάλιστα, ότι τα signaling μηνύματα έχουν μεγαλύτερες προτεραιότητες, σημαίνει ότι η αύξησή τους μπορεί να διαταράξει σημαντικά τις άλλες υπηρεσίες.

Για το πρόβλημα αυτό, προτάθηκε η λύση των Αναβεβλημένων Δεσμεύσεων [17] οι οποίες δίνουν τη δυνατότητα στους δρομολογητές να αναβάλλουν την μη αποδοχή μίας δέσμευσης όταν δεν υπάρχουν αρκετοί από άκρο – σε – άκρο πόροι. Έτσι, ενώ τυπικά το εύρος ζώνης μίας ζεύξης μπορεί να βρίσκεται σε δεσμευμένη ή ελεύθερη κατάσταση, ο προταθέν αλγόριθμος προσθέτει και μία τρίτη, την «προκρατημένη». Η περίοδος αναβολής υπολογίζεται από τους δρομολογητές λαμβάνοντας υπ' όψιν την διαθέσιμη γνώση της κατάστασης του δικτύου, ενώ ο δέκτης της υπηρεσίας μπορεί να ορίσει, πλέον, και το μέγιστο χρονικό διάστημα στο οποίο είναι διατεθειμένος να παραμείνει σε «αναμονή». Τα πειραματικά αποτελέσματα της εφαρμογής του εν λόγω μηχανισμού έδειξαν ότι η χρησιμοποίηση των ζεύξεων μπορεί να αυξηθεί μέχρι και δύο φορές, ενώ παράλληλα μεγιστοποιείται σημαντικά και η πιθανότητα αποδοχής μίας κλήσης. Επιπρόσθετα, παρ' όλο που υπό

συνθήκες αποστολής ριπών για μεγάλα χρονικά διαστήματα, η ποιότητα υπηρεσίας υφίσταται σχετική υποβάθμιση, αυτή είναι σημαντικά μικρότερη από εκείνη που θα παρατηρούνταν σε μη – Αναβεβλημένες Δεσμεύσεις. Τέλος, αξίζει να τονιστεί ότι ο προταθέν αλγόριθμος παρουσιάζει εξαιρετικά αποτελέσματα σε συνθήκες μεγάλου φόρτου ροών στο δίκτυο, προσφέροντας δυνατότητες κλιμάκωσης.

Εντούτοις, εάν και το πρόβλημα του «ή όλα ή τίποτα» που αναφέρθηκε παραπάνω, φαίνεται να δέχεται λύση, τα μειονεκτήματα του IntServ παραμένουν ακόμα αρκετά. Ο τρόπος λειτουργίας του το καθιστά ανεπίτρεπτο σε περιβάλλοντα πολλών συνδέσεων ή σε μεγάλα δίκτυα καθώς αφ' ενός δεν προσφέρει καμία δυνατότητα κλιμάκωσης και αφ' ετέρου οι αλλαγές που απαιτούνται στον κώδικα των δρομολογητών προκειμένου να το υποστηρίξουν είναι πολύ σημαντικές και συνεπώς ακριβές και χρονοβόρες. Ως απάντηση, λοιπόν, στους αναφερθέντες περιορισμούς του IntServ/RSVP, η IETF επινόησε μία απλούστερη προσέγγιση ως προς την ποιότητα υπηρεσιών, η οποία μπορεί να υλοποιηθεί εύκολα σε κάθε δρομολογητή χωρίς την εκ των προτέρων διευθέτηση και χωρίς να εμπλέκεται στο σύνολό της η διαδρομή των πακέτων. Η τεχνολογία αυτή είναι βασισμένη σε τάξεις (σε αντιδιαστολή με το βασισμένο σε ροές IntServ) και είναι γνωστή με το όνομα «Διαφοροποιημένες Υπηρεσίες» (Differentiated Services).

ΜΟΝΤΕΛΟ ΔΙΑΦΟΡΟΠΟΙΗΜΕΝΩΝ ΥΠΗΡΕΣΙΩΝ

Η αρχιτεκτονική Διαφοροποιημένων Υπηρεσιών σχεδιάστηκε για την υποστήριξη ποιότητας υπηρεσίας τοπικά σε κάθε δρομολογητή, αποφεύγοντας την εκ των προτέρων δέσμευση πόρων κατά μήκος της διαδρομής κάθε ροής και περιγράφεται στα έγγραφα RFC 2474, 2475 και πολλά άλλα. Έτσι, ενώ το IntServ προσφέρει εγγυήσεις ανά ροή, το DiffServ ακολουθεί την φιλοσοφία της άθροισης πολλαπλών ροών σε μερικά επίπεδα

υπηρεσίας προσφέροντας καλύτερη δυνατότητα κλιμάκωσης διατηρώντας, πλέον, την πληροφορία κατάστασης ανά κλάση και όχι ανά ροή.

Η βασική φιλοσοφία πίσω από το DiffServ μπορεί να περιγραφεί πολύ συνοπτικά ως εξής: Ένα σύνολο δρομολογητών σχηματίζει μία διαχειριστική περιοχή (π.χ. ISP). Η διαχειριστική αρχή, ορίζει ένα σύνολο τάξεων υπηρεσιών, με την κάθε τάξη να έχει συγκεκριμένους κανόνες προώθησης (δηλαδή, συγκεκριμένους μηχανισμούς ή τύπους PHBs ανά τάξη). Παράλληλα, τα πακέτα ενός πελάτη που επιθυμεί να χρησιμοποιήσει τις διαφοροποιημένες υπηρεσίες, διαφοροποιούνται από το πεδίο Τύπου Υπηρεσίας (Type Of Service, ToS) που φέρουν. Έτσι, πακέτα με τύπο υπηρεσίας υψηλότερης τάξης, θα τυγχάνουν καλύτερης εξυπηρέτησης από άλλα χαμηλότερης.

Το ToS πεδίο, το οποίο τοποθετείται στην IP επικεφαλίδα, έχει μήκος 1 byte και αποτελείται από δύο τμήματα: το πρώτο τμήμα, μήκους 6 bits αποτελεί το Differentiated Service Code Point (DSCP) και είναι αυτό που διαφοροποιεί τα πακέτα ως προς τις κλάσεις. Το δεύτερο τμήμα, μήκους 2 bits, μένει αχρησιμοποίητο. Για να κατασκευαστεί, λοιπόν, ένας τύπος υπηρεσίας, ακολουθείται η λογική δύο βημάτων που συνοψίζεται ως ακολούθως:

1. Τα άκρα του δικτύου καθορίζουν και καταχωρούν την τιμή του DSCP πεδίου. Ουσιαστικά, δηλαδή, μαρκάρουν τα πακέτα, βασισμένα στις απαιτήσεις και τους κανόνες κάθε υπηρεσίας, καθώς και στο Service Level Agreement (SLA) που έχει προσυμφωνηθεί μεταξύ του πελάτη και του δικτύου
2. Οι ενδιάμεσοι κόμβοι χρησιμοποιούν το πεδίο για να αποφασίσουν ως προς τον τρόπο προώθησης των πακέτων

Συνεπώς το DiffServ επιτυγχάνεται με την αντιστοίχιση του DSCP σε συγκεκριμένες μεταχειρίσεις ή Συμπεριφορές Προώθησης Ανά-Κόμβο (PHBs), σε κάθε κόμβο δικτύου κατά μήκος της διαδρομής. Για να υποστηριχθούν τα παραπάνω, λοιπόν, στο μοντέλο αυτό ορίζονται τρεις κλάσεις, ή PHBs: Η Κλάση Βέλτιστης Προσπάθειας (Best Effort), η Κλάση «Εσπευσμένης» προώθησης (Expedited Forwarding, EF) και η κλάση «Εξασφαλισμένης»

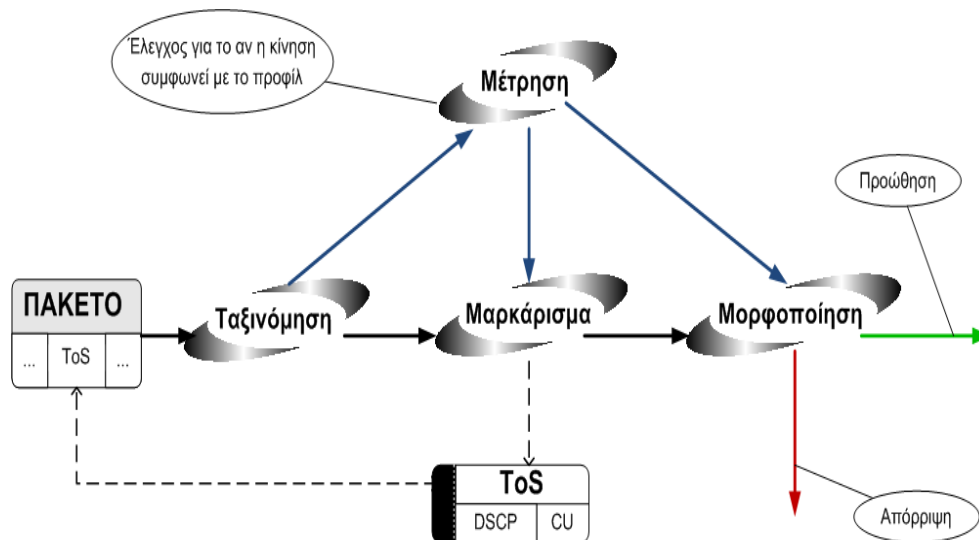
προώθησης (Assured Forwarding, AF) , οι οποίες θα περιγραφούν παρακάτω.

ΚΛΑΣΗ ΕΞΑΣΦΑΛΙΣΜΕΝΗΣ ΠΡΟΩΘΗΣΗΣ

Η Κλάση Εξασφαλισμένης Προώθησης περιγράφεται στο RFC 2597 και καθορίζει τέσσερις κλάσεις προτεραιότητας, με την κάθε μία να απαιτεί τους δικούς της συγκεκριμένους πόρους. Παράλληλα, ορίζει και τρεις πιθανότητες απόρριψης (χαμηλή, μεσαία, υψηλή) για τα πακέτα που αντιμετωπίζουν συνθήκες συμφόρησης. Το μαρκάρισμα των πακέτων για να υποστηρίξουν τα παραπάνω γίνεται ως εξής: τα τρία πρώτα bits της επικεφαλίδας, απεικονίζουν την κλάση, τα 2 επόμενα καθορίζουν την πιθανότητα απόρριψης και το τελευταίο αφήνεται πάντα μηδενικό. Με τον τρόπο αυτό, τελικά, προσδιορίζονται 12 τύποι υπηρεσιών.

Ένας από τους πιθανούς τρόπους επεξεργασίας των πακέτων στην εξασφαλισμένη προώθηση μπορεί να περιγραφεί ως εξής:

1. Ταξινόμηση των πακέτων σε μία από τις τέσσερις τάξεις προτεραιότητας. Το βήμα αυτό μπορεί να εκτελεστεί στον υπολογιστή υπηρεσίας αποστολής ή στο δρομολογητή εισόδου στο δίκτυο. Το πλεονέκτημα της εκτέλεσης της ταξινόμησης στον υπολογιστή υπηρεσίας αποστολής είναι ότι εκεί υπάρχουν περισσότερες διαθέσιμες πληροφορίες σχετικά με τα πακέτα που ανήκουν σε κάθε ροή.
2. Επισήμανση των πακέτων ανάλογα με την τάξη τους (δηλαδή, μαρκάρισμα της επικεφαλίδας).



Εικόνα 2. Επεξεργασία Πακέτων στο DiffServ

3. Πέρασμα των πακέτων από ένα φίλτρο μορφοποίησης / απόρριψης, το οποίο μπορεί να καθυστερεί ή να απορρίπτει μερικά από αυτά έτσι ώστε να μορφοποιήσει τις τέσσερις ροές σε αποδεκτές μορφές – χρησιμοποιώντας αλγορίθμους όπως ο τρύπιος κουβάς ή ο κουβάς κουπονιών. Αν υπάρχουν πάρα πολλά πακέτα, μερικά από αυτά μπορούν να απορρίπτονται στο σημείο αυτό, ανάλογα με την κατηγορία απόρριψης στην οποία ανήκουν. Είναι επίσης πιθανές και πιο σύνθετες μέθοδοι, οι οποίες συμπεριλαμβάνουν μετρήσεις ή ανάδραση, όπως ο αλγόριθμος Random Early Detection (Jacobson, 1993) και Random Early Detection with In or Out.

Το πού θα εκτελούνται τα τρία παραπάνω βήματα είναι επιλογή του σχεδιαστή του δικτύου. Αξίζει, ωστόσο, να σημειωθεί ότι υπάρχει μεγαλύτερο πλεονέκτημα της εκτέλεσης της ταξινόμησης στον υπολογιστή υπηρεσίας αποστολής, καθώς, όπως ήδη αναφέρθηκε, αυτός κατέχει περισσότερες διαθέσιμες πληροφορίες σχετικά με τα πακέτα που ανήκουν σε κάθε ροή. Βέβαια, οι ενέργειες αυτές μπορούν να εκτελούνται και από ειδικό λογισμικό δικτύου ή ακόμη και από το λειτουργικό σύστημα, έτσι ώστε να μη χρειάζεται να γίνουν αλλαγές στις υπάρχουσες εφαρμογές.

Η εσπευσμένη προώθηση είναι η απλούστερη τάξη, του συνόλου των τάξεων που έχει καθορίσει η IETF, και περιγράφεται στο RFC 3246. Η ιδέα πίσω από την εσπευσμένη προώθηση είναι απλή. Διατίθενται δύο τάξεις υπηρεσιών: απλή και εσπευσμένη. Η συντριπτική πλειοψηφία της κίνησης αναμένεται να είναι απλή, με ένα μικρό ποσοστό των πακέτων να είναι εσπευσμένα. Τα εσπευσμένα πακέτα θα πρέπει να μπορούν να διασχίζουν το υποδίκτυο σαν να μην υπήρχαν παρόντα άλλα πακέτα. Η τάξη αυτή, λοιπόν, υλοποιεί μία υπηρεσία που έχει συγκεκριμένο μέγιστο εύρος ζώνης (peakbit – rate), χαμηλές καθυστερήσεις και jitter, ασήμαντη αναμονή στις ουρές των δρομολογητών και απομόνωση. Ένας τρόπος υλοποίησης της στρατηγικής αυτής είναι ο εξής:

1. Ένας edge router, ο οποίος μέσω κάποιου πρωτοκόλλου σηματοδότησης είναι ενημερωμένος για όλες τις ροές πακέτων που επιθυμούν ένα συγκεκριμένο επίπεδο υπηρεσίας, ελέγχει τα πακέτα για τη συμμόρφωσή τους με το προφίλ της κίνησης της συγκεκριμένης ροής και μαρκάρει τα πακέτα ως Expedited ή Assured ανάλογα με την κλάση τους. Στην περίπτωση που το πακέτο δεν συμφωνεί με το προφίλ (Out of Profile), είτε δεν το μαρκάρει καθόλου είτε το ανάγει σε κλάση Βέλτιστης Προσπάθειας.
2. Ο δρομολογητής, ο οποίος έχει δύο ουρές εξόδου ανά εξερχόμενη γραμμή (μία για τα εσπευσμένα και μία για τα απλά πακέτα), τοποθετεί το εισερχόμενο πακέτο στην κατάλληλη ουρά. Σημαντικό πλεονέκτημα της προσέγγισης αυτής αποτελεί το γεγονός ότι η ουρά υψηλής προτεραιότητας δεν πρόκειται να υπερχειλίσει ποτέ, αφού ο αριθμός των Expedited Class πακέτων ελέγχεται με αυστηρά κριτήρια στην είσοδο καταλαμβάνοντας πάντα ένα κλάσμα του εύρους του διαύλου. Το υπόλοιπο εύρος, δε, μπορεί να δοθεί στις κλάσεις χαμηλότερης προτεραιότητας.

3. Εάν το μέγεθος ουράς περάσει ένα δοθέν κατώφλι τότε εκτελείται ο μηχανισμός απόρριψης των πακέτων σύμφωνα με την κλάση στην οποία ανήκουν. Συγκεκριμένα, πρώτα απορρίπτονται τα Best Effort πακέτα και κατόπιν τα πακέτα της Εγγυημένης Κλάσης, σύμφωνα πάντα με το πεδίο της Προτεραιότητας Απόρριψης που εμφανίζεται στο DSCP.
4. Παράλληλα με τα παραπάνω, ένας Μεσίτης Εύρους Ζώνης (Bandwidth Broker, BB) υλοποιεί μηχανισμούς αποδοχής κλήσεων, αρχικοποιεί τα στοιχεία του δικτύου, διασφαλίζει τους πόρους στο δίκτυο και προστατεύει τις γραμμές των γειτονικών παρόχων. Έτσι, προκειμένου ένας αποστολέας να εγκαταστήσει μία σύνδεση, πρώτα επικοινωνεί με τον BB ο οποίος εφαρμόζοντας προκαθορισμένες πολιτικές εξακριβώνει την γνησιότητα του χρήστη και εφαρμόζει μηχανισμούς ελέγχου αποδοχής. Εάν ο έλεγχος πραγματοποιηθεί επιτυχώς, η αίτηση του χρήστη προωθείται στον γειτονικό BB ο οποίος με τη σειρά του την επεξεργάζεται και την προωθεί στον επόμενο BB (ή στον παραλήπτη). Παράλληλα, ο κάθε BB ρυθμίζει τους δρομολογητές της περιοχής του, ώστε να υποστηρίξει το ζητούμενο περίγραμμα υπηρεσίας (service profile).

ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ DIFFSERV

Όπως ήδη αναφέρθηκε, στο DiffServ μοντέλο η πληροφορία κατάστασης διατηρείται ανά κλάση και όχι ανά ροή. Αυτό σημαίνει ότι ενώ στο IntServ οι δρομολογητές αποθηκεύουν σχετικά δεδομένα για την κάθε ροή που αναλαμβάνουν, το DiffServ το κάνει για κάθε κλάση. Διατηρώντας το πλήθος των διαφορετικών κλάσεων μικρό, μειώνει τη σηματοδότηση και συνεπώς και το overhead του δικτύου, παρουσιάζοντας έτσι εξαιρετική κλιμάκωση. Επιβάλλοντας, δε, λειτουργίες ελέγχου πολιτικών στα άκρα και παρέχοντας απλή μεταχείριση συσσωρευμένων δεδομένων στον πυρήνα του δικτύου, διασφαλίζει νέες υπηρεσίες IP χωρίς υπερβολική αποθήκευση πληροφορίας

κατάστασης ή ακριβών αποφάσεων προώθησης στους δρομολογητές του δικτύου πυρήνα.

Το γεγονός ότι η μέθοδος αυτή δεν απαιτεί ούτε εκ των προτέρων διευθέτηση, ούτε δέσμευση πόρων, ούτε χρονοβόρες διαπραγματεύσεις από άκρο εις άκρο για κάθε ροή (όπως συμβαίνει στις Ολοκληρωμένες Υπηρεσίες), κάνει τις διαφοροποιημένες υπηρεσίες σχετικά εύκολες στην υλοποίηση και τη διαχείριση. Επιπροσθέτως, τα βήματα εκτέλεσης του DiffServ μηχανισμού μπορούν να εκτελεστούν τόσο από ειδικό λογισμικό δικτύου όσο και από λειτουργικά συστήματα, ελαχιστοποιώντας έτσι την απαιτούμενη προσπάθεια και το κόστος πιθανών αλλαγών στην υπάρχουσα δικτυακή υποδομή.

Τέλος, εκτός από την ποιότητα που απολαμβάνουν οι τελικοί χρήστες, μεγάλο πλεονέκτημα δίνεται και στους παρόχους υπηρεσιών, καθώς αυτοί πλέον μπορούν να εφαρμόζουν διάφορες πολιτικές χρέωσης για την κάθε κλάση υπηρεσίας χρεώνοντας, για παράδειγμα, περισσότερο για κάθε πακέτο άμεσης εξυπηρέτησης που μεταφέρει ή επιτρέποντας μέχρι Ν τέτοια πακέτα το μήνα έναντι κάποιου σταθερού πρόσθετου μηνιαίου ποσού.

Βέβαια, όσο σημαντικά και πολλά κι αν είναι τα πλεονεκτήματα του υπό ανάλυση μοντέλου, πρέπει να ληφθούν υπ' όψιν και τα μειονεκτήματά του. Ένα από τα πιο βασικά είναι η διαχείριση. Παρ' όλο που η πληροφορία, όπως ο ρυθμός των πακέτων, είναι διαθέσιμη μέσω της βασισμένης σε τάξεις Βάση Διαχειριστικής Πληροφορίας (Management Information Base, MIB), οι λειτουργίες χρέωσης και ελέγχου είναι ακόμη δύσκολες, ειδικά εάν ληφθεί υπ' όψιν ότι ο ρόλος του Bandwidth Broker βρίσκεται ακόμη υπό μελέτη και τυποποίηση.

Επιπρόσθετα, άλλο ένα ελάττωμα του DiffServ έχει να κάνει με το ότι η σηματοδότηση εκτελείται ξεχωριστά από τον μηχανισμό δρομολόγησης. Έτσι, δεν είναι αμελητέα η πιθανότητα να υπάρχει μία διαδρομή (π.χ. OSPF, EIGPR) στο δίκτυο η οποία να μπορεί να προσφέρει τους απαραίτητους πόρους, αλλά το DiffServ να μην καταφέρει να την εντοπίσει.

Αξίζει να τονιστεί, βέβαια, ότι τουλάχιστον σε σύγκριση με το IntServ, τα μειονεκτήματα του DiffServ είναι ελάχιστα και σχετικά επιλύσιμα. Ωστόσο, το

γεγονός ότι αυτό λαμβάνει μέριμνα για την παρεχόμενη ποιότητα υπηρεσίας τοπικά σε κάθε κόμβο και όχι από άκρο – εις – άκρο, δεν το άφησε τελικά να επικρατήσει και επισκιάστηκε από την Μεταγωγή Ετικετών Πολλαπλών Πρωτοκόλλων που θα εξεταστεί στην ενότητα που ακολουθεί.

ΜΕΤΑΓΩΓΗ ΕΤΙΚΕΤΩΝ ΠΟΛΛΑΠΛΩΝ ΠΡΩΤΟΚΟΛΛΩΝ

Την ώρα που η IETF δούλευε πάνω στα μοντέλα IntServ και DiffServ, πολλοί κατασκευαστές δρομολογητών εξέταζαν καλύτερες μεθόδους προώθησης προκειμένου να αποφύγουν τις δυσχέρειες της παραδοσιακής δρομολόγησης η οποία: βασίζεται στην hop – by – hop προώθηση ενός πακέτου βάσει της διεύθυνσης προορισμού του, αναγκάζει κάθε δρομολογητή κατά μήκος του μονοπατιού να εκτελεί πρωτόκολλα εσωτερικών πυλών (Interior Gateway Protocols, IGP) όπως το RIP και το OSPF ή πρωτόκολλα εξωτερικών πυλών (Exterior Gateway Protocols, EGP) όπως το BGP^[16] και αυξάνει τοιουτοτρόπως την από άκρο – εις – άκρο καθυστέρηση και την πιθανότητα συμφόρησης.

Η δουλειά τους, λοιπόν, εμπνευσμένη κυρίως από την αρχιτεκτονική ATM δικτύων, εστιάστηκε στην προσθήκη μίας ετικέτας πριν από κάθε πακέτο και την δρομολόγηση του τελευταίου βάσει αυτής, αποφεύγοντας έτσι τις χρονοβόρες αναζητήσεις σε πίνακες δρομολόγησης και τις αποφάσεις για την προώθηση του πακέτου σε κάθε κόμβο. Αργότερα, το μοντέλο αυτό υιοθετήθηκε από την IETF, ονομάζοντάς το «Μεταγωγή Ετικετών Πολλαπλών Πρωτοκόλλων» (Multiprotocol Label Switching, MPLS, [RFC 3031 και άλλα]) και ξεκίνησε η προτυποποίησή του συνδυάζοντας τα καλύτερα χαρακτηριστικά των λειτουργιών μεταγωγής που εκτελούνται στον πυρήνα του δικτύου, με τις ωφέλειες των IP τεχνολογιών δρομολόγησης που τρέχουν στα άκρα. Με πιο απλά λόγια, αυτό που άρχισε να προτυποποιείται ήταν,

ουσιαστικά, η ενοποίηση των καλύτερων χαρακτηριστικών των παραδοσιακών τεχνολογιών επιπέδου 2 και 3, παράγοντας ένα υβριδικό σύνολο πρωτοκόλλων και διαδικασιών το οποίο επιτρέπει στις γρήγορες δυνατότητες μεταγωγής του ATM και του Frame Relay να χρησιμοποιούνται από IP δίκτυα.

Το MPLS, λοιπόν, μπορεί να χαρακτηριστεί σαν ένας κλιμακούμενος, ανεξάρτητος από πρωτόκολλα μηχανισμός μεταφοράς δεδομένων [23]. Οι αποφάσεις προώθησης πακέτων βασίζονται αποκλειστικά στο περιεχόμενο της ετικέτας, πράγμα που επιτρέπει την δημιουργία από άκρο – εις – άκρο κυκλωμάτων μέσω οποιουδήποτε μέσου και οποιουδήποτε πρωτοκόλλου. Στο θέμα αυτό, το πρωταρχικό πλεονέκτημα έγκειται στην εξάλειψη της εξάρτησης από συγκεκριμένες τεχνολογίες επιπέδου Ζεύξης Δεδομένων, όπως το ATM, η μεταγωγή πακέτων, το SONET ή το Ethernet και η απαλοιφή της ανάγκης για πολλαπλά δίκτυα επιπέδου 2 για την υποστήριξη διαφορετικών τύπων κίνησης.

Χρησιμοποιώντας ATM μεταγωγείς, οι οποίοι ήταν πολύ γρηγορότεροι από τους δρομολογητές του IP πρωτοκόλλου, η MPLS αρχιτεκτονική επιλύει προβλήματα σχετικά με την κίνηση, τη διαχείριση εύρους ζώνης και το IP management, τα οποία χαρακτηρίζουν άλλες τεχνολογίες [10]. Για παράδειγμα, το MPLS εξαλείφει τις πολύπλοκες διεργασίες διαχείρισης Εικονικών Ιδιωτικών Δικτύων (Virtual Private Networks, VPN) οι οποίες μαστίζουν τη μεταγωγή πλαισίων, κάνοντας πιο εύκολη την διαχείριση του εύρους ζώνης της ζεύξης πρόσβασης και επιτρέποντας πιο ευέλικτη παροχή bandwidth. Πρέπει να τονιστεί ότι η απλοποιημένη διαχείριση μεταφράζεται σε λιγότερα λάθη, λιγότερη προσπάθεια και υψηλότερη αξιοπιστία. Παράλληλα, προσφέρει εγγυήσεις σταθερής χωρητικότητας για συγκεκριμένες ροές, εξασφαλίζει χωρητικότητα για φωνή και ελέγχει την καθυστέρηση και τις διακυμάνσεις της. Δίνοντας, δε, δυνατότητες παροχής συγκεκριμένων, εγγυημένων και ποσοτικοποιημένων SLAs, διευθετεί ποικίλους βαθμούς ποιότητας υπηρεσίας για πολλαπλούς πελάτες.

Τα πολλά και σημαντικά πλεονεκτήματα του MPLS τόσο ως προς την ταχύτερη δρομολόγηση των πακέτων όσο ως και προς τις δυνατότητες

παροχής εγγυημένων επιπέδων ποιότητας υπηρεσιών, συγκέντρωσαν τελικά την προσοχή της επιστημονικής κοινότητας και επισκίασαν σε κάποιο βαθμό τις προηγούμενες τεχνολογίες των IntServ και DiffServ. Στις παρακάτω ενότητες, λοιπόν, θα παρουσιαστούν τα βασικά χαρακτηριστικά του υπό ανάλυση μοντέλου, εστιάζοντας κυρίως σε θέματα αρχιτεκτονικής και λειτουργίας τα οποία θα βοηθήσουν την κατανόηση των θεμάτων Ποιότητας Υπηρεσιών που θα εξεταστούν στο Τρίτο Κεφάλαιο.

ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ MPLS

Για την ευκολότερη κατανόηση πολλών εννοιών και λειτουργιών που θα αναφερθούν σε επόμενες ενότητες, σκόπιμη κρίνεται η περιγραφή των βασικών συστατικών της MPLS Αρχιτεκτονικής, σε ένα σχετικά υψηλό επίπεδο αφαίρεσης. Ζητήματα που δε θα απασχολήσουν τόσο την παρούσα εργασία, όπως ο ακριβής τρόπος προώθησης των πακέτων ή το Πρωτόκολλο Κατανομής Ετικετών, θα περιγραφούν συνοπτικά στις παραγράφους που ακολουθούν. Άλλα θέματα, δε, όπως η παροχή ποιότητας υπηρεσιών, η κατασκευή μονοπατιών διαμέσω ενός MPLS τομέα κ.τ.λ., δε θα αναφερθούν εδώ, αφήνοντας την ανάλυσή τους για το επόμενο κεφάλαιο που αφιερώνεται στην Ποιότητα Υπηρεσιών του MPLS.

ΚΛΑΣΗ ΙΣΟΔΥΝΑΜΗΣ ΠΡΟΩΘΗΣΗΣ

Μία Κλάση Ισοδύναμης Προώθησης (Forwarding Equivalence Class, FEC) είναι ένα σύνολο πακέτων που προωθούνται με τον ίδιο τρόπο εντός του δικτύου. Μία FEC μπορεί να περιλαμβάνει, για παράδειγμα, όλα τα πακέτα των οποίων η διεύθυνση προορισμού ταιριάζει με ένα συγκεκριμένο IP πρόθεμα (prefix) ή τα πακέτα τα οποία ανήκουν σε κάποια συγκεκριμένη εφαρμογή. Οι FECs συνήθως δημιουργούνται με τη χρήση της πληροφορίας που παρέχουν οι IGP τεχνολογίες (όπως το OSPF και το RIP). Πιο συγκεκριμένα, όταν ένα πακέτο εισέρχεται σε ένα MPLS δίκτυο, ο MPLS edge router το εξετάζει για να εξάγει πληροφορία όπως οι διευθύνσεις πηγής/

προορισμού, το interface από το οποίο έφτασε, οι απαιτήσεις του σε ποιότητα υπηρεσίας και άλλα. Βασισμένος στα παραπάνω, το ταξινομεί σε μια συγκεκριμένη Κλάση Ισοδύναμης Προώθησης και το προωθεί στο δίκτυο. Το παραπάνω σημαίνει ότι όλα τα πακέτα που ανήκουν στην ίδια κλάση θα ακολουθούν το ίδιο δρομολόγιο και θα υποβάλλονται στην ίδια μεταχείριση.

ΕΤΙΚΕΤΑ

Η ετικέτα είναι ένα μικρό, σταθερού μήκους, συνεχόμενο αναγνωριστικό το οποίο χρησιμοποιείται για την αναγνώριση μίας FEC και περιλαμβάνει όλη την πληροφορία που χρειάζεται για την προώθηση του πακέτου. Οι ετικέτες δημιουργούνται και ανατίθενται στα IP πακέτα, βασισμένες στην πληροφορία που παρατέθηκε προηγουμένως και έχουν συνήθως τοπική σημασία / εμβέλεια. Το μέγεθός τους είναι 32 bits και αποτελούνται από τα παρακάτω πεδία:

- Τιμή Ετικέτας: Καταλαμβάνει τα πρώτα 20 bits και προσδιορίζει την FEC, όπως περιγράφηκε πιο πάνω
- Experimental Range: Καταλαμβάνει τα επόμενα 3 bits και είναι δεσμευμένη για πειραματική χρήση, όπως για την πληροφορία των Διαφοροποιημένων Υπηρεσιών (π.χ. Κλάση Υπηρεσίας) ή για τις PHBs οδηγίες.
- S: Πεδίο μήκους 1 bit, χρησιμοποιείται για την υπόδειξη του εάν η στοίβα ετικετών (label stack) είναι παρούσα ή όχι. Εάν η ετικέτα είναι η μόνη υπάρχουσα, ή είναι η τελευταία στη στοίβα, το πεδίο λαμβάνει τιμή «1», αλλιώς «0».
- Time To Live: 8-bit πεδίο, χρησιμοποιείται για την υπόδειξη του αριθμού των MPLS κόμβων τους οποίους έχει διασχίσει το πακέτο για να φτάσει στον προορισμό του. Το πεδίο TTL λαμβάνει μία αρχική τιμή στην αρχή της δρομολόγησής του και μπορεί να προσδιοριστεί τόσο από έναν δρομολογητή για λόγους ελέγχου προώθησης, όσο και από την ίδια την εφαρμογή για λόγους περιορισμού των hops ή του χρόνου παραμονής του πακέτου στο δίκτυο (ως άλλος ένας περιορισμός

ποιότητας). Κάθε επόμενος δρομολογητής που θα παραλάβει το πακέτο, θα μειώσει την τιμή TTL κατά ένα. Έτσι, εάν η τιμή φτάσει στο 0 αλλά το πακέτο δεν έχει φτάσει στον προορισμό του, θα απορριφθεί. Ένας από τους βασικότερους λόγους της παραπάνω προσέγγισης, έγκειται στην αποφυγή loops του πακέτου ή της ροής κλάσης καθώς και στον έλεγχο του συνολικού χρόνου μετάδοσής του διαμέσω του MPLS τομέα, μειώνοντας έτσι την πιθανότητα συμφόρησης ή εκτός χρόνου παράδοσης του πακέτου.

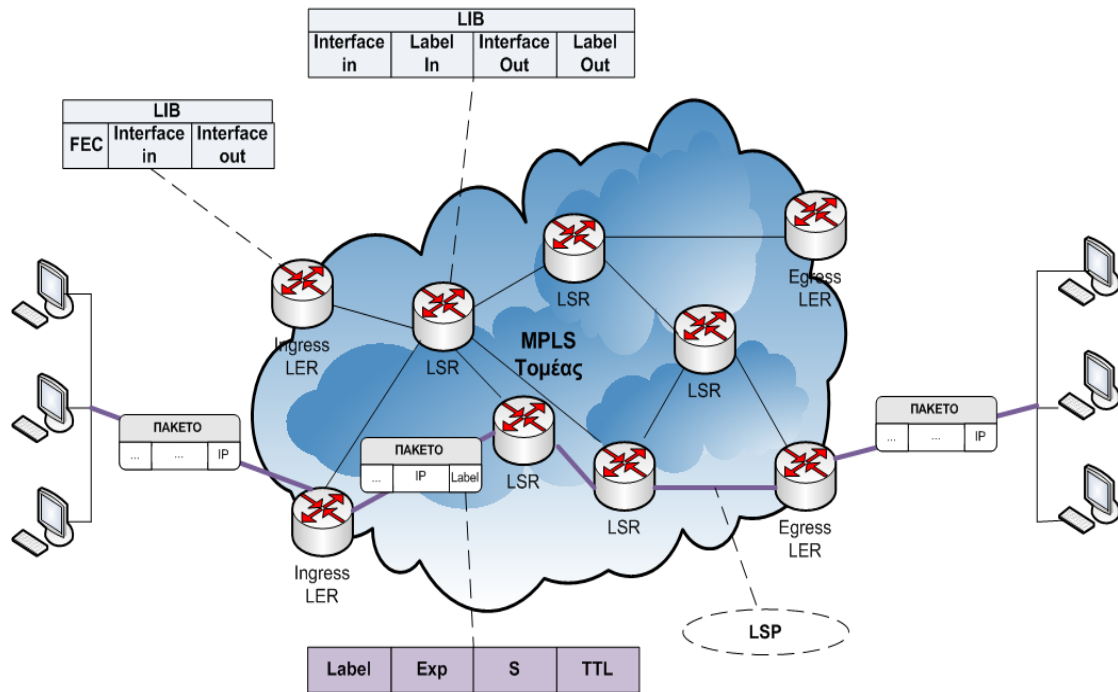
Η θέση τις ετικέτας (ή στοίβας, όπως θα περιγραφεί παρακάτω) βρίσκεται μετά την επικεφαλίδα του επιπέδου διασύνδεσης δεδομένων (Data Link Layer, DLL) και πριν την επικεφαλίδα του επιπέδου δικτύου (Network Layer, NL). Εάν υπάρχει στοίβα ετικετών, τότε η κορυφή της (δηλαδή η πρώτη ετικέτα) είναι η πιο κοντινή στην επικεφαλίδα του DLL. Παραδείγματα της θέσης που μπορεί να καταλάβει η ετικέτα αποτελούν το πεδίο VPI/VCI του ATM Header, το πεδίο DLCI στο FrameRelay και άλλα.

ΣΤΟΙΒΑ ΕΤΙΚΕΤΩΝ

Κάθε πακέτο δεν έχει κατ' ανάγκη μόνο μία ετικέτα, αλλά μπορεί να περιλαμβάνει μία στοίβα ετικετών (Label Stack), οργανωμένες σαν μία ουρά LIFO, δίνοντας έτσι τη δυνατότητα ιεραρχικής δρομολόγησης στο MPLS. Η επεξεργασία των ετικετών λαμβάνει χώρα πάντα στο ανώτερο επίπεδο της στοίβας και επιτρέπει τη συνάθροιση LSPs σε ένα μοναδικό LSP για ένα τμήμα της διαδρομής (φτιάχνοντας, έτσι, ένα ή περισσότερα tunnels, ανάλογα με τον αριθμό ετικετών της στοίβας).

Η διαχείριση της στοίβας ετικετών είναι απλή και μπορεί να περιγραφεί ως ακολούθως: Έστω ότι ένα πακέτο δρομολογείται στο MPLS και παρουσιάζεται η ανάγκη για την προώθησή του μέσα από ένα tunnel. Πριν εισέλθει στο πρώτο tunnel, το πακέτο είναι μαρκαρισμένο με την ετικέτα A η οποία προσδιορίζει την εγγενή FEC του. Στην αρχή του tunnel, ο LSR θα του προσθέσει (push) μία διαφορετική ετικέτα, έστω B, με αποτέλεσμα όλα τα πακέτα που εισέρχονται στο tunnel να τυγχάνουν ίδιας μεταχείρισης. Στο

τέλος του tunnel, ένας άλλος LSR θα βγάλει (pop) την ετικέτα της κορυφής και θα συνεχίσει την προώθηση του πακέτου χρησιμοποιώντας την επόμενη, στη στοίβα, ετικέτα, δηλαδή την A.



Εικόνα 3. Αρχιτεκτονική MPLS

MPLS ΔΡΟΜΟΛΟΓΗΤΕΣ

Οι δρομολογητές στο MPLS λειτουργούν σαν μεταγωγείς και χωρίζονται σε δύο τύπους ανάλογα με την τοποθεσία και τα καθήκοντά τους. Συγκεκριμένα, οι δρομολογητές – μεταγωγείς, γνωστοί ως Label Switching Routers (LSRs) βρίσκονται στο εσωτερικό ενός MPLS τομέα και εκτελούν τις βασισμένες στην ετικέτα λειτουργίες προώθησης πακέτων. Οι Δρομολογητές Ετικέτας Άκρου (Label Edge Routers, LERs) είναι μεταγωγείς που τοποθετούνται στα άκρα του MPLS domain και είναι ικανοί να χρησιμοποιούν την πληροφορία δρομολόγησης προκειμένου να αναθέσουν ετικέτες στα πακέτα και κατόπιν να τα προωθήσουν στο MPLS δίκτυο. Οι LERs, με τη σειρά τους, κατανέμονται σε Ingress και Egress LERs, με τους πρώτους να βρίσκονται στην είσοδο του

τομέα τοποθετώντας (push) μία ετικέτα στα εισερχόμενα πακέτα και τους δεύτερους να τοποθετούνται στην έξοδο απομακρύνοντάς την (pop).

ΠΡΩΤΟΚΟΛΛΟ ΚΑΤΑΝΟΜΗΣ ΕΤΙΚΕΤΩΝ

Παρ' όλο που αυτό καθ' αυτό το Πρωτόκολλο Κατανομής Ετικετών δε θα απασχολήσει την παρούσα εργασία, θα περιγραφεί σε υψηλό επίπεδο αφαίρεσης για λόγους συνοχής. Προκειμένου, λοιπόν, ένας LSR να γνωρίζει σε ποιό σημείο να αποστείλει το εκάστοτε πακέτο, χωρίς να χρειάζεται να αποφασίζει για/ υπολογίζει την διαδρομή εκ των προτέρων, πρέπει να υπάρχει ένας μηχανισμός ο οποίος αφ' ενός να του επιτρέπει να μεταγάγει την ετικέτα αλλάζοντας την παλιά τιμή της και τοποθετώντας της μία καινούρια η οποία να έχει την ίδια τιμή για κάθε πακέτο της ίδιας κλάσης, και αφ' ετέρου να του υποδεικνύει τον προορισμό στον οποίο αυτό πρέπει να προωθηθεί. Για το λόγο αυτό κάθε LSR διατηρεί μία Βάση Πληροφοριών Ετικέτας (Label Information Base, LIB) η οποία περιλαμβάνει τις αντιστοιχίσεις «εισερχόμενο σημείο προσαρμογής – τιμή ετικέτας» προς «εξερχόμενο σημείο προσαρμογής – τιμή ετικέτας».

Ο μηχανισμός ο οποίος, ουσιαστικά, «γεμίζει» τη LIB ονομάζεται Πρωτόκολλο Κατανομής Ετικετών (Label Distribution Protocol, LDP) και πρόκειται για ένα σύνολο διαδικασιών οι οποίες πρώτον, επιτρέπουν στους γειτονικούς LSRs να ανταλλάσουν πληροφορίες σχετικές με τις επιλεγμένες ετικέτες τους και δεύτερον, χρησιμοποιούνται από τους τελευταίους για την απ' ευθείας αντιστοίχιση της πληροφορίας δρομολόγησης επιπέδου δικτύου με τα μονοπάτια μεταγωγής του επιπέδου ζεύξης δεδομένων. Αξίζει, σε αυτό το σημείο, να τονιστεί ότι το πρωτόκολλο LDP επιτρέπει την ανταλλαγή ετικετών με κατεύθυνση αντίθετη από αυτή της ροής δεδομένων (ο κάθε upstream κόμβος ενημερώνει τον γειτονικό downstream του) και μπορεί να εκτελεστεί με δύο τρόπους: αυτόκλητα (ένας LSR εκπέμπει τις συνδεδεμένες με κάθε FEC ετικέτες του) ή κατ' απαίτηση (ένας LSR αιτεί από έναν άλλον ομότιμο LSR να μάθει την τιμή που αυτός έχει αντιστοιχήσει σε μία FEC).

Προκειμένου να εκτελεστούν με επιτυχία οι παραπάνω λειτουργίες, το LDP προσφέρει τέσσερα είδη μηνυμάτων τα οποία συνοψίζονται ως ακολούθως:

- Μηνύματα Ανίχνευσης (Discovery Messages)

Χρησιμοποιούνται για την αναγγελία και τη διατήρηση της παρουσίας ενός LSR.

- Μηνύματα Συνεδρίας/ Γειτνίασης (Session/ Adjacency Messages)

Χρησιμοποιούνται για την εγκατάσταση, τη διατήρηση και τον τερματισμό συνεδριών ανάμεσα σε ομότιμους LSRs.

- Μηνύματα Διαφήμισης (Advertisement Messages)

Χρησιμοποιούνται για τη δημιουργία, την αλλαγή και τη διαγραφή αντιστοιχίσεων ετικετών .

- Ειδοποιητικά Μηνύματα (Notification Messages)

Χρησιμοποιούνται για την παροχή συμβουλευτικής πληροφορίας και τη σηματοδότηση σφαλμάτων.

ΜΟΝΟΠΑΤΙ ΜΕΤΑΓΩΓΙΜΟ ΜΕ ΕΤΙΚΕΤΕΣ

Ως Label Switched Path (LSP) ορίζεται το μονοπάτι που ακολουθείται από ένα πακέτο, εντός του MPLS τομέα. Οι αποφάσεις που λαμβάνονται κατά τον προσδιορισμό του, καθώς και ο ακριβής τρόπος κατασκευής του θα αναλυθεί στα κεφάλαια που ακολουθούν.

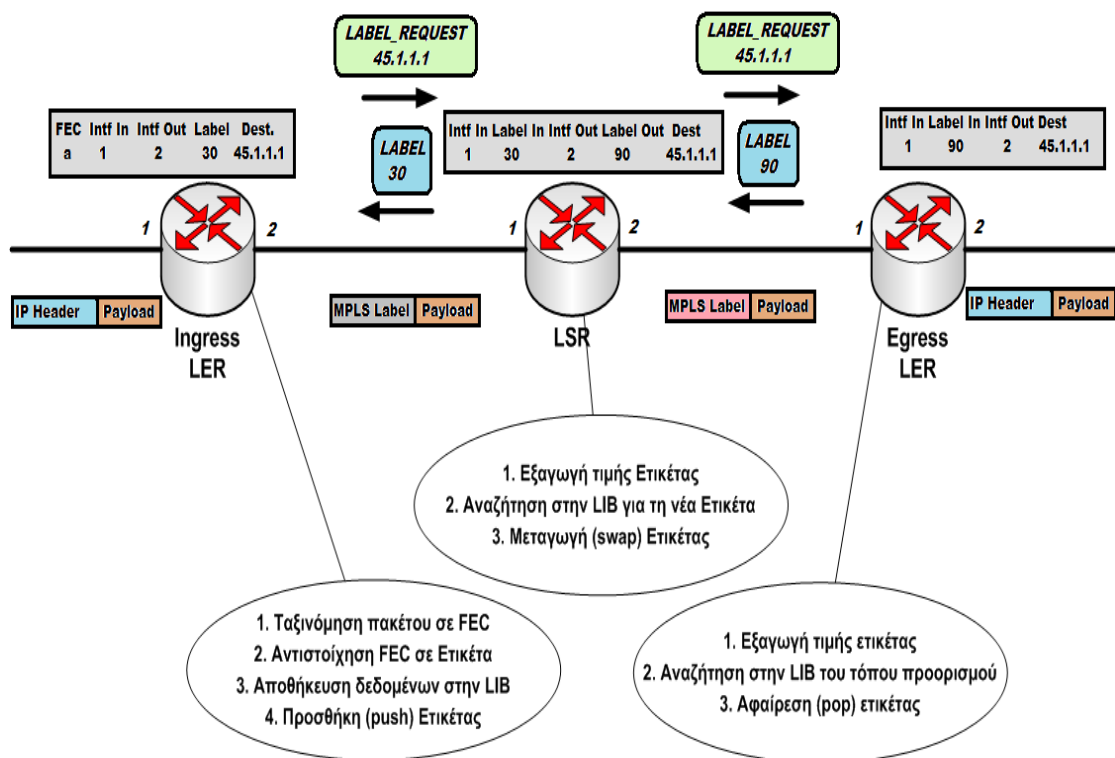
ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ MPLS

Στην ενότητα αυτή θα δοθεί μία σύντομη περιγραφή της λειτουργίας του MPLS στο γενικό πλαίσιο εφαρμογής του. Θα αποφευχθεί η ανάλυση των διαδικασιών και των αποφάσεων που λαμβάνονται προκειμένου να υποστηριχθεί η διαχείριση πόρων και κίνησης, καθώς και τα χαρακτηριστικά των MPLS οντοτήτων που συμμετέχουν σε αυτές, εμμένοντας σε πιο καθολικές διεργασίες και αφήνοντας τα τελευταία για μεγαλύτερη ανάλυση στο επόμενο κεφάλαιο.

Η διαδικασία λειτουργίας του MPLS, λοιπόν, ξεκινάει όταν ένας Ingress LER παραλαμβάνει ένα πακέτο. Όπως και στην IP αρχιτεκτονική, το πρώτο που κάνει είναι να το εξετάσει ως προς τα χαρακτηριστικά του. Έτσι, εξάγει πληροφορίες όπως η διεύθυνση του προορισμού, η μάσκα δικτύου, η διεύθυνση της διεπαφής από την οποία έφτασε, το πεδίο DS και άλλα. Βασισμένο σε αυτά, υπολογίζει τη βέλτιστη διαδρομή του χρησιμοποιώντας υπάρχοντα πρωτόκολλα (π.χ. OSFP). Επόμενο βήμα είναι η ενημέρωση του πίνακα δρομολόγησής του. Ενώ στο IP πρωτόκολλο, όλοι οι δρομολογητές διατηρούν έναν πίνακα με πεδία όπως η διεύθυνση προορισμού/ μάσκας/ διεπαφής/ πύλης, η μετρική, το πρωτόκολλο του πακέτου κ.ο.κ. και αναζητούν το επόμενο βήμα βασισμένοι σε αυτά, στο MPLS η διαδικασία απλοποιείται σημαντικά, με τον Ingress LER να πρέπει απλά να αντιστοιχίσει μία τιμή ετικέτας στη συγκεκριμένη FEC.

Έτσι, λοιπόν, αφού εξαχθεί η FEC για το πακέτο, ο Ingress LER, όπως και οι υπόλοιποι δρομολογητές κατά μήκος του μονοπατιού, θα πρέπει να της αναθέσουν μία τιμή ετικέτας προκειμένου να την αναγνωρίζουν. Παρ' όλο που για το σκοπό αυτόν έχουν προταθεί αρκετά πρωτόκολλα κατανομής ετικετών με διαφορετικές προσεγγίσεις και υλοποιήσεις, εδώ θα παρουσιαστεί μόνο το βασικό σκεπτικό αποφεύγοντας λεπτομέρειες που σχετίζονται με την επικοινωνία, τις οντότητες και τα μηνύματα που μπορεί να συμμετέχουν στην κάθε υλοποίηση. Συνεχίζοντας, λοιπόν, ο Ingress LER θα αποστείλει στον Egress LER ένα αίτημα ετικέτας (Label Request), το οποίο μπορεί να περιλαμβάνει, για παράδειγμα, την διεύθυνση προορισμού της FEC. Ο Egress

LER που θα λάβει το αίτημα, θα αναθέσει μία τιμή ετικέτας για τη συγκεκριμένη FEC, ενημερώνοντας τα πεδία «Διεύθυνση Προορισμού» και «Ετικέτα Εξόδου» της LIB του και θα αποστείλει την τιμή της ετικέτας προς την αντίθετη κατεύθυνση (upstream). Κάθε δρομολογητής κατά μήκος του μονοπατιού, λοιπόν, που θα λαμβάνει από τον προηγούμενό του πληροφορίες σχετικές με τη διεύθυνση της διεπαφής από την οποία θα φτάσει το πακέτο, καθώς και την τιμή της ετικέτας που αυτό θα έχει, θα παράγει μία άλλη τιμή ετικέτας για τη συγκεκριμένη FEC και θα συμπληρώνει αντίστοιχα την LIB του. Έτσι, στο τέλος της διαδικασίας όλοι οι δρομολογητές θα είναι ενημερωμένοι ως προς το «ποια νέα ετικέτα E2 να αναθέσουν στο πακέτο P που εισέρχεται από τη διεπαφή A έχοντας ετικέτα E1» και το «πού να την προωθήσουν».



Εικόνα 4. Λειτουργία του MPLS

Αφού κατανεμηθούν οι ετικέτες, ο Ingress LER θα εισάγει την ετικέτα στο πακέτο, θα ενημερώσει τα υπόλοιπα πεδία (όπως το TTL) και θα το δρομολογήσει. Κάθε δρομολογητής κατά μήκος της διαδρομής, θα εξετάσει την ετικέτα για να εξάγει από αυτήν την τιμή TTL και θα τη μειώσει (π.χ. κατά

ένα, εάν το TTL απεικονίζει το μέγιστο πλήθος κόμβων από τους οποίους θα πρέπει να περάσει το πακέτο). Εάν το TTL είναι μηδέν και ο εκάστοτε LSR δεν είναι ο κόμβος προορισμού, θα απορρίψει το πακέτο. Στην περίπτωση, δε, που δεν είναι μηδενικό, ο δρομολογητής με βάση την τιμή της ετικέτας και τη διεπαφή από την οποία έφτασε το πακέτο, θα αναζητήσει στην LIB τη νέα ετικέτα (Ετικέτα Εξόδου) την οποία έχει αντιστοιχήσει στη συγκεκριμένη τιμή ετικέτας και θα αντικαταστήσει (switch) την παλιά τιμή με τη νέα. Κατόπιν, θα το προωθήσει στη διεπαφή εξόδου που αναφέρεται στον πίνακα. Αξίζει να αναφερθεί ότι ανά FEC μπορούν να προσδιοριστούν και συγκεκριμένες PHBs, με αποτέλεσμα οι ενδιάμεσοι δρομολογητές να επεξεργάζονται ή να προωθούν με συγκεκριμένο τρόπο (π.χ. σειρά) το κάθε πακέτο (ανά FEC).

Όταν το πακέτο φτάσει τελικά στον Egress LER, αυτός θα εξετάσει την τιμή της ετικέτας του και θα εξάγει από τον πίνακα LIB τη διεπαφή στην οποία θα πρέπει να το προωθήσει. Γνωρίζοντας ότι είναι κόμβος απόληξης, θα αφαιρέσει την ετικέτα (pop), γυρνώντας το πακέτο στην κανονική του μορφή (π.χ. με την IP επικεφαλίδα να φαίνεται κανονικά) και θα το προωθήσει στο γειτονικό (υπο)δίκτυο. Από το σημείο αυτό και μετά, το πακέτο θα συνεχίσει κανονικά τη διαδρομή του μέχρι να φτάσει στον τελικό του προορισμό.

ΣΥΝΟΨΗ

Όπως ήδη παρουσιάστηκε, τρία βασικά μοντέλα κυριάρχησαν στον τομέα της Ποιότητας Υπηρεσιών: οι Ολοκληρωμένες Υπηρεσίες, οι Διαφοροποιημένες Υπηρεσίες και η Μεταγωγή Ετικετών Πολλαπλών Πρωτοκόλλων. Το πρώτο, εστιαζόμενο στην παροχή ποιότητας ανά ροή, προσφέρει δυνατότητες λεπτομερούς προδιαγραφής της ποιότητας υπηρεσίας που επιθυμεί ένας Client και από άκρο σε άκρο αστυνόμευση και τήρηση των προσυμφωνημένων παραμέτρων ποιότητας και κίνησης. Ωστόσο, διατηρώντας μεγάλο όγκο δεδομένων ανά ροή καθίσταται αποτρεπτικό για δίκτυα μεγάλου μεγέθους καθώς δεν προσφέρει δυνατότητες κλιμάκωσης ενώ ταυτόχρονα δύναται να υπερφορτώσει τα τελευταία με πολλαπλά signaling μηνύματα.

Ως απάντηση στα προβλήματα του πρώτου, το δεύτερο μοντέλο αποφεύγει την παροχή ποιότητας ανά ροή δεδομένων και στρέφεται στην συνάθροιση των τελευταίων εντός καλά ορισμένων τάξεων υπηρεσίας. Έτσι, διατηρώντας δεδομένα ανά τάξη, πλέον, και όχι ανά ροή, μειώνει δραστικά τον απαιτούμενο όγκο δεδομένων καθώς και το overhead που προσθέτουν οι Ολοκληρωμένες Υπηρεσίες. Παράλληλα, μην απαιτώντας εκ των προτέρων διευθέτηση και χρονοβόρες διαπραγματεύσεις από άκρο εις άκρο ανά ροή, καθίσταται σχετικά εύκολο στην υλοποίηση και τη διαχείριση ενώ δίνει και τη δυνατότητα εφαρμογής, με πολύ πιο εύκολο τρόπο, διαφορετικών πολιτικών χρέωσης ανά κλάση.

Εντούτοις, το γεγονός ότι και τα δύο μοντέλα εφαρμόζουν χρονοβόρες πολιτικές και αναζητήσεις σε κάθε κόμβο, ενώ παράλληλα βασίζονται σε πολύ μεγάλο βαθμό στις ήδη υπάρχουσες τεχνολογίες (π.χ. OSFP) για τη δρομολόγηση των πακέτων, πράγμα που σημαίνει ότι η διαδρομή που θα επιλεγεί μπορεί αφ' ενός να απαιτήσει σημαντικούς χρονικούς πόρους για την αναζήτησή της ενώ αφ' ετέρου να μην είναι η «βέλτιστη» ποιοτικά, έδωσε την ώθηση για την τελική επισκίαση των προηγούμενων μοντέλων από το MPLS. Το μοντέλο Μεταγωγής Ετικέτας Πολλαπλών Πρωτοκόλλων, όπως δηλώνει και το όνομά του, βασίζεται στη δημιουργία κυκλωμάτων αναθέτοντας ετικέτες ανά ροή, ενώ ταυτόχρονα είναι ανεξάρτητο από τα διαφορετικά πρωτόκολλα επιπέδων Ζεύξης Δεδομένων και Δικτύου.

Συνδυάζοντας τα καλύτερα χαρακτηριστικά των λειτουργιών μεταγωγής που εκτελούνται στον πυρήνα ενός δικτύου με τις ωφέλειες των IP τεχνολογιών δρομολόγησης που τρέχουν στα άκρα, μπορεί να χαρακτηριστεί σαν ένα ταχύτατο και εξαιρετικά κλιμακούμενο υβριδικό μοντέλο. Τα πολλαπλά πλεονεκτήματά του δίκαια συγκέντρωσαν την προσοχή της επιστημονικής κοινότητας η οποία πλέον, εκτός από την εκμετάλλευσή του για την ταχεία δρομολόγηση πακέτων, έχει ξεκινήσει την προτυποποίηση και την εύρεση μηχανισμών και τεχνολογιών προς παροχή Ποιότητας Υπηρεσιών και Διαχείρισης Κίνησης.

Στο κεφάλαιο που ακολουθεί, λοιπόν, θα πραγματοποιηθεί εμβάθυνση στα τελευταία αυτά ζητήματα, δίνοντας αρχικά μία καθολική περιγραφή του

προβλήματος Διαχείρισης Κίνησης και Πόρων, η οποία κατόπιν θα συγκεκριμενοποιηθεί με τα απαιτούμενα χαρακτηριστικά που εισάγει η MPLS τεχνολογία στο θέμα αυτό. Θα παρουσιαστεί, με άλλα λόγια, τί είναι η Διαχείριση Κίνησης και Πόρων, καθώς και το πώς πρέπει ή μπορεί να υλοποιηθεί μέσω του MPLS. Εν συνεχεία, θα λάβει χώρα μια αναφορά σε κάποιες τεχνολογίες και αλγορίθμους που έχουν προταθεί για το συγκεκριμένο πρόβλημα, εστιάζοντας τόσο στον τρόπο λειτουργίας τους όσο και στο βαθμό εκμετάλλευσής τους των δυνατοτήτων και απαιτήσεων του μοντέλου.

ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΗΣΗΣ ΚΑΙ ΠΟΡΩΝ ΣΤΟ MPLS

ΕΙΣΑΓΩΓΗ

Το κεφάλαιο αυτό στρέφεται γύρω από το πρόβλημα της Διαχείρισης Κίνησης και Πόρων εστιάζοντας στον ορισμό αυτού καθώς και των χαρακτηριστικών που πρέπει να έχει το μοντέλο Μεταγωγής Ετικετών Πολλαπλών Πρωτοκόλλων προκειμένου να το υποστηρίξει ορθά. Αρχικά, λοιπόν, θα δοθεί ο ορισμός του υπό ανάλυση προβλήματος ο οποίος θα διαιρεθεί περαιτέρω ως προς τα συστατικά του στοιχεία, όπως η διανομή των πληροφοριών ζεύξεων, ο υπολογισμός των διαδρομών και η σηματοδότηση των τελευταίων.

Συνεχίζοντας, θα εξεταστούν οι αρχές που εισάγει το MPLS μοντέλο για την υποστήριξη των παραπάνω. Με άλλα λόγια, θα παρουσιαστούν οι απαιτήσεις του MPLS καθώς και οι δυνατότητες που αυτό προσφέρει για την παροχή Ποιότητας Υπηρεσιών και Διαχείρισης της κυκλοφορίας, από άκρο – εις – άκρο. Παράλληλα, θα δοθεί μία πιο λεπτομερής απεικόνιση των λειτουργιών τόσο του MPLS συνολικά όσο και των επιμέρους οντοτήτων που το αποτελούν, με κατεύθυνση πάντα το αναφερθέν πρόβλημα.

Έχοντας αναλύσει το πρόβλημα Διαχείρισης Κίνησης και Πόρων, καθώς και τις απαιτούμενες και προτεινόμενες αρχές για την επίλυσή του στο MPLS, θα παρουσιαστούν τεχνικές και αλγόριθμοι που έχουν προταθεί για το εν λόγω ζήτημα. Έτσι, λοιπόν, στην τελευταία ενότητα του παρόντος κεφαλαίου θα εξεταστούν διάφορες προσεγγίσεις για τη Διαχείριση της Κίνησης και των Πόρων στο MPLS, από την σκοπιά των στόχων που αυτές θέτουν, του τρόπου της λειτουργίας τους καθώς και του βαθμού συμμόρφωσής τους με τις απαιτήσεις και τα χαρακτηριστικά του μοντέλου.

Στο σημείο αυτό, σκόπιμη κρίνεται η αναφορά κάποιων παραδοχών που ελήφθησαν στις ενότητες που ακολουθούν. Με τον όρο «πακέτο», αναφέρονται τα πακέτα πληροφορίας στα δίκτυα Μεταγωγής Πακέτου, όπως τα IP. Εντούτοις, από το σημείο αυτό και μετά, με τον όρο «πακέτο» θα γίνεται αναφορά στο οποιοδήποτε τμήμα δεδομένων ανταλλάσσεται ή προωθείται μεταξύ δύο δικτυακών οντοτήτων, όπως το δεδομενογράφημα, το πλαίσιο, το κύκλωμα και ούτω καθ' εξής.

Επιπρόσθετα, ο όρος «ροή» θα αναφέρεται στην συνάθροιση πακέτων της ίδιας κλάσης υπηρεσίας η οποία δρομολογείται από το ίδιο LSP. Κοινώς, μία ροή θα χρησιμοποιείται ως ένα traffic trunk πολλαπλών ροών πακέτων, που ανήκουν στην ίδια FEC σε μία δεδομένη χρονική στιγμή.

ΔΙΑΧΕΙΡΙΣΗ ΚΙΝΗΣΗΣ ΚΑΙ ΠΟΡΩΝ

Η ραγδαία διάδοση του Διαδικτύου, η οποία χαρακτηρίζεται από την αύξηση του πλήθους χρηστών, τον πολλαπλασιασμό των εφαρμογών και την ετερογένεια των τεχνολογιών, συντελεί σε μεγάλες διακυμάνσεις της δικτυακής κίνησης τόσο γεωγραφικά, μέσω της μετατόπισης της κυκλοφορίας σε διαφορετική γεωγραφική περιοχή, όσο και χρονικά, μέσω εκρηκτικών και ήρεμων περιόδων. Σε ένα τόσο ετερογενές και ακανόνιστο περιβάλλον, η αποδοτική κατανομή και εκμετάλλευση των πόρων είναι σχεδόν αδύνατη [21]. Απαραίτητος, λοιπόν, είναι ο συνεχής έλεγχος και η ρύθμιση της κίνησης με το βέλτιστο τρόπο. Η τεχνολογία που ασχολείται με το ζήτημα αυτό αναφέρεται στη βιβλιογραφία ως Διαχείριση Κίνησης (Traffic Engineering, TE) και έχει σαν γενικότερο στόχο την βελτιστοποίηση απόδοσης των δικτύων.

Οι Τεχνολογίες Διαχείρισης Κίνησης, χρησιμοποιούν τεχνολογικές και επιστημονικές αρχές για τη μέτρηση, τη μοντελοποίηση, το χαρακτηρισμό και τον έλεγχο της τηλεπικοινωνιακής κυκλοφορίας προκειμένου να πετύχουν το βασικό στόχο: την αποδοτική και αξιόπιστη λειτουργία δικτύου με την ταυτόχρονη βελτιστοποίηση κατανομής πόρων και κυκλοφοριακής επίδοσης. Ο αντικειμενικός στόχος της TE, συνεπώς, μπορεί να χωριστεί σε δύο τομείς: τον βασισμένο στην κυκλοφορία και τον βασισμένο στους πόρους.

Το βασισμένο στην κίνηση πρόβλημα βελτιστοποίησης έχει να κάνει με την ελαχιστοποίηση απωλειών πακέτων, την ελαχιστοποίηση της καθυστέρησης, την μεγιστοποίηση της διέλευσης στο δίκτυο και την ορθή εφαρμογή των SLAs. Πολύ σημαντικό ζήτημα, το οποίο λαμβάνεται υπ' όψιν από την Διαχείριση Κίνησης, είναι βέβαια και η συμφόρηση των δικτύων. Στο

[RFC2702] αναγνωρίζονται δύο χρονικοί τύποι συμφόρησης: η βραχύβια, που μπορεί να προκληθεί σε μικρές περιόδους εκρηκτικότητας και η παρατεταμένη η οποία μπορεί να παρουσιαστεί υπό καταστάσεις ανεπαρκών, για το εκάστοτε φορτίο, πόρων ή/και όταν οι ροές κυκλοφορίας κατανέμονται αναποτελεσματικά σε σχέση με τους διαθέσιμους πόρους. Το πρόβλημα της συμφόρησης λόγω ανεπάρκειας πόρων μπορεί να λυθεί με τεχνολογίες όπως το κυλιόμενο παράθυρο, ο περιορισμός του ρυθμού αποστολής και άλλα. Η άνιση κατανομή του φορτίου πάνω στους διαθέσιμους πόρους, δε, απασχολεί την TE και μπορεί να επιλυθεί μέσω της Εξισορρόπησης Φορτίου, στόχος της οποίας είναι η ελαχιστοποίηση της συμφόρησης ή της μέγιστης χρησιμοποίησης των πόρων. Αξίζει να τονιστεί ότι η μείωση της συμφόρησης μέσω της αποδοτικής κατανομής πόρων συνεπάγεται τη μείωση της απώλειας πακέτων ή/ και της καθυστέρησης, με αποτέλεσμα όχι μόνο να αυξάνεται η διέλευση στο δίκτυο αλλά να προσφέρεται και εγγυήσιμη ποιότητα στις υπηρεσίες.

Η βασισμένη στους πόρους διαχείριση κίνησης, θέτει σαν πρωταρχικό στόχο την βελτιστοποίηση της χρήσης πόρων. Δεδομένου ότι ο βασικότερος πόρος (άρα και μεγαλύτερη πηγή κόστους για τους ISPs) είναι το φάσμα [\[19\]](#), προφανής γίνεται η ανάγκη για την βέλτιστη και αποδοτικότερη κατανομή του έτσι ώστε να παρέχεται επαρκές εύρος ζώνης σε όσο το δυνατόν περισσότερους χρήστες (συνδρομητές). Μέθοδοι και πρακτικές για την εφαρμογή Ελέγχου Κίνησης υπάρχουν αρκετές, πολλές από τις οποίες αποτελούν επεκτάσεις των ήδη υπάρχοντων τεχνολογιών και πρωτοκόλλων και περιγράφονται στα RFCs 3630, 3970, 5543, 5329, 4105 και άλλα. Ωστόσο, όπως ήδη έχει αναφερθεί, τα τελευταία χρόνια η επιστημονική κοινότητα έχει στρέψει την προσοχή της στις δυνατότητες του MPLS, το οποίο παρουσιάζει εξαιρετικές δυνατότητες και προοπτικές για τον αποδοτικό έλεγχο της κίνησης και την ορθή κατανομή του φορτίου. Η προτίμηση στην συγκεκριμένη τεχνολογία, όπως αναφέρεται και στο RFC 2702, μπορεί να αποδοθεί σε αρκετούς παράγοντες όπως:

- Η δυνατότητα ρητού προσδιορισμού μονοπατιών

- Η εύκολη αντιστοίχιση ροών σε LSPs και η αποτελεσματική διαχείρισή τους
- Η δυνατότητα απόδοσης συγκεκριμένων χαρακτηριστικών στις ροές, οι οποίες μπορούν να καθορίσουν τις συμπεριφορές τους
- Η δυνατότητα τόσο συνάθροισης όσο και διαχωρισμού κίνησης (σε αντίθεση με την IP δρομολόγηση που επιτρέπει μόνο τη συσσώρευση ροών)
- Το σημαντικά χαμηλότερο overhead που παράγει, σε σύγκριση με άλλες τεχνολογίες Ελέγχου Κίνησης, και άλλα

Παρόλο που το θέμα του MPLS - TE ερευνάται ακόμη, σε διάφορες διαστάσεις και επίπεδα, οι βασικότερες λειτουργίες που αυτό πρέπει να εκτελέσει προκειμένου να μάθει, να υπολογίσει και να αποφασίσει το πώς θα χειριστεί το φορτίο των ροών που εισέρχονται στον MPLS τομέα έχουν, λίγο – πολύ, οριστεί. Οι λειτουργίες αυτές αποτελούνται από τη διανομή πληροφορίας ζεύξεων, τον υπολογισμό των διαδρομών και την LSP σηματοδότηση και θα περιγραφούν, χωρίς ιδιαίτερη λεπτομέρεια, παρακάτω.

ΔΙΑΝΟΜΗ ΠΛΗΡΟΦΟΡΙΑΣ ΖΕΥΞΕΩΝ

Στα σύγχρονα παραδοσιακά δίκτυα η πληροφορία των ζεύξεων διανέμεται μέσω πρωτοκόλλων όπως το OSPF και IS – IS και εκτελείται συνήθως πριν την άφιξη οποιουδήποτε πακέτου ή σε περιπτώσεις βλαβών στο δίκτυο. Ως αποτέλεσμα, κάθε δρομολογητής μπορεί να έχει μία σχετικά στατική απεικόνιση της τοπολογίας του δικτύου και να γνωρίζει όλα τα κόστη (π.χ. αριθμό βημάτων) για τη δρομολόγηση ενός πακέτου με οποιονδήποτε προορισμό.

Στο MPLS, ωστόσο, η διανομή της πληροφορίας των ζεύξεων μπορεί να γίνει και κατ' αίτηση, σε οποιαδήποτε χρονική στιγμή από οποιονδήποτε LSR(συνήθως, βέβαια, η αίτηση εκκινείται από τον Ingress LER) και να αφορά

το κόστος για τη δρομολόγηση ενός πακέτου (ή ροής) όχι απαραίτητα προς οποιονδήποτε, αλλά προς συγκεκριμένο προορισμό. Η πληροφορία αυτή, στους περισσότερους προταθέντες αλγορίθμους αφορά συνήθως τον αριθμό των βημάτων και το διαθέσιμο εύρος ζώνης στις ζεύξεις. Εντούτοις, επεκτάσεις των υπάρχοντων πρωτοκόλλων, όπως το OSPF – TE και το IS – IS – TE δύνανται να διανείμουν μία πολύ μεγαλύτερη λίστα χαρακτηριστικών η οποία τελικά μπορεί να χρησιμοποιηθεί από τον αιτούντα LSR για να υπολογίσει το βέλτιστο, σύμφωνα με συγκεκριμένους περιορισμούς, μονοπάτι για μία FEC. Παραδείγματα τέτοιων χαρακτηριστικών αποτελούν [\[13\]](#):

- Η διεύθυνση της διεπαφής: IP διεύθυνση της διεπαφής (interface) της ζεύξης του τρέχοντος LSR
- Διεύθυνση γείτονα: IP διεύθυνση της διεπαφής του γειτονικού δρομολογητή
- Μέγιστο bandwidth ζεύξης: Η πραγματική χωρητικότητα ζεύξης, στην κατεύθυνση του γειτονικού δρομολογητή
- Διαθέσιμο bandwidth ζεύξης : Η μέγιστη χωρητικότητα που μπορεί να δεσμευτεί, στην κατεύθυνση του γειτονικού δρομολογητή
- Διαθέσιμο bandwidth κλάσης: Διαθέσιμο bandwidth σε κάθε μία από τις 8 κλάσεις προτεραιότητας (βλ. παρακάτω)
- TE μετρική: Μετρική ζεύξης για τη Διαχείριση Κίνησης
- Διοικητική ομάδα: Τιμές που χαρακτηρίζουν μία ζεύξη ως προς το αν θα πρέπει να συμπεριληφθεί ή να αποκλειστεί κατά τον υπολογισμό της διαδρομής

Συνδυάζοντας περισσότερες μετρικές, ένας LSR μπορεί να κατασκευάσει ένα μονοπάτι το οποίο δεν περιορίζεται μόνο στο μήκος του αλλά προσφέρει πιο ποιοτική, αξιόπιστη και ελεγχόμενη μεταφορά δεδομένων.

ΥΠΟΛΟΓΙΣΜΟΣ ΔΙΑΔΡΟΜΗΣ

Ο υπολογισμός της Διαδρομής έγκειται στην αναζήτηση εκείνου του μονοπατιού το οποίο να συμφωνεί με τους περιορισμούς φορτίου και ποιότητας που ορίζει το MPLS – TE και το εκάστοτε SLA. Όπως αναφέρθηκε πιο πάνω, δεν απαιτεί τον εκ των προτέρων υπολογισμό διαδρομών από κάθε πηγή προς κάθε προορισμό, αλλά μπορεί να εκτελεστεί και κατ' αίτηση με πιο δυναμικό χαρακτήρα.

Ενώ, λοιπόν, στα παραδοσιακά δίκτυα ο υπολογισμός της διαδρομής αφορά τον υπολογισμό του μονοπατιού με τα λιγότερα βήματα, χρησιμοποιώντας μία απλή βάση τοπολογίας, στο MPLS – TE δίνεται η δυνατότητα εύρεσης εκείνου του μονοπατιού το οποίο να βελτιστοποιεί πιο συγκεκριμένους και περισσότερους στόχους από εκείνον της συντομότερης διαδρομής. Παραδείγματα τέτοιων στόχων μπορεί να είναι η μεγιστοποίηση της διέλευσης, η ελαχιστοποίηση της χρησιμοποίησης των ζεύξεων, η μείωση της καθυστέρησης ή της διακύμανσής της, η βέλτιστη και ισορροπημένη κατανομή του φορτίου και άλλα.

Ο υπολογισμός της διαδρομής εκκινείται, όπως παρουσιάστηκε πιο πάνω, από έναν Ingress LER. Η εκτέλεσή του, ωστόσο, μπορεί να γίνει οπουδήποτε. Συχνά, προκειμένου να ελαφρύνουν τους δρομολογητές απόληξης από διεργασίες οι οποίες απαιτούν σεβαστό κόπο και χρόνο, οι διαχειριστές δικτύων προτιμούν να θέτουν σε λειτουργία ειδικούς εξυπηρετητές οι οποίοι εκτελούν τέτοιες λειτουργίες. Έτσι, ένας LSR αντί να υπολογίζει μόνος του το εκάστοτε μονοπάτι και να διατηρεί ποικίλες πληροφορίες τοπολογίας στη μνήμη, μπορεί απλά να αποστείλει μία αίτηση υπολογισμού με τα κατάλληλα δεδομένα εισόδου σε έναν εξυπηρετητή, ο οποίος θα εκτελέσει τον αντίστοιχο υπολογισμό και θα του επιστρέψει την ακριβή διαδρομή ή ένα σύνολο εφικτών μονοπατιών. Αξίζει να τονιστεί, βέβαια, ότι η κάθε λύση κρύβει τις δικές της αδυναμίες. Για παράδειγμα, ενώ ο υπολογισμός διαδρομής στον Ingress LER, όπως αναφέρθηκε, απαιτεί πόρους από άποψη κύκλων επεξεργαστή και μνήμης, η ανάθεση του υπολογισμού σε έναν εξυπηρετητή μπορεί να αυξήσει την απαιτούμενη σηματοδосία (άρα και το αντίστοιχο overhead) στο δίκτυο

και απαιτεί τον καθορισμό και την τήρηση καλά ορισμένου πρωτοκόλλου για την επικοινωνία των δύο μερών.

Περισσότερα πάνω στον υπολογισμό των διαδρομών και την επιλογή των στόχων προς βελτιστοποίηση θα παρουσιαστούν σε παρακάτω ενότητα στην οποία θα εξεταστούν ποικίλοι προταθέντες αλγόριθμοι και η συμβολή τους στην βέλτιστη παροχή ποιότητας υπηρεσιών και διαχείρισης φορτίου.

LSPΣΗΜΑΤΟΔΟΣΙΑ

Τον υπολογισμό της διαδρομής διαδέχεται η εγκατάσταση του μονοπατιού, η οποία εν γένει περιλαμβάνει την διανομή ετικετών και τη δέσμευση των πόρων. Παρ' όλο που έχουν προταθεί διάφοροι τρόποι για την εκτέλεση της διαδικασίας αυτής, σαν πιο «κομψή» προσέγγιση επικράτησε η επέκταση του RSVP για τη σηματοδосία των LSPs, στο οποίο προστέθηκαν πέντε νέα αντικείμενα [\[4\]](#):

- LABEL_REQUEST

Χρησιμοποιείται με τα RSVPPATH μηνύματα για να αιτηθεί η δέσμευση μίας ετικέτας, σε κάθε κόμβο.

- LABEL

Αντικείμενο που χρησιμοποιείται από τα RSVP μηνύματα, μαζί με το LABEL_REQUEST για την κατανομή της ετικέτας.

- EXPLICIT_ROUTE

Το EXPLICIT_ROUTE αντικείμενο περιλαμβάνει μία λίστα βημάτων (hops) η οποία απεικονίζει το επιλεγθέν μονοπάτι

- RECORD_ROUTE

Το RECORD_ROUTE αντικείμενο συλλέγει την πληροφορία των ζεύξεων και των ετικετών, κατά μήκος της διαδρομής σηματοδосίας

- SESSION_ATTRIBUTE

Το SESSION_ATTRIBUTE απεικονίζει τις απαιτήσεις για το LSP, όπως την προτεραιότητα, την προστασία και άλλα.

Τα RSVPPATH μηνύματα, όπως και στην προηγούμενη υλοποίησή του RSVP, προωθούνται από τον κόμβο πηγής προς τον κόμβο προορισμού, μαζί με μία συλλογή αντικειμένων, τέσσερις από τις οποίες αντιστοιχούν στο MPLSTE (EXPLICIT_ROUTE, LABEL_REQUEST, SESSION_ATTRIBUTE, RECORD_ROUTE). Τα RESV μηνύματα, από την άλλη, αποστέλλονται προς τα πίσω, πάνω στην ζεύξη ανόδου και περιλαμβάνουν τα αντικείμενα LABEL και RECORD_ROUTE. Η ανταλλαγή αυτών των μηνυμάτων ανάμεσα στους δρομολογητές ολοκληρώνει, ως ένα βαθμό, την Διαχείριση της Κίνησης, με τους τελευταίους να εκτελούν ελέγχους αποδοχής, να κατασκευάζουν τις LSP Forwarding Bases (LFIB) και να κατανέμουν τις αντίστοιχες ετικέτες για την επερχόμενη προώθηση των πακέτων.

ΘΕΜΑΤΑ ΣΧΕΔΙΑΣΜΟΥ ΔΙΑΧΕΙΡΙΣΗΣ ΚΙΝΗΣΗΣ ΚΑΙ ΠΟΡΩΝ

Προκειμένου να υποστηρίξει την Διαχείριση Κίνησης και Πόρων, το MPLS, όπως δείχθηκε παραπάνω, επεκτάθηκε στο MPLS – TE ορίζοντας συγκεκριμένες απαιτήσεις που πρέπει να τηρούνται κατά την προκείμενη διαδικασία, καθώς και τα χαρακτηριστικά που πρέπει ή δύναται να έχουν τα συστατικά μέρη και οι οντότητες ενός MPLS τομέα. Τα θέματα αυτά θα πρέπει, στο σύνολό τους ή μερικώς, να λαμβάνονται υπ’ όψιν κατά το σχεδιασμό μηχανισμών και αλγορίθμων που έχουν στόχο την επίλυση του εν λόγω προβλήματος και περιλαμβάνουν ζητήματα όπως: οι δυνατοί περιορισμοί κίνησης που μπορούν να προσδιοριστούν, ο ρητός προσδιορισμός και η επαναβελτιστοποίηση των διαδρομών, οι απαιτήσεις ως προς την κατανομή των τελευταίων, οι προσδιορισμένες κλάσεις και τα επίπεδα προτεραιότητας που μπορούν να οριστούν, οι απαιτήσεις για την

προστασία και την αστυνόμευση των ροών, ο τρόπος εκτέλεσης των εν λειτουργία αλγορίθμων για την υπό ανάλυση διαδικασία και άλλα.

ΠΑΡΑΜΕΤΡΟΙ ΚΙΝΗΣΗΣ ΚΑΙ ΠΕΡΙΟΡΙΣΜΟΙ

Οι παράμετροι κίνησης αναφέρονται στα επιθυμητά χαρακτηριστικά που πρέπει να έχει η εκάστοτε FEC και μπορούν να περιλαμβάνουν πληροφορίες όπως ο μέγιστος και μέσος ρυθμός μετάδοσης, τα επίπεδα ανοχής σε απώλειες, η καθυστέρηση και η διακύμανση καθυστέρησης, τα μεγέθη πακέτων και άλλα. Οι παράμετροι κίνησης, ουσιαστικά, απεικονίζουν τις απαιτήσεις που έχει η εκάστοτε ροή σε πόρους και μπορούν να χρησιμοποιηθούν σε λειτουργίες δέσμευσης πόρων, αποφυγής συμφόρησης και πρόβλεψης απαιτούμενου εύρους ζώνης. Αξίζει να τονιστεί ότι ο υπολογισμός της βέλτιστης διαδρομής σύμφωνα με δύο ή περισσότερους αθροιστικούς (όπως καθυστέρηση) ή πολλαπλασιαστικούς (όπως λόγος απώλειας) περιορισμούς μπορεί να αναχθεί σε NP - complete (πολυωνυμικού χρόνου) πρόβλημα. Ωστόσο, η ανάκτηση ακριβών τιμών των μετρικών όπως η καθυστέρηση ή η διακύμανσή της, είναι δύσκολη. Έτσι, η γενική πρακτική συνίσταται στη μετατροπή αθροιστικών και πολλαπλασιαστικών μετρικών σε bandwidth απαίτηση.

ΡΗΤΟΣ ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΔΙΑΔΡΟΜΩΝ

Το MPLS προσφέρει δύο γενικούς τρόπους κατασκευής μονοπατιών: τον αυτόματο, ο οποίος μπορεί να χρησιμοποιήσει υπάρχοντα ή επεκτεταμένα πρωτόκολλα δρομολόγησης και αλγορίθμους για να υπολογίσει τη βέλτιστη διαδρομή, και το ρητό, ο οποίος δίνει τη δυνατότητα σε κόμβους και διαχειριστές δικτύου να προσδιορίζουν τμήματα ή ολόκληρη τη διαδρομή μίας ροής. Ο ρητός προσδιορισμός μονοπατιών μπορεί να χωριστεί σε τρεις κατηγορίες:

1. Διαχειριστικά προσδιορισμένες διαδρομές: Μπορούν να καθοριστούν από έναν διαχειριστή δικτύου με μερικό (τμήματα - ζεύξεις) ή καθολικό (από άκρο - εις - άκρο) τρόπο. Ο προσδιορισμός μίας ροής μπορεί να περιλαμβάνει ζεύξεις/κόμβους που πρέπει υποχρεωτικά να ακολουθήσει ή να αποφύγει μία ροή.
2. Ιεραρχικές διαδρομές: Προσδιορίζονται από έναν διαχειριστή δικτύου και μπορούν να απεικονίσουν ένα σύνολο «επιθυμητών» διαδρομών, ζεύξεων ή/και κόμβων το οποίο θα πρέπει να διασχίσει ή να αποφύγει μία ροή. Το σύνολο αυτό μπορεί να τύχει ιεράρχησης, έτσι ώστε η επιλογή να ξεκινάει πάντα από το πιο προτιμητέο και σε περίπτωση αποτυχίας (π.χ. μη εύρεση εφικτού μονοπατιού με δοθέντες περιορισμούς) να εξετάζεται η επόμενη σε σειρά προτίμησης διαδρομή.
3. Διαδρομές βασισμένες σε πόρους: Μπορούν να προσδιοριστούν τόσο από διαχειριστή όσο και από τους LSRs και απεικονίζουν εκείνους τους πόρους (ή τύπους/κλάσεις περιορισμών και πόρων) που πρέπει να χρησιμοποιηθούν ή να αποφευχθούν.

ΕΠΑΝΑΒΕΛΤΙΣΤΟΠΟΙΗΣΗ ΔΙΑΔΡΟΜΩΝ

Η αρχιτεκτονική του MPLS επιτρέπει σε ροές να τυγχάνουν επαναβελτιστοποίησης, εφ' όσον η ιδιότητα αυτή ορίζεται ρητά από το SLA ή άλλες μετρικές. Πιο συγκεκριμένα, εάν μία ροή επιτρέπει την επαναβελτιστοποίηση τότε στην περίπτωση κατάρρευσης κάποιου δικτύου ή εμφάνισης καλύτερης εναλλακτικής διαδρομής, θα υπολογιστεί εκ νέου το βέλτιστο μονοπάτι και η ροή θα ανακατευθυνθεί προς αυτό. Στην αντίθετη περίπτωση (δηλαδή, εάν δεν οριστεί ρητά), η ροή ουσιαστικά απαγορεύει την ανακατεύθυνση και βελτιστοποίηση του μονοπατιού της υπό οποιαδήποτε αλλαγή της τοπολογίας του δικτύου.

ΚΑΤΑΝΟΜΗ ΔΙΑΔΡΟΜΩΝ

Στις περιπτώσεις που μία μόνο ζεύξη δεν μπορεί να υποστηρίξει το φορτίο μίας FEC, σκόπιμος είναι ο διαμοιρασμός του σε δύο ή περισσότερες παράλληλες διαδρομές. Το MPLS, για το σκοπό αυτό, δίνει τη δυνατότητα προσδιορισμού του ποσοστού της τηλεπικοινωνιακής κίνησης της συγκεκριμένης ροής που θα ακολουθήσει το εκάστοτε παράλληλο μονοπάτι (για παράδειγμα, μπορεί να καθοριστεί ότι σε περιπτώσεις συμφόρησης ή έλλειψης πόρων, μία ροή A μπορεί να χωριστεί στις ροές A1 και A2 με ποσοστό φορτίου 40% και 60% αντίστοιχα ως προς το αρχικό μέγεθος, και να κατευθυνθεί μέσω παράλληλων μονοπατιών). Αξίζει να σημειωθεί στο σημείο αυτό ότι είναι επιθυμητό πακέτα που ανήκουν στην ίδια μεμονωμένη ροή (ίδιες διευθύνσεις πηγής/προορισμού και αριθμός θύρας) αλλά ακολουθούν διαφορετικά μονοπάτια, να μη χάνουν την ταξινόμησή τους. (Πρέπει να τονιστεί βέβαια, ότι άλλες πηγές διαφοροποιούν την συγκεκριμένη απαίτηση, ορίζοντας ότι οι μεμονωμένες ροές πακέτων δεν επιτρέπεται να διαχωρίζονται αλλά πρέπει πάντα να ακολουθούν την ίδια διαδρομή. Έτσι, εάν X είναι μία συσσώρευση πακέτων και Y μία μεμονωμένη ροή συγκεκριμένης διεύθυνσης και θύρας πηγής και προορισμού η οποία ανήκει στην X, τότε : η X μπορεί να διαχωριστεί σε Z μονοπάτια με τρόπο τέτοιο ώστε τα πακέτα της Y να ακολουθήσουν ένα μοναδικό μονοπάτι.)

ΤΕΚΛΑΣΕΙΣ ΚΑΙ ΕΠΙΠΕΔΑ ΠΡΟΤΕΡΑΙΟΤΗΤΑΣ

Στο MPLS, τα σύνολα των ροών μπορούν να χαρακτηριστούν από TE κλάσεις οι οποίες αποτελούνται από τον Τύπο της Κλάσης και το Επίπεδο Προτεραιότητας ανά κλάση [\[9\]](#). Ένα LSP, δηλαδή, που μεταφέρει φορτίο συγκεκριμένου Τύπου Κλάσης μπορεί να χρησιμοποιήσει το Επίπεδο Προτεραιότητας για να ορίσει την προτεραιότητα εγκατάστασης ή/και την προτεραιότητα διατήρησης μίας ροής. Για τις TE κλάσεις μπορούν να ισχύουν τα ακόλουθα: Ροές συγκεκριμένης TE κλάσης, που ανήκουν στον ίδιο Τύπο

Κλάσης, διαφέρουν ως προς την προτεραιότητά τους. Αντίστοιχα, ροές της ίδιας TE κλάσης, με τον ίδιο βαθμό προτεραιότητας, πρέπει να διαφέρουν ως προς τον Τύπο της Κλάσης τους.

Τα επίπεδα προτεραιότητας στο MPLS είναι 7 : αριθμούνται από το 0 έως το 7, με το επίπεδο 0 να είναι δεσμευμένο για μελλοντική ή ερευνητική χρήση. Κάθε επίπεδο απεικονίζει το βαθμό της προτεραιότητας που έχει μία ροή: όσο πιο μικρό το επίπεδο (ο αύξων αριθμός) τόσο υψηλότερη η προτεραιότητα. Αυτό σημαίνει ότι: εάν η προτεραιότητα αναφέρεται στην εγκατάσταση, τότε ροές με υψηλότερο επίπεδο προτεραιότητας (χαμηλός αύξων αριθμός) πρέπει να εγκατασταθούν πριν από τις χαμηλότερου επιπέδου. Εάν, δε, η προτεραιότητα αναφέρεται στην διατήρηση της ροής, αυτό σημαίνει ότι σε καταστάσεις συμφόρησης ή ανεπαρκών πόρων, μία ροή υψηλότερης προτεραιότητας μπορεί να δεσμεύσει τους πόρους ροών χαμηλότερης.

Τα επίπεδα προτεραιότητας διατήρησης μπορούν να δημιουργήσουν τέσσερις τύπους ροών: 1. Ροή που έχει δικαίωμα να καταλάβει μία άλλη, 2. Ροή που δεν έχει το δικαίωμα να καταλάβει μία άλλη, 3. Ροή που μπορεί να τύχει κατάληψης και 4. Ροή που δεν μπορεί να τύχει κατάληψης. Έτσι, λοιπόν, τα επίπεδα διατήρησης καθορίζουν κανόνες: για παράδειγμα, μία ροή μπορεί να διακοπεί μόνο εάν ανήκει στον τύπο 3 και εάν η ροή που αιτεί τη διακοπή της είναι τύπου 1. Έτσι, οι πιθανοί συνδυασμοί (σύμφωνα με τους παραπάνω τύπους) που επιτρέπεται να υλοποιηθούν είναι: (1, 3), (1, 4), (2, 3) και (2, 4).

Λαμβάνοντας υπ' όψιν τα παραπάνω, μία αίτηση εγκατάστασης LSP μπορεί να έχει, πλέον, δύο σημαντικές παραμέτρους: το bandwidth και το επίπεδο προτεραιότητας. Το σύνολο των LSPs που δύναται να τύχει κατάληψης, μπορεί να επιλεγεί με την βελτιστοποίηση μίας αντικειμενικής συνάρτησης η οποία απεικονίζει τους δύο αυτούς περιορισμούς καθώς και το πλήθος LSPs που μπορούν να καταληφθούν. Πιο συγκεκριμένα, μία τέτοια αντικειμενική συνάρτηση μπορεί να στοχεύει στην κατάληψη των LSPs με το χαμηλότερο επίπεδο προτεραιότητας, στην ελαχιστοποίηση του πλήθους κατειλημμένων LSPs, στην κατάληψη του ελαχίστου συνολικού bandwidth το οποίο να μπορεί να υποστηρίξει την νέα απαίτηση, στην ελαχιστοποίηση της πιθανότητας αποκλεισμού μίας διακεκομμένης ροής, στην ελαχιστοποίηση του cascading

(δηλαδή, του πλήθους των επόμενων ροών που διακόπτονται και χρήζουν επανεγκατάστασης, λόγω της εγκατάστασης μίας ροής υψηλότερου επιπέδου) και ούτω καθ' εξής.

ΠΡΟΣΤΑΣΙΑ ΚΑΙ ΕΥΕΛΙΞΙΑ

Η ιδιότητα της ευελιξίας (ή προσαρμοστικότητας) προσδιορίζει την συμπεριφορά μίας ροής υπό συνθήκες συμφόρησης ή κατάρρευσης μίας οντότητας κατά μήκος της LSP της. Παρόλο που έχουν προταθεί ποικίλοι μηχανισμοί αντιμετώπισης τέτοιων περιπτώσεων (όπως μη επαναδρομολόγηση της ροής, υπολογισμός διαφορετικού μονοπατιού εάν υπάρχουν οι απαραίτητοι πόροι, υπολογισμός διαδρομής ακόμα και αν δεν επαρκούν οι πόροι κ.ο.κ.), αυτό που φαίνεται να επικράτησε είναι η έννοια των «backup» διαδρομών.

Πιο συγκεκριμένα, το MPLS μπορεί να προσφέρει δύο τύπους αποκατάστασης: Προστασία από άκρο – εις – άκρο και τοπική προστασία. Ο πρώτος τύπος έγκειται στην εκ των προτέρων εγκατάσταση εφεδρικού μονοπατιού από τον LSR πηγής προς τον LSR προορισμού, για το κύριο LSP. Εάν παρουσιαστεί βλάβη, ο LSR πηγής θα πρέπει να ενημερωθεί σχετικώς από κάποιον ενδιάμεσο δρομολογητή, να απεγκαταστήσει την χρησιμοποιούμενη σύνδεση (LSR) και να προωθήσει τα πακέτα στην Backup διαδρομή.

Η τοπική προστασία, από την άλλη, μπορεί να χωριστεί σε προστασία ζεύξης και προστασία κόμβου. Είναι γνωστή ως MPLS Fast Reroute και προστατεύει με εφεδρικά μονοπάτια τα στοιχεία του δικτύου (όπως κόμβο ή ζεύξη). Γίνεται προφανές, λοιπόν, από τα παραπάνω, ότι ένας αλγόριθμος Διαχείρισης Κίνησης θα πρέπει να λάβει υπ' όψιν την ανάγκη για προστασία των διαδρομών και των ζεύξεων και να μεριμνήσει τόσο για τον υπολογισμό του βέλτιστου πρωτεύοντος μονοπατιού όσο και για την κατασκευή του backup του.

ΑΣΤΥΝΟΜΕΥΣΗ

Η αστυνόμευση προσδιορίζει τις ενέργειες στις οποίες θα πρέπει να υποβληθούν οι out of profile ροές και μπορεί να καθοριστεί τόσο από έναν διαχειριστή δικτύου όσο και από τις οντότητες του τελευταίου. Έτσι, αν μία ροή δε συμφωνεί με το προσυμφωνημένο SLA τότε, ανάλογα με την ιδιότητα αυτή, μπορεί να τύχει επεξεργασίας (π.χ. μορφοποίηση), να απορριφθεί ή να μην υποβληθεί σε καμία από τις προηγούμενες ενέργειες. Σημαντικό είναι να τονιστεί ότι είναι γενικά επιθυμητή η εκτέλεση διαδικασιών αστυνόμευσης στον Ingress κόμβο του δικτύου (προκειμένου να προσδιοριστεί το εάν η ροή συμφωνεί ή όχι με το SLA πριν την δρομολόγησή της) και η ελαχιστοποίησή της στον κορμό (εκτός από δυσμενείς περιπτώσεις όπως η συμφόρηση, η οποία μπορεί να οδηγήσει έναν δρομολογητή στο να μορφοποιήσει ή να απορρίψει πακέτα)

ONLINE / OFFLINE ΔΡΟΜΟΛΟΓΗΣΗ

Οι αλγόριθμοι Διαχείρισης Κίνησης μπορούν να χαρακτηριστούν σαν online ή offline, ανάλογα με το πλαίσιο της εφαρμογής τους. Η online δρομολόγηση επιλύει το πρόβλημα της δυναμικής εγκατάστασης μονοπατιών για κάθε απαίτηση που φτάνει σε έναν Ingress LSR. Η offline εγκατάσταση μονοπατιών, δε, ασχολείται με τον εκ των προτέρων υπολογισμό των διαδρομών για γνωστά ζεύγη πηγής και προορισμού. Για την επίλυση αυτού του προβλήματος απαιτείται όλο ή μέρος του πλαισίου των απαιτήσεων (SLAs, προφίλ δικτύου, στατιστικές μετρήσεις ή/και προβλέψεις) το οποίο να περιγράφει με λεπτομέρεια ή/και προσεγγιστικά το ποσό των δεδομένων που θα μεταδοθεί ανάμεσα σε κάθε ζεύγος Ingress και Egress LER. Ο βασικότερος στόχος της offline δρομολόγησης είναι, λοιπόν, η βελτιστοποίηση της χρησιμοποίησης των δικτυακών πόρων, με χρήση προηγούμενης γνώσης του τηλεπικοινωνιακού φορτίου.

ΑΛΓΟΡΙΘΜΟΙ ΔΡΟΜΟΛΟΓΗΣΗΣ ΚΑΙ ΚΑΤΑΝΟΜΗΣ ΦΟΡΤΙΟΥ

Όπως αναφέρθηκε πιο πάνω, οι ακριβείς υλοποιήσεις μηχανισμών κατανομής φορτίου και δρομολόγησης, τόσο ως προς την προσέγγιση της βελτιστοποίησης όσο και ως προς τις παραμέτρους που θα ληφθούν υπ' όψιν κατά την εκτέλεση των διαφόρων αποφάσεων και επιλογών, δεν έχουν προτυποποιηθεί αλλά είναι ανοιχτές σε προτάσεις και έρευνα. Οι προταθείσες προσεγγίσεις είναι αρκετές και συχνά μπορούν να περιλαμβάνουν όχι μόνο το MPLS αυτό καθ' αυτό, αλλά και συνδυαστικές λύσεις όπως MPLS με DiffServ, MPLS με RSVP και άλλα. Ο Tran Cong Hung [\[18\]](#) στην εργασία του διαχωρίζει αυτές τις προσεγγίσεις σε δύο τύπους: τις βασισμένες στην ποιότητα υπηρεσίας και τις βασισμένες στο φορτίο.

Η QoS based routing, βασίζεται στην δρομολόγηση των πακέτων με τρόπο τέτοιο ώστε να ικανοποιούνται οι απαιτήσεις σε ποιότητα υπηρεσίας ανάμεσα στο χρήστη και τον πάροχο δικτύου. Οι απαιτήσεις αυτές, ως επί των πλείστων, προσυμφωνούνται ανάμεσα στα δύο μέλη μέσω SLAs τα οποία μπορούν να χρησιμοποιηθούν ως μετρικές για την επιθυμητή ποιότητα υπηρεσίας και δύναται να περιλαμβάνουν παραμέτρους όπως η καθυστέρηση, ο βαθμός ανοχής σε διακύμανση καθυστέρησης και απώλειας, το κόστος που προτίθεται να διαθέσει ο χρήστης και άλλα. Βασισμένος στα παραπάνω, ο αντικειμενικός στόχος της βασισμένης στην ποιότητα δρομολόγησης, λοιπόν, είναι η εύρεση των βέλτιστων μονοπατιών και η προώθηση του φορτίου με τρόπο τέτοιο ώστε από τη μία πλευρά να ικανοποιούνται οι απαιτήσεις του χρήστη όπως αυτές προσδιορίζονται από το εκάστοτε SLA, και από την άλλη η παροχή των πόρων του δικτύου να γίνεται με βέλτιστο τρόπο αποφεύγοντας χαρακτηριστικά όπως η άνιση κατανομή πόρων, η συμφόρηση ή η μη συμφωνία οποιουδήποτε μέλους με το προσυμφωνημένο SLA.

Στη βιβλιογραφία παρατηρούνται δύο κύριες κατευθύνσεις για την επίτευξη των παραπάνω στόχων, οι οποίες ξεχωρίζουν από το μέγεθος του πλαισίου

εφαρμογής τους: η link optimization routing και η path optimization routing. Η πρώτη προσέγγιση στρέφει την προσοχή της στους περιορισμούς των links και ανάγει το πρόβλημα Διαχείρισης Κίνησης στην εύρεση εκείνου του μονοπατιού το οποίο να βελτιστοποιείται στις ζεύξεις με περιορισμένη χωρητικότητα ή χώρο αποθήκευσης. Δεδομένου ότι τέτοιες ζεύξεις (bottlenecks) αυξάνουν τον κίνδυνο της συμφόρησης, βασικοί στόχοι των εν λόγω προσεγγίσεων μπορεί να είναι ο υπολογισμός της διαδρομής που βελτιστοποιείται στα bottlenecks ή η ανίχνευση εκείνης της διαδρομής στην οποία τα bottlenecks συμφωνούν με προκαθορισμένους περιορισμούς.

Η δεύτερη προσέγγιση, η οποία συναντάται πιο συχνά στη βιβλιογραφία, λαμβάνει υπ' όψιν κάποιες ή όλες τις μετρικές όλων των ζεύξεων κατά μήκος της διαδρομής και ανάγει το πρόβλημα βελτιστοποίησης στην εύρεση εκείνου του μονοπατιού το οποίο να συμφωνεί με τους από άκρο – σε – άκρο περιορισμούς. Αξίζει να αναφερθεί ότι οι περισσότερες προτάσεις λαμβάνουν και τις δύο προσεγγίσεις υπ' όψιν, παρουσιάζοντας αλγορίθμους οι οποίοι υπολογίζουν διαδρομές που όχι μόνο βελτιστοποιούν την κατανομή πόρων από άκρο – εις – άκρο, αλλά λαμβάνουν υπ' όψιν και τους εκάστοτε περιορισμούς σε bottleneck για να αυξήσουν τη διέλευση και να μειώσουν την πιθανότητα απωλειών και συμφόρησης.

Δεδομένου ότι οι περισσότερες προτάσεις είναι συνδυαστικές, οι προταθέντες αλγόριθμοι μπορούν να διαιρεθούν σε δύο κατηγορίες: αυτούς που έχουν λύση εντός πολυωνυμικού χρόνου και αυτούς που δεν μπορούν να λυθούν με πολυωνυμικό χρόνο (NonPolynomial, NP). Η πρώτη ομάδα αλγορίθμων πρώτα επιλύει το πρόβλημα της δρομολόγησης σε πεδίο ζεύξεων (βελτιστοποίηση ζεύξης ή περιορισμοί ζεύξεων) καταλήγοντας σε έναν μικρότερο αριθμό εφικτών links. Έπειτα, εφαρμόζει λύσεις βελτιστοποίησης ή περιορισμού του μονοπατιού για να υπολογίσει την τελική διαδρομή. Η δεύτερη ομάδα αλγορίθμων, η οποία ασχολείται με ολόκληρη τη διαδρομή, στο πρώτο βήμα συλλέγει όλα τα δυνατά μονοπάτια από την πηγή στον προορισμό, και στο δεύτερο βήμα εφαρμόζει όλες τις μετρικές (διαδρομής και ζεύξεων) για να εντοπίσει εκείνο που συμφωνεί με αυτές. Βασικό μειονέκτημα της δεύτερης προσέγγισης αποτελεί το γεγονός ότι το πρόβλημα βελτιστοποίησης μπορεί να επιλυθεί σε εκθετικό χρόνο (ανάλογα με το

πλήθος των κόμβων) και δύναται να παράξει μία σχετικά προσεγγιστική βέλτιστη λύση.

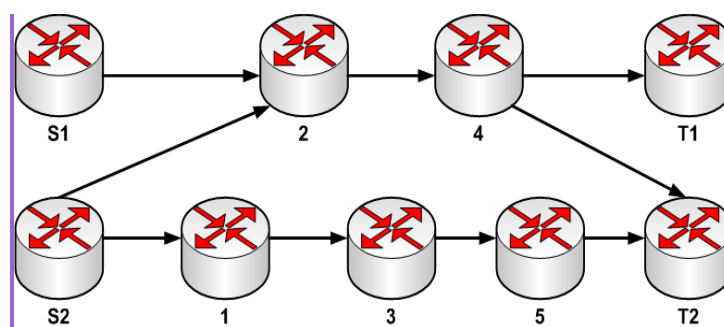
Επιστρέφοντας στις δύο κυριότερες κατευθύνσεις δρομολόγησης, κρίνεται σκόπιμο να παρουσιαστούν οι στόχοι της Traffic Based Routing. Η δρομολόγηση με βάση το φορτίο, λοιπόν, ασχολείται με τη βελτιστοποίηση των πόρων δικτύου όχι μόνο για τις τρέχουσες αλλά και για τις μελλοντικές απαιτήσεις προβλέποντας εκείνες τις ζεύξεις στις οποίες ενδέχεται να παρουσιαστεί συμφόρηση και αποτρέποντας ή μειώνοντας τη δρομολόγηση φορτίου προς αυτές. Για την ορθή υλοποίηση της προσέγγισης αυτής μπορεί να χρησιμοποιηθεί και το «προφίλ» του δικτύου, το οποίο αποτελείται ουσιαστικά από στατιστικές ή/και προσυμφωνημένες πληροφορίες που αφορούν το φορτίο και τις εκάστοτε απαιτήσεις.

Η τρέχουσα πληροφορία δικτύου μπορεί να χρησιμοποιηθεί για την εύρεση εκείνων των ζεύξεων που θα μειώσουν τη μέγιστη επιτρεπτή τιμή διέλευσης στο δίκτυο. Ένας τρόπος επίτευξης του αντικειμενικού αυτού στόχου είναι ο υπολογισμός (βάσει χωρητικότητας) όλων των μονοπατιών μεταξύ πηγής και προορισμού, τα οποία προσφέρουν μέγιστη διέλευση και μετά αφαίρεση εκείνων που περιλαμβάνουν ζεύξεις οι οποίες δύναται να τη μειώσουν (π.χ. ζεύξεις με μικρή διαθέσιμη χωρητικότητα ενταμιευτών που, όπως είναι λογικό, αυξάνουν την πιθανότητα απόρριψης μειώνοντας την τιμή της μέγιστης διέλευσης). Άλλοι τρόποι, πιο απλοί, μπορούν να περιλαμβάνουν την μέτρηση των ροών που διασχίζουν τις εκάστοτε ζεύξεις, χρησιμοποιώντας τις υπολογισμένες τιμές σαν «κόστος» σε έναν SPF αλγόριθμο. Έτσι, σαν λύση θα επιστραφούν εκείνες οι διαδρομές οι οποίες περιλαμβάνουν τις ζεύξεις με το μικρότερο αριθμό εγκατεστημένων συνδέσεων, μειώνοντας έτσι την πιθανότητα συμφόρησης.

Παρόλο, βέβαια, που οι προσεγγίσεις Διαχείρισης Κίνησης και Φορτίου διαιρούνται λογικά στις δύο κατευθύνσεις που παρατέθηκαν, οι περισσότεροι προταθέντες αλγόριθμοι συνίστανται στην ενοποίηση των δύο στόχων. Όπως θα παρουσιαστεί παρακάτω, η κοινή πρακτική τους έγκειται, σαν πρώτη φάση, στην βέλτιστη κατανομή της κίνησης με στόχο την μεγιστοποίηση της διέλευσης στο δίκτυο και, σαν δεύτερη φάση, την βελτιστοποίηση της χρήσης

και κατανομής πόρων. Συνενώνοντας τους δύο αντικειμενικούς στόχους σε έναν, καταφέρουν τοιουτοτρόπως να εφαρμόζουν πρακτικές και αποφάσεις για τη βέλτιστη κατανομή του φορτίου τόσο ποσοτικά όσο και ποιοτικά.

Ένας από τους πιο γνωστούς τέτοιους αλγορίθμους είναι ο MIRA (Minimum Interference Routing Algorithm), ο οποίος περιγράφεται στο [12] και βασίζεται στην επιλογή τέτοιων διαδρομών που να παρεμβάλλονται όσο το δυνατόν λιγότερο με άλλες διαδρομές. Η καινοτομία του, έναντι άλλων υπαρχόντων και προταθέντων αλγορίθμων, απορρέει από το γεγονός ότι δε λαμβάνει υπ' όψιν μόνο την τοπολογία του δικτύου καθολικά, αλλά εξετάζει και την σχετική θέση των Ingress και Egress LERs. Πιο συγκεκριμένα, στην εργασία [12] αποδεικνύεται το γεγονός ότι τεχνολογίες όπως η OSPF δύναται να δρομολογήσουν αρκετές ροές δεδομένων πάνω από την ίδια ζεύξη η οποία να ανήκει στις συντομότερες διαδρομές άνω του ενός ζεύγους Ingress – Egress LERs. Ένα προφανές ελάττωμα μίας τέτοιας προσέγγισης είναι η μείωση της διαθέσιμης χωρητικότητας της συγκεκριμένης ζεύξης, λόγω της αύξησης του δρομολογημένου μέσω αυτής φορτίου, και η επακόλουθη επιδείνωση της προσφερόμενης ποιότητας υπηρεσίας για τις ήδη εγκατεστημένες ροές. Ζεύξεις με τέτοια χαρακτηριστικά αναφέρονται στην εργασία ως «κρίσιμες ζεύξεις» και ορίζονται ως «ζεύξεις τέτοιες οι οποίες εάν μία ροή δρομολογηθεί διαμέσω αυτών, θα μειωθεί η μέγιστη διέλευση ενός η περισσότερων ζευγών Ingress – Egress LERs».



Εικόνα 5. Κρίσιμες Ζεύξεις βάσει MIRA

Η έννοια της «κρίσιμης ζεύξης» μπορεί να γίνει πολύ εύκολα κατανοητή από το παράδειγμα που παρουσιάζεται στην [Εικόνα 5](#). Έστω δύο ροές $f_1(S1, T1)$ και $f_2(S2, T2)$ που πρέπει να διασχίσουν το παραπάνω τμήμα δικτύου, με την

κάθε ροή να έχει απαίτηση ίση με τη μονάδα ($X_1 = X_2 = 1$). Επίσης, έστω ότι κάθε link του δικτύου έχει ελεύθερη χωρητικότητα ακριβώς ίση με 1. Εφαρμόζοντας στον παραπάνω γράφο έναν OSFP αλγόριθμο, διαπιστώνεται ότι τα επιστρεφόμενα μονοπάτια για τις ροές f_1 και f_2 θα είναι αντίστοιχα $P_1 = [S_1, 2, 4, T_1]$ και $P_2 = [S_2, 2, 4, T_2]$, περιέχοντας το ελάχιστο δυνατό πλήθος hops (ή κόμβων).

Εντούτοις, εάν επιχειρήσει κανείς να προωθήσει τις παραπάνω ροές, θα διαπιστώσει ότι δρομολογώντας την (οποιαδήποτε) πρώτη, θα οδηγήσει στον κορεσμό της ζεύξης (2, 4) με αποτέλεσμα η δεύτερη ροή να μην μπορεί να υποστηριχθεί. Φαίνεται, λοιπόν, ότι η (2, 4) είναι μία «κρίσιμη» ζεύξη στο συγκεκριμένο παράδειγμα, καθώς δρομολογώντας διαμέσω αυτής μία ροή, μειώνεται (στο προκείμενο παράδειγμα, μηδενίζεται) η μέγιστη διέλευση για το ζεύγος LERs της δεύτερης ροής. Έτσι, για το συγκεκριμένο παράδειγμα, θα ήταν προτιμότερη η επιλογή των διαδρομών $P_1 = [S_1, 2, 4, T_1]$ και $P_2 = [S_2, 1, 3, 5, T_2]$ παρ' όλο που το δεύτερο μονοπάτι δεν είναι το ελάχιστο.

Βασιζόμενος στις «κρίσιμες» ζεύξεις και έχοντας ως κύριο στόχο την αποφυγή τους, ο αλγόριθμος MIRA μπορεί να περιγραφεί ως εξής: Στο πρώτο βήμα υπολογίζονται όλες οι ζεύξεις (i, j) των οποίων η διαθέσιμη/ υπολειπόμενη χωρητικότητα είναι μεγαλύτερη ή ίση της απαιτούμενης για μία συγκεκριμένη ροή. Από το υποδίκτυο που παράγεται στο πρώτο βήμα ανιχνεύονται οι κρίσιμες ζεύξεις στις οποίες έχει ανατεθεί κόστος τέτοιο που να απεικονίζει το πόσο θα μειωθεί η τιμή της μέγιστης διέλευσης εάν μία ροή δρομολογηθεί μέσω αυτών. Τέλος, χρησιμοποιώντας τεχνικές όπως ο αλγόριθμος του Dijkstra ή των Bellman – Ford, υπολογίζεται το τελικό βέλτιστο μονοπάτι για τα υπολογισμένα βάση, παράγοντας τελικά εκείνη τη διαδρομή για δοθείσα ροή η οποία θα ελαχιστοποιήσει τη μείωση της μέγιστης διέλευσης.

Τα πειραματικά αποτελέσματα που εξήχθησαν από την εφαρμογή του αλγορίθμου σε σύγκριση με άλλους υπάρχοντες (όπως ο OSPF), έδειξαν ότι ο MIRA παράγει ελάχιστη μείωση των διαθέσιμων μονοπατιών και πολύ λίγες απορρίψεις κλήσεων, ενώ ταυτόχρονα αυξάνει την συνολική διέλευση και το ποσό της διαθέσιμης χωρητικότητας, τόσο υπό κανονικές συνθήκες όσο και υπό καταστάσεις επαναδρομολόγησης. Με τον τρόπο αυτόν, λοιπόν,

προσεγγίζει τους στόχους τόσο της βασισμένης στην κυκλοφορία όσο και της βασισμένης στους πόρους Διαχείριση Κίνησης και Πόρων. Εντούτοις, στην εργασία των [15] αποδεικνύεται ότι ο συγκεκριμένος αλγόριθμος έχει και δύο πολύ βασικές αδυναμίες:

Η πρώτη έχει να κάνει με τη μεγάλη πολυπλοκότητά του, πράγμα που σημαίνει ότι απαιτεί σημαντικούς χρονικούς πόρους για την επιτυχή εκτέλεσή του και συνεπώς δεν μπορεί να υποστηρίξει online, δυναμική εκτέλεση για την οποία είχε σχεδιαστεί. Δεύτερον, το μαθηματικό πρόβλημα που προσπαθεί να επιλύσει μπορεί να οδηγήσει σε ανισορροπία της χρησιμοποίησης του δικτύου. Το δεύτερο μπορεί να γίνει πιο κατανοητό εάν ληφθεί, για παράδειγμα, ένα ζεύγος Ingress – Egress LERs με δύο εφικτά μονοπάτια ανάμεσά τους: όταν στον Ingress LER φτάσει μία απαίτηση, αυτός θα τη δρομολογήσει πάνω από το πρώτο μονοπάτι, αφήνοντας το δεύτερο ελεύθερο προκειμένου να ελαχιστοποιήσει το φαινόμενο της παρεμβολής που εξετάστηκε παραπάνω. Με αυτόν τον τρόπο, όμως, το δεύτερο μονοπάτι στην επόμενη εκτέλεση του αλγορίθμου, θα χαρακτηριστεί σαν «κρίσιμο», καθώς εάν μία ροή δρομολογηθεί διαμέσω αυτού τότε θα μειωθεί η χωρητικότητά του, μειώνοντας και τη μέγιστη διέλευση ανάμεσα στους δύο LERs. Έτσι, μετά από μερικές επαναλήψεις του αλγορίθμου, το ένα μονοπάτι θα παρουσιάσει συμφόρηση ενώ το άλλο θα παραμένει άδειο, οδηγώντας στην τελικά άνιση κατανομή των πόρων.

Σε απάντηση στις αδυναμίες του MIRA, στην ίδια εργασία ([15]), προτάθηκε το έτος 2002 ο DORA (Dynamic Online Routing Algorithm) ο οποίος σχεδιάστηκε για online εκτέλεση, όπως δηλώνει και το όνομά του, ενώ βασίζεται στην τρέχουσα τοπολογία δικτύου για να αποφύγει ζεύξεις οι οποίες έχουν μεγαλύτερη πιθανότητα συμφόρησης. Εάν και φαινομενικά παρόμοιοι, η κύρια διαφορά μεταξύ των δύο αλγορίθμων έγκειται στο ότι ενώ ο MIRA βασίζεται στην μεγιστοποίηση της διέλευσης μέσω αποφυγής ζεύξεων που διασχίζονται από τις εκάστοτε ροές, ο DORA λαμβάνει υπ' όψιν το πλήθος των μονοπατιών (π.χ. χωρίς κανένα φορτίο να τα χρησιμοποιεί) που διασχίζουν ένα link και αποφεύγει τις ζεύξεις με το μεγαλύτερο πλήθος δυνατών διαδρομών μειώνοντας έτσι μία πιθανή μελλοντική συμφόρηση. Με πιο απλά λόγια, εάν για τον MIRA μία κρίσιμη ζεύξη είναι εκείνη από την

οποία δρομολογούνται αρκετές ροές με αποτέλεσμα να μειώνεται η μέγιστη τιμή διέλευσης, για τον DORA μία τέτοια ζεύξη θα είναι εκείνη την οποία διασχίζουν αρκετά εφικτά μονοπάτια τα οποία δύναται να αυξήσουν την πιθανότητα συμφόρησης εάν χρησιμοποιηθούν για τη δρομολόγηση φορτίου.

Για την επιτυχή εκτέλεση του παραπάνω σκεπτικού, ο DORA σαν πρώτο βήμα υπολογίζει όλα τα ανεξάρτητα μονοπάτια ανάμεσα σε κάθε ζεύγος πηγής και προορισμού. Έχοντας όλο το σύνολο των εφικτών διαδρομών, το χρησιμοποιεί για να υπολογίσει το πλήθος των μονοπατιών που διασχίζουν την κάθε ζεύξη του δικτύου, το οποίο ισούται με τη διαφορά του πλήθους των διαδρομών που διασχίζουν τη συγκεκριμένη ζεύξη και ανήκουν σε διαφορετικό ζεύγος Ingress – Egress LERs μείον το πλήθος των διαδρομών που διασχίζουν τη ζεύξη και ανήκουν στο εξεταζόμενο ζεύγος Ingress – Egress LERs. Τα βήματα αυτά, λοιπόν, κάνουν γνωστά όλα τα πιθανά μονοπάτια για οποιαδήποτε πηγή και οποιονδήποτε περιορισμό, καθώς και το βάρος κάθε ζεύξης το οποίο εκφράζεται μέσω των μονοπατιών που τη χρησιμοποιούν.

Από το βεβαρυμμένο υπογράφο που προκύπτει, αφαιρούνται εκείνες οι ζεύξεις των οποίων η διαθέσιμη χωρητικότητα είναι μικρότερη της απαιτούμενης και στο παραχθέν υποσύνολο εφαρμόζεται μία νέα συνάρτηση βάρους η οποία προέρχεται από το συνδυασμό του προηγουμένως υπολογισμένου πλήθους των διαδρομών που διασχίζουν την εκάστοτε ζεύξη, με τη διαθέσιμη χωρητικότητά της. Αξίζει να αναφερθεί ότι ενώ στην υλοποίηση του MIRA η διαθέσιμη χωρητικότητα λαμβάνεται στατικά σαν την διαφορά της δεσμευμένης από την ονομαστική, ο DORA υπολογίζει μία πιο δυναμική απεικόνισή της λαμβάνοντας υπ' όψιν την ονομαστική χωρητικότητα της ζεύξης και την τιμή διαθέσιμης χωρητικότητας της λιγότερο και της περισσότερο φορτωμένης ζεύξης στη δοθείσα χρονική στιγμή. Έχοντας, λοιπόν, κατάλληλα προσαρμοσμένα βάρη, εφαρμόζει τον αλγόριθμο Dijkstra και παράγει τη βέλτιστη διαδρομή η οποία περιέχει τα λιγότερα μονοπάτια διαφορετικών ζευγών Ingress – Egress LERs, επαρκείς διαθέσιμους πόρους και συντελεί στην καλή κατανομή του φορτίου.

Η εκτέλεση του προκείμενου αλγορίθμου, σε σύγκριση με άλλους όπως ο MIRA και ο OSPF, παρήγαγαν αρκετά θετικά αποτελέσματα τα οποία έδειξαν ότι ο DORA απορρίπτει σημαντικά λιγότερες κλήσεις και από τους δύο αναφερθέντες, ενώ ο χρόνος εκτέλεσής του πλησιάζει αυτόν του OSPF σε περιπτώσεις αραιών αλλαγών της τοπολογίας του δικτύου. Παράλληλα, ο DORA απαιτεί λιγότερα μονοπάτια να επαναδρομολογούνται (σε περιπτώσεις συμφόρησης ή εύρεσης καλύτερης διαδρομής) πράγμα λογικό εάν αναλογιστεί κανείς ότι οι ροές κατανέμονται ισορροπημένα.

Ωστόσο, ο βασικός στόχος του αλγορίθμου αυτού μπορεί τελικά να αποτελέσει και το σημαντικότερο τρωτό του σημείο. Όπως ήδη αναφέρθηκε, ο DORA επιλέγει εκείνες τις διαδρομές οι οποίες πρώτον αποτελούνται από ζεύξεις τις οποίες διασχίζει το ελάχιστο πλήθος εφικτών μονοπατιών και δεύτερον, οι οποίες μπορούν να υποστηρίξουν ισορροπημένα το απαιτούμενο φορτίο. Το δεύτερο σκέλος του στόχου είναι προφανώς επιθυμητό, σε οποιαδήποτε περίπτωση δικτύου ή τεχνολογίας. Το πρώτο, ωστόσο, μπορεί να θεωρηθεί μη απαραίτητο, έως και παραπανίσιο, αν ληφθούν υπ' όψιν τα MPLS δίκτυα. Πιο συγκεκριμένα: επιλέγοντας διαδρομές τέτοιες οι οποίες να αποτελούνται από ζεύξεις που εμφανίζονται στα λιγότερα μονοπάτια, κατανέμοντας, παράλληλα, ισορροπημένα το φορτίο στο δίκτυο, αυτό που ελαχιστοποιείται τελικά δεν είναι τόσο η συμφόρηση στις ζεύξεις, όσο οι ουρές αναμονής στους κόμβους.

Με ταυτόσημους παράγοντες φορτίου σε οποιαδήποτε ζεύξη του φορτίου, που σημαίνει ότι και ο λόγος συμφόρησης όλων των ζεύξεων θα είναι ο ίδιος (στην βέλτιστη, υποθετική περίπτωση), το επόμενο ζήτημα πιθανού λόγου συμφόρησης, επομένως, είναι η υπερχείλιση των ουρών αναμονής. Συνεπώς, κάποιος θα μπορούσε να θεωρήσει ότι το πρώτο και βασικότερο σκέλος του στόχου του DORA, επισκιαζόμενο προφανώς από το δεύτερο που κατανέμει με βέλτιστο τρόπο το απαιτούμενο φορτίο στο δίκτυο, δεν είναι τόσο η αποφυγή συμφόρησης των links όσο η αποφυγή δρομολόγησης αυξανόμενου πλήθους πακέτων προς συγκεκριμένους, «κρίσιμους» κόμβους. Όμως, όπως παρουσιάστηκε στα προηγούμενα κεφάλαια, οι λειτουργίες των LSRs είναι ελάχιστες και εκτελούνται σε ταχύτατο χρόνο καθώς βασίζονται καθαρά στη μεταγωγή ετικέτας και δεν εκτελούν χρονοβόρες διαδικασίες αναζήτησης και

υπολογισμού του επόμενου βήματος. Παράλληλα, χάριν στη συνεχή εξέλιξη της τεχνολογίας, οι σημερινοί δρομολογητές μπορούν να χαρακτηριστούν σαν ταχύτατες, μεγάλης χωρητικότητας ενταμιευτών μηχανές οι οποίες πλέον δεν περιορίζονται τόσο σημαντικά όσο οι παλαιότεροι routers ως προς τους κύκλους επεξεργαστή ή τη διαθέσιμη μνήμη. Συνεπώς, μία τόσο επαναληπτική διαδικασία όπως αυτή του υπολογισμού των «κρίσιμων» (βάσει του ορισμού του DORA) ζεύξεων, η οποία τελικά ελαφρύνει πιο πολύ τους κόμβους παρά τις ζεύξεις από το αυξανόμενο φορτίο, μπορεί να θεωρηθεί μη αναγκαία. Με τους σύγχρονους δρομολογητές η πιθανότητα υπερχειλίσις των ενταμιευτών ενός δικτύου με εξισορροπημένο φορτίο είναι πολύ μικρή για να ληφθεί υπ' όψιν μέσω τόσο επαναληπτικής διαδικασίας, οπότε μπορεί να θεωρηθεί ότι το πρώτο μέρος του DORA είναι, ουσιαστικά, μη απαραίτητα χρονοβόρο.

Αποφεύγοντας την παραπάνω αδυναμία του DORA και εστιάζοντας στο βασικότερο πρόβλημα του MIRA, σχεδιάστηκε ο MIH (Minimum Interference Hybrid) αλγόριθμος [\[11\]](#). Πιο συγκεκριμένα, όπως παρουσιάστηκε πιο πάνω, ο MIRA προσπαθεί να βελτιστοποιήσει τον παράγοντα της παρεμβολής, με αποτέλεσμα να υποβοηθά μεν στην εγκατάσταση περισσότερων συνδέσεων, αλλά χωρίς να λαμβάνει υπ' όψιν τη χρησιμοποίηση των ζεύξεων, γεγονός που μπορεί να οδηγήσει σε συμφόρηση. Ο DORA, απαντάει στην αδυναμία αυτή βελτιστοποιώντας πρώτα τον παράγοντα της παρεμβολής και έπειτα κατανέμοντας το φορτίο εντός του δικτύου για να εξισορροπήσει τη χρησιμοποίηση. Ο MIH, από την άλλη, συνεχίζοντας το σκεπτικό των MIRA και DORA, προσπαθεί να βελτιστοποιήσει τρεις παράγοντες: τον αριθμό των βημάτων, την χρησιμοποίηση των ζεύξεων και την παρεμβολή. Η μετρική που χρησιμοποιεί συνδυάζει την τρέχουσα χρησιμοποίηση μίας ζεύξης σε σχέση με τη μέγιστη τιμή της, καθώς και το βαθμό παρεμβολής που μπορεί να παρουσιαστεί από τη χρήση της ζεύξης αυτής. Παράλληλα, το πιο σημαντικό είναι ότι εισάγει την έννοια της «ρύθμισης» προσθέτοντας παράγοντες βάρους στα δύο αυτά στοιχεία, τους οποίους ένας διαχειριστής δικτύου μπορεί να ρυθμίζει προκειμένου να δηλώσει τη σημασία του κάθε παράγοντα. Για παράδειγμα, μία τιμή $T1 = 10$ ως πολλαπλασιαστής της χρησιμοποίησης της ζεύξης και $T2 = 5$ για την κρισιμότητα των ζεύξεων

δηλώνει ότι η αύξηση μίας μονάδας στη μέγιστη χρησιμοποίηση ισούται με 10 φορές αύξηση στη χρησιμοποίηση της ζεύξης και 2 φορές αύξηση στην κρισιμότητα της τελευταίας.

Σε αντίθεση με τον DORA ο οποίος ξεκινάει με τον υπολογισμό των κρίσιμων ζεύξεων και σε επόμενο βήμα εξετάζει τη χρησιμοποίηση αυτών, ο MIH εκκινεί υπολογίζοντας την τιμή της μέγιστης διέλευσης για κάθε μονοπάτι του δικτύου, εντοπίζοντας παράλληλα το σύνολο των κρίσιμων ζεύξεων σε κάθε ένα από αυτά. Έτσι, στο πρώτο βήμα γνωρίζει όχι μόνο τα μονοπάτια τα οποία αυξάνουν τη διέλευση, αλλά και την κρισιμότητα των ζεύξεων που τα απαρτίζουν. Κατόπιν, υπολογίζει την τιμή της κρισιμότητας και της χρησιμοποίησης κάθε ζεύξης για να εξάγει τα μέγιστά τους. Έχοντας όλες τις παραπάνω πληροφορίες, εφαρμόζει συνάρτηση βάρους σε κάθε link η οποία μαζί με τους παράγοντες ρύθμισης που αναφέρθηκαν παραπάνω, αντικατοπτρίζει τους λόγους (1) μέγιστης προς τρέχουσας κρισιμότητας και (2) μέγιστης προς τρέχουσας χρησιμοποίησης. Τέλος, εφαρμόζει τον αλγόριθμο του Dijkstra για να υπολογίσει το βέλτιστο μονοπάτι για την εκάστοτε ροή, το οποίο έχει το ελάχιστο, κατάλληλα ρυθμισμένο κόστος από άποψη κρισιμότητας και χρησιμοποίησης, οδηγώντας τελικά στην αύξηση της διέλευσης των ροών.

Αποδεικνύοντας ότι η βέλτιστη παραμετροποίηση για τον παραπάνω αλγόριθμο μπορεί να ρυθμιστεί στα $T1 = 8$ για την χρησιμοποίηση και $T2 = 4$ για την κρισιμότητα των ζεύξεων, τα πειραματικά αποτελέσματα της εφαρμογής του σε σύγκριση με άλλους αλγορίθμους έδειξαν ότι ο MIH επιτυγχάνει πολύ μικρότερες (σχεδόν στο μισό του MIRA) τιμές χρησιμοποίησης. Ο MIH, λοιπόν, αποφεύγει αποτελεσματικά εκείνες τις ζεύξεις οι οποίες αφ' ενός δύναται να επιφέρουν συμφόρηση από την άποψη ελεύθερης χωρητικότητας και αφ' ετέρου δύναται να χαρακτηριστούν κρίσιμες (με βάση τον ορισμό του MIRA) σε μελλοντικές απαιτήσεις, επιτυγχάνοντας έτσι μία πολύ καλή ισορροπία μεταξύ των τριών μετρικών που αναφέρθηκαν πιο πάνω. Αξίζει να αναφερθεί, για χάριν σύγκρισης, ότι ο MIH σε καταστάσεις ισορροπημένα κατανεμημένου φορτίου μπορεί να βελτιώσει τα αποτελέσματα του DORA καθώς, όπως φάνηκε παραπάνω, ο τελευταίος πρώτα υπολογίζει την κρισιμότητα και βασισμένος σε αυτήν εντοπίζει τα μονοπάτια που θα

μειώσουν τη χρησιμοποίηση, με αποτέλεσμα δοθέντος ενός ισορροπημένου πλαισίου, το πρώτο βήμα να είναι μη απαραίτητο από άποψη βελτιστοποίησης του αποτελέσματος. Ο MIH, δε, πρώτα βελτιστοποιεί την διέλευση και κατόπιν προσαρμόζει το αποτέλεσμα έτσι ώστε να έρχεται σε συμφωνία με την απαίτηση για τις κρίσιμες ζεύξεις. Έτσι, το δεύτερο βήμα βοηθάει στη βελτιστοποίηση του αντικειμενικού στόχου, χωρίς να υποπέφτει σε πιθανές ανώφελες επαναλήψεις.

Όπως εύκολα μπορεί να παρατηρήσει κανείς, οι αλγόριθμοι που περιγράφηκαν πιο πάνω φαίνεται να θεωρούν το MPLS μοντέλο πιο πολύ ως μέσο για την εφαρμογή τους και λιγότερο ως προσδιορισμένο πλαίσιο απαιτήσεων, αναγκών και χαρακτηριστικών για την επίτευξη της Διαχείρισης Φορτίου και Πόρων. Ενώ, δηλαδή, στοχεύουν στην βέλτιστη κατανομή του φορτίου και των δικτυακών πόρων, δεν λαμβάνουν υπ' όψιν τα θέματα σχεδιασμού που αναλύθηκαν παραπάνω με αποτέλεσμα παρ' όλο που σχεδιάστηκαν για το MPLS, να μην εκμεταλλεύονται τις σημαντικές δυνατότητές του και να αποφεύγουν την συμμόρφωσή τους με τις καθορισμένες αρχές και έννοιες που εξετάστηκαν. Έτσι, αρκετό ενδιαφέρον παρουσιάζει ο αλγόριθμος που περιγράφεται στο [\[2\]](#) ο οποίος στοχεύει στην μεγιστοποίηση της διέλευσης στο δίκτυο, λαμβάνει υπ' όψιν κάποιες από τις απαιτήσεις του MPLS για τους αλγορίθμους διαχείρισης πόρων, ενώ παράλληλα εισάγει νέες έννοιες και παραδοχές για το σχεδιασμό του.

Σαν πρώτο βήμα, ο προκείμενος αλγόριθμος ταξινομεί τις απαιτήσεις σύμφωνα με κάποιες παραδοχές. Παρ' όλο που δε λαμβάνει υπ' όψιν την Προτεραιότητα Εγκατάστασης που εξετάστηκε στην προηγούμενη ενότητα, τοποθετεί τις απαιτήσεις σε σειρά με βάση την ύπαρξη εκ των προτέρων προσδιορισμένων ζεύξεων, την ανάγκη για backup LSPs, την απαιτούμενη χωρητικότητα και τον υπολογισμένο αριθμό βημάτων της συντομότερης διαδρομής από την πηγή μέχρι τον προορισμό της εκάστοτε τρέχουσας, εξεταζόμενης απαίτησης. Όσον αφορά την απαιτούμενη χωρητικότητα, αυτή υπολογίζεται ως το άθροισμα της χωρητικότητας του πρωτεύοντος και του backup LSP (εάν υπάρχει), ενώ θεωρείται σαν μία αίτηση της οποίας το μήκος της διαδρομής ισούται με τη μικρότερη εκ των δύο.

Έχοντας ταξινομήσει τις τρέχουσες απαιτήσεις, αρχίζει να εξυπηρετεί την κάθε μία πρώτα βρίσκοντας όλες τις εφικτές διαδρομές από την πηγή μέχρι τον προορισμό της και έπειτα βελτιστοποιώντας την επιλεχθείσα αντικειμενική συνάρτηση για κάθε μονοπάτι καταλήγοντας τελικά στο βέλτιστο από αυτά. Για το πρώτο βήμα, προτείνεται μία προσεγγιστική, δυναμική συνάρτηση κόστους των ζεύξεων σαν είσοδος για τον αλγόριθμο Dijkstra η οποία λαμβάνει υπ' όψιν το πραγματικό φορτίο της ζεύξης, την συνολική χωρητικότητά της καθώς και την απαιτούμενη χωρητικότητα της επόμενης απαίτησης. Η συνάρτηση κόστους, τελικά, παράγει εκείνα τα μονοπάτια που περιλαμβάνουν αφενός το ελάχιστο πλήθος κόμβων γενικότερα και αφ' ετέρου τον μικρότερο αριθμό σχετικά πιο φορτωμένων ζεύξεων ειδικότερα. Το υποδίκτυο των ζεύξεων που παράγεται από το πρώτο βήμα, εξετάζεται ως προς τους υπόλοιπους αντικειμενικούς στόχους οι οποίοι προσπαθούν να βελτιστοποιήσουν το τελικό αποτέλεσμα. Συγκεκριμένα, μέσα από παραδοχές, πειραματικές προσεγγίσεις και κανονικοποιήσεις (οι οποίες δε θα παρουσιαστούν εδώ), επιλέγεται τελικά εκείνο το μονοπάτι το οποίο περιλαμβάνει τα λιγότερα βήματα και αποφεύγει τις περιοριστικές ζεύξεις.

Αυτό που παρουσιάζει ενδιαφέρον είναι η φιλοσοφία πίσω από την αποδέσμευση των μονοπατιών και την προσπάθεια εύρεσης πρωτεύοντος και backup LSP. Όσον αφορά την αποδέσμευση, η οποία εκτελείται στην περίπτωση που ο αλγόριθμος δεν καταφέρει να βελτιστοποιήσει την αντικειμενική συνάρτηση, μεγάλη σημασία δίνεται στην επιλογή των υποψηφίων μονοπατιών. Πιο συγκεκριμένα, τα υποψήφια μονοπάτια προς αποδέσμευση επιλέγονται να είναι εκείνα των οποίων η διαθέσιμη χωρητικότητα ζεύξης είναι μικρότερη της απαιτούμενης και δεύτερον η απόστασή τους από ζεύξεις με επαρκή χωρητικότητα είναι η ελάχιστη. Η απελευθέρωση διαδρομών από ζεύξεις με τα προηγούμενα χαρακτηριστικά, είναι πολύ πιθανόν να οδηγήσει στην καλύτερη εγκατάστασή τους σε επόμενο βήμα. Το πλήθος, δε, των προς απεγκατάσταση διαδρομών υπολογίζεται ως ανάλογο με τις αποτυχημένες επαναλήψεις του αλγορίθμου, κατά τις οποίες η αντικειμενική συνάρτηση δεν έτυχε βελτιστοποίησης, με αποτέλεσμα η επιτυχώς δρομολογημένη χωρητικότητα (τιμή τρέχουσας διέλευσης) να είναι αντιστρόφως ανάλογη με το σύνολο των υποψηφίων διαδρομών. Ακόμα

μεγαλύτερο ενδιαφέρον στο θέμα αυτό παρουσιάζει η επιλογή των εν προκειμένω μονοπατιών. Αξίζει να αναφερθεί ότι το 20% των υποψηφίων επιλέγονται με τυχαίο τρόπο, ενώ από το υπόλοιπο 80% τα μονοπάτια επιλέγονται από το «χειρότερο» προς το «καλύτερο», με τη μετρική του «χειρίστου» να εξαρτάται από τη διαφορά του δικού του μήκους με το ελάχιστο μήκος μονοπατιού στο δίκτυο, καθώς και από την απαιτούμενη χωρητικότητά τους. Στο τέλος της παραπάνω διαδικασίας, τα μονοπάτια αυτά εισέρχονται στην λίστα των «απαιτήσεων» με μέγιστη σειρά προτεραιότητας.

Όσον αφορά τα backup μονοπάτια, αυτά θεωρούνται σαν μία απαίτηση με τα πρωτεύοντά τους. Ωστόσο, οι δύο διαδρομές υπολογίζονται εντελώς ανεξάρτητα. Πρώτα υπολογίζεται η πρωτεύουσα διαδρομή και μετά, μαρκάροντας τις ζεύξεις της πρωτεύουσας σαν «απαγορευτικές» υπολογίζεται το backup. Η απλή αυτή μέθοδος, βέβαια, μπορεί να μην καταφέρει να καταλήξει σε εφικτά μονοπάτια. Σε τέτοιες περιπτώσεις, υπολογίζεται διαφορετικό μονοπάτι για την πρωτεύουσα διαδρομή και έπειτα ανιχνεύεται το backup μονοπάτι. Εάν και η δεύτερη μέθοδος αποτύχει, τότε ο αλγόριθμος αλλάζει τη σειρά υπολογισμού και πρώτα προσπαθεί να βρει μονοπάτι για την backup διαδρομή και κατόπιν για την κανονική.

Μέσω πραγματικών πειραμάτων και προσομοιώσεων, αποδεικνύεται στο [2] ότι ο υπό συζήτηση αλγόριθμος αποφέρει τα βέλτιστα (έως και 100%) αποτελέσματα σε πλαίσια εφαρμογής με μεγάλο αριθμό κόμβων και μικρή αρχική ποσότητα δεσμευμένης χωρητικότητας ζεύξεων, τόσο από άποψη της συνολικής διέλευσης όσο και από τη σκοπιά του απαιτούμενου χρόνου για τους παραπάνω υπολογισμούς παρουσιάζοντας, έτσι, εκπληκτικές δυνατότητες κλιμάκωσης. Παράλληλα, αν και προσεγγιστική, η λύση που υπολογίζεται πλησιάζει πάρα πολύ τη θεωρητική χάριν στις ποικίλλες παραμέτρους που συνυπολογίζονται στις συναρτήσεις και οι οποίες εκτιμήθηκαν ευρηστικά. Ωστόσο, ένα από τα πιο βασικά μειονεκτήματα του αλγορίθμου έγκειται στο γεγονός ότι ο τελευταίος δίνει μεγαλύτερη σημασία στην εύρεση του συντομότερου μονοπατιού παρά στην εξισορρόπηση του φορτίου. Οι ερευνητές, βέβαια, προτείνουν λύση για την αδυναμία αυτή, κατά την οποία κάθε εξεταζόμενο μονοπάτι βελτιστοποιείται επαναληπτικά με χρήση της μερικώς βέλτιστης τιμής που εξήχθη από κάθε προηγούμενη

εφαρμογή, μέχρι να φτάσει στο επιθυμητό ελάχιστο αλλά η λύση αυτή, ειδικά αν υπολογιστούν οι επαναληπτικές και αναδρομικές διαδικασίες των υπολοίπων βημάτων, αυξάνει σημαντικά την πολυπλοκότητα του αλγορίθμου. Επιπρόσθετα, ο αλγόριθμος λαμβάνει υπ' όψιν τους σχετικούς με το μήκος του μονοπατιού και την απαιτούμενη/διαθέσιμη χωρητικότητα περιορισμούς. Παρόλο που και άλλοι περιορισμοί θα μπορούσαν να εφαρμοστούν για να δώσουν πιο συγκεκριμένη παραμετρικά λύση, η εφαρμογή τους θα απαιτούσαν ακόμα περισσότερο χρόνο καθώς κάτι τέτοιο θα σήμαινε ακόμα μεγαλύτερη δυσκολία στους υπολογισμούς και μεγαλύτερη ανάγκη για επαναληπτική επαναβελτιστοποίηση κάθε δυνατής διαδρομής.

Η τελευταία αδυναμία του υπό ανάλυση αλγορίθμου απορρέει από το γεγονός ότι η αύξηση των παραμέτρων απόφασης και των περιορισμών σε ένα NP – hard πρόβλημα, συνεπάγεται την εκθετική αύξηση της πολυπλοκότητας και του απαιτούμενου χρόνου εκτέλεσής του. Βέβαια, όπως εύκολα μπορεί να παρατηρήσει κανείς, η προσπάθεια ανίχνευσης των βέλτιστων μονοπατιών βάσει λίγων μόνο περιορισμών, δεν είναι αδύναμο σημείο μόνο της προηγούμενης προσέγγισης. Θυμηθείτε ότι ο MIRA περιορίζεται στις κρίσιμες ζεύξεις, ο DORA στις κρίσιμες ζεύξεις και τη συμφόρηση, ο MIH προσθέτει τον αριθμό βημάτων και τη ρύθμιση των παραμέτρων, ενώ ο τελευταίος περιορίζεται στο μήκος των μονοπατιών και τη διέλευση. Οι δυνατές απαιτήσεις και οι περιορισμοί, ωστόσο, δεν είναι κατ' ανάγκη μόνο αυτοί. Όπως έχει ήδη αναφερθεί, περιορισμοί όπως το bandwidth, τα επίπεδα προτεραιότητας, οι διάφορες ιδιότητες των ζεύξεων, η ελεύθερη χωρητικότητα, οι προσδιορισμένοι σύνδεσμοι και άλλα είναι επίσης εφικτοί χάριν στην MPLS τεχνολογία.

Ένας πολύ ενδιαφέρον αλγόριθμος ο οποίος μπορεί να λάβει τους παραπάνω (και περισσότερους) περιορισμούς, ενώ ταυτόχρονα έχει μικρή πολυπλοκότητα και ευκολία στην υλοποίηση, είναι ο CSPF (Constrained Shorterst Path First) [\[13\]](#). Η φιλοσοφία του είναι πολύ απλή και μπορεί να περιγραφεί ως εξής: Εν αρχή ταξινομεί τις απαιτήσεις σύμφωνα με το επίπεδο της προτεραιότητας εγκατάστασης που αυτά κατέχουν. Εάν δύο απαιτήσεις έχουν την ίδια προτεραιότητα, επιλέγει εκείνη με το μεγαλύτερο απαιτούμενο bandwidth. Κατόπιν, αρχίζει να «αφαιρεί» ζεύξεις από την τρέχουσα

τοπολογία. Ανάλογα με τις εκάστοτε απαιτήσεις «κόβει» από την τοπολογία εκείνες τις ζεύξεις που δεν έχουν την απαιτούμενη χωρητικότητα, που δεν μπορούν να οδηγήσουν σε explicit links, που έχουν χαρακτηριστεί σαν «προς αποφυγή» και ούτω καθεξής. Τέλος, έχοντας αφαιρέσει όλες τις ζεύξεις που δεν συμφωνούν με τις αρχικές απαιτήσεις και περιορισμούς, εφαρμόζει τον αλγόριθμο συντομότερης διαδρομής στο εναπομείναν υποδίκτυο και εξάγει έτσι το βέλτιστο μονοπάτι. Αξίζει να αναφερθεί ότι παρ' όλο που η CSPF τεχνική δεν εγγυάται 100% βέλτιστη κατανομή της τηλεπικοινωνιακής κίνησης πάνω στους πόρους του δικτύου, θεωρείται γενικά μία πολύ επαρκής προσέγγιση.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Όπως ήδη παρουσιάστηκε, η Διαχείριση Κίνησης και Πόρων εστιάζει στην αποδοτική και αξιόπιστη λειτουργία των δικτύων με την παράλληλη, βέλτιστη κατανομή των πόρων και της κυκλοφοριακής επίδοσης. Η αρχιτεκτονική του MPLS, με τις σημαντικές δυνατότητες και τα χαρακτηριστικά της, δύναται να προσδώσει πολύ ισχυρά θεμέλια για την επιτυχή επίλυση του εν προκειμένω προβλήματος προσφέροντας δυνατότητες ρητού προσδιορισμού μονοπατιών, αποτελεσματικής διαχείρισης των ροών, δυνατότητες προσδιορισμού ποικίλλων περιορισμών για το εκάστοτε μονοπάτι, προστασία των διαδρομών συνολικά ή των ζεύξεων/κόμβων ειδικότερα και άλλες.

Παρ' όλο που η Διαχείριση Κίνησης και Πόρων διαχωρίζει το γενικό πρόβλημα σε δύο λογικούς τομείς, τον βασισμένο στην κίνηση και τον βασισμένο στους πόρους, η πλειοψηφία των αλγορίθμων που έχουν σχεδιαστεί για το προκειμένο θέμα προσπαθούν να υπολογίσουν μία βέλτιστη λύση η οποία να προσεγγίζει και τα δύο αυτά μαζί. Η συνήθης τακτική, είναι ο υπολογισμός των μονοπατιών με τρόπο τέτοιο ώστε αφ' ενός να μεγιστοποιείται η διέλευση ενώ αφ' ετέρου να ελαχιστοποιείται η χρησιμοποίηση των πόρων στο δίκτυο. Ο κάθε αλγόριθμος μπορεί να προσεγγίσει διαφορετικά το θέμα αυτό,

αναθέτοντας το πρόβλημα στην κρισιμότητα των ζεύξεων, στη συμφόρηση, στην ελεύθερη χωρητικότητα, στον αριθμό των βημάτων και ούτω καθ' εξής.

Εντούτοις, οι περισσότεροι αλγόριθμοι φαίνεται να χρησιμοποιούν το MPLS περισσότερο ως ένα μέσο εφαρμογής, παρά ως μία καλά ορισμένη, περιοριστική τεχνολογία για την επίτευξη τόσο της ταχύτερης μετάδοσης πακέτων όσο και της βέλτιστης διαχείρισης των πόρων και της κίνησης. Θέματα όπως ο απλός και ιεραρχικός προσδιορισμός των ζεύξεων, η προστασία, οι προτεραιότητες εγκατάστασης και διατήρηση της σύνδεσης, οι βαθμοί προτίμησης σε έναν περιορισμό έναντι ενός άλλου, καθώς και ο καθορισμός πολλαπλών, διαφορετικών απαιτήσεων ανά ροή φαίνεται να αποφεύγονται ή απλά να υπονοούνται ως «επίσης εφικτοί» στη συντριπτική πλειοψηφία των προταθέντων αλγορίθμων.

Από τους αλγορίθμους που εξετάστηκαν στην προηγούμενη ενότητα, μόνο ο CSPF και ο αλγόριθμος που περιγράφεται στο [2] φαίνεται να λαμβάνουν υπ' όψιν κάποια από τα θέματα σχεδιασμού που προσδιορίστηκαν παραπάνω, με τον πρώτο να εισάγει τη δυνατότητα προσδιορισμού πολλαπλών περιορισμών, προτεραιοτήτων και προστασίας μονοπατιών αυξάνοντας έτσι την απόκλιση από τη βέλτιστη τιμή της αντικειμενικής συνάρτησης και τον δεύτερο να προσδιορίζει κατόπιν παραδοχών την προτεραιότητα εγκατάστασης, τις backup διαδρομές καθώς και την δυνατότητα απεγκατάστασης και επανακαθορισμού μονοπατιών στην περίπτωση μη επιτυχών επαναλήψεων του αλγορίθμου, αυξάνοντας έτσι την πολυπλοκότητα του τελευταίου.

Ένα επιπρόσθετο σημαντικό σημείο μπορεί να αποτελέσει και το γεγονός ότι οι περισσότεροι αλγόριθμοι βασίζονται στην τρέχουσα τοπολογία του δικτύου, καθώς και στις τρέχουσες απαιτήσεις, μη λαμβάνοντας υπ' όψιν το προφίλ του δικτύου και τις προσυμφωνημένες ροές που θα το διασχίσουν. Αν και λογικό για αλγορίθμους που ανήκουν στην online κατηγορία, όπως οι MIRA, DORA και MIH, είναι ένα σχετικά (έως σημαντικά) περιοριστικό ζήτημα καθώς η βέλτιστη κατανομή του φορτίου τόσο από άποψη κίνησης όσο και από τη σκοπιά του φορτίου δύναται να υπολογιστεί πιο αποτελεσματικά εάν όλο το σύνολο των απαιτήσεων είναι γνωστό. Με άλλα λόγια, δοθέντος όλου του

συνόλου των απαιτήσεων που ενδέχεται να δρομολογηθούν στο δίκτυο σε οποιαδήποτε χρονική περίοδο, η κατανομή αυτών δύναται να πραγματοποιηθεί καλύτερα μιας και θα είναι γνωστοί εκ των προτέρων όλοι οι παράγοντες για την επιλογή οποιουδήποτε μονοπατιού (όπως προηγούμενες, ήδη εγκατεστημένες ροές ή μελλοντικά, εν αναμονή φορτία).

Για να μπορεί να πραγματοποιηθεί το παραπάνω, δύο προσεγγίσεις είναι δυνατές: (1) η διατήρηση στατιστικών πληροφοριών και προφίλ του δικτύου και (2) η εφαρμογή ενός offline αλγορίθμου με είσοδο όλες τις προσυμφωνημένες απαιτήσεις και τα χαρακτηριστικά τους για ένα συγκεκριμένο χρονικό διάστημα. Αξίζει να αναφερθεί ότι εκτός του προφανούς πλεονεκτήματος για την βέλτιστη κατανομή του φορτίου, η δεύτερη προσέγγιση μπορεί να εισάγει και έναν σημαντικό βαθμό χαλάρωσης στην απαιτούμενη πολυπλοκότητα του αλγορίθμου. Πιο συγκεκριμένα, εάν και λιγότερο πολύπλοκες λύσεις είναι πιο προτιμητέες, ειδικά σε online πλαίσια εφαρμογής, κάτι τέτοιο δεν είναι απαραίτητα περιοριστικό για έναν offline αλγόριθμο καθώς αυτός θα εφαρμοστεί πριν την άφιξη οποιουδήποτε πακέτου, χωρίς να διακόπτει ή να καθυστερεί οποιαδήποτε απαίτηση και έτσι δεν είναι τόσο ουσιώδης η ταχύτατη και απλή εκτέλεσή του. Μάλιστα, offline αλγόριθμοι που συναντώνται στη βιβλιογραφία μπορούν να φτάνουν και τις 10 – 15 ώρες εκτέλεσης, κατά μέσο όρο, παράγοντας ωστόσο θεαματικά αποτελέσματα.

Λαμβάνοντας υπ' όψιν όλα τα παραπάνω, στο σημείο αυτό μπορεί να προσδιοριστεί πλέον ο στόχος της προκείμενης εργασίας. Το θέμα στο οποίο εστιάζεται αυτή, λοιπόν, είναι ο σχεδιασμός αλγορίθμου διαχείρισης πόρων και κίνησης, ο οποίος μέσα στο προκαθορισμένο πλαίσιο του MPLS και σε offline κατάσταση εφαρμογής θα κατανείμει το απαιτούμενο, προσυμφωνημένο φορτίο κατά τρόπο τέτοιο ώστε να μεγιστοποιείται η διέλευση ενώ ταυτόχρονα ικανοποιούνται τα θέματα και οι απαιτήσεις σχεδιασμού όπως αυτά ορίζονται στο [RFC 2702] και άλλα. Το μαθηματικό μοντέλο, οι παραδοχές καθώς και τα εργαλεία για την επίτευξη του εν προκειμένω στόχου, θα εξεταστούν ενδελεχώς στο κεφάλαιο που ακολουθεί.

MAX – FLOW ΠΡΟΒΛΗΜΑ

ΕΙΣΑΓΩΓΗ

Το κεφάλαιο αυτό, ορμώμενο εκ των θεμάτων που εξετάστηκαν στις προηγούμενες ενότητες, παρουσιάζει το μαθηματικό πρόβλημα του υπό σχεδιασμό TEαλγορίθμου. Σε πρώτη φάση, θα παρουσιαστεί η γενικότερη προσέγγιση του προς επίλυση προβλήματος, παρουσιάζοντας τις γενικές πτυχές και τους περιορισμούς του και εξετάζοντας αναλυτικά τόσο τα θέματα του συνολικού υποβάθρου, όπως το δίκτυο και τις απαιτούμενες ροές, όσο και τα ζητήματα των περιορισμών στους οποίους υπόκειται αυτό.

Έχοντας ορίσει το μαθηματικό πρόβλημα, ακολουθεί η εισαγωγή των ζητημάτων που προσδιορίζει το MPLS. Στη δεύτερη φάση του κεφαλαίου αυτού, λοιπόν, θα προσδιοριστούν τα θέματα που θα ληφθούν υπ' όψιν, όπως η προστασία των μονοπατιών, ο προσδιορισμός των διαδρομών, οι κλάσεις προτεραιότητας κ.τ.λ., μέσω μαθηματικών τύπων και παραδοχών. Έτσι, οι δύο επόμενες ενότητες θα ορίσουν το συνολικό πλαίσιο του υπό σχεδιασμό αλγορίθμου για την επίλυση του TEπροβλήματος εντός των MPLSτομέων.

Δεδομένου ότι το υπό επίλυση πρόβλημα, από τη συνολική του σκοπιά, στρέφεται στην μεγιστοποίηση της διέλευσης, σκόπιμη κρίνεται η μελέτη της θεωρίας της μεγιστοποίησης της ροής σε δίκτυα μεταφοράς. Έτσι, στην τέταρτη ενότητα θα παρουσιαστεί το μαθηματικό πρόβλημα της MaxFlow θεωρίας εξετάζοντας το πρωτεύον και το δυϊκό της γραμμικό πρόβλημα. Η ανάλυση αυτή θα κάνει περισσότερο κατανοητά τα ζητήματα που θα επιλυθούν, καθώς θα προσδώσει τις βάσεις και το κατάλληλο θεωρητικό υπόβαθρο.

Τέλος, έχοντας δει το πρόβλημα κατανομής του φορτίου και των πόρων από τη μαθηματική και θεωρητική πλευρά, θα εξεταστούν οι πιο γνωστοί αλγόριθμοι για την επίλυση των Max – Flowπροβλημάτων που έχουν προταθεί στη βιβλιογραφία. Η ανάλυσή τους δύναται να ρίξει περισσότερο φως στους τρόπους, τις τεχνικές και τα εργαλεία για τον υπολογισμό των βέλτιστων μονοπατιών με στόχο την αύξηση της διέλευσης (ή τη μείωση της

συμφόρησης) και θα αποτελέσει τον ακρογωνιαίο λίθο, σε συνδυασμό με την προηγούμενη θεωρία, του τελικού σχεδιασμού του αλγορίθμου της προκείμενης εργασίας.

ΟΡΙΣΜΟΣ ΠΡΟΒΛΗΜΑΤΟΣ (ΓΕΝΙΚΗ ΠΡΟΣΕΓΓΙΣΗ)

ΒΑΣΙΚΟ ΠΛΑΙΣΙΟ

Έστω συνεκτικό κατευθυνόμενο γράφημα $G = (V, E, C)$ που αντιπροσωπεύει μία τοπολογία δικτύου, στην οποία V είναι το σύνολο των κόμβων (LSRs), E το σύνολο των ακμών (links) και C το συσχετισμένο με κάθε ακμή βάρος, με $C \in \mathbb{R}^+$, το οποίο δηλώνει την χωρητικότητα (capacity) και άλλους περιορισμούς του link. Οι ακμές προσδιορίζονται από την κατεύθυνσή τους, με μία ακμή $e_{ij}=(i, j)|i, j \in V$ να έχει για αρχική κορυφή τον κόμβο i και για τερματική κορυφή τον κόμβο j . Δύο ακμές $(i_0, i_1), (i_1, i_2)$ είναι συνδεδεμένες εάν η τερματική κορυφή της μίας (εδώι₁) αποτελεί την αρχική της άλλης. Το δίκτυο G έχει νόημα εάν το πλήθος των κόμβων είναι $|V| \geq 2$ (Ingress και Egress LERs), ενώ δεν υπάρχει περιορισμός στο πλήθος των ενδιάμεσων LSRs, με έναν LER να μπορεί να συνδέεται απ' ευθείας με έναν άλλον ακόμη και εάν ο πρώτος αποτελεί σημείο εισόδου ενώ ο δεύτερος σημείο εξόδου του δικτύου (περίπτωση σχεδόν απίθανη αλλά έγκυρη).

Έστω $D = \{d_1, \dots, d_k\}, k \in \mathbb{N}$ το σύνολο των απαιτήσεων (demands) με $d_i = \{s_i, t_i, b_i\}$ όπου s_i είναι η πηγή (Source/Ingress LER) του d_i , t_i είναι ο προορισμός του (Target/Egress LSR) και b_i είναι η απαίτηση του τελευταίου σε bandwidth. Παρακάτω θα φανεί ότι ένα demand d_i μπορεί να χαρακτηριστεί με περισσότερες παραμέτρους, σύμφωνες με τις απαιτήσεις του MPLS – TE. Ωστόσο, προς το παρόν, θα εξεταστεί η πιο γενική περίπτωση. Για κάθε

απαίτηση d_k ορίζεται η ροή της ως f_{ik} έτσι ώστε σε οποιαδήποτε χρονική στιγμή ΔT και σε οποιαδήποτε ακμή (i, j) η ποσότητα f_{ik} να υποδηλώνει το τμήμα της συνολικής απαίτησης d_k η οποία διασχίζει την εκάστοτε ακμή. Έτσι, εάν δε ληφθεί η ροή f_{ij} ως το ποσοστό, αλλά ως το τμήμα της πρωτεύουσας ροής, μπορεί να οριστεί ότι:

$$f_{ik} = b_i \quad (1)$$

Σε ένα κατευθυνόμενο γράφημα, ένα μονοπάτι ορίζεται ως μία ακολουθία από συνδεδεμένες ακμές. Ένα απλό μονοπάτι ονομάζεται εκείνο που δεν περιέχει την ίδια ακμή δύο ή περισσότερες φορές, ενώ χαρακτηρίζεται ως στοιχειώδες εάν δεν συναντά την ίδια κορυφή πάνω από μία φορά. Έτσι, έστω P το σύνολο των στοιχειωδών μονοπατιών του γράφου G όπου $P = (P_1, \dots, P_t)$, $t \in \mathbb{N}^+$ με $P_i = \{(i_0, i_1), (i_1, i_2), \dots, (i_x, i_y)\}$, $i \in V$, $(i, j) \in E$. Ο βασικός στόχος του παρόντος προβλήματος είναι ο προσδιορισμός των βέλτιστων μονοπατιών $OP \subseteq P$, έτσι ώστε το demand d_k να ακολουθεί το μονοπάτι OP_k που υπόκειται στους περιορισμούς και βελτιστοποιεί την αντικειμενική συνάρτηση που θα παρουσιαστεί παρακάτω.

ΠΕΡΙΟΡΙΣΜΟΣ ΧΩΡΗΤΙΚΟΤΗΤΑΣ

Εισάγεται η παράμετρος απόφασης X_{ij}^k η οποία δείχνει εάν το μονοπάτι OP_k του demand d_k διασχίζει τον κόμβο (i, j) έτσι ώστε :

$$X_{ij}^k = \begin{cases} 1 & \text{εάν το demand } d_k \text{ χρησιμοποιεί το link } i, j, \quad i, j \in V, \quad d_k \in D \\ 0, & \text{αλλιώς} \end{cases} \quad (2)$$

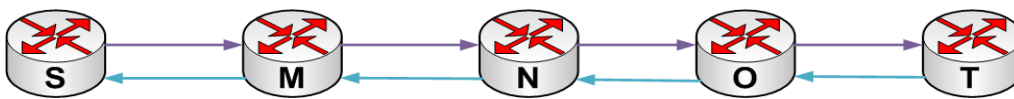
Σε συνδυασμό με το απαιτούμενο bandwidth b_k κάθε απαίτησης d_k , το φορτίο μίας ακμής ορίζεται ως το άθροισμα όλων των τμημάτων ροών εκείνων των απαιτήσεων που τη διασχίζουν. Έτσι προκύπτει ότι εάν l_{ij} είναι το φορτίο (δεσμευμένη χωρητικότητα) της ακμής (i, j) τότε

$$l_{ij} = \sum_{k=1}^{|D|} X_{ij}^k b_k, \quad i, j \in V \quad (3)$$

Το συνολικό φορτίο που μπορεί να υποστηρίξει μία οποιαδήποτε ακμή $(i, j) \in E$ δεν επιτρέπεται να υπερβαίνει την ονομαστική χωρητικότητα της τελευταίας. Έτσι προκύπτει ότι:

$$C_{ij} \geq l_{ij} \quad , \quad i, j \in V, \quad i, j \in E \quad 4$$

ΔΙΑΤΗΡΗΣΗ ΦΟΡΤΙΟΥ



Εικόνα 6. Διατήρηση Φορτίου

Το φορτίο οποιασδήποτε ροής πρέπει να διατηρείται και να μην υπόκειται σε απώλειες. Με πολύ απλά λόγια, ό,τι εισέρχεται σε έναν κόμβο θα πρέπει να εξέρχεται από αυτόν. Έστω, λοιπόν, ένας οποιοσδήποτε ενδιάμεσος κόμβος N . Στον N προσπίπτουν ένα ή περισσότερα links (m, n) με $m, n \in V, m \notin N, n \in N$ και εκπίπτουν ένα ή περισσότερα links (n, o) με $n \in N, o \notin N, n, o \in V$. Για ευκολία, ορίζονται δύο υποσύνολα $N_{in}, N_{out} \subseteq V$, τα οποία υποδηλώνουν εκείνους τους κόμβους εκ των οποίων προσπίπτουν στον/ εκπίπτουν από τον N παραπάνω ακμές. Έτσι, για τη διατήρηση του φορτίου μπορεί να λεχθεί ότι το άθροισμα των φορτίων οποιονδήποτε ροών που εισέρχονται στον κόμβο N πρέπει να ισούται με το άθροισμα των φορτίων των ίδιων ροών που εξέρχονται από αυτόν. Με το φορτίο όλων των ροών μίας ακμής να έχει οριστεί από την εξίσωση (3), προκύπτει ότι:

$$l_{mn} = l_{no} \quad , \quad m \in N_{in}, o \in N_{out}, n \in N \quad (5)$$

Για να ισχύει η εξίσωση (5), θα πρέπει το φορτίο να μην υπόκειται σε απώλειες αλλά ούτε και μεταβολές. Συνεπώς, εάν η (5) είναι αληθής, τότε χωρίς βλάβη της γενικότητας μπορεί να μετατραπεί στην :

$$l_{mn} = -l_{nm} \quad , \quad m \in N_{in}, n \in N \quad (6)$$

, η οποία ουσιαστικά δηλώνει ότι το εξερχόμενο φορτίο ισούται με την αλλαγή κατεύθυνσης του εισερχομένου. Συνεπώς, το φορτίο παραμένει ίσο, όπως ορίζει ο περιορισμός (5) αλλά η κατεύθυνσή του είναι η αντίστροφη.

Η(6), βέβαια, ισχύει για οποιονδήποτε ενδιάμεσο κόμβο ο οποίος λαμβάνει και προωθεί το φορτίο οποιασδήποτε ροής. Στους κόμβους πηγής και προορισμού, όμως, δεν ισχύει κάτι τέτοιο. Πιο συγκεκριμένα, θεωρείται ότι ο κόμβος πηγής μπορεί μόνο να παράξει φορτίο ενώ ο κόμβος προορισμού μπορεί μόνο να το καταναλώσει. Έτσι, ισχύει ότι : το εισερχόμενο φορτίο του κόμβου πηγής και το εξερχόμενο φορτίο του κόμβου προορισμού ισούνται με μηδέν. Γενικεύοντας, λοιπόν, την (6) για κάθε κόμβο στο δίκτυο προκύπτει ότι για τη διατήρηση του φορτίου πρέπει να ισχύει :

$$l_{ij} - l_{ji} = \begin{cases} l_{ij} \text{ αν } i = t_i \\ -l_{ji} \text{ αν } i = s_i \\ 0, \text{ αλλιώς} \end{cases} \quad (7)$$

ΠΕΡΙΟΡΙΣΜΟΣ ΜΟΝΟΠΑΤΙΟΥ

Προκειμένου να αποτραπεί η περίπτωση ένα μονοπάτι να διασχίζει ένα γράφο ατέρμονα, σκόπιμος είναι ο περιορισμός των βημάτων (hops) που αυτό μπορεί να εκτελέσει. Έτσι, η ανίσωση (8) υποδηλώνει ότι ένα μονοπάτι μπορεί να διασχίσει τόσους κόμβους όσοι υπάρχουν στο δίκτυο και όχι παραπάνω

$$X_{ij}^k \leq V - 1 \quad (8)$$

$i, j \in E, d_k \in D$

Παράλληλα, κάθε demand πρέπει να ακολουθεί ένα μοναδικό μονοπάτι, προκειμένου το φορτίο του να μη διαχωρίζεται. Αυτό σημαίνει ότι εάν το τμήμα ροής f_{km} της απαίτησης d_k διασχίζει ένα link (i, j) , τότε όλο το φορτίο του b_k θα πρέπει να ευρίσκεται κατά της ακμής. Δηλαδή:

$$\begin{aligned}
f_{km} X_{ij}^k &= \\
&= \begin{matrix} m \\ b_k, \text{ εάν το } d_k \text{ χρησιμοποιεί την ακμή } i, j, \\ 0, \text{ αλλιώς} \end{matrix} \quad \forall i, j \in E, d_k \in D \quad (9)
\end{aligned}$$

ΣΥΜΦΟΡΗΣΗ ΔΙΚΤΥΟΥ

Ένας από τους βασικότερους στόχους βελτιστοποίησης δικτύων που συναντάται στη βιβλιογραφία, είναι αυτός της μεγιστοποίησης της ροής. Γνωστό ως «max – flow» πρόβλημα, έχει σαν σκοπό την αύξηση της διέλευσης και υπολογίζει, ουσιαστικά, τη μέγιστη τιμή του ζέτσι ώστε τουλάχιστον το $z\%$ κάθε ροής να μπορεί να εισέλθει στο δίκτυο. Απλά αλλάζοντας την τελευταία δήλωση, το πρόβλημα μπορεί να διατυπωθεί και ως «υπολογισμός του ζέτσι ώστε το πολύ $100 - z\%$ οποιασδήποτε ροής να απορρίπτεται από το δίκτυο». Προφανές είναι ότι οι δύο δηλώσεις είναι ίδιες, καθώς η μεγιστοποίηση της διέλευσης σημαίνει ελαχιστοποίηση της απόρριψης και το αντίστροφο.

Οι λόγοι για να απορριφθεί μία ροή μπορούν να ποικίλλουν (π.χ. μη συμμόρφωση με το SLA, ανέφικτες explicit ζεύξεις ή μονοπάτια, αποτυχιές ελέγχων αστυνόμευσης κ.ο.κ.) και οι περισσότεροι δύναται να αποφευχθούν μέσω αλλαγών στα συμφωνητικά, διορθώσεων λογισμικού, τοπικών εγκαταστάσεων υλικού και άλλα. Ο πιο σημαντικός λόγος, ωστόσο, για την απόρριψη μίας κλήσης, και ταυτόχρονα ο πιο δύσκολος ως προς την αποφυγή, είναι η συμφόρηση του δικτύου. Δεδομένου ότι κάθε link έχει σταθερή ονομαστική χωρητικότητα, εύκολα γίνεται αντιληπτό ότι ακόμα και σε ένα δίκτυο χωρίς προβλήματα και βλάβες, η αύξηση της διέλευσης ροών που το διασχίζουν μπορεί να καταλήξει στη συμφόρηση ζεύξεων και στη συνεπαγόμενη απόρριψη των επόμενων κλήσεων.

Από τα παραπάνω, προκύπτει ότι εάν η μεγιστοποίηση της διέλευσης απαιτεί την ελαχιστοποίηση της απόρριψης τότε και η τελευταία, με τη σειρά της, απαιτεί την ελαχιστοποίηση της συμφόρησης. Η συνεπαγωγή των δηλώσεων

αυτών, μετατρέπει το «max – flow» πρόβλημα σε «min – congestion». Έτσι, έστω $u_{ij} \in \mathbb{R}$ ο βαθμός συμφόρησης μίας τυχαίας ακμής $(i, j) \in E$. Από τον ορισμό του link congestion, ισχύει ότι:

$$u_{ij} = \frac{l_{ij}}{C_{ij}} \quad (10)$$

,που σημαίνει ότι για C_{ij} αμετάβλητο, όσο πιο πολύ αυξάνεται το φορτίο σε ένα link (αύξηση ροής) τόσο πιο πολύ αυξάνεται ο δείκτης συμφόρησης και συνεπώς, αυξάνεται και η πιθανότητα απόρριψης μίας ροής. Η πιο προφανής λύση για την ελαχιστοποίηση της συμφόρησης είναι, απλά, η ελαχιστοποίηση της ροής. Όμως, κάτι τέτοιο είναι αφ' ενός αντίθετο με τον πρωταρχικό στόχο και αφ' ετέρου απαράδεκτο από την πλευρά των ISPs και των χρηστών. Η (10) ελαχιστοποιείται πλήρως όταν $u_{ij} = 0 \Rightarrow l_{ij} = 0$ που σημαίνει ότι το δίκτυο δε χρησιμοποιείται καθόλου και συνεπώς δεν υπάρχει καν νόημα στην ύπαρξή του.

Προκειμένου, λοιπόν, να υποστηρίζεται η εξυπηρέτηση όσο το δυνατόν περισσότερων ροών απορρίπτοντας όσο το δυνατόν λιγότερες από αυτές λόγω συμφόρησης, ο στόχος είναι η ελαχιστοποίηση του μέγιστου βαθμού συμφόρησης έτσι ώστε οποιαδήποτε απαίτηση d_k να μπορεί να εξυπηρετηθεί. Δηλαδή, εάν

$$u = \text{Max } u_{ij} \quad , \quad i, j \in E \quad (11)$$

, τότε

$$\sum_{k=1}^{|D|} b_k X_{ij}^k \leq u C_{ij} \Leftrightarrow l_{ij} \leq u C_{ij}, \quad d_k \in D, \quad i, j \in E \quad (12)$$

Παρατηρείται ότι η (12) είναι ισχυρότερη ανίσωση από την (4), καθώς το πρόβλημα που τίθεται έγκειται στην ελαχιστοποίηση της αντικειμενικής συνάρτησης ενώ το αποτέλεσμα της (12) είναι μικρότερο της (4). Έτσι, ο περιορισμός (4) μπορεί να αφαιρεθεί εντελώς.

ΠΡΟΒΛΗΜΑ ΒΕΛΤΙΣΤΟΠΟΙΗΣΗΣ

Από τις (1) - (12) προκύπτει το παρακάτω πρόβλημα βελτιστοποίησης:

Min(u)

Όταν :

$$l_{ij} - l_{ji} = l_{ij}, \quad i = t_i, \quad \forall d_k \in D$$

$$l_{ij} - l_{ji} = -l_{ji}, \quad i = s_i, \quad \forall d_k \in D$$

$$l_{ij} - l_{ji} = 0, \quad i \neq t_i, s_i, \quad \forall d_k \in D$$

$$l_{ij} \leq uC_{ij}, \quad d_k \in D, \quad i, j \in E$$

$$f_{km} X_{ij}^k = b_k, \quad X_{ij}^k = 1, \quad \forall i, j \in E, d_k \in D$$

m

$$f_{km} X_{ij}^k = 0, \quad X_{ij}^k = 0, \quad \forall i, j \in E, d_k \in D$$

m

$$X_{ij}^k \leq V - 1$$

$i, j \in E, d_k \in D$

$$X_{ij}^k \in [0, 1]$$

$$V \geq 2$$

$$u \geq 0$$

$$C_{ij} \geq 0$$

$$b_i \geq 0$$

ΕΠΕΚΤΑΣΗ ΒΑΣΙΚΗΣ ΠΡΟΣΕΓΓΙΣΗΣ

ΠΡΟΣΔΙΟΡΙΣΜΟΣ ΔΙΑΔΡΟΜΩΝ

Το MPLS – TE δίνει τη δυνατότητα του ρητού προσδιορισμού ολόκληρης ή τμήματος μίας διαδρομής, επιτρέποντας τον καθορισμό στοιχείων όπως ο μέγιστος αριθμός hops του εκάστοτε μονοπατιού, τα επιθυμητά links και οι κόμβοι τους οποίους θα πρέπει μία ροή να διασχίσει ή να αποφύγει. Για λόγους ευκολίας, από εδώ και πέρα θα ονομάζεται «προκαθορισμένη» μία ακμή (ή έναν κόμβο) η οποία έχει προσδιοριστεί σαν «απαιτούμενη» ή «προς αποφυγή», για να αποφευχθεί ο συνεχής δυαδικός χαρακτηρισμός της.

Ένας προκαθορισμένος κόμβος μπορεί να χαρακτηριστεί μέσω του συνόλου των ακμών που προσπίπτουν σε / εκπίπτουν από αυτόν. Ένας υποχρεωτικός κόμβος για ένα μονοπάτι P_i , λοιπόν, είναι εκείνος του οποίου η μία (τυχαία ή μη) ακμή χρησιμοποιείται υποχρεωτικά από το demand d_i . Επειδή, όμως, η προσέγγιση αυτή αυξάνει το βαθμό πολυπλοκότητας για την απόφαση της τελικής επιλογής του μονοπατιού για μία οποιαδήποτε ροή, θα αγνοηθεί ο ρητός προσδιορισμός των κόμβων και θα τεθεί καθαρά ο καθορισμός των ακμών.

Για κάθε ακμή στο γράφο, ορίζονται τρεις χαρακτηρισμοί τους οποίους αυτή μπορεί να λάβει: απαγορευμένη, απλή και υποχρεωτική. Μία ακμή χαρακτηριζόμενη σαν «απαγορευμένη» ορίζεται ως εκείνη η ακμή την οποία η εκάστοτε ροή πρέπει να αποφύγει. Σαν «υποχρεωτική» ορίζεται η ακμή την οποία υποχρεωτικά πρέπει να διασχίσει μία ροή, ενώ σαν «απλή» ορίζεται οποιαδήποτε ακμή δεν υπάγεται στις παραπάνω (είναι προφανές ότι μία ροή μπορεί να λάβει ακριβώς έναν χαρακτηρισμό ανά demand). Εάν, λοιπόν, M είναι το σύνολο των παραπάνω χαρακτηρισμών, ορίζονται δείκτες m έτσι ώστε

$$m = \begin{cases} -1, & \text{αν η ακμή είναι απαγορευμένη} \\ 1, & \text{αν η ακμή είναι υποχρεωτική} \\ 0, & \text{αλλιώς} \end{cases} \quad (13)$$

, με αποτέλεσμα μία ακμή (i, j) να μπορεί να διατυπωθεί πλέον ως $(i, j)_m$.

Λαμβάνοντας υπ' όψιν τα παραπάνω, η παράμετρος επιλογής X_{ij}^k που ορίσαμε στην εξίσωση (2) ισχύει, πλέον, ως εξής:

$$X_{ij}^k = \begin{cases} 1 & \text{εάν το } d_k \text{ χρησιμοποιεί το } link(i, j)_m, i, j \in V, d_k \in D, m \in [0, 1] \\ 0, & \text{αλλιώς} \end{cases} \quad (14)$$

Για τον προσδιορισμό των προκαθορισμένων ακμών, ορίζεται ένα σύνολο πινάκων με τον κάθε πίνακα να σχετίζεται με μία απαίτηση : $Z_{axb} = \{z_{axb}^1, \dots, z_{axb}^k\} \quad d_k \in D$. Κάθε στήλη του πίνακα περιλαμβάνει ένα σύνολο από $links(i, j)_{ma} \quad m \in \{-1, 1\}$ ενώ κάθε γραμμή προσδιορίζει το βαθμό προτεραιότητας των προσδιορισμένων ακμών, έτσι ώστε για $a_1 < a_2$ προτίμηση στο $(i, j)_{ma_1}$ είναι μεγαλύτερη από την προτίμηση για το $(i, j)_{ma_2}$. Παράλληλα, η σειρά με την οποία δηλώνονται οι προκαθορισμένες ακμές σε μία γραμμή δεν έχει καμία σημασία, καθώς λαμβάνονται υπ' όψιν σαν μεμονωμένες ζεύξεις, αγνοώντας οποιαδήποτε πιθανή σύνδεση ανάμεσά τους κατά τη διάρκεια των υπολογισμών.

$$z_{axb}^k = \begin{matrix} (i, j)_{m0_a} & \cdots & (i, j)_{m0_b} \\ \vdots & \ddots & \vdots \\ (i, j)_{ma_a} & \cdots & (i, j)_{ma_b} \end{matrix}$$

Η σημασία των γραμμών του πίνακα, μπορεί εύκολα να γίνει αντιληπτή εάν θεωρηθεί ένας πίνακας z_{axb}^k με $a > 1$ (δηλαδή πίνακας με τουλάχιστον 2 γραμμές). Το γεγονός ότι η πρώτη γραμμή ($a = 0$) χαρακτηρίζεται πιο σημαντική από τη δεύτερη ($a = 1$) σημαίνει ότι κατά τη διάρκεια του υπολογισμού της διαδρομής P_k μίας απαίτησης d_k , θα ληφθούν πρώτα υπ' όψιν ως υποχρεωτικές ακμές αυτές της πρώτης γραμμής και εάν δε βρεθεί λύση θα δοκιμαστούν οι επόμενες κατά σειρά προτεραιότητας.

Στην περίπτωση, λοιπόν, που μία απαίτηση $d_k \in D$ προσδιορίζει έναν πίνακα z_{axb}^k , ως εφικτό μονοπάτι ορίζεται εκείνο το οποίο πρώτον, υπόκειται στους περιορισμούς της παραπάνω υποενότητας και δεύτερον, περιλαμβάνει (ή αντίστοιχα δεν περιλαμβάνει) τις προσδιορισμένες ακμές. Για υποστήριξη

backward compatibility, θεωρείται ότι εάν δεν προσδιοριστεί $z \in Z$, τότε οποιαδήποτε ακμή $(i, j)_m$ του γράφου θεωρείται απλή, με $m = 0$.

Όπως ήδη αναφέρθηκε, εκτός από τις ακμές μπορεί να προκαθοριστεί και ο μέγιστος αριθμός των κόμβων τους οποίους επιτρέπεται να διασχίσει μία ροή. Έτσι, αν $h_k \in \mathbb{N}^+$ είναι το άνω όριο των hops για το μονοπάτι $P_k \in P$ οποιασδήποτε απαίτησης $d_k \in D$ τότε η (8) μπορεί να γραφεί ως:

$$X_{ij}^{km} \leq h_k \quad (15)$$

$i, j \in E, d_k \in D, m \in \{0, 1\}$

Εάν το h_k δεν προσδιοριστεί, θεωρείται ότι $h_k = |V| - 1$.

ΠΡΟΤΕΡΑΙΟΤΗΤΑ

Κάθε demand μπορεί να χαρακτηριστεί από το επίπεδο της προτεραιότητάς του με $pl_k \in PL = \{0, \dots, 7\}$ να προσδιορίζει το preemption level του demand d_k . Το πρώτο, μηδενικό επίπεδο, δε χρησιμοποιείται, ενώ για τα υπόλοιπα ισχύει ότι $pl_i > pl_j$ για $i < j$.

Δεδομένου ότι η πιθανότητα συμφόρησης αυξάνεται με την αύξηση του φορτίου στο δίκτυο, εύκολα αντιληπτό γίνεται ότι όσο λιγότερη η ροή τόσο μικρότερη η πιθανότητα απόρριψης. Παράλληλα, τόσο από την πλευρά του δικτύου όσο και από την πλευρά των χρηστών, επιθυμητή είναι η απόρριψη κλήσεων με το χαμηλότερο βαθμό προτεραιότητας, έτσι ώστε πρώτες να εξυπηρετούνται αυτές με τον υψηλότερο βαθμό, παρέχοντας μία καλύτερη εγγύηση ότι όσο το δυνατόν περισσότερες ροές υψηλού επιπέδου θα εξυπηρετηθούν.

Για να υποστηριχτεί η παραπάνω παραδοχή, λοιπόν, θεωρούνται οι απαιτήσεις σαν ένας διατεταγμένος πίνακας με τη σειρά προτεραιότητας να δίνεται με βάση:

1. Το επίπεδο προτεραιότητας

2. Το μέγεθος του b_k
3. Το μέγεθος του z_{axb}^k
4. Λοιποί περιορισμοί

Υπολογίζοντας μονοπάτια και ξεκινώντας από την πρώτη απαίτηση του ταξινομημένου, πλέον, πίνακα και συνεχίζοντας με τις επόμενες, αυξάνεται η πιθανότητα μία ροή υψηλού επιπέδου και υψηλών απαιτήσεων να εξυπηρετηθεί ταχέως και βέλτιστα. Αντίστοιχα, στην περίπτωση απόρριψης κλήσεων, η πιθανότητα να απορριφθούν μη απαιτητικά demands αυξάνεται.

Εκτός από το βαθμό προτεραιότητας για την ταξινόμηση των demands, το preemption level προσδιορίζει τις ροές που μπορούν να διακοπούν. Ωστόσο, ο αλγόριθμος που αναλύεται είναι offline, οπότε όλες οι απαιτήσεις είναι γνωστές εκ των προτέρων, ενώ παράλληλα ταξινομώντας τις με τη σειρά που αναφέρθηκαν, εξασφαλίζεται ότι η σειρά προτεραιότητας μίας οποιασδήποτε ροής της οποίας η διαδρομή υπολογίζεται, είναι χαμηλότερη από αυτή των ήδη εγκατεστημένων. Συνεπώς, η θεώρηση του επιπέδου προτεραιότητας από τη σκοπιά διακοπής των ροών δεν είναι τόσο απαραίτητη στο παρόν πλαίσιο.

ΑΠΟ ΑΚΡΟ – ΕΙΣ – ΑΚΡΟ ΠΡΟΣΤΑΣΙΑ

Το MPLS – ΤΕόπως ήδη έχει αναφερθεί, προσδιορίζει τρεις κατηγορίες προστασίας: προστασία κόμβου, προστασία ζεύξης και προστασία μονοπατιού. Ωστόσο, επειδή η παρούσα εργασία πραγματεύεται τον προσδιορισμό των βέλτιστων μονοπατιών, θα αγνοηθούν οι απαιτήσεις προστασίας κόμβων και ζεύξεων και εξεταστεί καθαρά η από άκρο – εις – άκρο προστασία.

Δεδομένου ότι ένα backup path δεν είναι τίποτα άλλο παρά άλλο ένα μονοπάτι στο γράφο, προφανές γίνεται ότι και αυτό απαιτεί τη δέσμευση των απαραίτητων πόρων, με την απαίτηση σε χωρητικότητα του backup να ισούται της αντίστοιχης του πρωτεύοντος path. Όμως, όπως αποδεικνύεται και στο [14], η πιθανότητα να χρειαστεί τελικά ένα backup μονοπάτι είναι η μικρότερη,

καθώς οι περισσότερες βλάβες μπορούν να προκληθούν και να επιδιορθωθούν σε καθαρά τοπικό επίπεδο. Παράλληλα, η αντιμετώπιση ενός backup path μίας απαίτησης d_k σαν τον κλώνο της τελευταίας διαφοροποιώντας τη μόνο ως προς τις απαγορευτικές ζεύξεις, σημαίνει ότι όχι μόνο οι απαιτούμενοι πόροι θα είναι πλέον οι διπλάσιοι, αλλά και οι ροές (πρωτεύουσα και backup) θα έχουν ακριβώς το ίδιο επίπεδο προτεραιότητας. Από τις παραπάνω παραδοχές εμφανές γίνεται ότι η κατασκευή backupμονοπατιών δύναται να αποτελέσει λόγο άσκοπης δέσμευσης πόρων, οδηγώντας έτσι στην πιθανή απόρριψη επόμενων στη σειρά ροών. Σκόπιμος, λοιπόν, είναι ο περιορισμός των backup paths, τόσο από άποψη πλήθους όσο και από άποψη επιπέδου προτεραιότητας.

Για να μειωθεί το πλήθος των απαιτούμενων backups, έτσι ώστε να μην δεσμεύεται ένα backupγια κάθε απαίτηση, λαμβάνεται η παραδοχή ότι τα εφεδρικά μονοπάτια θα υπολογίζονται και θα κατασκευάζονται μόνο για τις ροές που το απαιτούν ρητά στο SLAτους.

Από όλα τα παραπάνω, λοιπόν, δύναται πλέον ο διαχωρισμός του συνόλου των μονοπατιών του δικτύου στα εξής:

- $P = (P_1, \dots, P_t)$, $t \in \mathbb{N}^+$ με $P_i = \{(i_0, i_1), (i_1, i_2), \dots, (i_x, i_y)\}$, $i \in V$, $(i, j) \in E$ το σύνολο όλων των μονοπατιών του δικτύου
- $OP \subseteq P = (OP_1, \dots, OP_k)$, $d_k \in D$ το σύνολο των βέλτιστων πρωτεύοντων μονοπατιών, με κάθε μονοπάτι να σχετίζεται με μία ροή
- $BP \notin OP = (BP_1, \dots, BP_i) \mid (u_1, u_2)_{m=\{0,1\}} \in OP_i \Rightarrow (u_1, u_2)_{m=\{-1\}} \in BP_i$, $i < k$, $pl < 5$, το σύνολο των βέλτιστων backup μονοπατιών για τις ροές d_i που το απαιτούν, έτσι ώστε οι ακμές του πρωτεύοντος μονοπατιού να είναι «απαγορευμένες» για το backup.

και ο ορισμός μίας απαίτησης ως :

$$D = \{d_1, \dots, d_k\}, k \in \mathbb{N}$$

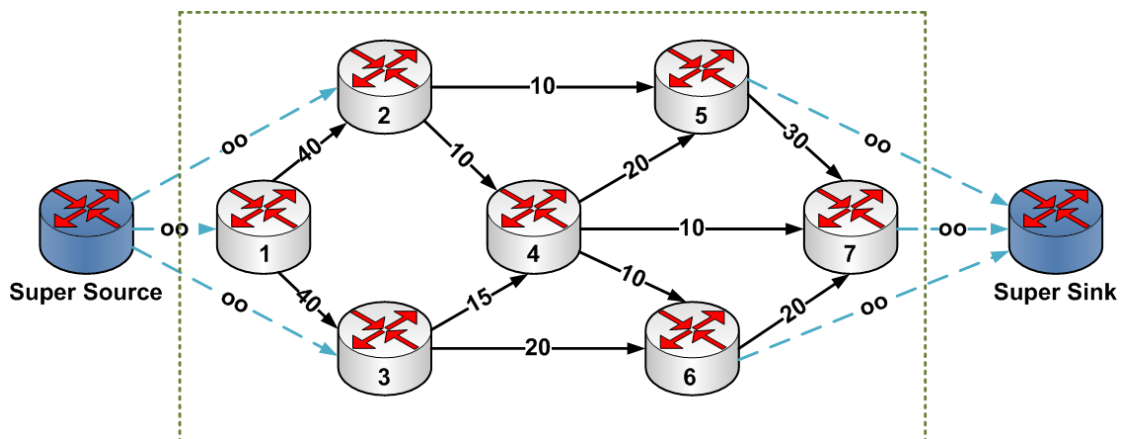
$$\text{με } d_i = \{s_i, t_i, b_i, h_i, pl_i, z_{axb}^i\}$$

ΠΡΟΒΛΗΜΑ ΜΕΓΙΣΤΗΣ ΡΟΗΣ

ΒΑΣΙΚΟ ΠΛΑΙΣΙΟ

Ένα χωρίς βρόχους γράφημα ονομάζεται δίκτυο μεταφοράς (transport network) όταν :

1. Έχει μία μόνο «είσοδο», δηλαδή μία μόνο κορυφή η οποία δεν είναι πέρας κανενός τόξου του. Η κορυφή αυτή ονομάζεται πηγή (source) και συμβολίζεται συνήθως με a .
2. Έχει μία μόνο «έξοδο», δηλαδή μία και μόνο κορυφή η οποία δεν είναι αρχική κανενός τόξου του. Η κορυφή αυτή ονομάζεται προορισμός ή καταβόθρα (sink) και συμβολίζεται με z .
3. Σε κάθε ακμή αντιστοιχεί ένας μη αρνητικός πραγματικός αριθμός, ο οποίος αντιπροσωπεύει την «χωρητικότητα» της και συμβολίζεται με $C_{ij} \forall i, j \in E$.



Εικόνα 7. Δίκτυο μεταφοράς

Ένα παράδειγμα δικτύου μεταφοράς παρουσιάζεται στην [Εικόνα 7](#). Το βασικό δίκτυο είναι αυτό που απεικονίζεται εντός του παραλληλογράμμου και περιλαμβάνει μία πηγή (1), έναν προορισμό (7) και ενδιάμεσους δρομολογητές (2, 3, 4, 5, 6). Η πηγή έχει άπειρο φορτίο το οποίο αποστέλλει

προς έναν μοναδικό προορισμό διαμέσω των ενδιάμεσων κόμβων. Ένα κλασσικό δίκτυο μεταφοράς μπορεί πολύ εύκολα να μετασχηματιστεί σε δίκτυο μεταφοράς πολλαπλών αγαθών (multi – commodity) έτσι ώστε να υποστηρίζει τη δρομολόγηση ροών από πολλαπλές πηγές προς πολλαπλούς προορισμούς. Το μόνο που αρκεί είναι να προστεθεί μία υπέρ – πηγή (super source) η οποία αποστέλλει άπειρη ποσότητα αγαθών προς την αρχική πηγή και τους άμεσα συνδεδεμένους με αυτήν ενδιάμεσους κόμβους, καθώς και μία υπέρ – καταβόθρα η οποία καταναλώνει άπειρη ποσότητα φορτίου και συνδέεται με την αρχική καταβόθρα και τους άμεσα συνδεδεμένους με αυτήν δρομολογητές.

Έστω, λοιπόν, $G(V, E, C)$ ένα δίκτυο μεταφοράς στο οποίο η πηγή a διαθέτει άπειρη ποσότητα αγαθών (π.χ. πακέτων) τα οποία επιθυμεί να αποστείλει κατά μήκος του γράφου. Ένα από τα πιο βασικά προβλήματα έγκειται στην εύρεση της μέγιστης ροής που μπορεί να πραγματοποιηθεί μέσα από το G , δηλαδή στον εντοπισμό μονοπατιών από a έως z έτσι ώστε η ροή (άρα η διέλευση) να λάβει τη μέγιστη ροή της.

Η πιο απλή, ίσως, προσέγγιση για το παραπάνω, είναι να λάβει κανείς με τυχαία σειρά όλα τα μονοπάτια από a προς z και να υπολογίσει την ποσότητα αγαθού που μπορεί να στείλει πάνω από το καθένα. Δεδομένου, βέβαια, ότι κάθε ακμή στο μονοπάτι έχει συγκεκριμένη χωρητικότητα, προφανές γίνεται ότι η μέγιστη ροή που μπορεί να διασχίσει το μονοπάτι θα ισούται με την ελάχιστη χωρητικότητα του τελευταίου. Για παράδειγμα, έστω το μονοπάτι $[1, 2, 4, 5, 7]$ της [Εικόνας 7](#). Οι χωρητικότητες των ακμών $[(1,2) (2,4) (4,5) (5,7)]$ ισούνται με $[40, 10, 20, 30]$ αντίστοιχα. Δεδομένου ότι για να «χωρέσει» μία οποιαδήποτε ροή κατά μήκος του μονοπατιού αυτού θα πρέπει να έχει τιμή μικρότερη ή ίση του 10, η οποία είναι η ελάχιστη χωρητικότητα της διαδρομής, η τιμή της μέγιστης ροής για το συγκεκριμένο, πάντα, μονοπάτι θα ισούται με 10.

Έτσι, λοιπόν, συνεχίζοντας, κάποιος θα μπορούσε να υπολογίσει όλα τα μονοπάτια ένα προς ένα, με οποιαδήποτε σειρά και να μετρήσει την τιμή της μέγιστης ροής για το κάθε ένα από αυτά. Γνωρίζοντας, πλέον, τα μονοπάτια και τις τιμές μέγιστης ροής για το καθένα, μπορεί κανείς να προωθήσει ροές

πάνω σε αυτά με τρόπο τέτοιο ώστε κάθε μονοπάτι να υπερπληρώνεται και να μην μπορεί να μεταφέρει περισσότερη ροή. Το άθροισμα όλων των ροών που διασχίζουν όλα τα διαθέσιμα μονοπάτια του G , θα δώσει τελικά τη μέγιστη δυνατή ροή του δικτύου.

Δεδομένου, λοιπόν, ότι η μέγιστη δυνατή ροή σε ένα μονοπάτι ισούται με την ελάχιστη διαθέσιμη χωρητικότητα αυτού, μπορεί να δοθεί ο ορισμός ενός αυξητικού μονοπατιού (augmenting path) ως το μονοπάτι που αποτελείται από συνδεδεμένες ακμές - βέλη $i, j \in E$, διασυνδέει την πηγή με τον προορισμό και η υπολοιπόμενη χωρητικότητα κάθε ακμής είναι αυστηρά θετική ($C_{ij} - X_{ij} > 0$).

Σύμφωνα με τον παραπάνω ορισμό, ο στόχος της υπό ανάλυσης προσέγγισης, λοιπόν, είναι η εύρεση αυξητικών μονοπατιών και η δρομολόγηση φορτίου πάνω από αυτά, το οποίο να ισούται με την ελάχιστη υπολειπόμενη χωρητικότητα οποιασδήποτε ακμής των τελευταίων. Ακολουθώντας το αναφερθέν σκεπτικό, όμως, εύκολα μπορεί να παρατηρηθεί ότι η διαφορετική επιλογή σειράς των μονοπατιών δύναται να δώσει διαφορετικές τιμές της μέγιστης δυνατής ροής στο δίκτυο (υπενθυμίζεται ότι η μέγιστη ροή του δικτύου ισούται με το άθροισμα των μεγίστων ροών κάθε μονοπατιού). Παράλληλα, εάν στον υπολογισμό ληφθούν υπ' όψιν και εκείνες οι ακμές των οποίων η κατεύθυνση είναι αντίθετη προς την κατεύθυνση της πηγής, η τελική τιμή της μέγιστης ροής δύναται να είναι μεγαλύτερη εκείνης η οποία υπολογίστηκε με τον ορισμό των αυξητικών μονοπατιών που δώσαμε προηγουμένως. Για παράδειγμα, έστω Z η μέγιστη ροή του δικτύου της [Εικόνας 7](#), η οποία υπολογίστηκε με τον παραπάνω ορισμό (με όλες τις ακμές να έχουν κατεύθυνση προς τα εμπρός). Και, έστω ότι στα υπολογισμένα μονοπάτια προστίθεται και το μονοπάτι $P = [1, 3, 6, 4, 7]$ του οποίου η ακμή με την ελάχιστη χωρητικότητα έχει $C_{ij} = x > 0$. Προκύπτει ότι μπορεί να αποσταλεί ροή μεγέθους x στο μονοπάτι P , με αποτέλεσμα η μέγιστη ροή του δικτύου να γίνεται τελικά $Z^* = Z + x > Z$.

Τα παραπάνω, αποδεικνύουν ότι ο ορισμός που δόθηκε για τα αυξητικά μονοπάτια δεν είναι επαρκής και η βέλτιστη τιμή του προβλήματος σύμφωνα

με αυτόν, εξαρτάται από εξωγενείς παράγοντες και από τυχαιότητα. Έτσι, παρουσιάζεται ένας εναλλακτικός ορισμός του αυξητικού μονοπατιού:

Ορισμός : Ένα αυξητικό μονοπάτι ροής (flow augmenting path) είναι ένα μονοπάτι από την πηγή έως τον προορισμό κατασκευασμένο από forward και backward ακμές. Για κάθε forward ακμή στο μονοπάτι, η υπολειπόμενη χωρητικότητα $C_{ij} - X_{ij}$ (με C_{ij} να είναι η ονομαστική χωρητικότητα της ακμής (i, j) και X_{ij} η τιμή της ροής που διασχίζει την ακμή (i, j)) είναι αυστηρά θετική. Για κάθε backward ακμή, υπάρχει μία ροή X_{ji} στην πρόσθια κατεύθυνση με τιμή ροής $X_{ji} > 0$ (αυστηρά θετική).

Από τον παραπάνω ορισμό, η ελάχιστη τιμή των $(C_{ij} - X_{ij})$ για forward ακμή και X_{ji} για backward ακμή, είναι η βέλτιστη τιμή η οποία θα αυξήσει την ροή. Με άλλα λόγια, $\text{Max}(f_i) = \text{Min}[(C_{ij} - X_{ij}), X_{ji}]$ για μία οποιαδήποτε ροή f_i . Αξίζει να σημειωθεί, στο σημείο αυτό, ότι όποτε μία backward ακμή εισέρχεται στο augmenting path, δε διαγράφει τη ροή, όπως θα περίμενε κανείς, αλλά την ανακατανέμει.

ΓΡΑΜΜΙΚΟ ΠΡΟΒΛΗΜΑ ΜΕΓΙΣΤΗΣ ΡΟΗΣ

Έχοντας δει, λοιπόν, το βασικό σκεπτικό πίσω από την εύρεση της μέγιστης ροής, καθώς και τον πλήρη ορισμό των αυξητικών μονοπατιών, μπορεί να δοθεί το μαθηματικό μοντέλο του προβλήματος.

Έστω δίκτυο $G(V, E, C)$. Οι $V_1, \dots, V_n \in N$ είναι οι κόμβοι του δικτύου με V_1 να αντιπροσωπεύει την πηγή και V_n τον προορισμό. Οι κόμβοι διασυνδέονται με τόξα (i, j) με $i, j \in N$, με το κάθε τόξο να χαρακτηρίζεται από χωρητικότητα $C_{ij} > 0$. Έστω X_{ij} η ποσότητα της μέγιστης ροής που διέρχεται από το τόξο (i, j) . Το γραμμικό πρόβλημα που προκύπτει είναι:

$$\text{Max}(f)$$

$$f = \sum_j X_{1j}, \forall j \in N \quad (17)$$

$$X_{ij} = X_{jk} \quad i \neq 1, n \quad (18)$$

$$\sum_j X_{1j} = \sum_i X_{in}, \forall j, i \in N \quad (19)$$

$$X_{ij} \leq C_{ij}, \forall i, j \in N \quad (20)$$

$$X_{ij} \geq 0, \forall i, j \in N \quad (21)$$

$$f \text{ χωρίς περιορισμό} \quad (22)$$

Η εξίσωση (17) δηλώνει ότι στο δίκτυο αντιστοιχεί ροή $\sum_j X_{1j}$, η οποία είναι το άθροισμα των ροών που εξέρχονται από την πηγή, της οποίας ζητείται η μέγιστη τιμή. Από την εξίσωση (19), προκύπτει, αντίστοιχα, ότι η ροή μπορεί να υπολογιστεί ως το άθροισμα των ροών που εισέρχονται στον προορισμό. Με άλλα λόγια, μπορεί να λεχθεί ότι :

$f = \sum_j X_{1j}$ ή $\sum_i X_{in} = -f$, όπου 1 είναι η πηγή, n ο προορισμός και j, i οι γειτονικοί κόμβοι τους.

Παράλληλα, η (19) εκφράζει και το γεγονός ότι δεν υπάρχουν διαρροές στο δίκτυο, καθώς η ποσότητα που εισέρχεται σε αυτό είναι ακριβώς ίση με την ποσότητα που εξέρχεται εξ' αυτού. Η εξίσωση (18), δε, εκφράζει το γεγονός ότι η ποσότητα που εισρέει σε κάθε κόμβο διάφορο της πηγής και του προορισμού είναι ακριβώς ίση με την ποσότητα που εκρέει από αυτόν.

Οι ανισώσεις (20) και (21) υποδηλώνουν ότι η αυστηρά θετική ποσότητα που διέρχεται από κάθε ακμή δεν δύναται να υπερβαίνει την χωρητικότητά του. Στο σημείο αυτό δύναται να προστεθεί ένας παραπάνω περιορισμός $f \geq 0$ για ναδειχθεί ότι η ροή δεν μπορεί να είναι αρνητική. Ωστόσο, η ροή f είναι το άθροισμα των αυστηρά μη αρνητικών ποσοτήτων X_{ij} οι οποίες προσδιορίζονται από την (21) και που θα την αναγκάσουν να λάβει επίσης αυστηρά μη αρνητική τιμή. Έτσι, χωρίς βλάβη της γενικότητας δύναται να δηλωθεί ως «unrestricted» μεταβλητή κάνοντας έτσι πιο εύκολη την εύρεση του δυϊκού προβλήματος που θα παρουσιαστεί παρακάτω. Παράλληλα, αξίζει να τονιστεί ότι το πρόβλημα μεγιστοποίησης θα μπορούσε να γραφτεί ως

$$\text{Max} \quad \sum_j X_{1j}$$

αφαιρώντας, έτσι, μία παραπάνω μεταβλητή απόφασης. Εντούτοις, η εισαγωγή της μεταβλητής f θα βοηθήσει στον πιο εύκολο μετασχηματισμό του πρωτεύοντος γραμμικού προβλήματος στο δυϊκό του, οπότε θα αποφευχθεί η αφαίρεσή της από το πρόβλημα.

ΓΡΑΜΜΙΚΟ ΠΡΟΒΛΗΜΑ ΕΛΑΧΙΣΤΗΣ ΤΟΜΗΣ

Πριν ακολουθήσει η επεξήγηση της Ελάχιστης Τομής ενός δικτύου μεταφοράς, θα εξετασθεί το δυϊκό πρόβλημα του γραμμικού προβλήματος που αναλύθηκε παραπάνω. Το πρωτεύον πρόβλημα αποτελείται από $|N|=n$ κόμβους και $|E|=m$ ακμές, άρα μαζί με τη μεταβλητή f περιλαμβάνει m μεταβλητές και $m+n+1$ περιορισμούς. Οι μεταβλητές του πρωτεύοντος προβλήματος θα είναι οι περιορισμοί του δυϊκού και αντίστροφα, οπότε μπορεί να δηλωθούν w_1, \dots, w_n δυϊκές μεταβλητές για τους περιορισμούς των κόμβων και h_{ij} δυϊκές μεταβλητές για τους περιορισμούς των ακμών του πρωτεύοντος.

Δεδομένου ότι πρώτον, το πρωτεύον γραμμικό πρόβλημα είναι ένα πρόβλημα μεγιστοποίησης και δεύτερον, όλες οι μεταβλητές του είναι μη αρνητικές, το δυϊκό πρόβλημα θα είναι ένα πρόβλημα ελαχιστοποίησης με μη αρνητικούς περιορισμούς. Έτσι, λοιπόν, εάν μετασχηματιστεί η εξίσωση (19) του πρωτεύοντος έτσι ώστε ο περιορισμός αυτός πλέον να έχει τη μορφή:

$${}_j X_{1j} - f = 0 \text{ και } {}_i X_{in} + f = 0$$

και εφαρμοστούν σε όλους τους περιορισμούς οι δυϊκές μεταβλητές, προκύπτει ότι:

$$w_i - w_j + h_{ij} \geq 0, \forall i, j \neq 1, n \quad (23)$$

$$w_n - w_1 = 1 \quad (24)$$

$$w_j \text{ unrestricted} \quad (25)$$

$$h_{ij} \geq 0 \quad (26)$$

και η αντικειμενική συνάρτηση γίνεται:

$$\text{Min}_{i,j \in N} h_{ij} U_{ij} \quad (27)$$

, όπου U_{ij} είναι η δυϊκή μεταβλητή που υποδηλώνει την χωρητικότητα της ακμής (i, j) .

Ας εξεταστεί τώρα, όμως, το πρόβλημα της Ελάχιστης Τομής και το πώς συνδέεται με το δυϊκό πρόβλημα που αναφέρθηκε παραπάνω και, ακολούθως, με το πρόβλημα της Μέγιστης Ροής. Στην [Εικόνα 7](#), παρατηρείται ότι στην πηγή καταλήγουν 3 ακμές $[(5,7) (4,7) (6,7)]$ με χωρητικότητες $[30, 10, 20]$ αντίστοιχα. Το άθροισμα των χωρητικοτήτων δίνει 60, άρα 60 είναι η τιμή της μέγιστης ροής που μπορεί να εισέλθει στον προορισμό. Παράλληλα, από την πηγή εκκινούν 2 ακμές $[(1,2) (1,3)]$ με χωρητικότητες $[40, 40]$ δίνοντας τιμή μέγιστης ροής ίση με 80. Το 80 αντιπροσωπεύει ένα άνω όριο για την μέγιστη ροή, καθώς η ροή στην πλευρά της πηγής δεν μπορεί να υπερβαίνει τις 80 μονάδες. Αντίστοιχα, το 60 αντιπροσωπεύει το άνω όριο της μέγιστης ροής από την πλευρά του προορισμού, καθώς η ροή στην πλευρά του προορισμού δεν μπορεί να υπερβαίνει τις 60 μονάδες. Παρατηρείται, λοιπόν, ότι όσο μικρότερο το άθροισμα των χωρητικοτήτων τόσο σημαντικότερη η πληροφορία για το άνω όριο της μέγιστης ροής.

Ένας τρόπος, λοιπόν, για να ληφθεί ένα άνω όριο για την μέγιστη ροή του δικτύου είναι η χρήση της Τομής Δικτύου, η οποία ορίζεται ως εκείνες οι ακμές οι οποίες εάν αφαιρεθούν από το γράφο θα αποσυνδέσουν την πηγή από τον προορισμό. Έτσι, εάν εντοπιστούν τέτοιες ακμές και αφαιρεθούν τότε αφ' ενός η πηγή θα αποσυνδεθεί από τον προορισμό και αφ' ετέρου το άθροισμα των χωρητικοτήτων των τετμημένων ακμών θα αποτελέσει ένα άνω όριο για την τιμή της μέγιστης ροής.

Έτσι, μπορούν να οριστούν δύο σύνολα κόμβων (cuts ή cutsets) $S_1, S_2 \subseteq N$ με $S_1 \cap S_2 = \emptyset$ έτσι ώστε $a \in S_1$ και $z \in S_2$ και οι υπόλοιποι κόμβοι μπορούν να βρίσκονται σε οποιοδήποτε από τα δύο σύνολα. Η χωρητικότητα της τομής $W(S_1, S_2)$ ορίζεται ως το άθροισμα των χωρητικοτήτων εκείνων των ακμών που προσπίπτουν από τις κορυφές του S_1 στις κορυφές του S_2 , δηλαδή

$$W(S_1, S_2) = \sum_{i \in S_1, j \in S_2} C_{ij}$$

Για παράδειγμα, εάν $S_1 = \{1, 2, 3\}$ και $S_2 = \{4, 5, 6, 7\}$ για το γράφο της [Εικόνας 7](#), τότε $W_{S_1, S_2} = C_{25} + C_{24} + C_{34} + C_{36} = 55$, που αυτόματα σημαίνει ότι 55 είναι ένα άνω όριο για τη μέγιστη ροή του δικτύου, άρα $f \leq 55$.

Το γεγονός ότι η κάθε τομή παρουσιάζει μία εφικτή λύση του προβλήματος της Μέγιστης Ροής, σημαίνει ότι υπάρχει κάποια ισχυρή σχέση ανάμεσα στην τομή του δικτύου και το δυϊκό πρόβλημα που παρουσιάστηκε παραπάνω. Η σχέση αυτή μπορεί να γίνει εύκολα εμφανής εάν θεωρηθεί η παρακάτω παραδοχή:

$$w_j = \begin{cases} 0, & \text{αν } j \in S_1 \\ 1, & \text{αν } j \in S_2 \end{cases}$$

Δηλαδή, η δυϊκή μεταβλητή θα λαμβάνει τιμή 0 εάν ο κόμβος που εξετάζεται βρίσκεται στο cut S_1 και τιμή 1 σε διαφορετική περίπτωση. Με τον τρόπο αυτό, ο περιορισμός [\(24\)](#) του δυϊκού προβλήματος θα ικανοποιείται πάντα, καθώς το w_1 αντιστοιχεί στον κόμβο πηγής και βρίσκεται πάντα στη διαμέριση S_1 ενώ το w_n αντιστοιχεί στον κόμβο προορισμού και βρίσκεται πάντα στο S_2 με αποτέλεσμα $w_n - w_1 = 1 - 0 = 1$.

Παράλληλα, με $w_i \in S_1$ και $w_j \in S_2$ προκύπτει ότι $w_i - w_j + h_{ij} = 1 \geq 0$ άρα και ο περιορισμός [\(23\)](#) ικανοποιείται. Τέλος, με το h_{ij} να λαμβάνει τιμή 1 για κάθε ακμή που εκκινεί από το S_1 και καταλήγει στο S_2 , προκύπτει ότι: το άθροισμα των χωρητικοτήτων των ακμών αυτών θα ισούται με την χωρητικότητα της τομής, η οποία δεν είναι τίποτα άλλο από την αντικειμενική συνάρτηση [\(27\)](#). Δεδομένου, δε, ότι η τιμή της χωρητικότητας μίας τομής δίνει ένα άνω όριο για την τιμή της μέγιστης ροής που μπορεί να διασχίσει το γράφο, συμπεραίνεται ότι εάν Z^* είναι η χωρητικότητα της ελάχιστης τομής, τότε Z^* θα είναι η τιμή της μέγιστης ροής. Με άλλα λόγια, στο βέλτιστο σημείο, η ελάχιστη τομή ισούται με τη μέγιστη ροή. Το θεώρημα αυτό αποδίδεται στους Ford και Fulkerson οι οποίοι βασισμένοι στις παραπάνω παρατηρήσεις και προτάσεις έδωσαν τον ομώνυμο αλγόριθμο που επιλύει το πρόβλημα της μέγιστης ροής.

ΑΛΓΟΡΙΘΜΟΙ ΕΥΡΕΣΗΣ ΜΕΓΙΣΤΗΣ ΡΟΗΣ

FORD - FULKERSON

Η διατύπωση του αλγορίθμου βασίζεται στην έννοια της ετικέτας L που σημειώνεται σε κάθε κόμβο του δικτύου και περιλαμβάνει έναν άλλο προσημασμένο κόμβο του και έναν μη αρνητικό αριθμό. Για την είσοδο του δικτύου, δηλαδή την πηγή, η ετικέτα ορίζεται ως $L(1) = [-, \infty]$. Ο αλγόριθμος περιγράφεται ως εξής [3], [20], [24]:

Βήμα 1.

Αρχικοποίηση: Η πηγή σημειώνεται με ετικέτα $L(1) = [-, \infty]$. Οι υπόλοιποι κόμβοι δεν έχουν καμία ετικέτα.

Βήμα 2.

Για κάθε ακμή (i, j) του δικτύου με ετικέτα στον κόμβο i και $X_{ij} \leq C_{ij}$ ανατίθεται ετικέτα $L(j) = [+i, \delta_j]$ όπου $\delta_j = \min(\delta_i, C_{ij} - X_{ij})$. Για κάθε ακμή (j, i) με ετικέτα στον κόμβο i και $X_{ji} > 0$ ανατίθεται ετικέτα $L(j) = [-i, \delta_j]$ με $\delta_j = \min(\delta_i, X_{ji})$. Το βήμα 2 επαναλαμβάνεται έως ότου συναντηθεί ο κόμβος προορισμού.

Βήμα 3.

Εάν δεν υπάρχει $L(n)$ τότε η ροή του δικτύου είναι η μέγιστη ροή. Εάν υπάρχει $L(n)$, τότε η μέγιστη ροή δεν έχει επιτευχθεί. Το δίκτυο ενημερώνεται με τη ροή X_{ij} που προέκυψε από το Βήμα 2 και εκκινώντας από την πηγή, επαναλαμβάνεται το προηγούμενο βήμα με νέες ετικέτες.

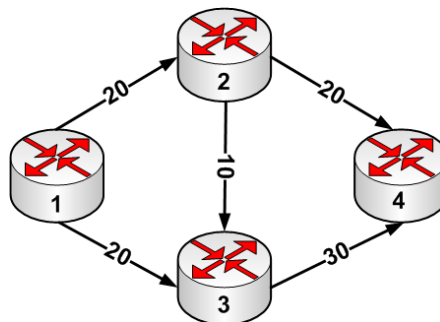
ΔΙΕΥΚΡΙΝΗΣΕΙΣ

1. Η ετικέτα $L(1) = [-, \infty]$ σημαίνει ότι από την πηγή του δικτύου διοχετεύεται άπειρη ποσότητα ροής.

2. Η ετικέτα $L(j) = [+i, \delta_j]$ σημαίνει ότι η ποσότητα X_{ij} που διέρχεται από την ακμή (i, j) μπορεί να αυξηθεί κατά δ_j .
3. Η ετικέτα $L(j) = [-i, \delta_j]$ σημαίνει ότι η ποσότητα X_{ji} που διέρχεται από την ακμή (j, i) μπορεί να μειωθεί κατά δ_j .
4. Η ετικέτα των κόμβων δεν ορίζεται πάντοτε μονοσήμαντα. Εάν ένας κόμβος μπορεί να λάβει ετικέτα με άνω του ενός τρόπου, επιλέγεται αυθαίρετα ένας από αυτούς.
5. Ο κόμβος ο οποίος ανήκει σε κορεσμένες ακμές, δηλαδή (i, j) με $X_{ij} = C_{ij}$ δεν έχει ετικέτα.
6. Η ετικέτα $L(n)$ σημαίνει ότι η πραγματοποιηθείσα εφικτή ροή δεν είναι άριστη, γιατί $L(n) = [+k, \delta_n]$ που σημαίνει ότι η ποσότητα X_{kn} δύναται να αυξηθεί κατά δ_n .
7. Η οπισθοδρόμηση, από τον προορισμό προς την πηγή με βάση τις ετικέτες, δίνει τις διαδρομές των εφικτών ροών.
8. Η μέγιστη ροή του δικτύου ισούται με το άθροισμα όλων των ροών κάθε διαδρομής που επιστρέφει ο αλγόριθμος.

ΠΑΡΑΔΕΙΓΜΑ

Για να γίνει πιο κατανοητό το σκεπτικό του εν προκειμένω αλγορίθμου, θα παρουσιαστούν τα βήματα και οι παραδοχές του μέσα από ένα παράδειγμα. Έστω, λοιπόν, το δίκτυο της [Εικόνας 8](#) το οποίο αποτελείται από 4 κόμβους και 5 ακμές.



Εικόνα 8. Παράδειγμα δικτύου μεταφοράς

Αρχικοποίηση

Για την αρχικοποίηση του αλγορίθμου, η πηγή θα λάβει ετικέτα $L1 = [-, \infty]$

Επανάληψη 1

Από την πηγή, επιλέγεται τυχαία ένας τυχαίος κόμβος με ακμή πρόσθιας κατεύθυνσης, έστω ο 2. Στο σημείο αυτό ισχύει ότι: Ο κόμβος 1 έχει ετικέτα ποσότητας ∞ , ενώ η ακμή (1, 2) έχει υπολειπόμενη χωρητικότητα ίση με $20 - 0$ (αφού δεν υφίσταται ακόμα καμία ροή που να τη χρησιμοποιεί). Επιλέγοντας την ελάχιστη τιμή των δύο ποσοτήτων, προκύπτει ότι η 2 μπορεί να καταναλώσει ποσότητα ίση με 20 η οποία να προέρχεται από τον πηγαίο κόμβο 1, οπότε θα λάβει ετικέτα $L2 = [+1, 20]$.

Κατόπιν, επιλέγεται μία τυχαία ακμή η οποία να εκπίπτει από την προηγούμενη (η οποία έχει ετικέτα), έστω 3. Ο κόμβος 2 έχει ετικέτα ποσότητας 2, ενώ η υπολειπόμενη χωρητικότητα της ακμής (2, 3) είναι 10. Η ελάχιστη τιμή ισούται με 10, οπότε η ετικέτα του 3 γίνεται $L3 = [+2, 10]$.

Από τον κόμβο 3, επιλέγεται ο κόμβος 4 ο οποίος είναι ο κόμβος προορισμού. Ο 4, λοιπόν, μπορεί να λάβει ροή ίση με 30 μονάδες από τον κόμβο 3, μέσω της ακμής (3, 4) η οποία έχει υπολειπόμενη χωρητικότητα ίση με 30, και η ετικέτα του γίνεται $L4 = [+3, 10]$

Παρατηρείται ότι ο κόμβος προορισμού έχει ετικέτα. Αυτό σημαίνει ότι η ροή προς αυτόν μπορεί να αυξηθεί κατά 10 μονάδες και συνεπώς η βέλτιστη ροή δεν έχει βρεθεί. Παράλληλα, η τιμή της μέγιστης ροής στην παρούσα φάση θα ισούται με την τιμή που έχει λάβει η ετικέτα του κόμβου 4 (10 μονάδες). Το μονοπάτι που σχηματίζεται, δε, πηγαίνοντας προς τα πίσω ξεκινώντας από τον προορισμό και λαμβάνοντας υπ' όψιν τις ετικέτες, είναι: (4, 3, 2, 1) άρα το η επανάληψη αυτή δίνει:

$$F_1 = 10, P_1 = [1, 2, 3, 4]$$

Οι ζεύξεις του υπολογισμένου μονοπατιού ενημερώνονται με τη νέα ροή (αφαιρώντας ποσότητα 10 από την ονομαστική τους χωρητικότητα) και ο αλγόριθμος προχωρά στη δεύτερη επανάληψη.

Επανάληψη 2

Στη δεύτερη επανάληψη, εξετάζονται οι ίδιοι κόμβοι με την πρώτη, ενώ λαμβάνεται η υπόθεση ότι κανένας κόμβος δεν έχει ετικέτες. Έτσι, ξεκινώντας, ο κόμβος πηγής θα έχει ετικέτα $L1 = [-, \infty]$ ενώ προς εξέταση θα επιλεγεί ο κόμβος 2. Ο κόμβος 2 μπορεί να λάβει ροή ίση με ∞ από την πηγή, αλλά η ακμή (1, 2) έχει υπολειπόμενη χωρητικότητα ίση με $20 - 10 = 10$, αφού έχει ήδη δρομολογηθεί ροή ποσότητας 10 διαμέσω αυτού, στο προηγούμενο βήμα. Έτσι, η ετικέτα που θα λάβει είναι $L2 = [+1, 10]$.

Από τον κόμβο 2, θα εξεταστεί ο κόμβος 3. Ο 3 μπορεί να λάβει ροή μεγέθους 10 από τον 2, όμως η υπολειπόμενη χωρητικότητά του είναι πλέον $10 - 10 = 0$, αφού η ονομαστική του είναι 10 ενώ ήδη εξυπηρετεί ροή μεγέθους 10 από το προηγούμενο βήμα. Έτσι, ο 3 δεν μπορεί να λάβει ετικέτα από τον 2. Ο 2, ωστόσο, μπορεί να δρομολογήσει ροή στον κόμβο 4. Η ακμή (2, 4) έχει υπολειπόμενη χωρητικότητα ίση με 10 μονάδες, οπότε η ετικέτα που θα λάβει ο κόμβος προορισμού είναι $L4 = [+2, 10]$.

Ο προορισμός έχει λάβει ετικέτα, άρα η μέγιστη ροή δεν έχει υπολογιστεί ακόμα. Η ροή της επανάληψης αυτής ισούται με 10 (όπως φαίνεται από την ετικέτα του κόμβου 4) και το μονοπάτι που προκύπτει μελετώντας τις ετικέτες προς τα πίσω είναι το (4, 2, 1). Άρα ισχύει ότι:

$$F_2 = 10, P_2 = [1, 2, 4]$$

, και η νέα ροή μεγέθους 10 προστίθεται στο γράφο.

Επανάληψη 3

Ξανά, λαμβάνει χώρα η υπόθεση ότι κανένας κόμβος δεν έχει ετικέτα. Ο κόμβος πηγής, όπως πάντα, λαμβάνει ετικέτα $L1 = [-, \infty]$. Η ακμή (1, 2) έχει υπολειπόμενη χωρητικότητα ίση με $20 - 20 = 0$, καθώς εξυπηρετεί τις δύο πρώτες ροές μεγέθους 10 η κάθε μία. Οπότε, εξετάζεται ο κόμβος 3. Ο 3 μπορεί να λάβει άπειρη ροή από τον 1, ενώ η υπολειπόμενη χωρητικότητα της ακμής (1, 3) ισούται με $20 - 0 = 20$. Έτσι, η ετικέτα του 3 γίνεται $L3 = [+1, 20]$.

Ο 3 έχει ετικέτα και συνδέεται με τον 4. Η ακμή (3, 4) έχει υπολειπόμενη χωρητικότητα ίση με $30 - 10 = 20$ και έτσι ο κόμβος 4 θα λάβει ετικέτα $L_4 = [+3, 20]$.

Έτσι, για την τρίτη ροή προκύπτει ότι :

$$F_3 = 20, P_3 = [1, 3, 4]$$

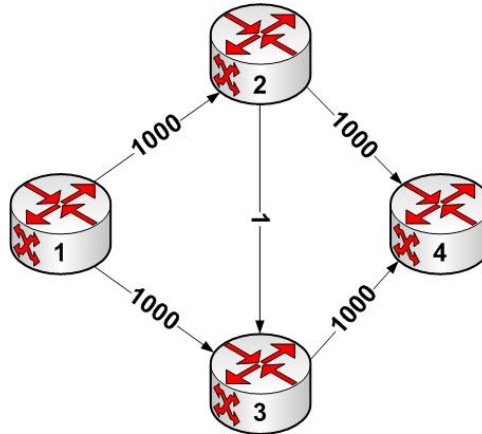
Αφού εφαρμοστεί η νέα ροή στο δίκτυο, παρατηρείται ότι δεν υπάρχει άλλο αυξητικό μονοπάτι από την πηγή ως τον προορισμό, καθώς οι ζεύξεις έχουν κορεστεί. Στο σημείο αυτό, ο αλγόριθμος σταματά, έχοντας εντοπίσει τις ροές που αναφέρθηκαν παραπάνω, και καταλήγοντας στη μέγιστη ροή μεγέθους:

$$\text{Max}(f) = F_1 + F_2 + F_3 = 10 + 10 + 20 = 40.$$

Στο παράδειγμα αυτό, βέβαια, παρουσιάστηκε η δρομολόγηση ροών εξετάζοντας μόνο ακμές με πρόσθια κατεύθυνση, χωρίς να λαμβάνονται οι backward ζεύξεις οι οποίες θα ανακαταλείμουν τη ροή. Ακολούθως εξετάζεται, λοιπόν, η χείριστη περίπτωση εφαρμογής του αλγορίθμου Ford–Fulkerson, υπολογίζοντας και τις backward ακμές, για να εμφανισθεί η βασική του αδυναμία.

ΠΕΡΙΟΡΙΣΜΟΙ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Για να γίνει κατανοητός ένας από τους πιο βασικούς περιορισμούς του εν λόγω αλγορίθμου, θεωρείται ένα δίκτυο όπως αυτό που φαίνεται στην παρακάτω εικόνα:



Εικόνα 9. Σενάριο χειρίστης περίπτωσης για τον Ford - Fulkerson αλγόριθμο

Δεδομένου ότι ο αλγόριθμος Ford – Fulkerson επιλέγει τις διαδρομές τυχαία, έστω η διαδρομή $[1, 2, 4]$. Στην πρώτη επανάληψη, η τιμή της ροής θα είναι $f = 1$, με αποτέλεσμα οι ακμές $[(1,2) (2,3) (3,4)]$ να λάβουν τιμές $[999, 0, 999]$ αντίστοιχα. Στη δεύτερη επανάληψη της ίδιας διαδρομής, η ακμή $(2,3)$ θα θεωρηθεί κορεσμένη, οπότε θα εξεταστεί η περίπτωση της ανακατανομής της ροής της, αφού $X_{32} = 1 > 0$, οπότε οι ακμές $[(1,2) (2,3) (3,4)]$ να λάβουν τιμές $[998, 1, 998]$. Συνεχίζοντας με τον ίδιο ακριβώς τρόπο και αυξομειώνοντας τη ροή της ακμής $(2,3)$ προφανές γίνεται ότι ο αλγόριθμος θα λήξει μετά από $1000+1000 = 2000$ επαναλήψεις, που είναι η μέγιστη ροή του δικτύου.

Όπως παρατηρείται, η έκταση του υπολογισμού εξαρτάται από το μέγιστο των χωρητικότητων της πηγής και του προορισμού, με αποτέλεσμα η πολυπλοκότητα του αλγορίθμου να είναι ανάλογη της ποσότητας $\max(f)$. Παράλληλα, δεδομένου ότι η αρχικοποίηση του αλγορίθμου (ανάθεση σε κάθε ακμή μηδενική τιμή ροής) και η βασική επανάληψη γίνεται για κάθε ακμή στο γράφο, η πολυπλοκότητα ισούται με $O(E \max | f |)$ χαρακτηρίζοντας τον προκείμενο αλγόριθμο ως μη αποδοτικό.

Εστιάζοντας στον παραπάνω περιορισμό του αλγορίθμου Ford – Fulkerson, οι Edmonds και Karp πρότειναν μία μικρή διόρθωση στον τελευταίο[25]. Ο αλγόριθμός τους είναι πανομοιότυπος με αυτόν των Ford και Fulkerson με μόνη διαφορά ότι προσδιορίζουν την σειρά αναζήτησης κατά τον υπολογισμό του αυξητικού μονοπατιού. Το μονοπάτι που θα υπολογιστεί τελικά είναι το συντομότερο μονοπάτι με διαθέσιμη χωρητικότητα. Ένας τρόπος για να εκτελεστεί το τελευταίο είναι να πραγματοποιηθεί μία breadth – first αναζήτηση (π.χ. με τον αλγόριθμο του Dijkstra) θέτοντας τις τιμές κάθε ακμής ίση με τη μονάδα.

Ο χρόνος εκτέλεσης του αλγορίθμου είναι $O(|V| |E|^2)$, μιας και κάθε αυξητικό μονοπάτι μπορεί να βρεθεί σε χρόνο $O(|E|)$ ενώ κάθε φορά που κάποια ακμή χαρακτηρίζεται ως κορεσμένη, η απόσταση μεταξύ πηγής και προορισμού της επόμενης διαδρομής θα είναι τουλάχιστον κατά μία μονάδα μεγαλύτερη χωρίς να ξεπερνάει τις $|V|$ μονάδες.

Ένα βασικό μειονέκτημα της προσέγγισης αυτής, πάνω στο οποίο στηρίχθηκε ο Dinic για να προτείνει τον αλγόριθμο που θα αναλυθεί παρακάτω, έγκειται στον χωρίς – μνήμη υπολογισμό των μονοπατιών. Πιο συγκεκριμένα, κάθε φορά που μία διαδρομή υπερφορτώνεται, υπολογίζεται εκ νέου το επόμενο βέλτιστο μονοπάτι από την πηγή στον προορισμό χωρίς να «θυμάται» προηγούμενους υπολογισμούς και ενημερώσεις. Σε πολύπλοκα δίκτυα κάτι τέτοιο σημαίνει, εκτός από περίσσεια πολυπλοκότητα, αύξηση στον απαιτούμενο χρόνο υπολογισμού της τελικής ροής, καθώς όσες περισσότερες οι ακμές του δικτύου τόσο περισσότερο θα αυξάνεται ο συντελεστής $|E|^2$ κάνοντας τον αλγόριθμο μη αποδοτικό.

DINIC' BLOCKINGFLOW

Ο αλγόριθμος του Dinic συνδυάζει τους αλγορίθμους των Ford – Fulkerson και Edmons – Karp , ενώ εστιάζει στην πιο βέλτιστη αναζήτηση μονοπατιών[26]. Όπως παρουσιάστηκε παραπάνω, ο Edmonds – Karraλγόριθμος υπολογίζει κάθε φορά το ελάχιστο μονοπάτι από την πηγή προκειμένου να το υπερφορτώσει, με αποτέλεσμα να εξετάζεται σε κάθε επανάληψη μόνο ένα μονοπάτι χωρίς να υπολογίζεται η προηγούμενη ή η σύγχρονη γνώση για άλλα βέλτιστα μονοπάτια. Κατευθύνοντας την προσοχή του στον περιορισμό αυτόν, λοιπόν, ο αλγόριθμος του Dinic εισάγει νέες έννοιες οι οποίες αποτελούνται από το blocking flow και το γράφο επιπέδων (level graph), μειώνοντας την τελική πολυπλοκότητα στο $O(E \bar{V})$.

ΥΠΟΛΕΙΠΟΜΕΝΟΣ ΓΡΑΦΟΣ

Όπως παρατέθηκε παραπάνω, η υπολειπόμενη χωρητικότητα δίνεται από τον τύπο:

$$U_{ij} = \begin{cases} C_{ij} - X_{ij} & \text{εάν } i, j \text{ είναι forward ark} \\ X_{ji} & \text{εάν υπάρχει } X_{ij} > 0 \end{cases}$$

Με βάση το παραπάνω, ένας υπολειπόμενος γράφος είναι ο γράφος $G_f = (V, E_f, U_f)$ με $E_f = \{(i,j) \in E \mid U_{ij} > 0\}$. Πιο απλά, ο υπολειπόμενος γράφος G_f είναι ένα υποσύνολο του αρχικού γράφου, ο οποίος περιλαμβάνει εκείνες τις ακμές του αρχικού των οποίων η υπολειπόμενη χωρητικότητα είναι αυστηρά θετική.

BLOCKINGFLOW

Το Blocking Flow ορίζεται ως μία ροή από την πηγή προς τον προορισμό, έτσι ώστε ο γράφος $G' = (V, E_L', s, t)$ με $E_L' = \{(i, j) \mid X_{ij} < U_{ij}\}$ να μην περιλαμβάνει κανένα μονοπάτι. Πιο απλά, εάν G' είναι ένας υπογράφος του υπολειπόμενου γράφου με μη αρνητική υπολειπόμενη χωρητικότητα, τότε δεν υπάρχει μονοπάτι από την πηγή προς τον προορισμό. Παράλληλα, ο G' είναι ένας layered γράφος ο οποίος προκύπτει από τον υπολειπόμενο γράφο

ταξινομημένο σε επίπεδα έτσι ώστε το επίπεδο L ενός κόμβου να ισούται με την απόστασή του από τον προορισμό.

ΑΛΓΟΡΙΘΜΟΣ

Βήμα 1.

Κατασκευή του υπολειπόμενου γράφου G_f . (Στην πρώτη εκτέλεση, $X_{ij} = 0$, $\forall i, j \in N$.) Εάν ο γράφος δεν μπορεί να κατασκευαστεί, έχει υπολογιστεί η μέγιστη ροή.

Βήμα 2.

Κατασκευή του G_L από τον G_f . Ο G_L είναι ένας layered γράφος που υπολογίζεται με breadth – first αλγόριθμο ως εξής: Τοποθετούνται στο γράφο οι γειτονικοί κόμβοι επιπέδου $L+1$ του κόμβου με επίπεδο L , έως ότου συναντηθεί ο κόμβος προορισμού. Όταν συμπληρωθεί το επίπεδο κόμβου προορισμού, η διαδικασία σταματά αγνοώντας, έτσι, κόμβους οι οποίοι ευρίσκονται μακρύτερα του προορισμού.

Βήμα 3.

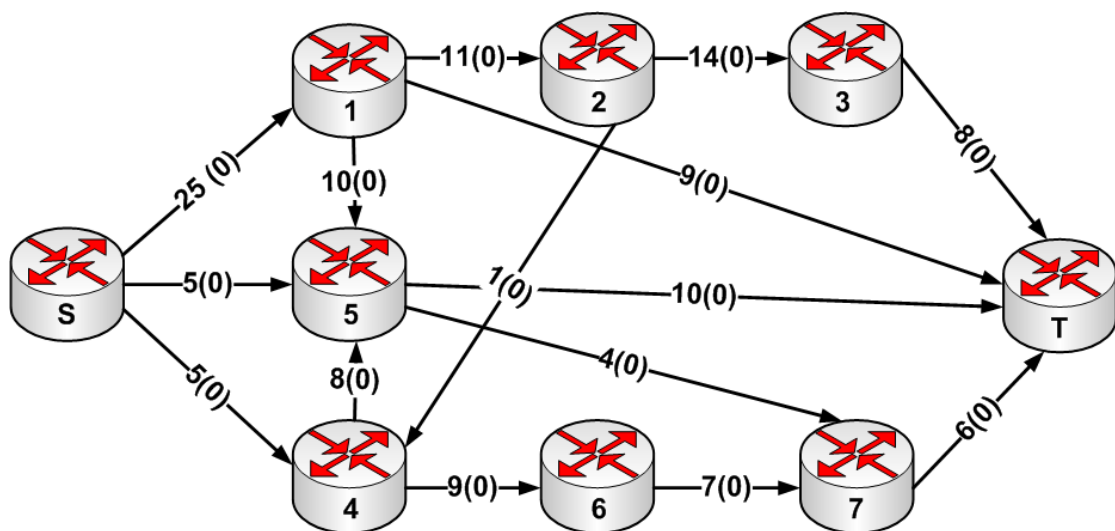
Για κάθε διαδρομή προς τα πίσω, από τον προορισμό προς την πηγή, υπολογίζεται η μέγιστη ροή. Κάθε διαδρομή του G_L που έχει προσθέσει μία ροή $X_{ij} > 0$, διαγράφεται προς τα εμπρός (από την πηγή προς τον προορισμό) από τον G_L και εξετάζονται οι υπόλοιπες backward διαδρομές. Όταν τέτοια διαδρομή δεν υπάρχει (άρα βρέθηκε το blocking flow), ο αλγόριθμος επιστρέφει στο βήμα 1, ενημερώνοντας το γράφο με τις νέες ροές και συνεχίζει με τις επόμενες επαναλήψεις.

Για να γίνουν πιο κατανοητές οι νέες έννοιες που εισάγει ο αλγόριθμος του Dinic, καθώς και ο τρόπος υλοποίησής του, εξετάζεται ένα παράδειγμα. Πρέπει να τονιστεί, στο σημείο αυτό, ότι έχοντας εξετάσει ενδελεχώς τον αλγόριθμο Ford – Fulkerson, δε θα δούμε όλες τις επαναλήψεις του Dinic, παρά μόνο το βασικό σκεπτικό του. Ο αναγνώστης, έχοντας τις πληροφορίες για τον αλγόριθμο του Dinic καθώς και το αλγοριθμικό υπόβαθρο για τον

Ford– Fulkerson, εύκολα θα μπορεί να εξάγει και τα υπόλοιπα βήματα του πρώτου.

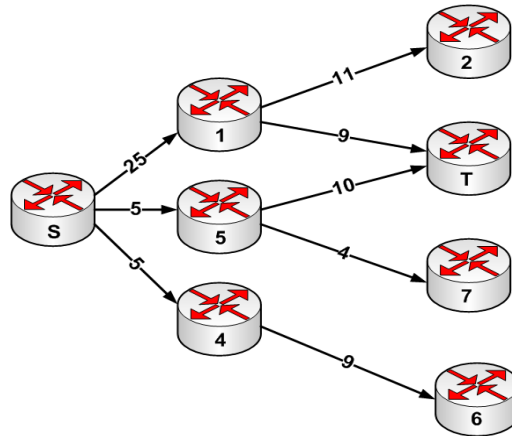
ΠΑΡΑΔΕΙΓΜΑ

Έστω ένας γράφος G και G_f να είναι ο υπολειπόμενος του, όπως φαίνεται στην [Εικόνα 10](#). Η πρώτη τιμή κάθε ακμής παρουσιάζει την ονομαστική χωρητικότητά της, ενώ οι παρενθέσεις δείχνουν την ποσότητα της ροής της. Στην προκειμένη περίπτωση, ο υπολειπόμενος γράφος έχει σε κάθε ακμή ροή ίση με 0 (αρχικοποίηση).



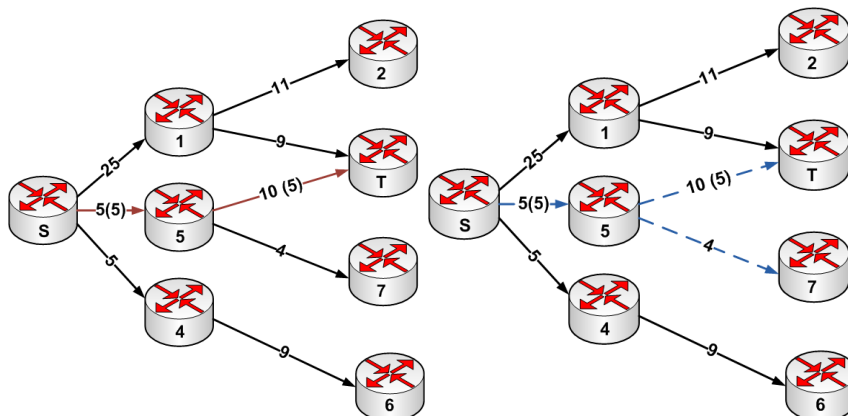
Εικόνα 10. Υπολειπόμενος γράφος μηδενικής ροής

Από τον παραπάνω υπολειπόμενο γράφο, θα κατασκευαστεί ο γράφος επιπέδων. Όπως φαίνεται και από την [Εικόνα 11](#), ο G_L είναι ένα δένδρο, κατασκευασμένο από τον υπολειπόμενο γράφο G_f μέσω Breadth – First αναζήτησης, έτσι ώστε στο τελευταίο επίπεδό του (εδώ, στο τρίτο) να ευρίσκεται ο κόμβος προορισμού και οι υπόλοιποι κόμβοι του ίδιου επιπέδου. Κόμβοι οι οποίοι έχουν επίπεδο 4 και άνω δε λαμβάνονται υπ’ όψιν.



Εικόνα 11.Γράφος Επιπέδων Μηδενικής Ροής

Από τον γράφο επιπέδων, λοιπών, και με βήματα προς τα πίσω, υπολογίζονται τα εφικτά μονοπάτια και οι αντίστοιχες τιμές των ροών τους. Επιλέγεται για παράδειγμα το μονοπάτι [S, 5, T]. Η ελάχιστη χωρητικότητα των ακμών του είναι αυτή της (S, 5)η οποία ισούται με 5 μονάδες. Άρα, το μονοπάτι αυτό μπορεί να αυξήσει τη ροή κατά 5. Έτσι, στο βήμα αυτό ενημερώνεται προς τα πίσω το δένδρο, όπως φαίνεται στην [Εικόνα 12](#), ενώ διαγράφονται προς τα μπροστά οι ζεύξεις που εκκινούν από την (S, 5), όπως δείχνει η [Εικόνα 13](#). Η (νοητή) διαγραφή των ζεύξεων, λαμβάνει χώρα λόγω του ότι η ακμή (S, 5)έχει πλέον κορεστεί και τα μονοπάτια που έχουν επισημανθεί με μπλε χρώμα δεν μπορούν να υποστηρίξουν πλέον παραπάνω ροή.

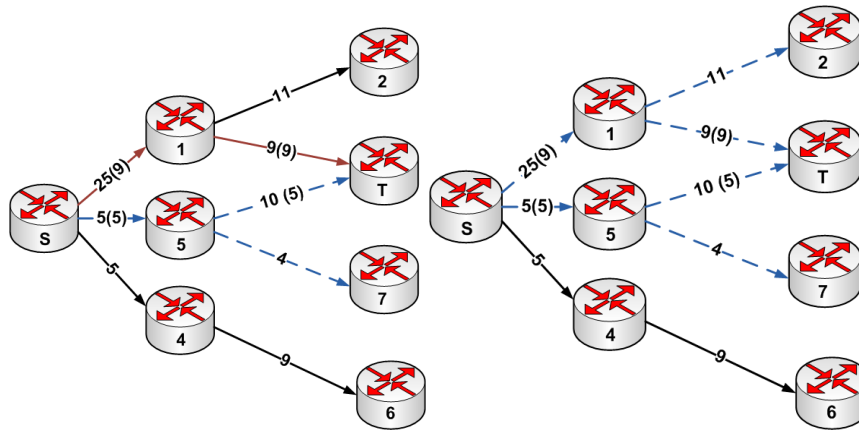


Εικόνα 12.Προσθήκη ροής προς τα πίσω

Εικόνα 13. Αφαίρεση διαδρομών προς τα εμπρός

Συνεχίζοντας με τον ίδιο υπογράφο, παρατηρείται ότι υπάρχει και άλλη μία διαδρομή η οποία εκκινεί από τον S και καταλήγει στον T: (S, 1, T). Η

διαδρομή αυτή μπορεί να υποστηρίξει ροή ίση με 9 μονάδες (η ελάχιστη χωρητικότητα των ακμών της). Με τον ίδιο τρόπο, λοιπόν, ενημερώνονται προς τα πίσω, ξεκινώντας από τον Τ οι ακμές με τη νέα ροή, ενώ διαγράφονται προς τα μπροστά όλα τα μονοπάτια που εκκινούν από την (S, 1).



Εικόνα 14. Προσθήκη νέας ροής προς τα πίσω

Εικόνα 15. Αφαίρεση νέων διαδρομών προς τα εμπρός

Όπως μπορεί να παρατηρήσει κανείς από την [Εικόνα 15](#), ο γράφος δεν περιλαμβάνει άλλα μονοπάτια τα οποία να οδηγούν στον προορισμό, καθώς οι ζεύξεις με μπλε χρώμα θεωρούνται αποκομμένες πλέον από αυτόν. Αυτό σημαίνει ότι έχει εντοπιστεί η λεγόμενη «blocking flow» που ορίστηκε πιο πάνω και ο αλγόριθμος μπορεί να προχωρήσει στην ενημέρωση του αρχικού γράφου.

Έτσι, στο επόμενο βήμα, ο γράφος της [Εικόνας 11](#) θα παρουσιάζει πλέον τον ενημερωμένο υπολειπόμενο γράφο με τις ροές μεγέθους 5 και 9 που υπολογίστηκαν και ο αλγόριθμος θα συνεχίσει με την ίδια φιλοσοφία, κατασκευάζοντας το νέο γράφο επιπέδων και υπολογίζοντας ξανά τις blocking ροές. Όταν ο υπολειπόμενος γράφος δεν θα μπορέσει να κατασκευαστεί, επειδή δε θα υπάρχει μονοπάτι από την πηγή ως τον προορισμό, ο αλγόριθμος σταματά και η τιμή της μέγιστης ροής που επιστρέφεται είναι ίση με το άθροισμα των επιμέρους ροών που ανιχνεύονται σε κάθε βήμα του.

Από τον παραπάνω αλγόριθμο μπορεί να παρατηρηθεί το εξής: Οι αρχικοποιήσεις και οι ενημερώσεις των γράφων γίνονται συγκεντρωτικά και

όχι ανά διαδρομή/ επανάληψη. Παράλληλα, κατά τον υπολογισμό μίας ροής μπορούν να ληφθούν υπ' όψιν άνω της μίας διαδρομής ταυτόχρονα, εάν ο γράφος GL περιλαμβάνει πάνω από μία διαδρομές που ξεκινούν από την πηγή και καταλήγουν στον προορισμό. Έτσι, τα βήματα 2 και 3 μπορούν να επιστρέψουν πολλές ροές ταυτόχρονα, ελαττώνοντας τον απαιτούμενο χρόνο υπολογισμών και αναζητήσεων και καθιστώντας τον αλγόριθμο αυτό πιο αποδοτικό από τον αντίστοιχο των Edmonds και Karp.

ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΚΑΙ ΚΙΝΗΣΗΣ

ΕΙΣΑΓΩΓΗ

Το παρόν κεφάλαιο πραγματεύεται την ανάλυση, το σχεδιασμό και τη δημιουργία ενός Αλγορίθμου Διαχείρισης Πόρων και Κίνησης σε τοπολογίες MPLS δικτύων. Ο προκείμενος αλγόριθμος, ονομάστηκε κατά τη διάρκεια της εκπόνησης της εργασίας ως «TiReD Algorithm», όνομα το οποίο έχει διπλή σημασία: Πρώτον, προέρχεται από τη φράση «Traffic – Resources Distribution» χαρακτηρίζοντας με τον τρόπο αυτόν τη βασική του λειτουργία, δηλαδή την offlineκατανομή της κίνησης και των πόρων. Δεύτερον, εκφράζει την κουραστική και περιπετειώδη διαδικασία του σχεδιασμού του, η οποία αποτελέστηκε από αλληπάλληλες προσπάθειες, διαγραφές, επανεκκινήσεις, διορθώσεις με την κάθε προσπάθεια να σχεδιάζει έναν εντελώς διαφορετικό αλγόριθμο μέχρι τελικά να κατασκευαστεί και να διατηρηθεί ως «καλύτερος» ο αλγόριθμος που θα εξεταστεί παρακάτω.

Ορμώμενος από όλα αυτά που εξετάστηκαν στα δύο προηγούμενα κεφάλαια, ο TiReD έρχεται να προτείνει λύση σε κάποια από τα πιο βασικά μειονεκτήματα των υπολοίπων ερευνών, εμπνευσμένος, παράλληλα, από τη βέλτιστη (κατά την κρίση του συγγραφέα) προσέγγιση Max – Flow αλγορίθμου από τους παραπάνω : του Dinic. Με διαφορετικά λόγια, ο προκείμενος αλγόριθμος θα προσπαθήσει να βρει λύσεις τέτοιες ώστε:

- Να ικανοποιούνται όλοι οι, τυχόν, περιορισμοί του MPLS – TE όσον αφορά στις ροές : η ανάγκη για εφεδρικά μονοπάτια, ο κλιμακωτός προσδιορισμός των ζεύξεων και των hops, καθώς και ο περιορισμός στο πλήθος παράλληλων ροών που μπορούν να απορρέουν από την ίδια αρχική ροή.
- Να κατανέμεται το φορτίο στο δίκτυο με τρόπο τέτοιο ώστε να αποφεύγονται κρίσιμες ζεύξεις, να ελαχιστοποιείται η συμφόρηση και να μεγιστοποιείται η ροή στο δίκτυο
- Να υπολογίζονται μονοπάτια τα οποία να πληρούν τις παραπάνω προϋποθέσεις και ταυτόχρονα να είναι όσο το δυνατόν μικρότερα σε μήκος.

Όπως ήδη αναφέρθηκε παραπάνω, σε κάποια σημεία του, ο TiReD έχει επηρεαστεί από τον αλγόριθμο του Dinic. Πιο συγκεκριμένα, έχει «δανειστεί» δύο από τις έννοιές του, τις οποίες έχει προσαρμόσει και χρησιμοποιήσει διαφορετικά διατηρώντας, ωστόσο, το αρχικό σκεπτικό: την έννοια του Υπολειπόμενου και του Layered γράφου. Ενώ κατά τον Dinic ο Υπολειπόμενος Γράφος είναι αυτός ο οποίος περιλαμβάνει μόνο εκείνες τις ακμές που έχουν θετική υπολειπόμενη χωρητικότητα, ο TiReD δημιουργεί δύο νέες κατηγορίες υπολειπομένων γράφων:

- Γράφος που υπόκειται στις MPLS – TE απαιτήσεις και αποτελείται από ακμές με υπολειπόμενη χωρητικότητα μικρότερη ή ίση της απαιτούμενης από την εκάστοτε ροή
- Γράφος που υπόκειται στις MPLS – TE απαιτήσεις και αποτελείται από ακμές με αυστηρά θετική υπολειπόμενη χωρητικότητα.

Αξίζει, βέβαια, να τονιστεί ότι ο Dinic Algorithm έχει σημασία μόνο σε δίκτυα μεταφοράς, τα οποία έχουν μόλις ένα ζεύγος (πραγματικό ή ιδεατό) πηγής και προορισμού, ασχέτως του πλήθους των πραγματικών Ingress και Egress LERs. Ο TiReD, δε, ο οποίος μπορεί να εφαρμοστεί σε δίκτυα πολλών πηγών και προορισμών, αφαιρεί την ανάγκη για την κατασκευή δικτύου με ιδεατούς κόμβους. Ο λόγος για την απόφαση αυτή προέρχεται από το γεγονός ότι :

- Κάθε κύκλος του αλγορίθμου σχεδιάζει τον υπολειπόμενο γράφο αποκλειστικά και μόνο για τη ροή την οποία εξετάζει τη στιγμή εκείνη. Η ροή μπορεί να έχει μόνο μία πηγή και έναν προορισμό και συνεπώς, ο υπολειπόμενος γράφος θα έχει και αυτός την πηγή και τον προορισμό της ροής. Οι υπόλοιποι LERs οι οποίοι δύναται να υπάρχουν στην τοπολογία θεωρούνται ως απλοί δρομολογητές.
- Εάν προστεθούν οι ιδεατοί κόμβοι με άπειρη χωρητικότητα ακμών, μεταμορφώνοντας το δίκτυο σε Δίκτυο Μεταφοράς, μπορεί να γίνει αδύνατη η κατεύθυνση της ροής έτσι ώστε να εκκινεί στην πραγματική της πηγή και να τερματίζει στον πραγματικό της προορισμό. Για παράδειγμα, δεδομένου ότι στο Δίκτυο Μεταφοράς με Super Source, η τελευταία θα ενώνεται με αρκετά πραγματικά Sources, μπορεί να γίνει σημαντικά πιο δύσκολος ο εξαναγκασμός της ροής να ακολουθήσει

μονοπάτι το οποίο να ξεκινάει από το Super Source και να διασχίζει την πραγματική του πηγή, καθ' ότι μπορεί ο αλγόριθμος να θεωρήσει καλύτερη ή περισσότερο εφικτή μία διαδρομή η οποία διασχίζει κάποιο άλλο LER, άσχετο με τον ζητούμενο από τη ροή. Προφανώς, τα ίδια ισχύουν και στην άλλη πλευρά του δικτύου όπου ευρίσκεται το Super Sink.

- Ο σχεδιασμός και η χρήση δύο παραπάνω κόμβων στο δίκτυο και αρκετών ακμών άπειρης χωρητικότητας, εισάγει την ανάγκη για περισσότερους ελέγχους στον αλγόριθμο (για παράδειγμα χαρακτηρισμός ακμής ως πραγματική ή ιδεατή προκειμένου να ληφθούν συγκεκριμένες αποφάσεις) και αυξάνει σε κάποιο βαθμό (εάν και μικρό) την απαίτηση σε μνήμη και χρόνο για τον συνολικό υπολογισμό της βέλτιστης λύσης.

Έτσι, ο TiReD μπορεί να εφαρμοστεί σε δίκτυα που δεν υπάγονται απαραίτητως στην κατηγορία των Δικτύων Μεταφοράς, ενώ παράλληλα εμπνέεται από τον υπολειπόμενο γράφο του Dinic, προσαρμόζοντάς τον με διαφορετικό τρόπο, όπως αναφέρθηκε και προηγουμένως, εισάγοντας έτσι δύο νέες κατηγορίες γράφων οι οποίες χρησιμοποιούνται για την εύρεση εφικτών (από άποψη απαιτήσεων και χωρητικότητας) μονοπατιών.

Επιπρόσθετα, δανείζεται, όπως αναφέρθηκε και παραπάνω, και την έννοια του Layered γράφου ο οποίος χρησιμοποιείται αφ' ενός για να εξεταστεί το εάν υπάρχει εφικτό (από άποψη χωρητικότητας) μονοπάτι στο δίκτυο και αφ' ετέρου προκειμένου να εισαχθεί περιορισμός στο μήκος του. Με άλλα λόγια, δεδομένου ότι στον Layered γράφο δεν εισάγονται κόμβοι οι οποίοι βρίσκονται «μακρύτερα» από τον κόμβο προορισμού, αμέσως εισάγεται η εγγύηση ότι τα μονοπάτια που θα εξαχθούν από αυτόν θα είναι και τα μικρότερα εφικτά.

Μία άλλη σημαντική προσθήκη η οποία λαμβάνει χώρα στον προκείμενο αλγόριθμο, είναι και ένα είδος «μνήμης» των προηγούμενων αποτυχημένων του προσπαθειών. Με πιο απλά λόγια, δίδεται στον αλγόριθμο η δυνατότητα να «θυμάται» για ποια ζεύγη πηγής και προορισμού και για ποια ποσότητα Bandwidth είχε δυσκολευτεί, σε προηγούμενους κύκλους, να βρει τη βέλτιστη

λύση με την πρώτη προσπάθεια, προκειμένου να αποφύγει αναζητήσεις και ελέγχους για ροές οι οποίες είναι βέβαιο ότι δε θα εξυπηρετηθούν με την πρώτη, και να προβεί σε αποφάσεις βασιζόμενος πλέον στην προηγούμενη εμπειρία του.

Το κεφάλαιο αυτό, λοιπόν, παρουσιάζει σε πολύ μεγαλύτερη λεπτομέρεια και βάθος όλα αυτά που αναφέρθηκαν πιο πάνω. Στην πρώτη ενότητα θα δοθεί λεπτομερής ανάλυση των δεδομένων εισόδου και εξόδου του αλγορίθμου, μέσα από τα οποία θα γίνουν γνωστές αρκετές από τις πιο σημαντικές παραδοχές που ελήφθησαν κατά το σχεδιασμό του αλγορίθμου. Στην δεύτερη ενότητα θα παρουσιαστούν ενδελεχώς οι οντότητες που λαμβάνουν χώρα στον αλγόριθμο καθώς και οι νέες έννοιες που αυτός εισάγει. Τέλος, στην τρίτη θα περιγραφούν τα ακριβή βήματα του εν προκειμένω, τα οποία μαζί με τα σχετικά Activity Diagrams θα παρουσιάσουν ολόκληρο τον αλγόριθμο, σε όλες τις πιθανές του καταστάσεις και περιπτώσεις.

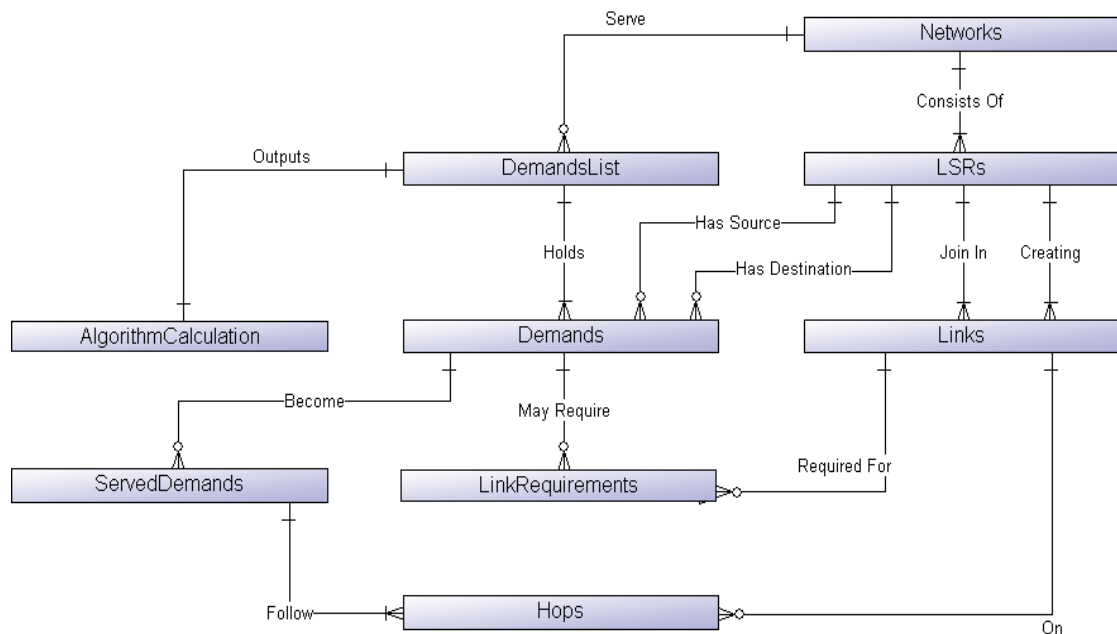
ΔΕΔΟΜΕΝΑ ΕΙΣΟΔΟΥ - ΕΞΟΔΟΥ

Για την εκτέλεση του προταθέντος αλγορίθμου θεωρούνται γνωστά η τοπολογία του δικτύου καθώς και όλες οι ροές των δεδομένων που αιτούν τη διάσχισή του. Για λόγους γρήγορης καταχώρησης και ανάκτησης, εύκολης επεξεργασίας, διατήρησης της λογικής των σχέσεων καθώς και προστασίας της ακεραιότητας των δεδομένων, θεωρήθηκε ως καλύτερη λύση η διατήρησή τους εντός μίας Βάσεως Δεδομένων.

Η Βάση Δεδομένων σχεδιάστηκε και υλοποιήθηκε με τρόπο τέτοιο ώστε να απεικονίζει και να διατηρεί, ουσιαστικά, τη λογική και πολλά από τα πιο βασικά στοιχεία του εν προκειμένω αλγορίθμου. Για το λόγο αυτό, θεωρείται ουσιώδης μια εις βάθος ανάλυση των σχέσεων καθώς και των δεδομένων αυτής, προκειμένου όχι μόνο να γίνει κατανοητό το περιβάλλον της εφαρμογής του αλγορίθμου, αλλά και οι λεπτομέρειες του σχεδιασμού και των παραδοχών που έγιναν κατά το σχεδιασμό του.

Στις επόμενες υποενότητες, λοιπόν, θα εξεταστεί το μοντέλο της Βάσης Δεδομένων και τα πεδία των πινάκων που τη συντελούν. Στην πρώτη υποενότητα θα αρκεστούμε στο μοντέλο Οντοτήτων – Συσχετίσεων, ώστε να γίνει κατανοητό το γενικό πλαίσιο σχεδιασμού και εφαρμογής του αλγορίθμου. Στην επόμενη υποενότητα, δε, εξετάζοντας τα πεδία των πινάκων, θα αναφερθούμε στις έννοιες και τις παραδοχές που ελήφθησαν και ενσωματώθηκαν στον τελικό αλγόριθμο.

ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ – ΣΥΣΧΕΤΙΣΕΩΝ



Εικόνα 16. Μοντέλο Οντοτήτων - Συσχετίσεων

ΤΟΠΟΛΟΓΙΑ

Για την εύκολη διατήρηση και καταχώρηση άνω της μίας τοπολογίας, καθώς και για την εξάλειψη της απαίτησης συνεχούς καθαρισμού των δεδομένων κατά την πραγματοποίηση συνεχών εκτελέσεων του αλγορίθμου για διαφορετικά πλαίσια αναφοράς, ελήφθη η παραδοχή ότι ένα οποιοδήποτε περιβάλλον έχει ένα ή περισσότερα δίκτυα (Networks). Να σημειωθεί, βέβαια,

πώς για κάθε εκτέλεση του αλγορίθμου μόνο ένα δίκτυο εκ των αποθηκευμένων λαμβάνει χώρα.

Κάθε δίκτυο αποτελείται από δύο ή περισσότερους δρομολογητές (LSRs) κάθε ένας από τους οποίους μπορεί να ανήκει φυσικά σε ένα δίκτυο και μόνο. Οι δρομολογητές επικοινωνούν μεταξύ τους μέσω φυσικών ζεύξεων (Links). Κάθε ζεύξη, εντός των δεδομένων εισόδου θεωρείται μονόδρομη. Ωστόσο, παρότι για την αμφίδρομη επικοινωνία δύο δρομολογητών, προφανές είναι ότι θα απαιτηθούν δύο φυσικά links, η εισαγωγή αυτών εντός της Βάσεως Δεδομένων θεωρείται εσφαλμένη, καθ' ότι θα προτιμηθεί η διατήρηση των ίδιων των δεδομένων μονόδρομα, υπονοώντας την αμφιδρόμηση εντός του ίδιου του αλγορίθμου (όπως θα δούμε παρακάτω).

ΡΟΕΣ

Η εκμετάλλευση κάθε δικτύου αιτείται από συνδρομητές διαμέσω συνδέσεων (ροών). Οι ροές που προσπελαίνουν κάθε συγκεκριμένο δίκτυο ομαδοποιούνται (Demands List) με αποτέλεσμα κάθε ομάδα ροών να ανήκει σε ένα και μόνο δίκτυο. Ακόμα και στην περίπτωση που μία ροή χρήζει διάσχισης δύο και παραπάνω γειτονικών δικτύων, θεωρείται ότι ανήκει σε δύο και παραπάνω αντίστοιχες ομάδες ροών, με την κάθε μία να καταλαμβάνει χώρο στο δίκτυο στο οποίο έχει καταχωρηθεί και μόνο.

Όπως δηλώνει και ο όρος, κάθε ομάδα ροών (Demands) αποτελείται από ανεξάρτητες και αυτόνομες μεταξύ τους ροές. Κάθε μία από αυτές εκκινεί και τερματίζει σε δρομολογητές του δικτύου στο οποίο ανήκει η ομάδα τους. Εκτός από απαιτήσεις και περιορισμούς που θα αναλυθούν εις βάθος παρακάτω, κάθε ροή, όπως σημειώθηκε και στις απαιτήσεις του MPLS – TE, δύναται να ορίσει περιορισμούς ζεύξεων (Link Requirements). Με άλλα λόγια, μπορεί να απαιτήσει τη διάσχιση ή την αποφυγή ενός ή μίας σειράς links εντός του δικτύου στο οποίο ανήκει η ομάδα της. Προφανές γίνεται πως όπως κάθε ροή μπορεί να ορίσει πλειάδα ζεύξεων προς χρήση ή αποφυγή, έτσι και κάθε μία ζεύξη μπορεί να αποτελέσει τμήμα περιορισμών καμίας ή και περισσοτέρων ροών.

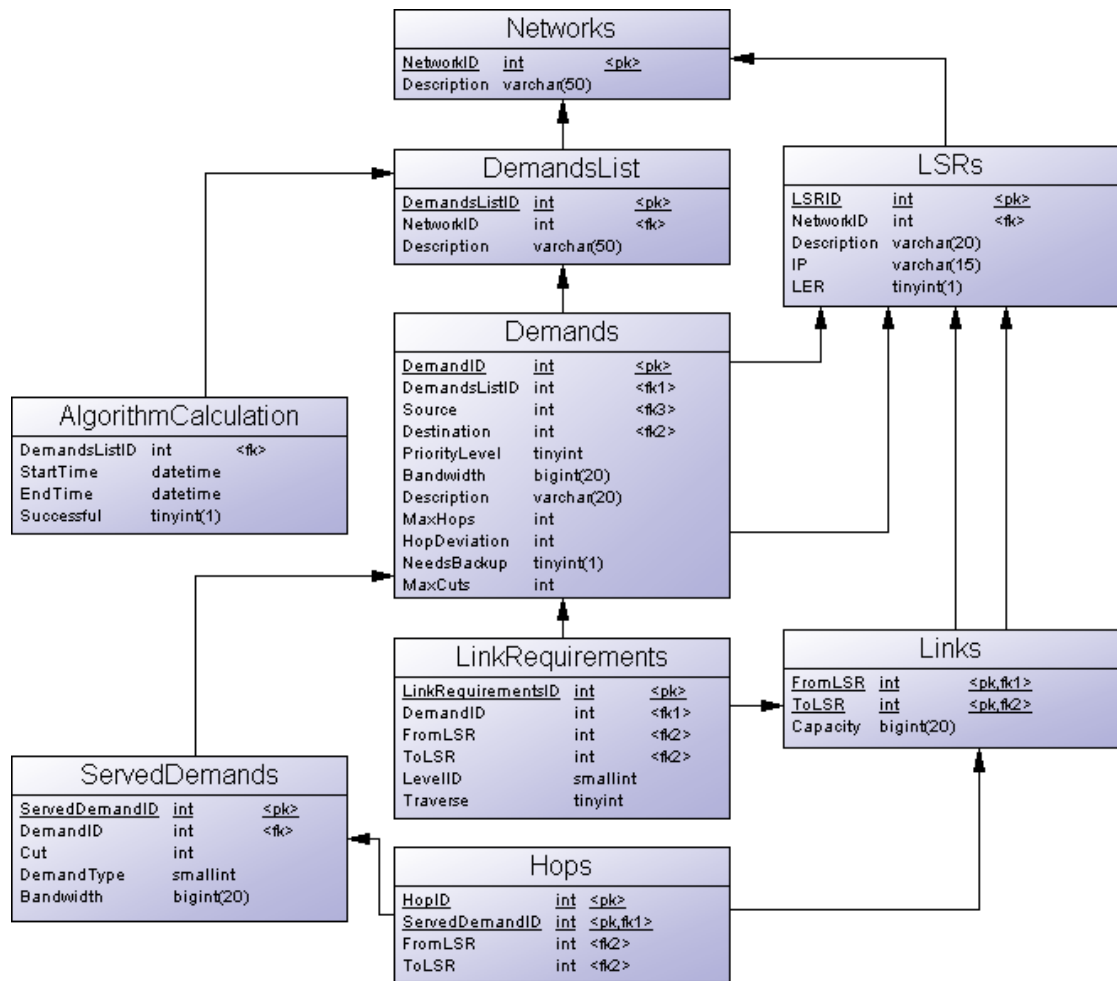
ΕΞΥΠΗΡΕΤΗΘΕΙΣΕΣ ΡΟΕΣ

Κάθε ροή μίας ομάδας μπορεί να εξυπηρετηθεί είτε ολόκληρη ως είναι είτε σε τμήματα (Served Demands), με το κάθε τμήμα να ακολουθεί το δικό του μονοπάτι το οποίο υπακούει σε όλες τις απαιτήσεις που ορίζει η πρώτη. Συνεπώς, μία ή περισσότερες Εξυπηρετηθείσες Ροές αποτελούν το σύνολο μίας και μόνο αρχικής ροής εντός της συγκεκριμένης ομάδας και τοπολογίας δικτύου. Κάθε μία από της εξυπηρετηθείσες ροές καταλαμβάνει ένα δικό της μονοπάτι (Hops) το οποίο διασχίζει μία ή περισσότερες ζεύξεις του δικτύου. Προφανές γίνεται ότι και κάθε ζεύξη μπορεί να εξυπηρετεί κανένα, ένα ή και περισσότερα τμήματα διαφορετικών ροών.

ΣΧΕΣΙΑΚΟ ΜΟΝΤΕΛΟ

Στην υποενότητα αυτή θα αναλυθούν οι λεπτομέρειες του σχεδιασμού του αλγορίθμου, με τη βοήθεια του Σχεσιακού Μοντέλου της Βάσης Δεδομένων του τελευταίου, προκειμένου να γίνουν γνωστές και κατανοητές κάποιες από τις πιο βασικές παραδοχές και έννοιες που θα χρησιμοποιηθούν παρακάτω. Μετά από αυτή την υποενότητα, λοιπόν, ο αναγνώστης θα κατέχει τη γνώση τόσο των παραδοχών όσο και των λέξεων – κλειδιών, προκειμένου να κατανοήσει το σχεδιασμό του αλγορίθμου ο οποίος θα ακολουθήσει στην επόμενη ενότητα.

Έχοντας ήδη αναλύσει το γενικό πλάνο σχεδιασμού και δεδομένου ότι στην παρούσα υποενότητα στόχος είναι η επεξήγηση των εννοιών – κλειδιών, θα προσπεραστούν τυχόν περιγραφικές αναλύσεις και θα επεξηγηθούν τα πεδία των πινάκων της Βάσεως Δεδομένων, δίδοντας έτσι και ένα σχετικό «λεξικό» των όρων που θα χρησιμοποιηθούν παρακάτω.



Εικόνα 17. Σχεσιακό Μοντέλο

NETWORKS

- NetworkID

Κλειδί αυξανόμενης αρίθμησης του πίνακα Networks, που χαρακτηρίζει μοναδικά τα δίκτυα που βρίσκονται εντός του περιβάλλοντος εφαρμογής.

- Description

Πεδίο περιγραφής για την ονομασία του κάθε δικτύου. Χρησιμοποιείται για την εύκολη αναζήτηση και αναγνώριση των δικτύων από το χρήστη του αλγορίθμου.

LSRS

- LSRID

Κλειδί αυξανόμενης αρίθμησης του πίνακα LSRs, που χαρακτηρίζει μοναδικά τους δρομολογητές που βρίσκονται σε κάθε δίκτυο.

- NetworkID

Εξωτερικό κλειδί του πίνακα Networks, το οποίο δηλώνει το δίκτυο στο οποίο ανήκει ο συγκεκριμένος δρομολογητής.

- Description

Πεδίο περιγραφής για την ονομασία κάθε δρομολογητή. Χρησιμοποιείται για την εύκολη αναγνώριση του κάθε δρομολογητή από το χρήστη του αλγορίθμου, στην περίπτωση που ο τελευταίος επιθυμεί να αποφύγει τη χρήση διευθύνσεων (για παράδειγμα, δρομολογητής «Α»)

- IP

Η IP διεύθυνση του δρομολογητή. Χρησιμοποιείται απλά για την αναγνώρισή του, όπως και το πεδίο Description.

- LER

Boolean πεδίο το οποίο ορίζει τον τύπο του κάθε δρομολογητή. Εάν το πεδίο είναι true, ο δρομολογητής είναι τύπου LER. Στην αντίθετη περίπτωση, είναι ένας απλός LSR. Να σημειωθεί ότι για να είναι έγκυρο το εκάστοτε δίκτυο, πρέπει να έχει τουλάχιστον 2 δρομολογητές τύπου LER.

$$LER = \begin{cases} 1, & \text{LSR is a LER} \\ 0, & \text{Plain LSR} \end{cases}$$

LINKS

- FromLSR

Εξωτερικό κλειδί του πίνακα LSRs το οποίο δηλώνει το δρομολογητή από τον οποίο εκκινεί η ζεύξη

- ToLSR

Εξωτερικό κλειδί του πίνακα LSRs το οποίο δηλώνει το δρομολογητή στον οποίο καταλήγει η ζεύξη.

Να σημειωθεί ότι το ζεύγος FromLSR – ToLSR είναι μοναδικό εντός του πίνακα, απαγορεύοντας την διπλή εισαγωγή της ίδιας ζεύξεως δύο φορές. Παράλληλα, απαγορεύεται και η εισαγωγή εγγραφής τύπου ToLSR – FromLSR, καθώς είναι επιθυμητό η αμφίδρομη επικοινωνία μεταξύ των LSRs να εννοείται από τον ίδιο τον αλγόριθμο και όχι από τα δεδομένα εισόδου. Τέλος, ζεύξεις του τύπου FromLSR – FromLSR και ToLSR – ToLSR θεωρούνται μη έγκυρες.

- Capacity

Μεγάλος θετικός ακέραιος αριθμός που διατηρεί την πληροφορία της χωρητικότητας της κάθε ζεύξης. Η μονάδα μέτρησης θεωρείται αδιάφορη (π.χ. Kbps, Mbps, κ.ο.κ.) καθώς δε συμμετέχει στους υπολογισμούς. Ο μόνος περιορισμός είναι να υπονοείται παντού η ίδια μονάδα (για παράδειγμα, εάν εκχωρείται τιμή χωρητικότητας εκφρασμένη σε Kbps, τότε και το Bandwidth που θα αιτήσεται παρακάτω η ροή θα πρέπει και αυτό να εκφράζεται στην ίδια μονάδα).

DEMANDSLIST

- DemandsListID

Κλειδί αυξανόμενης αρίθμησης του πίνακα DemandsList. Μοναδικοποιεί τις ομάδες εντός του πλαισίου εφαρμογής.

- NetworkID

Εξωτερικό κλειδί του πίνακα Networks που δηλώνει σε ποιο δίκτυο ανήκει η ομάδα ροών. Να σημειωθεί ότι το ζεύγος DemandsListID – NetworkID είναι μοναδικό, διατηρώντας τον κανόνα που τέθηκε ότι κάθε ομάδα μπορεί να ανήκει σε ένα δίκτυο και μόνο.

- Description

Περιγραφικό πεδίο για την ονοματοδοσία των ομάδων και την εύκολη αναγνώρισή τους.

DEMANDS

- DemandID

Κλειδί αυξανόμενης αρίθμησης του πίνακα Demands που χαρακτηρίζει μοναδικά κάθε ροή.

- DemandsListID

Εξωτερικό κλειδί του πίνακα DemandsList το οποίο δηλώνει την ομάδα στην οποία ανήκει η συγκεκριμένη ροή.

- Source

Εξωτερικό κλειδί του πίνακα LSRs το οποίο δείχνει στο δρομολογητή από το οποίο επιθυμεί να εκκινήσει η ροή. Να σημειωθεί ότι ο συγκεκριμένος LSR απαιτείται να είναι τύπου LER.

- Destination

Εξωτερικό κλειδί του πίνακα LSRs το οποίο δηλώνει το δρομολογητή στον οποίο επιθυμεί να καταλήξει η ροή. Όπως και με το Source, ο δρομολογητής του Destination πρέπει να είναι τύπου LER.

- PriorityLevel

Επίπεδο προτεραιότητας της ροής, όπως ορίζει το MPLS – TE. Δηλώνεται μέσω ενός θετικού ακεραίου στο κλειστό διάστημα [1, 7].

- Bandwidth

Αυστηρά θετικός ακέραιος ο οποίος δηλώνει το Bandwidth που επιθυμεί να χρησιμοποιήσει η ροή στο συγκεκριμένο δίκτυο. Όπως και στην περίπτωση του Capacity των ζεύξεων, η μονάδα μέτρησης δεν είναι τόσο σημαντική όσο η συνέπειά της εντός της εφαρμογής του αλγορίθμου.

- Description

Πεδίο περιγραφής για την ονοματοδοσία κάθε ροής, με σκοπό την εύκολη

αναγνώρισή της από τον χρήστη του αλγορίθμου.

- **MaxHops**

Ακέραιος αριθμός μέσω του οποίου η ροή προσδιορίζει, εάν το επιθυμεί, το μέγιστο αριθμό Hops που μπορεί να διανύσει εντός του δικτύου. Να σημειωθεί ότι εάν αφεθεί κενό (δε δηλωθεί τιμή), σαν μέγιστος αριθμός Hops για το οποιοδήποτε μονοπάτι υπονοείται το συνολικό πλήθος των δρομολογητών εντός του δικτύου.

- **HopsDeviation**

Θετικός ακέραιος ο οποίος χρησιμοποιείται για να προσδώσει ελαστικότητα στην τιμή MaxHops. Ουσιαστικά δηλώνει το για πόσα hops πάνω και κάτω από την τιμή που ορίζει το MaxHops θα θεωρηθεί έγκυρο το μονοπάτι (π.χ. +/- 2). Εάν αφεθεί κενό, θεωρείται μηδέν, οπότε και ο αλγόριθμος θα λάβει την απόλυτη τιμή του πεδίου MaxHops για το μονοπάτι της συγκεκριμένης ροής.

- **NeedsBackup**

Boolean πεδίο μέσω του οποίου γίνεται γνωστό το εάν η ροή απαιτεί backup μονοπάτι ή όχι. Η τιμή true σημαίνει ότι πρέπει να υπολογιστεί και να κατασκευαστεί δευτερεύον path, ενώ η τιμή false ή null δηλώνει ότι μόνο η πρωτεύουσα διαδρομή είναι επιθυμητή. Στο σημείο αυτό, αξίζει να τονιστεί ότι τόσο οι παραπάνω περιορισμοί όσο και αυτοί που έπονται στις επόμενες παραγράφους, οφείλουν να ικανοποιούνται τόσο για το πρωτεύον μονοπάτι της συγκεκριμένης ροής, όσο και για το δευτερεύον της (στην περίπτωση, βέβαια, που NeedsBackUp = true).

- **MaxCuts**

Θετικός ακέραιος αριθμός μέσω του οποίου η ροή ορίζει το μέγιστο αριθμό παράλληλων διαδρομών που μπορεί να ακολουθήσει. Για παράδειγμα, εάν MaxCuts = 5, σημαίνει πως στην περίπτωση που η ροή μπορεί να εξυπηρετηθεί μόνο μετά της κατάτμησής της σε άνω του ενός μονοπατιού, τότε το πλήθος των τελευταίων δεν επιτρέπεται να ξεπεράσει τα 5. Στην περίπτωση μη δήλωσης της τιμής, δίδεται από τον αλγόριθμο, τυχαία, η τιμή 3.

LINKREQUIREMENTS

- LinkRequirementID

Κλειδί αυξανόμενης αρίθμησης για την μοναδικοποίηση των εγγραφών του πίνακα.

- DemandID

Εξωτερικό κλειδί του πίνακα Demands που δηλώνει τη ροή η οποία αιτείται του περιορισμού της ζεύξης.

- FromLSR - ToLSR

Εξωτερικό κλειδί του πίνακα Linksto οποίο υποδεικνύει τη ζεύξη της οποίας τον περιορισμό αιτείται η ροή DemandID.

- LevelID

Μη μηδενικός θετικός ακέραιος που δηλώνει το επίπεδο προτεραιότητας του συγκεκριμένου περιορισμού. Χρησιμοποιείται με αύξουσα σειρά προτεραιότητας έτσι ώστε για δύο περιορισμούς 1 και 2, με LevelID 3 και 5 αντίστοιχα, να ισχύει ότι $1.3 > 2.5$. Χρησιμοποιείται από τον αλγόριθμο για να υπολογίσει το όσο το δυνατόν καλύτερο μονοπάτι, δηλαδή το μονοπάτι των οποίων οι εφικτοί περιορισμοί ζεύξεων να έχουν το ελάχιστο LevelID εκ των καταχωρημένων. Προφανές γίνεται ότι εάν μία ροή δηλώσει περιορισμούς με σειρά προτεραιότητας και ο αλγόριθμος δεν καταφέρει να κατασκευάσει μονοπάτι με ένα δοθέν LevelID, θα αναζητήσει ένα άλλο το οποίο να ικανοποιεί το αμέσως επόμενο επίπεδο κ.ο.κ.

Εάν δε δηλωθεί τιμή στο πεδίο αυτό, θεωρείται πως είναι «1» (μέγιστη προτεραιότητα)

- Traverse

Boolean πεδίο που δείχνει εάν η ζεύξη έχει δηλωθεί ότι πρέπει να διασχιστεί ή να αποφευχθεί. Εάν η τιμή είναι true σημαίνει ότι το μονοπάτι (στο συγκεκριμένο επίπεδο LevelID) υποχρεωτικά πρέπει να περιλαμβάνει την καταχωρημένη ζεύξη, ενώ στην αντίθετη περίπτωση ότι πρέπει να μην εμφανιστεί το link εντός του.

$$\text{Traverse} = \begin{cases} 1, & \text{Link } A - B \text{ must be included in the path} \\ 0, & \text{Link } A - B \text{ must be excluded from the path} \end{cases}$$

Επιλογή μη δήλωσης τιμής στο πεδίο αυτό δεν υπάρχει. Να σημειωθεί ότι τόσο το πεδίο Traverse όσο και όλα τα προηγούμενα έχουν ισχύ και πρέπει να ικανοποιούνται όχι μόνο στο πρωτεύον μονοπάτι της ροής που αιτείται του περιορισμού, αλλά και από το δευτερεύον της (εάν υπάρχει).

SERVEDDEMANDS

- ServedDemandID

Κλειδί αυξανόμενης αρίθμησης για τη μοναδικοποίηση των εγγραφών εξυπηρετημένων ροών. Ουσιαστικά, για κάθε τμήμα ροής (πρωτεύουσας ή backup) παράγεται μία μοναδική εξυπηρετημένη (Served) ροή.

- DemandID

Εξωτερικό κλειδί από τον πίνακα Demandsto οποίο υποδεικνύει την ροή η οποία εξυπηρετείται τμηματικά ή εξ' ολοκλήρου από το συγκεκριμένο ServedDemand.

- Cut

Πεδίο το οποίο υποδεικνύει το τμήμα της ροής για την οποία γίνεται λόγος. Λαμβάνει οποιοδήποτε θετικό ακέραιο αριθμό, υπακούοντας στους περιορισμούς MaxCutsπου ορίζει η εκάστοτε αρχική ροή. Για παράδειγμα, εάν μία ροή DemandID τμηθεί τελικά σε 3 ξεχωριστές ροές ServedDemandID, τότε το πεδίο Cutτης κάθε μία από αυτές θα λάβει αριθμούς 1, 2 και 3 δείχνοντας ότι αποτελούν το πρώτο, δεύτερο και τρίτο τμήμα της αρχικής. Η περίπτωση που ένα DemandID έχει μόνο μία εγγραφή στον πίνακα με Cut = 1, υποδεικνύει ότι εξυπηρετήθηκε ως είναι χωρίς την ανάγκη τμηματοποίησης της.

- DemandType

Πεδίο συνάρτησης τύπου :

$$DemandType = \begin{matrix} 1, & ServedDemandID = Primary \\ 0, & ServedDemandID = Backup \end{matrix}$$

- Bandwidth

Θετικός ακέραιος, ο οποίος δείχνει το εύρος ζώνης του κάθε ServedDemandID που εξυπηρετείται. Εάν η ροή δεν τετμηθεί τότε το εύρος ζώνης που εξυπηρετείται είναι ίσο με το απαιτούμενο. Στις περιπτώσεις κατάτμησης της ροής, το πεδίο αυτό καταχωρεί το τμήμα εύρους ζώνης που καταλαμβάνει το κάθε μέρος της αρχικής ροής. Το άθροισμα όλων των επιμέρους Bandwidths των εξυπηρετηθέντων ροών, λοιπόν, δίνει το Bandwidth που αιτεί η αρχική ροή.

Συνοψίζοντας, τα πεδία που αναφέρθηκαν παραπάνω ουσιαστικά δίνουν την πληροφορία «Μία αρχική ροή, η οποία είναι πρωτεύουσα ή backup, σε πόσα τμήματα διαιρέθηκε και τί εύρος ζώνης καταλαμβάνει το κάθε ένα από αυτά».

HOPS

- HopID

Αριθμός αυξανόμενης αρίθμησης ο οποίος μοναδικοποιεί το «Hop» της διαδρομής της κάθε εξυπηρετημένης ροής.

- ServedDemandID

Εξωτερικό κλειδί από τον πίνακα ServedDemands το οποίο υποδεικνύει το τμήμα ποιας ροής εκτελεί το Hop πάνω στην ζεύξη.

- FromLSR – ToLSR

Εξωτερικό κλειδί του πίνακα Links που δηλώνει την ίδια τη ζεύξη από την οποία περνά το τμήμα της εξυπηρετηθείσας ροής.

Μαζί με τα δεδομένα από τον πίνακα ServedDemands, η πληροφορία που δίδεται είναι «Πώς ετμήθη η αρχική ροή, σε τί τμήματα εύρους ζώνης και ποιο το μονοπάτι που ακολουθεί το κάθε ένα από αυτά».

ALGORITHM CALCULATION

Στον πίνακα AlgorithmCalculation διατηρούνται κάποια στατιστικά στοιχεία για λόγους ελέγχου της εκτέλεσης του αλγορίθμου. Ουσιαστικά διατηρεί την πληροφορία του για ποια ομάδα ροής εκτελείται, ποιος ο συνολικός χρόνος εκτέλεσης και πώς ολοκληρώθηκε η εκτέλεση (επιτυχώς ή με σφάλμα). Να σημειωθεί ότι για κάθε ομάδα ροών διατηρείται μία μοναδική εγγραφή στον προκείμενο πίνακα.

ΟΝΤΟΤΗΤΕΣ ΤΟΥ ΑΛΓΟΡΙΘΜΟΥ

Ως οντότητες του αλγορίθμου θεωρούνται στοιχεία και έννοιες οι οποίες παίζουν τον πιο καθοριστικό ρόλο κατά τη διάρκεια των υπολογισμών της Βέλτιστης Κατανομής της Κίνησης και των Πόρων. Στον τελικό κώδικα του αλγορίθμου έχουν τη μορφή Κλάσεων των οποίων τα Αντικείμενα απεικονίζουν τις διάφορες μορφές και κατηγορίες που αυτές μπορούν να λάβουν. Οι πιο σημαντικές οντότητες είναι : Οι διάφορες κατηγορίες του γράφου της τοπολογίας, οι διάφορες κατηγορίες της ομάδας ροών καθώς και η «μνήμη» του αλγορίθμου. Πιο αναλυτικά οι τρεις αυτές έννοιες παρουσιάζονται παρακάτω.

NETWORK

Η οντότητα Network αποτελείται από γράφο ο οποίος απεικονίζει το δίκτυο στις διάφορες μορφές που αυτό μπορεί να λάβει κατά τη διάρκεια της εκτέλεσης του αλγορίθμου. Περιλαμβάνει κόμβους και ακμές οι οποίες αναπαριστούν τους πραγματικούς δρομολογητές και ζεύξεις της τοπολογίας, αντίστοιχα. Κάθε κόμβος μπορεί να είναι είτε απλός LSR είτε LER. Παράλληλα,

μπορούν να υπάρχουν άνω του ενός LERσπηγής και προορισμού, χωρίς την προσθήκη ιδεατών ζεύξεων άπειρης χωρητικότητας που συζητήθηκαν σε προηγούμενη ενότητα.

Κάθε ακμή, πέραν των χαρακτηριστικών που αναφέρθηκαν κατά την ανάλυση της Βάσης Δεδομένων, διατηρεί και πληροφορίες σχετικές με τις δρομολογηθείσες ροές. Τα δεδομένα που κρατούνται σε όλες τις φάσεις εκτέλεσης, συνοψίζονται ως εξής:

- Δεσμευμένη Χωρητικότητα

Αποτελεί το άθροισμα του Bandwidthτων ροών/ τμημάτων ροών που έχει ήδη δεσμευτεί στο συγκεκριμένο Link

- Δρομολογηθείσες Ροές

Πρόκειται για διατηρούμενη ανά ζεύξη πληροφορία, η οποία περιλαμβάνει όλη την απαραίτητη γνώση για τις ροές για τις οποίες έχει δεσμευτεί χωρητικότητα. Πιο αναλυτικά, διατηρείται το αναγνωριστικό της ροής που έχει δεσμεύσει χώρο στη ζεύξη (DemandID), ο αύξων αριθμός του τμήματος της ροής (Cut), η χωρητικότητα που καταλαμβάνει (Bandwidthτης ροής/ τμήματος ροής) και ο τύπος της (πρωτεύον ή δευτερεύον).

Η οντότητα Network, ανάλογα με τη φάση και τις συνθήκες εκτέλεσης του αλγορίθμου, μπορεί να λάβει διαφορετικές μορφές : OriginalNetwork, SubResidualNetwork και ExtendedSubResidualNetwork. Τα χαρακτηριστικά του κάθε τύπου δικτύου, καθώς και οι λόγοι για την χρήση μίας τέτοιας διαφοροποίησης, θα αναλυθούν ευθύς αμέσως.

ORIGINALNETWORK

Ως OriginalNetwork θεωρείται η τοπολογία του αρχικού δικτύου, όπως έχει εξαχθεί από τη βάση δεδομένων. Αποτελεί, ουσιαστικά, την έξοδο του αλγορίθμου καθώς είναι αυτό το οποίο διατηρεί όλες τις πληροφορίες

δρομολόγησης των ροών, ενώ παράλληλα τροφοδοτεί συνεχώς τον αλγόριθμο με τις λεπτομέρειες της πραγματικής τοπολογίας προκειμένου να σχεδιαστούν οι υπόλοιπες μορφές δικτύων και να ληφθούν οι εκάστοτε αποφάσεις. Σε κάθε κύκλο του αλγορίθμου, αναλόγως της έκβασής του, ενημερώνεται με την προσθήκη ή την αφαίρεση ροών μέσω της επεξεργασίας των στοιχείων Flow και RoutedDemands της κάθε ακμής του. Από άποψη τοπολογίας, παραμένει σταθερό με τους ίδιους δρομολογητές και ακμές, από την αρχή της εκτέλεσής του αλγορίθμου μέχρι και τον τερματισμό αυτού.

SUBRESIDUALNETWORK

Το SubResidual δίκτυο αποτελεί υποτμήμα του ResidualNetwork (του οποίου ο ορισμός δόθηκε σε προηγούμενη ενότητα) και σχεδιάζεται βασισμένο στη ροή που εξετάζεται κάθε φορά. Για το σχεδιασμό του απαιτούνται οι εξής πληροφορίες :

- Η πηγή της ροής
- Ο προορισμός της ροής
- Η απαίτηση της ροής σε Bandwidth
- Το είδος της ροής (δηλαδή εάν πρόκειται για πρωτεύουσα ή δευτερεύουσα ροή)
- Οι απαιτήσεις της ροής σε ζεύξεις

Τα παραπάνω δεδομένα εξάγονται και χρησιμοποιούνται για το σχεδιασμό του συγκεκριμένου τύπου δικτύου. Πιο αναλυτικά, η δημιουργία του συγκεκριμένου δικτύου πραγματοποιείται ως εξής:

- Όλοι οι κόμβοι που εμφανίζονται στο OriginalNetwork, θα εμφανιστούν και στο SubResidual
- Από τις ακμές του Αρχικού Δικτύου θα προστεθούν μόνο εκείνες οι οποίες:
 - Δε χρησιμοποιούνται από την πρωτεύουσα ροή, στην

περίπτωση που η ροή που εξετάζεται είναι Backup

- Μπορούν να εξυπηρετήσουν την απαίτηση της ροής σε Bandwidth. Αυτό σημαίνει ότι θα προστεθούν εκείνες οι ακμές για τις οποίες ισχύει:

$$Capacity - Flow \geq Requested Bandwidth$$

Με τις παραπάνω παραδοχές, εάν βρεθεί μονοπάτι στο δίκτυο σημαίνει αυτόματα ότι είναι έγκυρο από γεωγραφικής πλευράς (ξεκινά από την πηγή και τερματίζει τον προορισμό της εκάστοτε ροής), επαρκές από άποψη χωρητικότητας, μιας και οι ακμές που δεν μπορούν να υποστηρίξουν τη ροή απουσιάζουν από το υποδίκτυο, και ανεξάρτητο από πρωτεύουσες ροές στην περίπτωση που εξετάζεται η Backup κάποιας προηγούμενης.

EXTENDED SUBRESIDUAL NETWORK

Το είδος αυτού του δικτύου αποτελεί υποτμήμα του OriginalNetwork και υπερτμήμα του SubResidual. Σχεδιάζεται και αυτό βασισμένο στις απαιτήσεις της εκάστοτε ροής, με ακριβώς τον ίδιο τρόπο και τις ίδιες παραδοχές όπως και το SubResidual, αλλά με τη διαφοροποίηση ότι από τις ακμές του αρχικού δικτύου θα προστεθούν όχι μόνο εκείνες οι οποίες μπορούν να υποστηρίξουν τη ροή, αλλά όλες εκείνες για τις οποίες ισχύει:

$$Capacity - Flow > 0$$

Από τις παραδοχές που τέθηκαν για το SubResidual Network καθώς και από την παραπάνω σχέση, προκύπτει ότι : εάν βρεθεί μονοπάτι στο είδος αυτού του δικτύου τότε είναι έγκυρο γεωγραφικά, ανεξάρτητο από την πρωτεύουσα ροή (στην περίπτωση που εξετάζεται backup path) και πιθανόν ανεπαρκές για τη ροή, αλλά επαρκές για τμήμα αυτής. Το τελευταίο, με πιο απλά λόγια, σημαίνει ότι ένα μονοπάτι που θα βρεθεί στο ExtendedSubResidual μπορεί να είναι εφικτό από κάθε άποψη πλην της ελεύθερης χωρητικότητας η οποία αν και είναι μικρότερη της απαιτούμενης, είναι εντούτοις θετική και θα μπορούσε να χρησιμοποιηθεί για ένα τμήμα της αρχικής ροής.

DEMANDS

Η οντότητα Demands μοντελοποιείται μέσω μίας λίστας η οποία διατηρεί τις ροές που επιθυμούν να διασχίσουν το δίκτυο μαζί με τις απαιτούμενες, για την εκτέλεση του αλγορίθμου, λεπτομέρειες. Συγκεκριμένα, για κάθε Demand καταχωρούνται σχεδόν όλα τα στοιχεία που αποθηκεύονται στον πίνακα Demands της Βάσεως Δεδομένων καθώς και τα δεδομένα HasLinksRequirements, ExtSubResCalculated, CalculateExtSubRes, IsBackUp και Cut τα οποία ορίζονται ως εξής:

- HasLinksRequirements

Boolean πεδίο το οποίο δηλώνει το εάν η συγκεκριμένη ροή έχει περιορισμούς ζεύξεων. Χρησιμοποιείται προκειμένου να απαλειφθεί η ανάγκη συνεχούς διατήρησης του πίνακα LinkRequirements στη μνήμη του αλγορίθμου. Έτσι, εάν μία ροή έχει θετική τιμή στη συγκεκριμένη παράμετρο, ο αλγόριθμος όποτε αυτό είναι απαραίτητο, αναζητά τα κατάλληλα δεδομένα εντός της ΒΔ, τα καταχωρεί προσωρινά στη μνήμη, τα επεξεργάζεται και τα απελευθερώνει.

- ExtSubResCalculated

Boolean πεδίο το οποίο δηλώνει εάν για τη συγκεκριμένη ροή έχει ήδη υπολογιστεί η ExtendedSubResidual μορφή του δικτύου.

- CalculateExtSubRes

Boolean πεδίο το οποίο ορίζει το εάν για τη συγκεκριμένη ροή πρέπει να υπολογιστεί το ExtendedSubResidual Network προκειμένου να βρεθεί εφικτό μονοπάτι. Εάν είναι θετική, όπως θα αναλυθεί και παρακάτω, ο αλγόριθμος προσπερνά το βήμα κατασκευής του SubResidual δικτύου και προχωράει απ' ευθείας στον Extended του τύπο.

- IsBackup

Boolean πεδίο το οποίο διατηρεί την πληροφορία για τον τύπο της ροής. Εάν IsBackup είναι αληθές, σημαίνει ότι η ροή αποτελεί τη δευτερεύουσα ροή της αντίστοιχής της. Σε αντίθετη περίπτωση, είναι η

πρωτεύουσα ροή.

- Cut

Παράμετρος η οποία δείχνει το τμήμα της ροής. Όπως αναφέρθηκε και παραπάνω, κάθε ροή δύναται να διασπαστεί σε δύο ή περισσότερα μονοπάτια. Σε κάθε ένα από τα μονοπάτια, λοιπόν, δρομολογείται ένα τμήμα της αρχικής ροής. Κάθε τμήμα θεωρείται σαν άλλη μία ροή η οποία έχει τα ίδια ακριβώς χαρακτηριστικά με την αρχική αλλά διαθέτει τη δική της τιμή Bandwidth καθώς και το δικό της αναγνωριστικό Cut. Για παράδειγμα, έστω η ροή A η οποία αιτείται 10 μονάδες Bandwidth. Αλλά αντί για ένα μόνο μονοπάτι το οποίο να μπορεί να υποστηρίξει την απαίτηση αυτή, υπάρχουν δύο με χωρητικότητες 6 και 4 μονάδες αντίστοιχα. Αυτό σημαίνει ότι η ροή A θα αντικατασταθεί από δύο ροές A.1 και A.2 (όπου 1 και 2 είναι οι τιμές της παραμέτρου Cut) με Bandwidth 6 και 4 και τις υπόλοιπες απαιτήσεις ίδιες με την αρχική (και πλέον ανύπαρκτη, μιας και αντικαθίσταται) A.

Η λίστα των Demands χαρακτηρίζεται από τον τύπο της ανάλογα με τη χρήση της. Συγκεκριμένα, στον αλγόριθμο ορίζονται τρεις τύποι συνόλων των ροών:

- Demands List

Η λίστα των ροών που αιτούνται χρήσης του δικτύου. Λαμβάνεται κυρίως μέσα από τη Βάση Δεδομένων, ενώ διατηρεί και τις δευτερεύουσες ροές. Πιο συγκεκριμένα, για κάθε ροή που αιτείται BackUp, δημιουργείται άλλη μία με ακριβώς τα ίδια χαρακτηριστικά αλλά τιμή true στην παράμετρο IsBackUp και τοποθετείται στη λίστα DemandsList. Έτσι, κάθε δευτερεύουσα ροή αντιμετωπίζεται σαν μία οποιαδήποτε άλλη ροή με μοναδική προϋπόθεση να αποφεύγει κάθε ζεύξη του μονοπατιού της πρωτεύουσας.

Η προσθήκη των ροών στην DemandsList γίνεται ταξινομημένα λαμβάνοντας υπ' όψιν (με την αναγραφόμενη σειρά) τα στοιχεία:

- Συνολικό αιτούμενο Bandwidth, το οποίο υπολογίζεται ως το Bandwidth της ροής εάν δεν αιτεί BackUp Route και το

Bandwidth * 2 της ροής εάν χρήζει δευτερεύουσας.

- Το πλήθος των περιορισμών ζεύξεων, δηλαδή το ύψος του πίνακα LinkRequirements, ασχέτως του LevelID του.
- Την τιμή MaxHops που ορίζει η ροή για το μονοπάτι
- Το μέγεθος ελαστικότητας Max Cuts

Με τον τρόπο αυτό, οι ροές που θα εξυπηρετηθούν πρώτα θα είναι αυτές που έχουν τη μεγαλύτερη απαίτηση σε Bandwidth και τις περισσότερους περιορισμούς, με τις πιο «απλές» ροές να έπονται.

Να σημειωθεί ότι κάθε ροή που προστίθεται εντός της λίστας DemandsList, κατά την αρχικοποίηση των δεδομένων του αλγορίθμου, έχει πάντα τιμή Cutίση με «1», δηλώνοντας ότι είναι το πρώτο (και προς το παρόν μοναδικό) τμήμα.

- ServedList

Η ServedList διατηρεί τις ροές (και τα τμήματα αυτών) που έχουν εξυπηρετηθεί. Ουσιαστικά αποτελεί ένα σημείο αναφοράς για εκείνα τα Demands για τα οποία έχει βρεθεί το εφικτό και βέλτιστο μονοπάτι. Σε αντίθεση με την DemandsList, δε διατηρεί απαραίτητως ολόκληρες τις ροές, αλλά και τα τμήματά τους εάν αυτές έχουν υποστεί διαχωρισμό σε δύο ή παραπάνω μονοπάτια. Αξίζει να τονιστεί ότι προκειμένου μία ροή να θεωρηθεί εξυπηρετημένη και να μετακινηθεί από την DemandsList στην ServedList, θα πρέπει να έχει βρεθεί μονοπάτι τόσο για την ίδια (ή όλα τα τμήματά της) όσο και για την δευτερεύουσά της (ή όλα τα τμήματά της) όπου ζητείται.

- TrashList

Στην TrashList αποθηκεύονται όλες οι ροές (και τα τμήματα αυτών) τα οποία δεν έτυχε να εξυπηρετηθούν. Ουσιαστικά, εάν για ένα οποιοδήποτε τμήμα μίας ροής (και της δευτερευούσας της) δε βρεθεί μονοπάτι, θεωρείται μη εξυπηρετήσιμη και μετακινείται από την DemandsList στην TrashList.

NETWORK MEMO

Όπως θα εξηγηθεί και παρακάτω, ο αλγόριθμος εκτελεί έως δύο απόπειρες εύρεσης μονοπατιού για κάθε ροή ή τμήμα αυτής. Πρώτα δοκιμάζει να βρει μία διαδρομή στο SubResidual δίκτυο και εάν δεν τα καταφέρει αναζητά στο ExtendedSubResidual. Ωστόσο, η αποτυχημένη απόπειρα εύρεσης path στο SubResidual σημαίνει ότι δε βρέθηκε μονοπάτι από μία συγκεκριμένη πηγή προς έναν συγκεκριμένο προορισμό με γνωστή ποσότητα Bandwidth. Το παραπάνω συνεπάγεται ότι εάν δε βρέθηκε διαδρομή για ένα DemandX με πηγή S, προορισμό D και Bandwidth B_x τότε και μία άλλη ροή Υπου επιθυμεί να διασχίσει το δίκτυο από το S στο D με Bandwidth $B_y \geq B_x$ δε θα βρει μονοπάτι στο SubResidual.

Προκειμένου, λοιπόν, να αυξηθεί η ευφυΐα του αλγορίθμου και να αφαιρεθεί ένας αποτυχημένος κύκλος αναζήτησης στο SubResidual τύπο δικτύου, στον αλγόριθμο αποδίδεται ένα είδος «μνήμης» για να κρατάει τις αποτυχημένες του προσπάθειες. Αυτό γίνεται μέσω της οντότητας NetworkMemo η οποία κρατά για κάθε ζεύγος πηγής και προορισμού την ελάχιστη ποσότητα Bandwidth που δεν κατάφερε να εξυπηρετηθεί.

Έτσι, έστω το προηγούμενο DemandX το οποίο δεν κατάφερε να εξυπηρετηθεί εντός της αναζήτησης στο SubResidual. Ο αλγόριθμος θα καταχωρήσει στο NetworkMemo την τριάδα [S, D, B_x]. Όταν λοιπόν φτάσει η σειρά για το Υ, ο αλγόριθμος θα «θυμάται» ότι για το ίδιο ζεύγος πηγής και προορισμού και με Bandwidth μικρότερο ή ίσο του B_y ήδη έχει αποτύχει, και θα κατασκευάσει απ' ευθείας το ExtendedSubResidual. Εάν αργότερα μία ροή Z με πηγή και προορισμό S και D αντίστοιχα, αλλά Bandwidth $B_z < B_x$ δεν εξυπηρετηθεί, τότε στο NetworkMemo η τιμή B_x θα αντικατασταθεί από την μικρότερη B_z .

ΑΛΓΟΡΙΘΜΟΣ ΔΙΑΧΕΙΡΙΣΗΣ ΠΟΡΩΝ ΚΑΙ ΚΙΝΗΣΗΣ

ΕΙΣΟΔΟΣ

- Γράφος ON (OriginalNetwork) εξαγόμενος από τους πίνακες LSRs και Links της Βάσης Δεδομένων.
- Ταξινομημένη λίστα απαιτήσεων DemandsList ληφθείσα από τους πίνακες Demands, LinksRequirements της Βάσης Δεδομένων μαζί με την δημιουργία και προσθήκη των BackUpDemands

ΈΞΟΔΟΣ

- Γράφος ON με ακμές (Flow και RoutedDemands) κατάλληλα ενημερωμένες με τις τιμές Demand ID, Bandwidth, Cut και IsBackUp των εξυπηρετηθέντων ροών

ΑΡΧΙΚΟΠΟΙΗΣΗ

- \forall ακμή στο ON (Original Network) τίθεται Flow = 0
- \forall Demand στην Demand List τίθεται ExtSubResCalculated = false
- \forall Demand στην Demand List τίθεται CalculateExtSubRes = false
- \forall Demand στην Demand List τίθεται Cut = 1

ΦΑΣΗ 1 – ΈΛΕΓΧΟΣ ΛΙΣΤΑΣ DEMANDSLIST

1. Εάν η λίστα είναι κενή
 - 1.1. Τερμάτισε
2. Αλλιώς
 - 2.1. Πήγαινε στη Φάση 2

ΦΑΣΗ 2 – ΛΗΨΗ (ΡΟΡ) ΤΗΣ ΠΡΩΤΗΣ ΡΟΗΣ ΣΤΗΝ ΚΟΡΥΦΗ ΤΗΣ DEMANDSLIST

1. Εάν για τη ροή έχει ήδη υπολογιστεί το ExtendedSubResidual δίκτυο (ESRN)
 - 1.1. Απόρριψε τη ροή (Φάση 7)
 - 1.2. Πήγαινε στη Φάση 1
2. Εάν η ροή έχει CalculateExtSubRes = true ή υπάρχει εγγραφή στο NetworkMemo για το Source, το Destination και Bandwidth μικρότερο ή ίσο της ροής
 - 2.1. Κατασκεύασε Extended SubResidual δίκτυο
 - 2.2. Θέσε ExtSubResCalculated = true για τη ροή
3. Αλλιώς
 - 3.1. Κατασκεύασε SubResidual δίκτυο (SRN)

ΦΑΣΗ 3 – ΚΑΤΑΣΚΕΥΗ LAYERED ΓΡΑΦΟΥ G_L ΑΠΟ ΤΟ ΔΙΚΤΥΟ

1. \forall κόμβο στο (ESRN ή SRN) δίκτυο θέσε Height = -1
2. Υπολόγισε το Height κάθε κόμβου έτσι ώστε αν E το σύνολο των ακμών του δικτύου, V το σύνολο των κόμβων, s η πηγή του δικτύου (πηγή του Demand) και t ο προορισμός του δικτύου (προορισμός του Demand) τότε ισχύει
$$\forall (u, v) \in E : Height(v) = Height(u) + 1$$
3. Εάν Height(t) = -1

- 3.1. Εάν το δίκτυο είναι ESRN
 - 3.1.1. Απόρριψε τη ροή (Φάση 7)
 - 3.1.2. Πήγαινε στη Φάση 1
- 3.2. Εάν το δίκτυο είναι SRN
 - 3.2.1. Θέσε CalculateExtSubRes = true
 - 3.2.2. Πήγαινε στη Φάση 1
- 4. Αλλιώς
 - 4.1. Πήγαινε στη Φάση 4

ΦΑΣΗ 4 – ΑΝΑΖΗΤΗΣΗ ΒΕΛΤΙΣΤΟΥ ΜΟΝΟΠΑΤΙΟΥ

- 1. \forall μονοπάτι που βρίσκεις στο δίκτυο, υπολόγισε
 - 1.1. Το LevelID των περιορισμών ζεύξεων που μπορεί να εξυπηρετηθεί
 - 1.2. Τη μέγιστη χωρητικότητα ακμής
 - 1.3. Την ελάχιστη χωρητικότητα ακμής
 - 1.4. Το πλήθος των Hops του μονοπατιού
- 2. Εάν το μονοπάτι δεν υπακούει στους περιορισμούς της ροής ή η ελάχιστη χωρητικότητα είναι μηδενική
 - 2.1. Εάν έχεις άλλα μονοπάτια να εξετάσεις
 - 2.1.1. Πήγαινε στο βήμα 1
 - 2.2. Αλλιώς
 - 2.2.1. Επέστρεψε τιμή Bandwidth = -1
- 3. Αλλιώς
 - 3.1. Υπολόγισε το βάρος του μονοπατιού :
$$PathWeight = (Hops \text{ μονοπατιού} / \text{Ζητούμενα Hops}) + (LevelID \text{ του τελευταίου ζητούμενου περιορισμού ζεύξεων} / LevelID \text{ μονοπατιού})$$
 - 3.2. Επέλεξε εκείνο το μονοπάτι που έχει τη μεγαλύτερη τιμή MaximumBandwidth και το μεγαλύτερο PathWeight
 - 3.3. Επέστρεψε σαν ευρεθείσα τιμή, τη μικρότερη τιμή Capacityτων Linkstou μονοπατιού
 - 3.4. Πήγαινε στη Φάση 5

ΦΑΣΗ 5 – ΕΠΕΞΕΡΓΑΣΙΑ ΒΕΛΤΙΣΤΟΥ ΜΟΝΟΠΑΤΙΟΥ

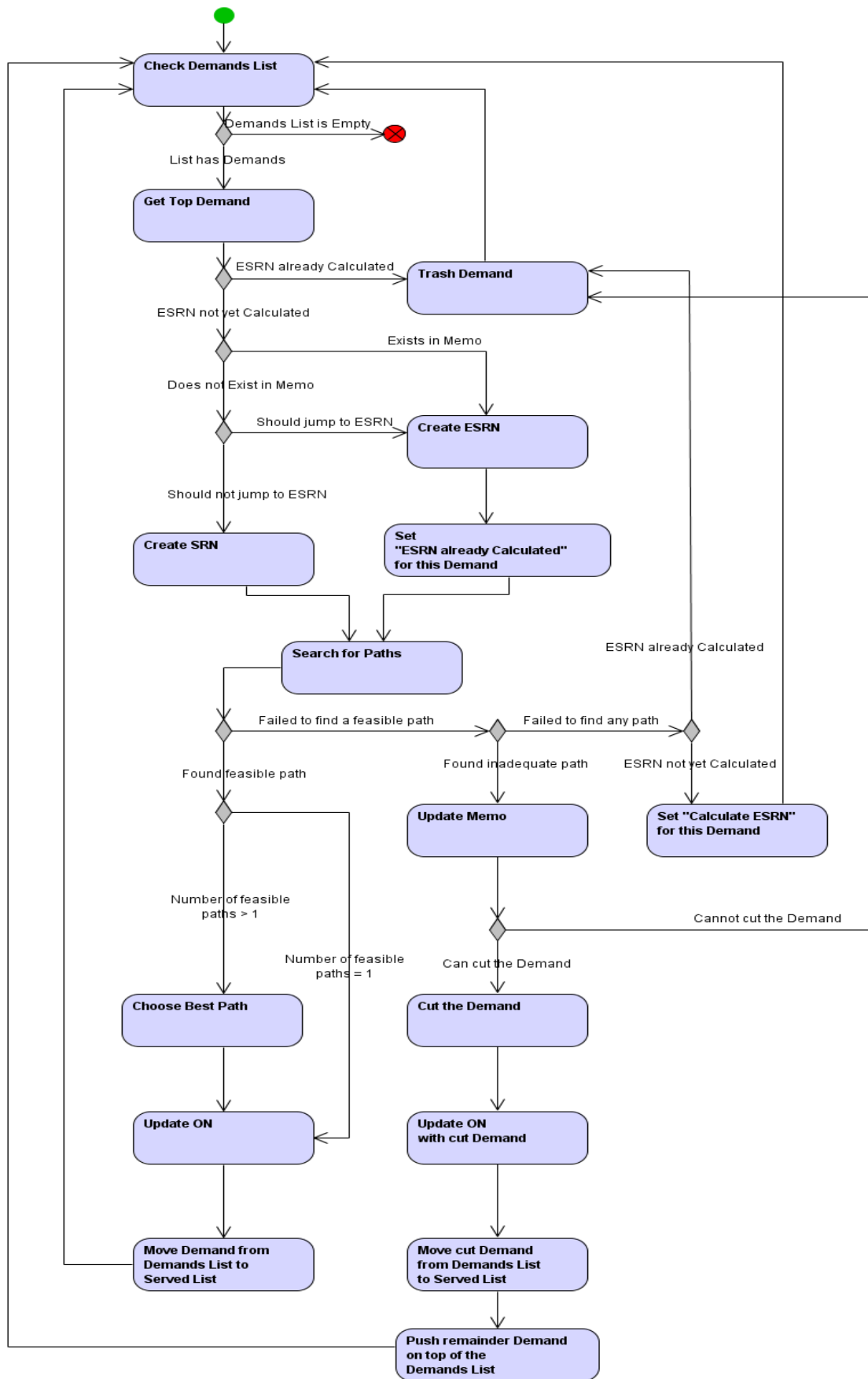
1. Εάν η χωρητικότητα του μονοπατιού είναι επαρκής (μεγαλύτερη ή ίση της απαιτούμενης)
 - 1.1. Εξυπηρέτησε τη ροή (Φάση 6)
 - 1.2. Πήγαινε στη Φάση 1
2. Αλλιώς αν η χωρητικότητα του μονοπατιού είναι θετική αλλά μη επαρκής
 - 2.1. Καταχώρησε/ Ενημέρωσε στο NetworkMemo την τριάδα [Source, Destination, Ελάχιστο Bandwidth μονοπατιού]
 - 2.2. Εάν το Cut της ροής είναι ίσο με το MaxCut που αιτεί (δεν μπορεί να διαχωριστεί περαιτέρω)
 - 2.2.1. Απόρριψε τη ροή (Φάση 7)
 - 2.2.2. Πήγαινε στη Φάση 1
 - 2.3. Αλλιώς
 - 2.3.1. Δημιούργησε δύο ροές
 - 2.3.2. Στην πρώτη ροή θέσε ως ζητούμενη χωρητικότητα την ελάχιστη χωρητικότητα του μονοπατιού
 - 2.3.3. Εξυπηρέτησε την πρώτη ροή (Φάση 6)
 - 2.3.4. Στη δεύτερη ροή θέσε ως ζητούμενη χωρητικότητα τη διαφορά μεταξύ ζητούμενης και ελάχιστης ευρεθείσας χωρητικότητας
 - 2.3.5. Θέσε στη δεύτερη ροή ExtSubResCalculated = false
 - 2.3.6. Αύξησε το μετρητή Cut της ροής κατά 1
 - 2.3.7. Τοποθέτησε τη δεύτερη ροή στην κορυφή της λίστας DemandsList
 - 2.3.8. Πήγαινε στη Φάση 1
3. Αλλιώς εάν η ελάχιστη χωρητικότητα ζεύξεως του μονοπατιού είναι μηδενική
 - 1.1. Εάν το ExtSubResCalculated της ροής είναι true
 - 1.1.1. Απόρριψε τη ροή (Φάση 7)
 - 1.1.2. Πήγαινε στη Φάση 1
 - 1.2. Αλλιώς
 - 1.2.1. Θέσε CalculateExtSubRes = true
 - 1.2.2. Πήγαινε στη Φάση 1

ΦΑΣΗ 6 – ΕΞΥΠΗΡΕΤΗΣΗ ΡΟΗΣ

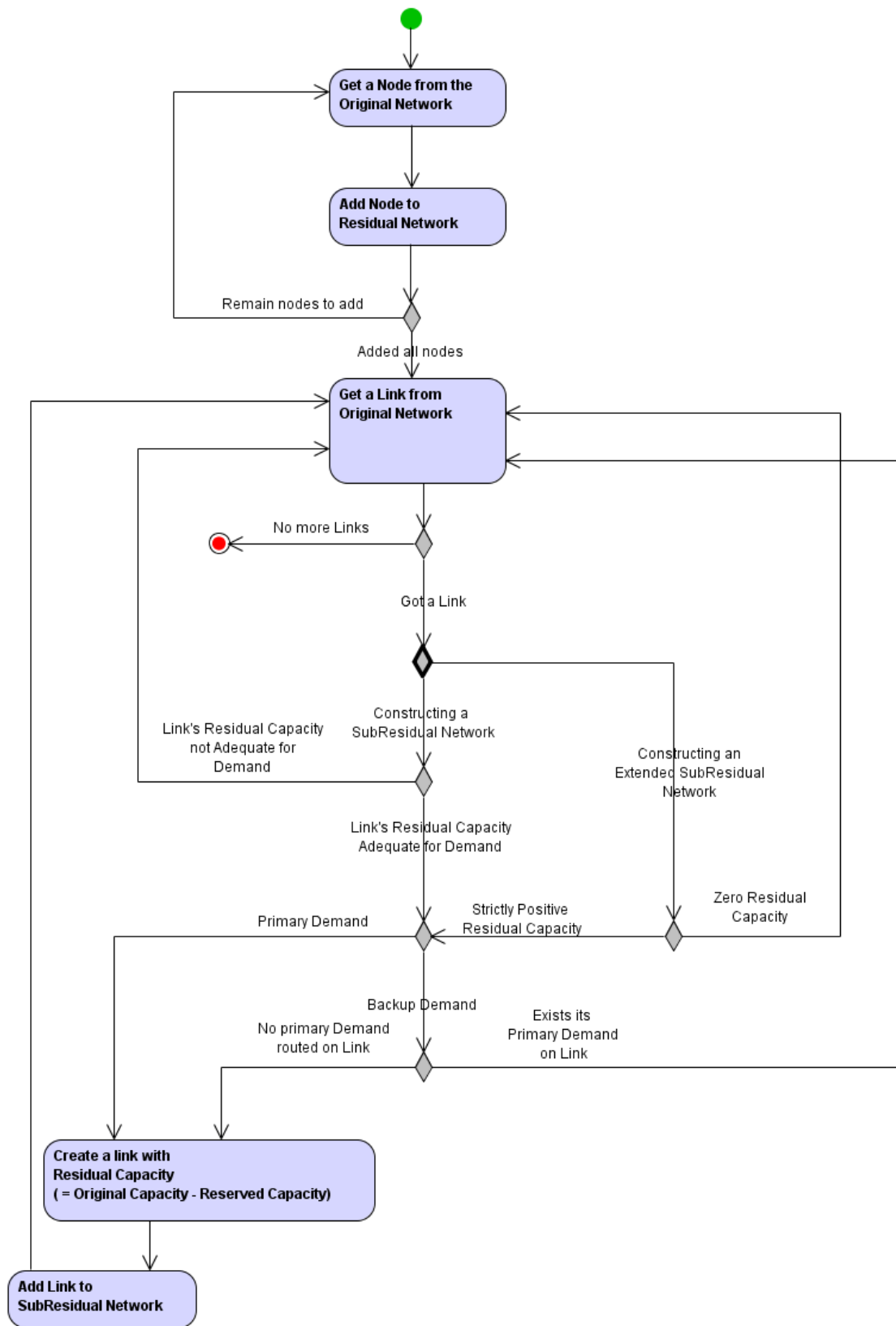
1. √ ακμή του μονοπατιού
 - 1.1. Πρόσθεσε στη λίστα Routed Demands του Αρχικού Δικτύου τα στοιχεία [DemandID, Cut, Bandwidth, IsPrimary]
 - 1.2. Αύξησε την τιμή Flow της ακμής κατά Bandwidth μονάδες
2. Πρόσθεσε τη ροή στη λίστα ServedDemands
3. Αφαίρεσε τη ροή από τη λίστα DemandsList
4. Επέστρεψε

ΦΑΣΗ 7 – ΑΠΟΡΡΙΨΗ ΡΟΗΣ

1. Αν το Cut της ροής έχει τιμή 1
 - 1.1. Πρόσθεσε τη ροή στη λίστα TrashList
 - 1.2. Αφαίρεσε τη ροή από τη λίστα DemandsList
 - 1.3. Επέστρεψε
2. Αλλιώς
 - 2.1. √ ακμή του δικτύου
 - 2.1.1. Εάν διασχίζεται από την τρέχουσα ροή
 - 2.1.1.1. Θέσε το Flow της ακμής ίσο με $\text{Flow} + \text{Bandwidth}$ της ροής
 - 2.1.1.2. Αφαίρεσε την εγγραφή της ροής από την RoutedDemands λίστα της ακμής
 - 2.2. Πρόσθεσε όλα τα τμήματα της πρωτεύουσας και της δευτερεύουσας (εάν υπάρχει) ροής στην TrashList
 - 2.3. Αφαίρεσε όλα τα τμήματα της πρωτεύουσας και της δευτερεύουσας (εάν υπάρχει) ροής από την DemandsList
 - 2.4. Αφαίρεσε όλα τα τμήματα της πρωτεύουσας και της δευτερεύουσας (εάν υπάρχει) ροής από την ServedList
 - 2.5. Επέστρεψε



Εικόνα 18. ActivityDiagram Συνολικού Αλγορίθμου



Εικόνα 19. ActivityDiagram Κατασκευής Υπολειπόμενου Γράφου

ΑΠΟΤΕΛΕΣΜΑΤΑ

ΧΡΟΝΟΙ ΕΚΤΕΛΕΣΗΣ

Ο παρακάτω πίνακας εμφανίζει τις λεπτομέρειες των τοπολογιών και των απαιτήσεων στις οποίες εφαρμόστηκε ο αλγόριθμος, καθώς και τον συνολικό χρόνο εκτέλεσής του. Κάθε γραμμή του πίνακα απεικονίζει μία εκτέλεση ανά δίκτυο και ομάδα ροών. Η σημασία των στηλών, πιο συγκεκριμένα, έχει ως εξής:

Demands : Πλήθος ροών της ομάδας που αιτείται διάσχισης στο συγκεκριμένο δίκτυο.

LSRs : Πλήθος των δρομολογητών που ανήκουν στο Δίκτυο. Αποτελούνται από απλούς LSRs καθώς και LERs.

Links : Πλήθος ακμών από τις οποίες αποτελείται το δίκτυο.

MiDB : (Minimum Demand's Bandwidth) Το ελάχιστο απαιτούμενο Bandwidth από τις ροές της συγκεκριμένης ομάδας.

MaDB : (Maximum Demand's Bandwidth) Το μέγιστο απαιτούμενο Bandwidth ανάμεσα στις ροές της συγκεκριμένης ομάδας.

MiLC : (Minimum Link's Capacity) Η ελάχιστη αρχική χωρητικότητα των ακμών του δικτύου.

MaLC : (Maximum Link's Capacity) Η μέγιστη αρχική χωρητικότητα των ακμών του δικτύου

Demands	LSRs	Links	MiDB	MaDB	MiLC	MaLC	Time (sec)
6	9	16	2	5	1	123	2
4	5	7	10	15	20	20	1
1	2	1	2	2	5	5	0,5
1	4	4	2	2	2	765	1
1	4	4	2	2	2	765	0,5
1	6	10	1	1	9	9	1
7	10	38	10	100	72	99	1
19	30	157	2	140	125	199	45
20	50	231	1	20	100	199	5
44	50	165	1	5	100	199	52
188	70	241	1	1	120	200	90

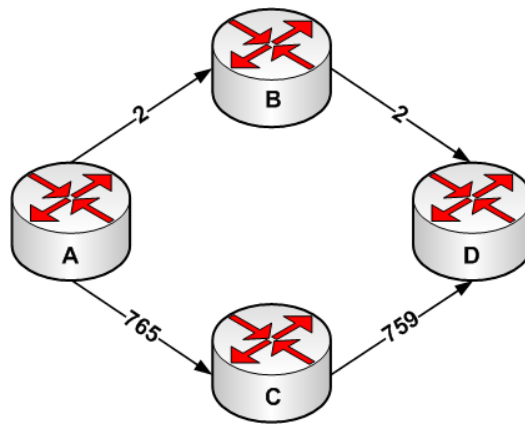
100	100	170	1	1	100	101	69
8	8	15	8	9	10	200	2
10	100	197	10	100	302	499	123

Πίνακας 1. Συγκεντρωτικός Πίνακας Τοπολογιών, Απαιτήσεων και Χρόνων

Όπως παρατηρείται και από τον παραπάνω πίνακα, ο ελάχιστος χρόνος εκτέλεσης στα συγκεκριμένα παραδείγματα είναι τα 0.5 δευτερόλεπτα, ενώ ο μέγιστος τα 123.

Ο χρόνος εκτέλεσης της τάξης του μισού δευτερολέπτου παρατηρείται σε δύο διαφορετικά πλαίσια εφαρμογής: κατά την τρίτη και πέμπτη εκτέλεση. Πιο συγκεκριμένα, αν ληφθεί το παράδειγμα της τρίτης εκτέλεσης του αλγορίθμου, παρατηρείται ότι ο αλγόριθμος εκτελέστηκε στο ελάχιστο δυνατό δίκτυο με μόλις δύο δρομολογητές και μία ακμή ανάμεσά τους. Η ακμή έχει 5 διαθέσιμες μονάδες capacity, ενώ αιτείται της διάσχισής της μόλις μία ροή με 2 μονάδες bandwidth. Όπως είναι προφανές, μόνο ένα μονοπάτι μπορεί να υπολογιστεί, στο οποίο μάλιστα χωρά να περάσει ολόκληρη η ροή. Συνεπώς, για ένα τόσο μικρό δίκτυο και τόσο μικρή ροή, ο χρόνος του αλγορίθμου δικαιολογείται να είναι τόσο μικρός.

Εάν συγκριθεί η τοπολογία της τρίτης εκτέλεσης με την αντίστοιχη της πέμπτης, παρατηρούνται μεγάλες διαφορές την ώρα που ο χρόνος εκτέλεσης παραμένει ίδιος. Όπως φαίνεται, το δίκτυο αποτελείται από 4 δρομολογητές οι οποίοι ενώνονται με 4 ακμές. Η ακμή με την ελάχιστη χωρητικότητα έχει 2 μονάδες Capacity, ενώ η αντίστοιχη με τη μέγιστη χωρητικότητα έχει 765. Παράλληλα, αιτείται διάσχισης του συγκεκριμένου δικτύου μόλις μία ροή η οποία επιθυμεί να καταλάβει μόλις 2 μονάδες bandwidth. Αν και δε φαίνεται από τον παραπάνω πίνακα, αξίζει να τονιστεί ότι η ροή απαιτεί και backup route. Από τα προηγούμενα προκύπτει ότι δύο, τελικά, ροές οι οποίες αιτούνται 2 μονάδων bandwidth διασχίζουν ένα δίκτυο το οποίο περιλαμβάνει 4 ακμές. Η ακριβής τοπολογία του δικτύου φαίνεται στην εικόνα που ακολουθεί.



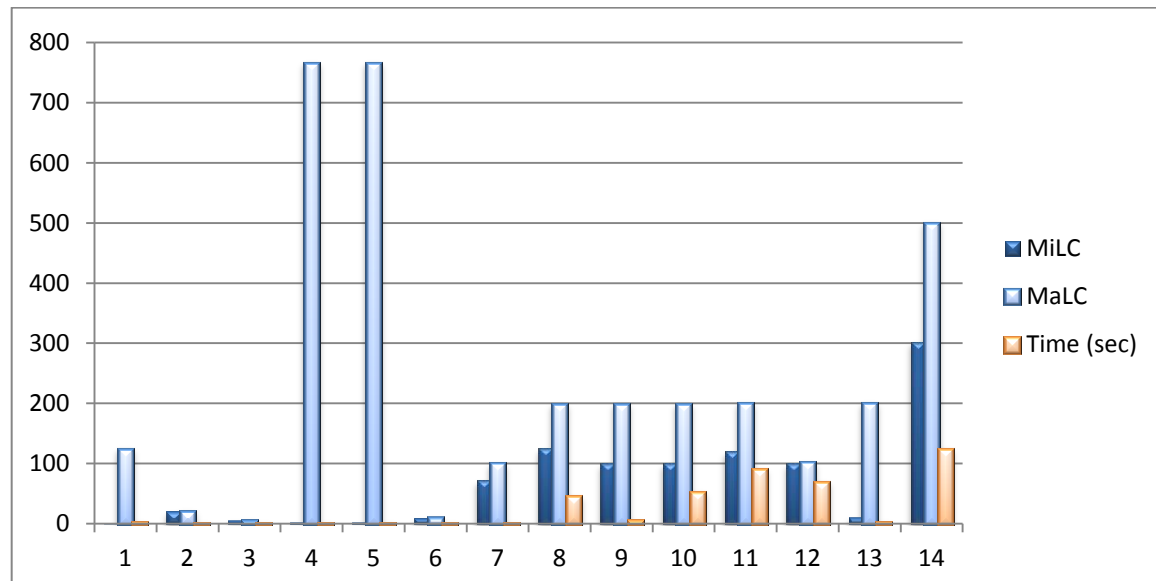
Εικόνα 20. Χείριστη Περίπτωση Τοπολογίας

Υπολογίζοντας τα βέλτιστα μονοπάτια, ο αλγόριθμος εξάγει πρώτα τα μονοπάτια [A – B – D] και [A – C – D]. Το δεύτερο έχει μεγαλύτερη ελεύθερη χωρητικότητα από το πρώτο, οπότε θα το επιλέξει για την πρώτη ροή. Η δεύτερη ροή, δε, είναι η backup της πρώτης. Έτσι μόνο ένα μονοπάτι μπορεί να εξαχθεί: το [A – B – D]. Και τα δύο μονοπάτια επαρκούν για τις ροές, οπότε κανένας άλλος έλεγχος δε χρειάζεται να γίνει. Έτσι, παρ' όλο που η τοπολογία αυτή διαφέρει αρκετά από την πρώτη, δικαιολογείται να έχει τον ίδιο ελάχιστο χρόνο εκτέλεσης.

Το μεγαλύτερο χρόνο εκτέλεσης χρειάστηκε το τελευταίο πλαίσιο εφαρμογής, για το οποίο οι υπολογισμοί διήρκησαν 123 δευτερόλεπτα. Στο εν προκειμένω παράδειγμα, 10 ροές με τουλάχιστον 10 και το πολύ 100 μονάδες bandwidth αιτούνται διάσχισης δικτύου το οποίο αποτελείται από 100 δρομολογητές συνδεδεμένους με 197 ακμές. Η ακμή με την μικρότερη χωρητικότητα έχει 302 μονάδες Capacity, ενώ εκείνη με τη μέγιστη, 499. Παρατηρείται ότι λίγες ροές με μικρές απαιτήσεις σε χωρητικότητα επιθυμούν να διασχίσουν μία μεγάλη τοπολογία δικτύου. Προφανές γίνεται, ότι για τόσο λίγες και μικρές ροές, τα εφικτά μονοπάτια που θα εξεταστούν για να ευρεθεί το καλύτερο, θα είναι πάρα πολλά. Έτσι και ο απαιτούμενος χρόνος εκτέλεσης για την βέλτιστη κατανομή των ροών στην προκειμένη τοπολογία δικτύου, αυξάνεται κατά πολύ σε σχέση με τα άλλα παραδείγματα πολύ απλών δικτύων που παρουσιάστηκαν, φτάνοντας τα 123 δευτερόλεπτα.

ΧΡΟΝΟΙ ΣΥΝΑΡΤΗΣΕΙ LINKCAPACITIES

Το παρακάτω διάγραμμα απεικονίζει τους χρόνους εκτέλεσης των 14^{ων} παραδειγμάτων συναρτήσεως των μέγιστων και ελάχιστων Capacities, προκειμένου να φανεί το πώς η διαφορά των χωρητικότητων των ακμών ενός δικτύου επιδρά στο συνολικό χρόνο εκτέλεσης του αλγορίθμου.

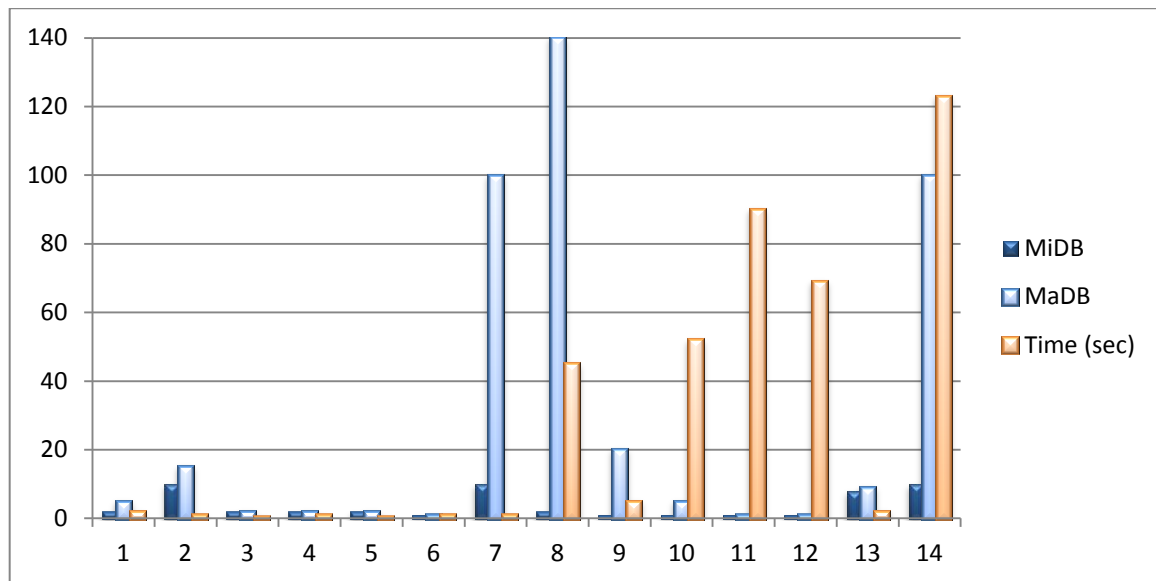


Διάγραμμα 1. Χρόνοι Εκτέλεσης Βάσει Capacity

Όπως φαίνεται και από το διάγραμμα, ο μέγιστος χρόνος εκτέλεσης εμφανίζεται στο δέκατο τέταρτο παράδειγμα εφαρμογής, την ώρα που στα πρώτα 7 είναι πολύ μικρός. Παρατηρείται ότι ο χρόνος εκτέλεσης είναι πολύ μικρός τόσο σε δίκτυα των οποίων οι ακμές έχουν πολύ μεγάλες διαφορές χωρητικότητας (4, 5) όσο και σε τοπολογίες με σχετικά ίσες και μικρές χωρητικότητες (2, 3). Παράλληλα, παρατηρείται ότι από την εφαρμογή 8 και μετά οι χρόνοι αυξάνονται, χωρίς να μπορεί να αιτιολογηθεί κάπως δοθέντων των χωρητικότητων των ακμών και μόνο. Για παράδειγμα, στις εφαρμογές 9 και 10, η ελάχιστη χωρητικότητα ακμής είναι η μισή της μέγιστης και ίση και για τις δύο τοπολογίες. Ωστόσο, οι χρόνοι εφαρμογής του αλγορίθμου για τις δύο αυτές, χωρητικά ίσες τοπολογίες, είναι αρκετά διαφορετικοί. Φαίνεται, λοιπόν, ότι οι χωρητικότητες ενός δικτύου, όπως επίσης και η διασπορά των τιμών τους, δεν επιδρούν καθόλου στον απαιτούμενο, για την επιτυχή εκτέλεση, χρόνο του αλγορίθμου.

ΧΡΟΝΟΙ ΣΥΝΑΡΤΗΣΕΙ ΑΠΑΙΤΟΥΜΕΝΟΥ BANDWIDTH

Συνεχίζοντας την προηγούμενη εξέταση όσον αφορά στον χρόνο εκτέλεσης του αλγορίθμου, στο παρακάτω διάγραμμα παρουσιάζονται οι ίδιοι χρόνοι αλλά συναρτήσει των μέγιστων και ελάχιστων απαιτούμενων μονάδων Bandwidth ανά Ομάδα Demands.



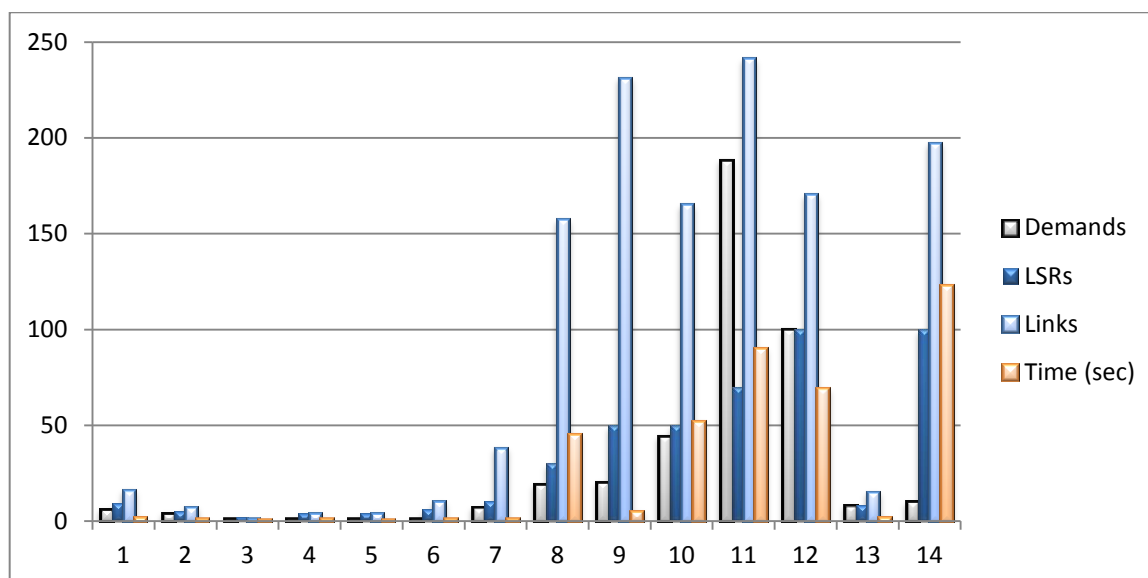
Διάγραμμα 2. Χρόνοι Εκτέλεσης Βάσει Bandwidth

Όπως παρατηρείται στα παραδείγματα 6 και 11, οι ροές αιτούνται πολύ μικρών και σχετικά ίσων μέγιστων και ελάχιστων τιμών Bandwidth. Ωστόσο, οι χρόνοι εκτέλεσης διαφέρουν εκπληκτικά. Και στις δύο εφαρμογές, τα ζητούμενα Bandwidths ισούνται με μία μόλις μονάδα. Όπως φαίνεται και σε αυτή την περίπτωση οι απαιτούμενες τιμές σε Bandwidth δεν έχουν απολύτως καμία επίδραση στο χρόνο εκτέλεσης του αλγορίθμου.

ΧΡΟΝΟΙ ΣΥΝΑΡΤΗΣΕΙ ΤΟΠΟΛΟΓΙΑΣ ΚΑΙ ΠΛΗΘΟΥΣ DEMANDS

Αφού, όπως φάνηκε και στις προηγούμενες υποενότητες, τα μεγέθη των Capacities και Bandwidths, καθώς και η διασπορά αυτών, δεν επιδρούν στον απαιτούμενο χρόνο εκτέλεσης του υπό ανάλυση αλγορίθμου, στην εν προκειμένω υποενότητα θα εξεταστεί ο ρόλος της ίδιας της τοπολογίας καθώς και του πλήθους των ροών στους τελικούς χρόνους.

Το διάγραμμα που ακολουθεί παρουσιάζει σχηματικά το πλήθος των ροών που αιτούνται χρήσης του δικτύου, σε κάθε εκτέλεση, καθώς και το πλήθος των δρομολογητών και των ζεύξεων που τους ενώνουν.



Διάγραμμα 3. Χρόνοι Εκτέλεσης Βάσει Τοπολογίας

Με μια πρώτη εξέταση, φαίνεται ότι το πλήθος των ροών συναρτήσε του πλήθους των ακμών, δεν παίζει ρόλο στον τελικό χρόνο εκτέλεσης του αλγορίθμου. Για παράδειγμα, στην εκτέλεση υπ' αριθμόν 9 η τοπολογία αποτελείται από σχετικά λίγες ροές οι οποίες διασχίζουν ένα πολύ μεγάλο δίκτυο πολλών ζεύξεων. Ο χρόνος εκτέλεσης, ωστόσο, είναι πάρα πολύ μικρός σε αντίθεση με το αναμενόμενο. Παράλληλα, στην εκτέλεση 11 μεγάλος αριθμός demands διασχίζει πάλι, σχετικά μεγάλο δίκτυο, αλλά αυτή τη φορά ο χρόνος εκτέλεσης του αλγορίθμου είναι σημαντικά μεγαλύτερος.

Ενώ, λοιπόν, φαίνεται ότι ούτε τα πλήθη ακμών και ροών επηρεάζουν το χρόνο εκτέλεσης, αν εξεταστούν και τα τρία διαγράμματα συνδυαστικά με την τοπολογία του εκάστοτε δικτύου, η εικόνα αλλάζει. Στην εκτέλεση 9 για παράδειγμα, το δίκτυο έχει μεγάλο πολύ μεγάλο μέγεθος ενώ οι ροές που επιθυμούν να το προσπελάσουν είναι λίγες. Η διασπορά του απαιτούμενου bandwidth από τις ροές είναι μικρή, ενώ η αντίστοιχη διασπορά για το Capacity έχει πολύ μεγαλύτερη τιμή, με τη χωρητικότητα επίσης να παίρνει συγκριτικά μεγάλες τιμές. Το αναμενόμενο, σε τέτοια περίπτωση, θα ήταν ο χρόνος εκτέλεσης να κατέληγε σε μεγάλη τιμή, καθώς λίγες και πολύ μικρές ροές θα αναζητήσουν μονοπάτι σε μεγάλο και επαρκές δίκτυο.

Ωστόσο, κάτι που δε φαίνεται από τον πίνακα και τα διαγράμματα, είναι η ίδια η τοπολογία του δικτύου. Το δίκτυο κατασκευάστηκε με τυχαίο τρόπο, αλλά έτσι ώστε να διαιρείται σε δύο μέρη. Οι δρομολογητές του πρώτου μισού ενώνονται με τυχαίο τρόπο με τους αντίστοιχους του ίδιου μισού, ενώ οι δρομολογητές του δεύτερου μισού ενώνονται με τους αντίστοιχους του δευτέρου. Στην προκειμένη τοπολογία, για την τυχαία κατανομή των ακμών, χρησιμοποιήθηκε η συνάρτηση `rand()` η οποία τείνει να επιλέγει τη μεγαλύτερη τιμή από ένα δοθέν εύρος. Έτσι, στην τοπολογία υπάρχουν αρκετές περιπτώσεις ένας Ingress LER να ενώνεται απευθείας με κεντρικό LSR, ο οποίος με τη σειρά του επικοινωνεί με Egress LER. Ως εκ τούτου, δίδεται η δυνατότητα κατασκευής μονοπατιών με μόλις 2, 3 και 4 hops σε ένα κατά τα άλλα πολύ μεγάλο δίκτυο.

Όπως είναι λογικό, για τόσες λίγες ροές, ο αλγόριθμος θα προτιμήσει μικρά μονοπάτια μεγάλης χωρητικότητας. Παράλληλα, η πιθανότητα άνω της μίας ροής να μπορέσει να περάσει από το ίδιο μονοπάτι, είναι μεγάλη μιας και το ζητούμενο bandwidth είναι αρκετά μικρότερο από το capacity. Έτσι, από τους πρώτους ήδη κύκλους εκτέλεσης, είναι πολύ εύκολο να ευρεθούν επαρκή μονοπάτια μικρού μήκους και μεγάλης χωρητικότητας, χωρίς να χρειαστεί να γίνει αναζήτηση στο σύνολο του δικτύου. Συνεπώς, ο χρόνος εκτέλεσης, ο οποίος εξαρτάται προφανώς από την τοπολογία, είναι πάρα πολύ μικρός.

Στην εκτέλεση υπ' αριθμόν 11, πάρα πολλές ροές πολύ μικρής απαίτησης σε χώρο, αναζητούν μονοπάτια σε μεγάλο δίκτυο μέσω των τιμών χωρητικότητας.

Ήδη γίνεται εμφανές ότι ο χρόνος εκτέλεσης θα πρέπει να είναι μεγαλύτερος. Όταν οι πρώτες ροές μειώσουν σε κάποια μονοπάτια το διαθέσιμο Capacity, ο αλγόριθμος θα αναζητήσει μονοπάτια μεγαλύτερης χωρητικότητας στο υπόλοιπο δίκτυο, προκειμένου να κατανείμει την κίνηση, αυξάνοντας έτσι το χρόνο για την αναζήτηση και επιλογή του βέλτιστου LSP. Παράλληλα, αυτό που πάλι δε φαίνεται στον πίνακα, το δίκτυο είναι πιο κατανεμημένο από ότι το προηγούμενο. Με άλλα λόγια, δεν υπάρχουν διαδρομές 2 και 3 hops που να ενώνουν τα άκρα του. Ως εκ τούτου, οι διαδρομές θα είναι μεγαλύτερες και η αναζήτηση σχετικά πιο δύσκολη.

Στην τελευταία εκτέλεση, υπ' αριθμόν 14, λίγες ροές με μεγάλες απαιτήσεις σε Bandwidth απαιτούνται χρήση πολύ μεγάλο δικτύου με μεγάλη χωρητικότητα και μεγάλη διασπορά στις τιμές της τελευταίας. Το δίκτυο σχεδιάστηκε με τυχαίο τρόπο, αλλά έτσι ώστε να είναι πλήρως κατανεμημένο. Δηλαδή, στη χειρότερη περίπτωση, ένα μονοπάτι από άκρο σε άκρο του δικτύου, μπορεί να διασχίσει τους μισούς και περισσότερους δρομολογητές.

Ήδη από την αρχή του, ο αλγόριθμος κάνει αρκετές αναζητήσεις για να βρει το βέλτιστο μονοπάτι, καθ' ότι το δίκτυο είναι πολύ μεγάλο και τέλεια κατανεμημένο. Αργότερα, δε, τα πράγματα δυσκολεύουν. Οι πρώτες ροές που θα εξυπηρετηθούν είναι οι μεγαλύτερες. Έτσι, θα αλλάξουν σημαντικά την ελεύθερη χωρητικότητα των ζεύξεων, με αποτέλεσμα σε κάθε κύκλο οι αναζητήσεις να αλλάζουν κατεύθυνση. Όσο το δίκτυο «γεμίζει» με demands, τόσο η ελεύθερη χωρητικότητα μειώνεται. Γίνεται προφανές ότι μία μεγάλη ροή σε κάποια στιγμή δε θα καταφέρει να βρει ένα βέλτιστο μονοπάτι για να εξυπηρετηθεί, αλλά θα διαιρεθεί σε δύο και περισσότερα LSPs. Έτσι, ενώ ο αλγόριθμος εκκινεί με λίγες, μεγάλες ροές, καταλήγει τελικά να έχει πολλά μικρότερα τμήματα των αρχικών demands, κάθε ένα από τα οποία αιτείται βέλτιστου μονοπατιού το οποίο να κατανείμει την κίνηση χωρικά πάνω στην τοπολογία. Το τελευταίο, μαζί με την ίδια την κατανεμημένη και μεγάλη τοπολογία του δικτύου, δικαιολογούν πλήρως το μεγάλο χρόνο εκτέλεσης ο οποίος φτάνει στα 123 δευτερόλεπτα.

Από όλα τα παραπάνω στοιχεία, γίνεται εμφανές ότι ο χρόνος εκτέλεσης του αλγορίθμου εξαρτάται πρώτα από την τοπολογία του δικτύου (δηλαδή από

την κατανομή των ακμών ανάμεσα στους δρομολογητές) και έπειτα από τα χαρακτηριστικά των ροών και των ζεύξεων. Όσο πιο κατανεμημένο το δίκτυο, τόσο μεγαλύτερη η δυσκολία του αλγορίθμου στο να κατανείμει με βέλτιστο τρόπο τις ροές. Αντίθετα, όσο πιο συμπυκνωμένες οι ζεύξεις ανάμεσα στους δρομολογητές, τόσο πιο εύκολο είναι να ευρεθούν οι κατάλληλες διαδρομές προκειμένου οι απαιτήσεις να κατανέμονται εντός της τοπολογίας βέλτιστα από άποψη κίνησης και πόρων. Αξίζει, δε, να τονιστεί ότι ακόμα και σε μία τόσο δύσκολη τοπολογία με λίγες αλλά μεγάλες ροές όπως αυτή της 14^{ης} εκτέλεσης, ο χρόνος των 123^{ωv} δευτερολέπτων είναι σχεδόν μηδενικός εν συγκρίσει με τον αντίστοιχους χρόνους που μετρήθηκαν στο [2]. Πιο συγκεκριμένα, για ένα δίκτυο 200 ακμών στο [2] απαιτήθηκαν αρκετές ώρες εκτέλεσης για την εύρεση της καλύτερης κατανομής του φορτίου. Ο προτεινόμενος αλγόριθμος που εξετάζεται, δε, κατάφερε να κατανείμει το φορτίο σε δίκτυο 197 τέλεια κατανεμημένων ζεύξεων σε μόλις 123 δευτερόλεπτα. Γίνεται προφανές, λοιπόν, ότι τουλάχιστον από άποψη ταχύτητας ο προκείμενος αλγόριθμος είναι επιτυχής.

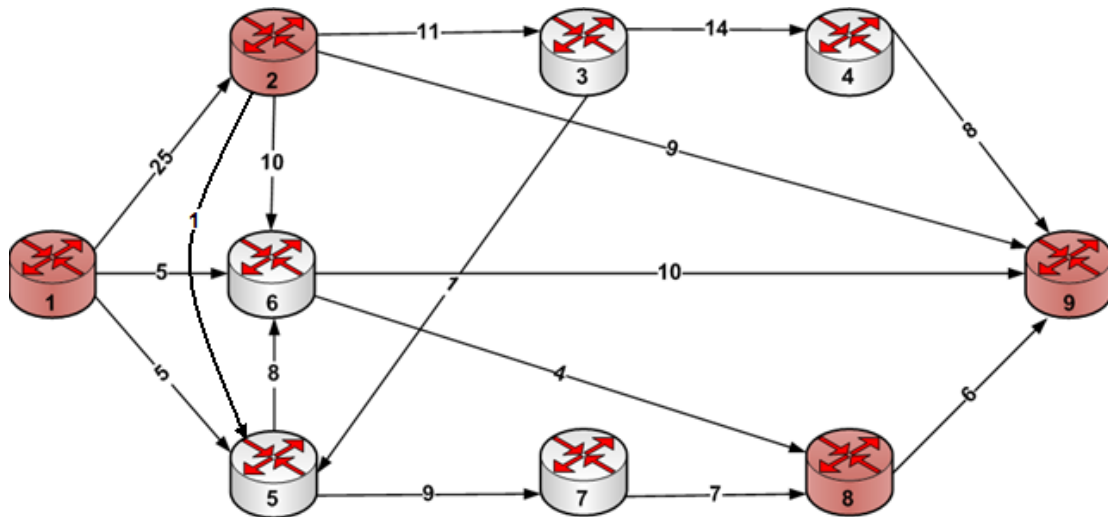
ΚΑΤΑΝΟΜΗ ΦΟΡΤΙΟΥ

Στην παρούσα ενότητα θα παρουσιαστεί ο τρόπος με τον οποίο κατανέμεται το φορτίο στις ζεύξεις του εκάστοτε δικτύου. Από τις τοπολογίες στις οποίες εφαρμόστηκε και εξετάστηκε ο προκείμενος αλγόριθμος, επιλέχθησαν δύο : ένα μικρό, ασύμμετρο δίκτυο με πλήθος και μεγέθη ροών κοντά στην χωρητικότητα των ζεύξεων, καθώς και ένα μεγάλο, συμμετρικό δίκτυο με μικρό πλήθος ροών μικρού μεγέθους.

ΑΠΛΟ ΔΙΚΤΥΟ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ

Το δίκτυο της απλής, σχετικά, εφαρμογής αποτελείται από 9 δρομολογητές εκ των οποίων δύο παίζουν το ρόλο των Ingress LERs και άλλοι δύο των Egress. Οι LSRs συνδέονται μεταξύ τους με συνολικά 16 ζεύξεις από τις

οποίες η ζεύξη με τη μικρότερη χωρητικότητα έχει 1 μονάδα Capacity, η αντίστοιχη με τη μέγιστη χωρητικότητα 25 μονάδες ενώ ο μέσος όρος χωρητικοτήτων υπολογίζεται στις 15,9375 μονάδες.



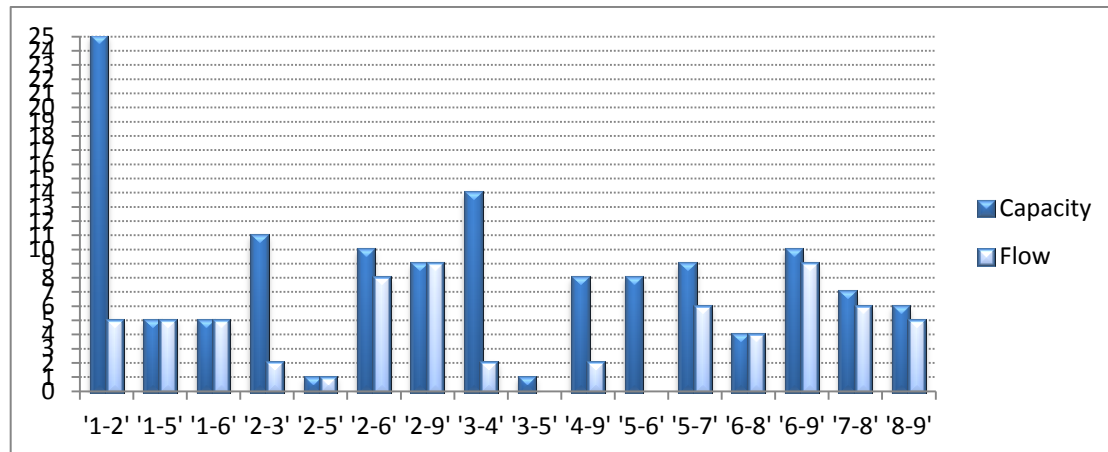
Εικόνα 21. Απλό Δίκτυο Μικρού Μεγέθους

Τη διάσχιση του συγκεκριμένου δικτύου αιτούνται 6 ροές. Τρεις από αυτές εκκινούν από τον έναν εκ των δύο Ingress LERs και οι άλλες τρεις από τον άλλον. Αντίστοιχα, τέσσερις από αυτές καταλήγουν στον έναν Egress LER και δύο στον άλλο. Το συνολικό απαιτούμενο Bandwidth υπολογίζεται στις 25 μονάδες. Ωστόσο, δύο από τις ροές αιτούνται Backup routes, οπότε το τελικό απαιτούμενο Bandwidth για το συγκεκριμένο δίκτυο ανέρχεται στις 34 μονάδες. Τέλος, μία από την ομάδα ροών, αιτείται υποχρεωτικής διάσχισης από ένα συγκεκριμένο link (7-8). Άλλες απαιτήσεις, όπως μέγιστο μήκος μονοπατιού, μέγιστο πλήθος παράλληλων διαδρομών κτλ δεν λαμβάνουν χώρα στο συγκεκριμένο πλαίσιο εφαρμογής.

Demand ID	1	2	3	4	5	6
Source	2	2	2	1	1	1
Destination	9	9	8	9	9	8
Priority Level	7	7	7	7	7	7
Bandwidth	4	2	5	5	5	4
MaxHops	0	0	0	0	0	0
Hops Deviation	0	0	0	0	0	0
Needs Backup	1	0	0	1	0	0
Max Cuts	0	0	0	0	0	0

Πίνακας 2. Ιδιότητες και Χαρακτηριστικά των Ροών

Τα παραπάνω στοιχεία αποτελούν δεδομένα εισόδου για τον αλγόριθμο ο οποίος υπολογίζει τα καλύτερα μονοπάτια για κάθε μία ροή, σύμφωνα με τις προδιαγραφές του MPLS – TE και κατανέμει την κίνηση και τους πόρους εντός της δοθείσας τοπολογίας. Το αποτέλεσμα της κατανομής φαίνεται το διάγραμμα που ακολουθεί.



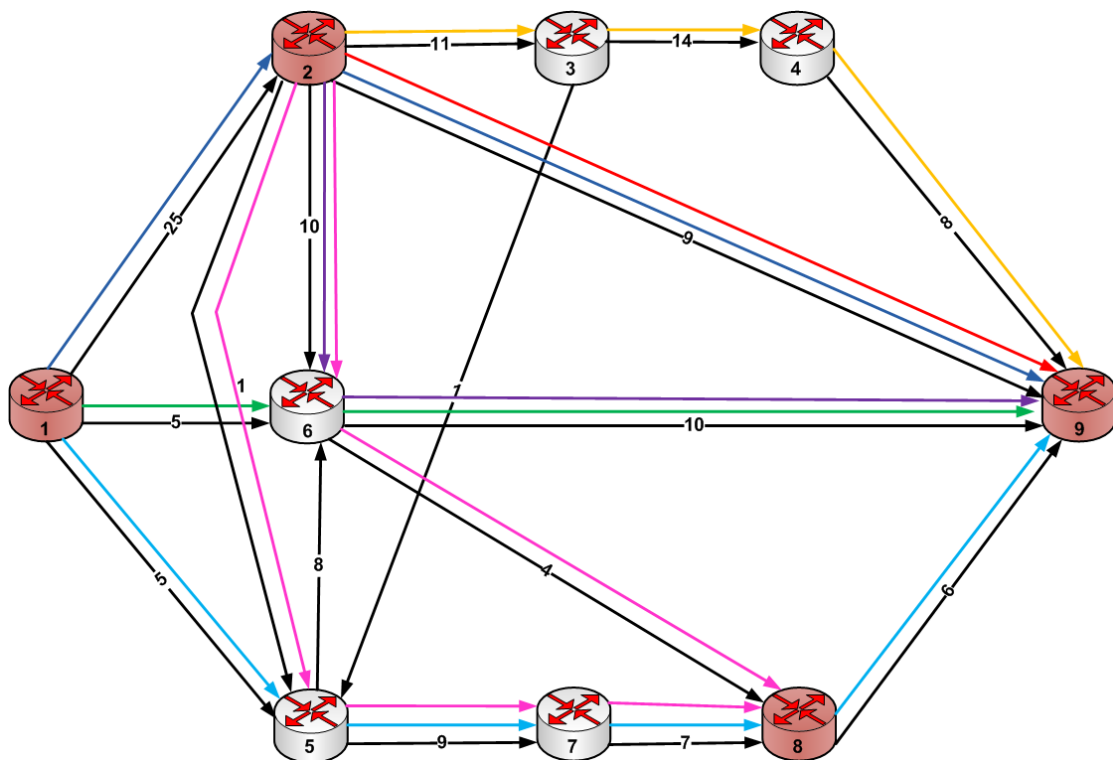
Διάγραμμα 4. Κατανομή Φορτίου στο Μικρό Δίκτυο

Ο άξονας Y διατηρεί την σχετική με την χωρητικότητα πληροφορία, ενώ ο X παρουσιάζει όλες τις ακμές της τοπολογίας. Με μπλε χρώμα εμφανίζεται η αρχική χωρητικότητα της κάθε ζεύξης, ενώ με γαλάζιο η δεσμευμένη χωρητικότητα από τις ροές τις οποίες εξυπηρέτησε και κατάνειμε ο αλγόριθμος. Όπως φαίνεται και από το διάγραμμα, οι 10 από τις 16 ακμές έχουν ποσοστό δεσμευμένης χωρητικότητας μεταξύ 66,667% και 100%, 4 ακμές κυμαίνονται ανάμεσα σε 14% και 25% και δύο δεν χρησιμοποιούνται από τη συγκεκριμένη ομάδα demands. Ο μέσος όρος του ποσοστού δέσμευσης συνολικά στο δίκτυο υπολογίζεται στα 61,44886%.

Δεδομένου ότι ο αλγόριθμος προσπαθεί να κατανείμει βέλτιστα την κίνηση και τους πόρους στο δίκτυο, θα περίμενε κανείς να μην υπάρχουν ακμές με ακραία ποσοστά δέσμευσης στα 0 και 100%, αλλά αντίθετα όλες οι ακμές να περιλαμβάνουν μέσες αναλογίες δεσμευμένης προς αρχικής αναλογίας. Ωστόσο, δεδομένου ότι

- Το δίκτυο είναι ασύμμετρο
- Οι χωρητικότητες των ακμών είναι διαφορετικές μεταξύ τους
- Οι ροές εκκινούν και τερματίζουν σε διαφορετικούς δρομολογητές
- Οι ροές έχουν διαφορετικές τιμές απαιτούμενου Bandwidth
- Οι ροές έχουν ποικίλες απαιτήσεις σε θέματα τοπολογίας, όπως προσδιορισμοί ζεύξεων/μονοπατιών και δευτερευόντων διαδρομών

προφανές γίνεται ότι δεν μπορεί ολόκληρη η τοπολογία να περιλαμβάνει ίσα ποσοστά δέσμευσης με το φορτίο να κατανέμεται πλήρως ομοιόμορφα. Παρακάτω, παρουσιάζεται μία εικόνα της τελικής μορφής της κατανομής της κίνησης εντός του δικτύου, η οποία μπορεί να ρίξει περισσότερο φως στον τρόπο της κατανομής και στο εάν τελικά είναι ο βέλτιστος ή όχι, στο συγκεκριμένο πλαίσιο εφαρμογής:



Εικόνα 22. Κατανομή Κίνησης στο Μικρό Δίκτυο

Οι μαύρες, αριθμημένες γραμμές της τοπολογίας απεικονίζουν τις πραγματικές ζεύξεις του δικτύου καθώς και την αρχική τους χωρητικότητα. Οι έγχρωμες ακμές δείχνουν το μονοπάτι κάθε ροής. Πιο συγκεκριμένα και σε

σειρά εξυπηρέτησης της κάθε μίας, τα μονοπάτια και οι ροές για τις οποίες αυτά έχουν δεσμευτεί έχουν ως εξής:

Ροή	Χρώμα	Bandwidth	Μονοπάτι
4, Primary (1 μονοπάτι)	Μπλε	5	1-2-9
4, Backup (1 μονοπάτι)	Πράσινο	5	1-6-9
1, Primary (1 μονοπάτι)	Κόκκινο	4	2-9
1, Backup (1 μονοπάτι)	Μωβ	4	2-6-9
5, Primary (1 μονοπάτι)	Γαλάζιο	5	1-5-7-8-9
3, Primary (1 ^ο μονοπάτι)	Ροζ	4	2-6-8
3, Primary (2 ^ο μονοπάτι)	Ροζ	1	2-5-7-8
2, Primary (1 μονοπάτι)	Κίτρινο	2	2-3-4-9

Πίνακας 3. Εξυπηρέτηση Ροών στο Μικρό Δίκτυο

Όπως φαίνεται και από τον πίνακα, όλες οι απαιτήσεις του MPLS –TE πληρούνται: οι ροές που αιτούνται Backup route έχουν εξυπηρετηθεί με το δευτερεύον μονοπάτι να αποφεύγει τις ακμές του πρωτεύοντος. Παράλληλα, η ροή η οποία απαιτούσε τη διάσχιση της ακμής [7 – 8] έχει λάβει ένα μονοπάτι το οποίο περνάει από το link αυτό. Το προκείμενο Demand, με DemandID = 5, θα μπορούσε τη στιγμή που υπολογιζόταν το μονοπάτι του, να λάβει ένα πολύ καλύτερο μονοπάτι με μεγαλύτερη χωρητικότητα, πράγμα που θα συντελούσε σε καλύτερη κατανομή του φορτίου: το [1 – 2 – 3 – 4 – 9], με minimum residual capacity να ισούται με 8 και maximum με 25. Ωστόσο, επειδή έχει ορίσει ότι πρέπει να περάσει από συγκεκριμένη ακμή, του ανατέθηκε τελικά το μονοπάτι [1 – 5 – 7 – 8 – 9] με ελάχιστη ελεύθερη χωρητικότητα (για τη στιγμή εκείνη) ίση με 5 και μέγιστη με 9.

Παράλληλα, παρατηρείται ότι η ροή υπ’ αριθμόν 6 δεν κατάφερε να εξυπηρετηθεί. Ο αλγόριθμος την έβαλε στην προτελευταία θέση της σειράς εξυπηρέτησης, προκειμένου να εξυπηρετήσει πρώτα τις πιο μεγάλες και πιο απαιτητικές ροές, με αποτέλεσμα όταν έφτασε η στιγμή για την 6 δεν υπήρχε αρκετή ελεύθερη χωρητικότητα στο δίκτυο για το συγκεκριμένο Demand με τα συγκεκριμένα χαρακτηριστικά πηγής και προορισμού. Έτσι, μία ροή μεγέθους 4 μονάδων Bandwidth, έμεινε «απ’ έξω», προκειμένου ροές οι οποίες θα δώσουν περισσότερο κέρδος στον ISP του δικτύου και καλύτερη ποιότητα στον τελικό χρήστη, να εξυπηρετηθούν.

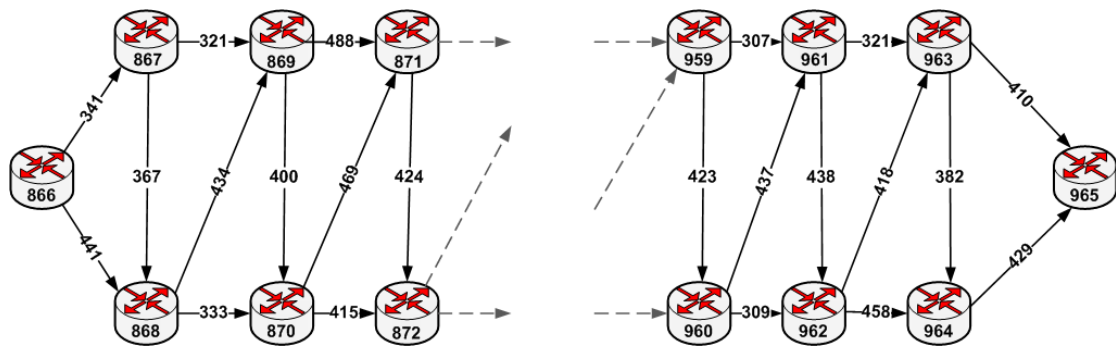
Λόγοι όπως οι παραπάνω, δηλαδή αποφυγή ή απαιτούμενη προσπέραση ακμών ή ολόκληρων μονοπατιών, καθώς και η ταξινόμηση των ροών με τρόπο τέτοιο ώστε να αυξάνεται η χρησιμοποίηση του δικτύου και, συνεπαγόμενα, το κέρδος του ISP και η ποιότητα του χρήστη, συντελούν σε σχετικά χειρότερη κατανομή του φορτίου εντός της τοπολογίας. Για το λόγο αυτό, ενώ θα μπορούσαν να είχαν επιλεχθεί διαφορετικά μονοπάτια και η κατανομή να είναι η βέλτιστη, τελικά παρατηρείται ότι η κατανομή είναι «η καλύτερη δυνατή, δεδομένων των συνθηκών, των απαιτήσεων και των αποφάσεων».

ΣΥΜΜΕΤΡΙΚΟ ΔΙΚΤΥΟ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ

Η τοπολογία της χειρίστης περίπτωσης, από αυτές στις οποίες εφαρμόστηκε ο αλγόριθμος, αποτελείται από 100 δρομολογητές εκ των οποίων 10 αποτελούν τους Ingress LERs και άλλοι 10 τους Egress. Οι LSRs συνδέονται μεταξύ τους με 197 ακμές εκ των οποίων η ακμή με τη μικρότερη χωρητικότητα έχει 302 μονάδες Capacity και η αντίστοιχη με τη μεγαλύτερη, 499.

10 ροές μεγέθους από 10 έως 100 μονάδες αιτούνται τη διάσχιση της τοπολογίας, με την κάθε μία να εκκινεί και να τερματίζει σε μοναδικά ζεύγη Πηγής και Προορισμού. Με άλλα λόγια, υπάρχει μόλις μία ροή η οποία να έχει SourceX και μόλις μία ροή με DestinationY. Παράλληλα, καμία από τις ροές δεν έχει απαιτήσεις ως προς δευτερεύοντα μονοπάτια, αριθμό Hops και ούτω καθ' εξής. Σύμφωνα, δηλαδή, με τις προδιαγραφές του MPLS – TE, είναι απλές ροές, χωρίς ιδιαίτερους περιορισμούς.

Η τοπολογία του δικτύου φαίνεται στην [Εικόνα 23](#). Για προφανείς λόγους μεγέθους του δικτύου, εμφανίζονται μόνο οι αρχικοί και οι τελικοί κόμβοι, καθαρά για να γίνει αντιληπτό το «σχήμα» του δικτύου. Με κόκκινο χρώμα συμβολίζονται κάποιοι από τους LERs προκειμένου να γίνει εμφανής η σχετική τους θέση στο δίκτυο.



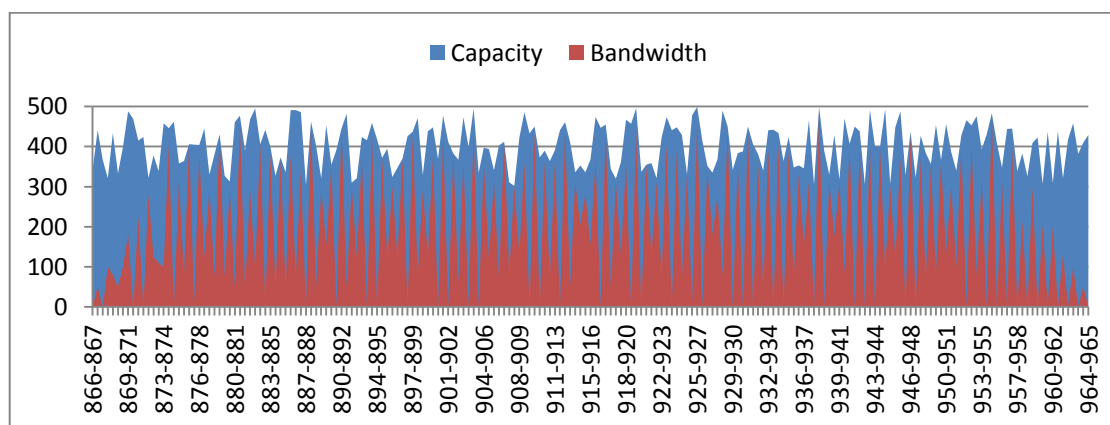
Εικόνα 23. Δίκτυο Μεγάλου Μεγέθους

Τη χρήση της παραπάνω τοπολογίας αιτούνται ροές, τα χαρακτηριστικά των οποίων παρουσιάζονται στον [Πίνακα 4](#).

Demand ID	661	662	663	664	665	666	667	668	669	670
Source	866	867	868	869	870	871	872	873	874	875
Destination	965	964	963	962	961	960	959	958	957	956
Bandwidth	50	100	80	100	75	100	10	100	100	80

Πίνακας 4. Ιδιότητες και Χαρακτηριστικά των Ροών

Η εφαρμογή του αλγορίθμου στην παραπάνω τοπολογία και στα αναφερθέντα Demands έχει σαν αποτέλεσμα την κατανομή του φορτίου με τον τρόπο που παρουσιάζεται στο [Διάγραμμα 5](#).



Διάγραμμα 5. Κατανομή Φορτίου στο Μεγάλο Δίκτυο

Όπως δείχνει το παραπάνω διάγραμμα, η δέσμευση των πόρων είναι σχετικά μικρότερη στα άκρα. Ο λόγος για αυτό είναι τα μοναδικά ζεύγη πηγών – προορισμού, όπως αναφέρθηκε και πιο πάνω. Η τοποθεσία των LERs είναι τέτοια ώστε: οι 10 Ingress LERs βρίσκονται στην αρχή της τοπολογίας που φαίνεται στην [Εικόνα 23](#), ενώ οι Egress LERs στο τέλος. Επειδή το διάγραμμα παρουσιάζει την κατανομή των πόρων πάνω στις ακμές οι οποίες είναι ταξινομημένες βάσει του ονόματός τους (άξονας X), στην αρχή και στο τέλος του διαγράμματος βρίσκονται οι Ingress και οι Egress LERs αντίστοιχα. Και αφού το κάθε ζεύγος χρησιμοποιείται από ακριβώς μία ροή, προφανές γίνεται ότι το φορτίο στα σημεία αυτά θα είναι κοντά στο απαιτούμενο Bandwidth της ροής που αιτείται προσπέλασης ανάμεσα στο συγκεκριμένο ζεύγος και όχι μεγαλύτερο.

Από τις 197 ακμές που λαμβάνουν χώρα στην προκείμενη τοπολογία, οι 46 παραμένουν αχρησιμοποίητες από τις ροές. Οι περισσότερες από τις ζεύξεις αυτές είναι διαγώνιες είναι μικρές σε σχέση με τις υπόλοιπες. Δεδομένου ότι οι ροές δεν έχουν απαιτήσεις ζεύξεων, ο αλγόριθμος επιλέγει τα βέλτιστα μονοπάτια με κριτήρια τέτοια ώστε το καλύτερο, από ένα δοθέν σύνολο εφικτών μονοπατιών, να έχει τη μεγαλύτερη ελεύθερη χωρητικότητα και το μικρότερο αριθμό Hop από την πηγή έως τον προορισμό. Έτσι, γίνεται προφανές ότι ζεύξεις οι οποίες, σε σχέση με τις υπόλοιπες, έχουν μικρότερο μέγεθος ελεύθερης χωρητικότητας ή αυξάνουν το μήκος του τελικού μονοπατιού, δεν βελτιστοποιούν την Αντικειμενική Συνάρτηση και αποφεύγονται κατά την εύρεση της βέλτιστης λύσης.

Από τις ακμές μη μηδενικής χρησιμοποίησης, 4 δεσμεύονται στο 100% τους ενώ οι υπόλοιπες λαμβάνουν ποσοστά δέσμευσης ανάμεσα στα 2,09205 και 99,75369%. Εάν και τα μικρότερα και μεγαλύτερα ποσοστά έχουν ακραίες τιμές, αξίζει να αναφερθεί πως ο μέσος όρος των μη μηδενικών και μη εξ' ολοκλήρου δεσμευμένων ακμών υπολογίζεται στα 61,85875%, ενώ ο αντίστοιχος συνολικός μέσος όρος δέσμευσης για όλη την τοπολογία υπολογίζεται στα 48,18902%. Η κατανομή του φορτίου των ακμών ανάμεσα στα μη ακραία ποσοστά έχει ως εξής: 64 ακμές βρίσκονται στο πεδίο μεταξύ 2,09205 και 61,85875%, ενώ 83 μεταξύ 64,7492 και 99,75369% με τις μισές από το δεύτερο σύνολο να δεσμεύουν λιγότερο από το 92% των πόρων.

Ροή	Bandwidth	Μονοπάτι
666	100	[871-873-875-877-879-881-883-885-887-889-891-893-895-897-899-901-903-905-907-909-911-913-915-917-919-921-923-925-927-928-930-932-934-936-938-940-942-944-946-948-950-952-954-956-958-960]
662	100	[867-869-871-873-875-877-879-881-883-885-887-889-891-893-895-897-899-901-903-905-907-909-911-913-915-917-919-921-923-925-927-929-931-933-935-937-939-940-942-944-946-948-950-952-954-956-958-960-962-964]
669	100	[874-876-878-880-882-883-885-887-889-891-893-895-897-899-901-903-905-907-909-911-913-915-917-919-921-923-925-927-929-931-933-935-937-939-941-943-945-947-949-951-953-955-957]
664	100	[869-870-872-874-876-878-880-882-884-886-888-890-892-894-896-898-900-902-904-906-908-910-912-914-916-918-920-922-924-926-928-930-932-934-936-938-940-942-944-946-948-950-952-954-956-958-960-962]
668	100	[873-874-876-878-880-882-884-886-888-890-892-894-896-898-900-902-904-906-908-910-912-914-916-918-920-922-924-926-928-930-932-934-936-938-940-942-944-945-947-949-951-952-954-956-958]
670	80	[875-876-878-880-882-884-886-888-890-892-894-896-898-900-902-904-906-908-910-912-914-916-918-920-922-924-926-928-929-931-933-935-937-939-941-943-945-947-949-951-953-954-956]
663	80	[868-869-871-873-875-877-879-881-883-885-887-889-890-892-894-896-898-899-900-902-904-906-908-910-912-914-915-916-918-919-921-922-923-925-927-928-930-932-934-936-937-939-941-943-945-947-949-951-953-955-957-959-961-693]
665	75	[870-872-873-875-877-878-879-880-882-884-886-887-889-890-892-893-895-896-897-899-901-903-905-906-907-908-909-911-912-914-915-916-918-920-922-924-925-927-929-931-933-935-936-937-939-940-941-943-945-946-948-949-950-951-953-955-957-959-961]
661	50	[866-868-870-872-873-875-877-878-880-881-882-884-885-886-888-889-891-892-893-895-896-897-899-900-902-903-905-906-908-909-911-913-914-915-917-918-919-921-922-924-926-928-930-932-933-935-937-939-941-943-945-946-948-950-951-953-955-957-959-961-963-965]
667	10	[872-874-876-877-879-881-883-884-886-888-890-892-894-896-898-900-902-903-905-907-908-909-911-913-915-917-919-921-923-925-926-928-930-932-934-936-938-940-941-943-945-947-949-950-951-953-955-957-959]

Πίνακας 5. Εξυπηρέτηση Ροών στο Μεγάλο Δίκτυο

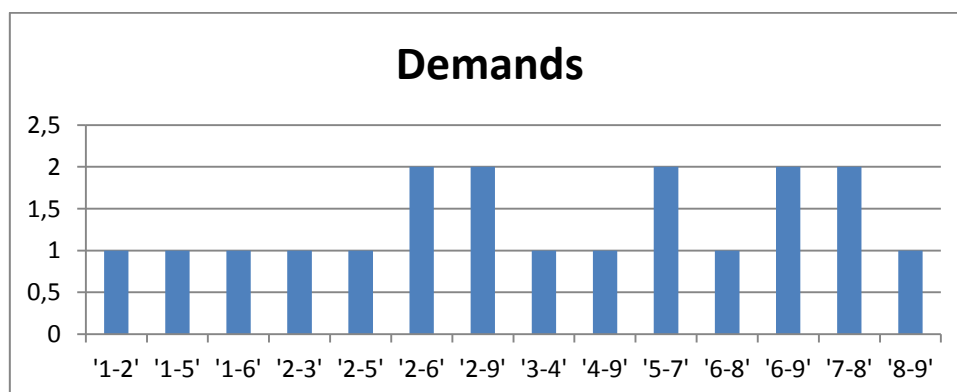
Όπως δείχνει και ο [Πίνακας 5](#), όλες οι ροές εξυπηρετούνται, με σειρά προτεραιότητας βασισμένη στο μέγεθος της απαίτησης σε Bandwidth. Τα Hops του κάθε μονοπατιού κυμαίνονται μεταξύ 42 και 61, περνώντας από το 21 και 31% των ακμών της τοπολογίας, αντίστοιχα.

ΚΑΤΑΝΟΜΗ ΚΙΝΗΣΗΣ (ΣΕ ΣΧΕΣΗ ΜΕ ΠΑΡΕΜΒΟΛΗ)

Η προκείμενη ενότητα έχει σαν στόχο την εξέταση, για τις ίδιες τοπολογίες που αναλύθηκαν προηγουμένως, του τρόπου με τον οποίο κατανέμονται οι ροές όχι ως προς το μέγεθος της δεσμευμένης χωρητικότητάς τους, αλλά ως προς το πλήθος των ροών που περνάνε από κάθε ακμή.

ΑΠΛΟ ΔΙΚΤΥΟ ΜΙΚΡΟΥ ΜΕΓΕΘΟΥΣ

Όπως δείχνει και το [Διάγραμμα 6](#), από το οποίο έχουν αφαιρεθεί οι ακμές μηδενικής χρησιμοποίησης, η κατανομή των ροών λαμβάνει τιμές από 1 έως 2 ροές ανά ζεύξη. Το σύνολο των ροών της συγκεκριμένης ομάδας υπολογίζεται στις 6 ροές, οι οποίες μαζί με τις δευτερεύουσες φτάνουν στις 8. Από τα 8 μονοπάτια προς υπολογισμό, ένα μένει «απ' έξω» λόγω αδυναμίας εύρεσης εφικτής διαδρομής (-1), ενώ μία ροή διακλαδίζεται σε δύο τμήματα (+1). Έτσι, τελικά, το πλήθος των υπολογισμένων μονοπατιών που διασχίζουν το δίκτυο φτάνει τα 8.



Διάγραμμα 6. Κατανομή Κίνησης στο Μικρό Δίκτυο

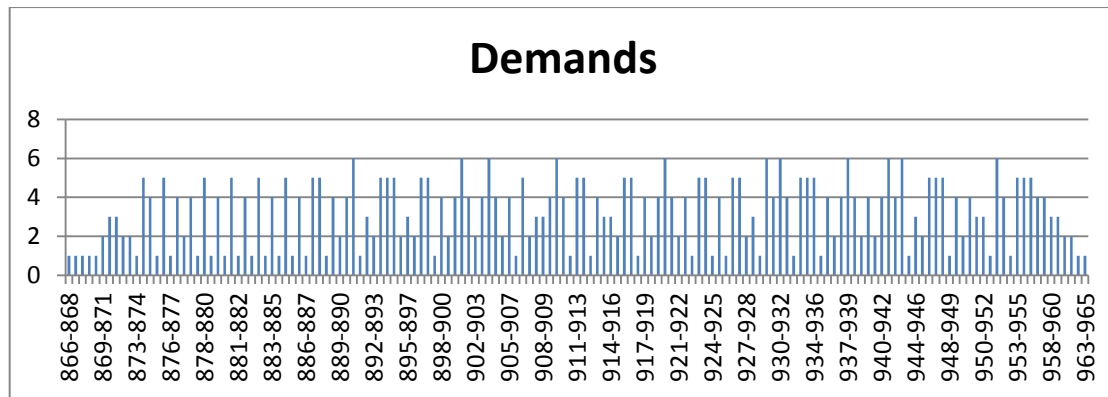
Για τα 8 αυτά μονοπάτια, λοιπόν, 9 από τις 14 χρησιμοποιούμενες ακμές της τοπολογίας χρησιμοποιούνται αποκλειστικά από μία μόλις ροή, ενώ τα υπόλοιπα 5 μοιράζονται από δύο ροές. Ως αποτέλεσμα, το 64,3% της χρησιμοποιούμενης τοπολογίας έχει «αποκλειστικότητα» για τις ροές, ενώ το υπόλοιπο 35,7% χρησιμοποιείται από μόλις 2 διαφορετικές ροές ταυτόχρονα. Με άλλα λόγια, οι ροές που διαμοιράζονται κοινά links είναι, συγκριτικά, λίγες μειώνοντας έτσι την παρεμβολή των διαδρομών από ξένα, μεταξύ τους, πακέτα.

ΣΥΜΜΕΤΡΙΚΟ ΔΙΚΤΥΟ ΜΕΓΑΛΟΥ ΜΕΓΕΘΟΥΣ

Στο μεγάλο, συμμετρικό σχετικά δίκτυο, που παρουσιάστηκε παραπάνω, η εικόνα είναι λίγο διαφορετική. Οι ροές που αιτούνται χρήσης του δικτύου ανέρχονται στις 10, ενώ ταυτόχρονα δεν έχουν άλλες απαιτήσεις με αποτέλεσμα να υπολογίζονται ακριβώς 10 μονοπάτια.

Όσον αφορά στο βαθμό παρεμβολής των διαδρομών ανά ζεύξεις, τα αποτελέσματα έχουν ως εξής:

- 36 ακμές χρησιμοποιούνται από 1 ροή αποκλειστικά
- 23 ακμές χρησιμοποιούνται από 2 ροές
- 14 ακμές χρησιμοποιούνται από 3 ροές
- 36 ακμές χρησιμοποιούνται από 4 ροές
- 31 ακμές χρησιμοποιούνται από 5 ροές
- 11 ακμές χρησιμοποιούνται από 6 ροές



Διάγραμμα 7. Κατανομή Κίνησης στο Μεγάλο Δίκτυο

Σχεδόν οι μισές από τις χρησιμοποιούμενες ακμές διασχίζονται από 1 έως 3 διαφορετικές ροές και οι υπόλοιπες από 4 έως 6. Όπως είναι λογικό για ροές που έχουν μοναδικά ζεύγη πηγής – προορισμού, τα πρώτα και τα τελευταία linkστης τοπολογίας διασχίζονται αποκλειστικά από ένα μονοπάτι. Παράλληλα, δεδομένου ότι οι περισσότερες από τις διαγώνιες ακμές δε χρησιμοποιούνται από τον αλγόριθμο για λόγους ελαχιστοποίησης των hops, η κίνηση επικεντρώνεται στα παράλληλα μονοπάτια, όπως αυτά προκύπτουν και από την [Εικόνα 23](#). Έτσι, προφανές γίνεται ότι οι ζεύξεις αυτές θα υποστηρίζουν περισσότερες από μία ροές με αποτέλεσμα τα νούμερα που υπολογίστηκαν παραπάνω να δικαιολογούνται.

Τέλος, οι ακμές με τη μεγαλύτερη παρεμβολή είναι μόλις 11. Σε σύγκριση με το συνολικό δίκτυο των 197 ακμών και ακόμα και με το χρησιμοποιούμενο δίκτυο των 151, το ποσοστό ακμών με τη μέγιστη παρεμβολή φτάνει τα 5,58 και 7,28% αντίστοιχα. Με άλλα λόγια, το 5,58% των συνολικών ακμών του δικτύου μεταφέρει το πολύ 60% των ξεχωριστών, μεταξύ τους, ροών. Την ίδια ώρα, το ένα τρίτο ολόκληρης της τοπολογίας έχει παρεμβολή στις ζεύξεις από μόλις 1 με 2 ροές. Από τους παραπάνω υπολογισμούς, προκύπτει ότι στο σύνολο ο βαθμός παρεμβολής στο δίκτυο είναι, γενικά, μικρός με μέγιστο τις 6 ροές ανά ζεύξη σε πολύ μικρό τμήμα του δικτύου με τις τελευταίες να είναι απομακρυσμένες, γεωγραφικά, μεταξύ τους.

ΕΠΙΣΚΟΠΗΣΗ

ΣΥΜΠΕΡΑΣΜΑΤΑ

Στα προηγούμενα κεφάλαια παρουσιάστηκε το θεωρητικό υπόβαθρο των MPLS δικτύων καθώς και οι προδιαγραφές τους στο πεδίο του Traffic Engineering. Εξετάστηκε ο τρόπος λειτουργίας των δικτύων αυτών, ενώ αναλύθηκαν ενδελεχώς οι απαιτήσεις που μπορούν ή/ και πρέπει να πληρούνται κατά τη δρομολόγηση των ροών διαμέσου ενός οποιουδήποτε MPLS τομέα. Εν συνεχεία, περιγράφηκε ο στόχος του προκείμενου ερευνητικού έργου, ο οποίος έγκειται στο σχεδιασμό και την κατασκευή αλγορίθμου ο οποίος θα κατανέμει τους πόρους και την κίνηση σε MPLS δίκτυα με τρόπο τέτοιο ώστε να επιτυγχάνεται η Μέγιστη Διέλευση τηρώντας τις προδιαγραφές του MPLS – TE.

Παράλληλα, εξετάστηκαν διάφοροι υπάρχοντες αλγόριθμοι οι οποίοι επιλύουν το MaxFlow πρόβλημα βελτιστοποίησης υπολογίζοντας το μέγιστο φορτίο το οποίο μπορεί να διασχίσει ένα δίκτυο μεταφοράς. Από τους αναφερθέντες, ξεχωρίστηκε ως καλύτερος ο αλγόριθμος του Dinic, τα βήματα του οποίου αποτέλεσαν πηγή έμπνευσης για τον TiReD αλγόριθμο.

Ο TiReD αλγόριθμος, όπως ήδη τονίστηκε, έχει σαν στόχο τη μεγιστοποίηση της κίνησης στο δίκτυο, με τρόπο τέτοιο ώστε η κίνηση και οι πόροι να κατανέμονται βέλτιστα ενώ παράλληλα να πληρούνται οι απαιτήσεις και οι προδιαγραφές που εισηχθήκαν στο MPLS – TE. Έχει σχεδιαστεί έτσι ώστε η σημασία που δίδεται σε κάθε ένα τμήμα του τελικού στόχου να είναι το ίδιο μεγάλη και να μη δίνεται προτίμηση σε κάποιο από τα τρία αυτά τμήματα.

Πιο συγκεκριμένα, όπως φάνηκε και από το Πέμπτο Κεφάλαιο, οι τρεις υπο-στόχοι είναι άρρηκτα συνδεδεμένοι. Στο πρώτο στάδιο του αλγορίθμου, οι ροές ταξινομούνται βάσει Επιπέδου Προτεραιότητας, τελικής απαιτούμενης χωρητικότητας και απαιτήσεων MPLS – TE. Ως αποτέλεσμα, παρέχεται η εγγύηση ότι οι πρώτες ροές που δύναται να τύχουν εξυπηρέτησης, είναι αυτές που έχουν τη μέγιστη προτεραιότητα στον MPLS τομέα, τη μέγιστη απαιτούμενη χωρητικότητα και τις περισσότερες και πιο ανελαστικές απαιτήσεις ακμών και διαδρομών. Με άλλα λόγια, εξετάζονται πρώτα εκείνες

οι απαιτήσεις οι οποίες θα αυξήσουν όσο το δυνατόν περισσότερο τη διέλευση στο δίκτυο και έχουν μεγαλύτερη σημασία, βάσει Priority Level, από τις υπόλοιπες. Έχοντας, παράλληλα, μεγαλύτερες απαιτήσεις διαδρομών και ακμών από τις άλλες, παρέχεται και μια σχετική εγγύηση ότι θα βρουν μονοπάτι, παρ' ότι περιορίζουν το εύρος των δυνατών διαδρομών, ακριβώς επειδή τη στιγμή που θα τύχουν επεξεργασίας, πιο πριν, δηλαδή, από άλλες, το δίκτυο θα έχει μεγαλύτερες πιθανότητες να προσφέρει ελεύθερα και έγκυρα μονοπάτια τα οποία δε θα έχουν ήδη δεσμευτεί από άλλες ροές. Τα παραπάνω, από μια καθαρά οικονομική σκοπιά, σημαίνουν ότι και το κέρδος των ISPs θα αυξάνεται λόγω της αυξημένης χρησιμοποίησης του δικτύου αλλά και η τελική ποιότητα που θα απολαμβάνουν οι χρήστες θα είναι η βέλτιστη, καθώς θα υπάγεται στις δικές τους προτιμήσεις και απαιτήσεις.

Έχοντας ταξινομήσει τις ροές βάσει της προτεραιότητας και της απαιτούμενης χωρητικότητάς τους, ο TiReD αλγόριθμος ξεκινά να τις επεξεργάζεται και να υπολογίζει έγκυρα μονοπάτια για την κάθε μία. Από τα μονοπάτια που ανιχνεύονται, επιλέγεται ένα βέλτιστο με τρόπο τέτοιο ώστε να γίνεται η κατανομή του φορτίου και των πόρων με τον καλύτερο δυνατό τρόπο ενώ παράλληλα να πληρούνται όλες οι απαιτήσεις της ροής που εξετάζεται τη στιγμή εκείνη. Με πιο απλά λόγια, ως «έγκυρα» θεωρούνται τα μονοπάτια τα οποία δύναται να εξυπηρετήσουν τη ροή (ή τμήμα αυτής) από θέμα χωρητικότητας και προσδιορισμένων ακμών/ διαδρομών. Ως «βέλτιστο» θεωρείται το «έγκυρο» μονοπάτι το οποίο θα διατηρήσει το φορτίο εντός του δικτύου κατανεμημένο: θα χρησιμοποιήσει, δηλαδή, ακμές τέτοιες οι οποίες εάν δρομολογηθεί φορτίο X διαμέσου αυτών, δε θα υπάρχουν υποχρησιμοποιούμενα ή υπερφορτωμένα links στο δίκτυο. Για άλλη μια φορά, λοιπόν, παρατηρείται ότι ο αλγόριθμος και κατά την εύρεση των διαδρομών, λαμβάνει υπ' όψιν του και τους τρεις υπο-στόχους που αναφέρθηκαν : εξισορροπεί το φορτίο στο δίκτυο, αυξάνει τη χρησιμοποίηση και τηρεί τις MPLS – TE απαιτήσεις των ροών.

Οι προσπάθειες που εκτελεί ο αλγόριθμος για να βρει το βέλτιστο μονοπάτι για μια ροή ή τμήμα αυτής, είναι δύο. Την πρώτη φορά ανιχνεύει να βρει μονοπάτι το οποίο να μπορεί να εξυπηρετήσει το απαιτούμενο φορτίο στο σύνολό του, ενώ εάν αποτύχει τέμνει το φορτίο σε δύο τμήματα τέτοια ώστε το

πρώτο να δρομολογηθεί στο καλύτερο δυνατό μονοπάτι ενώ το δεύτερο να υπολογιστεί αργότερα με το ίδιο ακριβώς σκεπτικό. Έτσι, ακόμη και εάν μία ροή καταλήξει να ακολουθήσει άνω της μίας διαδρομής (εφ' όσον, φυσικά, η ίδια το επιτρέπει) οι παράλληλες διαδρομές υπόκεινται στους ίδιους στόχους με αυτούς που αναφέρθηκαν παραπάνω, κατανέμοντας με ισορροπία το φορτίο, αυξάνοντας τη διέλευση και τηρώντας τις απαιτήσεις της αρχικής ροής.

Ενδιαφέρον παρουσιάζει η εισαγωγή «μνήμης» στον αλγόριθμο ο οποίος μπορεί να «θυμάται» τις προηγούμενες αποτυχίες του. Όταν, δηλαδή, αντιληφθεί ότι για δοθείσα τριάδα χωρητικότητας, πηγής και προορισμού δεν υπάρχει ένα έγκυρο, χωρικά, μονοπάτι το οποίο να μπορεί να την υποστηρίξει, την επόμενη φορά που θα εξετάσει μία απαίτηση με τα ίδια χαρακτηριστικά και χωρητικότητα μεγαλύτερη ή ίση της προηγούμενης, θα εκτελέσει απ' ευθείας το δεύτερο κύκλο προσπάθειας ανεύρεσης μονοπατιού. Έτσι, αποκτά ένα είδος ευφυΐας προσαρμόζοντας τις αποφάσεις και τα βήματά του ανάλογα με την πληροφορία που έχει ήδη συλλέξει από προηγούμενες ροές και αποφεύγοντας να εκτελέσει αναζητήσεις και επεξεργασίες οι οποίες σίγουρα θα καταλήξουν άκαρπες.

Όπως φάνηκε και από τα αποτελέσματα, ο χρόνος εκτέλεσης του αλγορίθμου είναι πάρα πολύ μικρός. Ακόμα και στις πιο δύσκολες περιπτώσεις στις οποίες οι ροές είναι πολύ μικρές και λίγες ενώ η τοπολογία είναι τεράστια και κατανεμημένη, ο χρόνος για την ολοκλήρωση όλου του αλγορίθμου δεν ξεπερνά τις μερικές δεκάδες δευτερόλεπτα. Σε σύγκριση με άλλους offline αλγορίθμους (για παράδειγμα, [\[2\]](#)) οι οποίοι έχουν χρόνο εκτέλεσης μέχρι και 25 ώρες, μπορεί κανείς να θεωρήσει ότι ακόμα και τα 123 δευτερόλεπτα που χρειάστηκε ο TiReD για την πιο δύσκολη τοπολογία από όλες στις οποίες εφαρμόστηκε, είναι σχεδόν μηδαμινός.

Παράλληλα και η κατανομή που επιτυγχάνει ο προκείμενος αλγόριθμος, είναι εντυπωσιακή με τον μέσο όρο του ποσοστού δέσμευσης όλων των ακμών να κυμαίνεται στα 40 – 60 %. Φυσικά, υπάρχουν ακμές οι οποίες είναι ή εντελώς μη χρησιμοποιούμενες ή διασχίζονται από μεγαλύτερο φορτίο σε σχέση με άλλες. Ωστόσο, εάν αναλογιστεί κανείς ότι η κατανομή του φορτίου

περιορίζεται σημαντικά από τις ίδιες τις απαιτήσεις οι οποίες έχουν διαφορετικά ζεύγη πηγής – προορισμού, διαφορετικές απαιτήσεις χωρητικότητας, αιτούνται κατωφλίου hops ή απαιτούν την διάσχιση ή την αποφυγή ακμών και ολόκληρων διαδρομών, φανερό γίνεται το ότι δεν μπορούν όλες οι ακμές της ίδιας τοπολογίας να έχουν ακριβώς το ίδιο ποσοστό δεσμευμένης χωρητικότητας. Αξίζει, όμως, να τονιστεί ότι σε ένα δίκτυο με μόλις μία πηγή, έναν προορισμό, ροές ίσου φορτίου και καμίας απαίτησης και με κατανομημένα links στην τοπολογία, η κατανομή του φορτίου πραγματοποιείται όντως σχεδόν εξισορροπημένα χωρίς υπέρ και υπό – φορτωμένες ζεύξεις.

Λαμβάνοντας πάντα υπ' όψιν τις ήδη εξυπηρετηθείσες ροές, ο αλγόριθμος με στόχο να κατανείμει το φορτίο και την κίνηση επιτυγχάνει και το Minimum Interference όπως αυτό ορίστηκε από τους αλγορίθμους MIRA[12] και DORA[15]. Δεδομένου ότι σε κάθε κύκλο προσπαθεί να βρει μονοπάτια με ακμές τέτοιες οι οποίες να έχουν το ελάχιστο ποσοστό δέσμευσης από προηγούμενες ροές, καταλήγει στο να κατανείμει όχι μόνο το φορτίο αυτό καθ' αυτό, αλλά και να εξισορροπεί τις ίδιες τις ροές έτσι ώστε να μην υπάρχουν ακμές οι οποίες να χρησιμοποιούνται από πάρα πολύ λίγες ή πάρα πολλές ροές ταυτόχρονα. Όπως φάνηκε και στο Έκτο Κεφάλαιο, ακόμη και σε μία τοπολογία μεγάλου μήκους, πολλών ακμών και 10 ροών οι οποίες κατευθύνονται προς την ίδια γεωγραφική περιοχή, μόλις 6 από αυτές εμφανίζονται να χρησιμοποιούν τα ίδια links τα οποία, παράλληλα, βρίσκονται μακριά το ένα από το άλλο.

Συνοψίζοντας όλα τα παραπάνω, ο αλγόριθμος TiReD

- Μεγιστοποιεί τη διέλευση φορτίου στο δίκτυο
- Πληροί τις απαιτήσεις του MPLS – TE γενικότερα
- Πληροί τους περιορισμούς MPLS – TE των ροών
- Κατανείμει το φορτίο και την κίνηση ισορροπημένα στην τοπολογία
- Κατανείμει τις ροές ισορροπημένα στις ακμές
- Έχει ελάχιστο απαιτούμενο χρόνο εκτέλεσης (της τάξης των μερικών δευτερολέπτων)

Φυσικά, το γεγονός ότι ο αλγόριθμος προσπαθεί, δοθέντων των περιορισμών, να μεγιστοποιήσει τη διέλευση του φορτίου στο δίκτυο, σε καμία περίπτωση αυτό δε σημαίνει όλες οι ροές, απαραίτητα, θα βρουν μονοπάτι και θα εξυπηρετηθούν. Θα εξυπηρετηθούν όσο το δυνατόν περισσότερες, ωστόσο υπάρχουν πιθανότητες κάποιες από αυτές να απορριφθούν. Οι λόγοι, αν και προφανείς, μπορούν (και φάνηκαν από τις δοκιμές του αλγορίθμου) να είναι οι εξής:

- Η απαιτούμενη χωρητικότητα των ροών υπερβαίνει την MaxFlow τιμή του δικτύου
- Μία ροή μπορεί να «περάσει» στο δίκτυο, από άποψη μεγέθους, αλλά δεν υπάρχει έγκυρο μονοπάτι το οποίο να υπόκειται στους περιορισμούς της (προσπέλαση/ αποφυγή links, υπάρχει πρωτεύον μονοπάτι αλλά όχι δευτερεύον, μικρή τιμή MaxCuts, μονοπάτια μακρύτερα από MaxHops και ούτω καθ' εξής)
- Το δίκτυο έχει ελεύθερα μονοπάτια επαρκούς χωρητικότητας αλλά όχι για το ζεύγος πηγής – προορισμού που αιτεί μια ροή
- Οι περιορισμοί της ροής είναι άκυροι για τη δοθείσα τοπολογία (π.χ. αίτηση υποχρεωτικής προσπέλασης link το οποίο δεν μπορεί να καταλήξει στον προορισμό της πηγής, τιμή MaxHops μικρότερη από το μήκος του μικρότερου μονοπατιού στο δίκτυο για τη συγκεκριμένη πηγή και προορισμό κ.ο.κ.)

Από τα παραπάνω μπορεί να προκύψει το συμπέρασμα ότι δεν υπάρχει εγγύηση ότι όλο το απαιτούμενο φορτίο θα εξυπηρετηθεί και θα διασχίσει το δίκτυο. Ωστόσο, μπορεί να χαρακτηριστεί σχεδόν σίγουρο ότι ο αλγόριθμος θα δρομολογήσει όσο το δυνατόν περισσότερο φορτίο δίνοντας προτεραιότητα στις ροές οι οποίες θα αυξήσουν τη χρησιμοποίηση του δικτύου καθώς και τα κέρδη (οικονομικά, QoS κ.ο.κ.) χρηστών και ISPs.

ΠΕΡΑΙΤΕΡΩ ΜΕΛΕΤΗ

Ένα από τα βασικότερα μειονεκτήματα του TiReD, το οποίο αποτελεί και σχεδιαστική παραδοχή, είναι ότι έχει σχεδιαστεί για εφαρμογή σε δίκτυα τοπολογίας κατευθυνόμενου γράφου με μοναδικές ακμές. Αυτό σημαίνει ότι εάν σε ένα δοθέν δίκτυο υπάρχει η ακμή $[A - B]$ τότε δεν υπάρχει η αντίστροφη της $[B - A]$. Προφανώς κάτι τέτοιο, σε πραγματικά πλαίσια εφαρμογής, σπάνια συναντάται. Από τα παραπάνω προκύπτει ότι σε ένα δίκτυο στο οποίο υπάρχει μονοπάτι από την πηγή A στον προορισμό Z :

- Μία ροή η οποία επιθυμεί να διασχίσει το δίκτυο από το A στο Z μπορεί να βρει μονοπάτι
- Μία ροή η οποία επιθυμεί να διασχίσει το δίκτυο από το Z στο A μπορεί να βρει μονοπάτι, αλλά μόνο εάν αυτό υπολογιστεί αντίστροφα, δηλαδή από το A στο Z .

Το δεύτερο σημείο, με διαφορετικά λόγια, σημαίνει ότι κάθε μονοπάτι το οποίο εκκινεί από μία πηγή και τερματίζει σε μία καταβόθρα μπορεί να προσπελαστεί και ανάποδα, αλλά μόνο εάν υπολογιστεί με τη δοθείσα φορά της τοπολογίας. Ο υπολογισμός απευθείας διαδρομής από το Z προς το A δε θα δώσει μονοπάτι και η ροή δε θα εξυπηρετηθεί.

Παράλληλα, με κατευθυνόμενες ακμές, από τα παραπάνω προκύπτει ότι ένα μονοπάτι δεν μπορεί να διασχίζει μία ζεύξη από την αντίστροφη φορά της. Έτσι, την ώρα που σε ένα MH κατευθυνόμενο γράφημα το πλήθος μονοπατιών από το A στο Z μπορεί να είναι X , στο αντίστοιχο κατευθυνόμενο θα είναι Y , με $Y \leq X$. Κάτι τέτοιο μπορεί να σημαίνει την απόρριψη ροών οι οποίες δεν μπορούν να βρουν μονοπάτι, καθ' ότι τα εφικτά μονοπάτια θα είναι λιγότερα από τα πραγματικά (σε ένα μη κατευθυνόμενο γράφημα), ενώ παράλληλα μπορεί να οδηγήσει σε υποχρησιμοποίηση ακμών οι οποίες ενώ μπορούν να χρησιμοποιηθούν σε κάποια διαδρομή με αντίστροφη φορά, στο κατευθυνόμενο γράφημα δεν θεωρούνται έγκυρες οπότε και δεν επιλέγονται. Από τα παραπάνω προκύπτει το συμπέρασμα ότι παρ' ότι ο TiReD σχεδιάστηκε για κατευθυνόμενα γραφήματα, εάν ανασχεδιαστεί ώστε να

λειτουργεί και σε μη κατευθυνόμενα, θα μπορούσε, ενδεχομένως, να παράξει ακόμα καλύτερα αποτελέσματα τόσο από την άποψη της κατανομής του φορτίου και της χρησιμοποίησης του δικτύου, όσο και από τη σκοπιά της ενδεχόμενης εξυπηρέτησης πολύ περισσότερων ροών.

Παράλληλα, ιδιαίτερο ενδιαφέρον θα είχε η μέτρηση των αποτελεσμάτων χρησιμοποιώντας τις δευτερεύουσες διαδρομές για την εξυπηρέτηση άλλων ροών. Για παράδειγμα, έστω μία ροή F1 η οποία επιθυμεί τη διάσχιση ενός δικτύου από την πηγή A στον προορισμό Z στα 10Mbps και αιτείται και εφεδρικής διαδρομής. Εάν εξυπηρευτεί τότε θα κατασκευαστούν και θα δεσμευτούν για την F1 δύο μονοπάτια από το A στο Z. Έστω, τώρα, ότι άλλη μία ροή χαμηλότερου Priority Level ζητά διάσχιση του δικτύου από το A στο Z, χωρίς εφεδρική διαδρομή και στα 10Mbps επίσης (ή λιγότερα). Και έστω ότι το δίκτυο, τη στιγμή του υπολογισμού της διαδρομής της F2 δεν καταφέρνει να υπολογίσει κάποιο μονοπάτι και την απορρίπτει. Προκύπτει, λοιπόν, ότι ενώ υπάρχει ένα δεσμευμένο, αλλά στην πραγματικότητα από άποψη χωρητικότητας ελεύθερο, μονοπάτι από το A στο Z στα 10Mbps, δε χρησιμοποιείται. Θα χρησιμοποιηθεί όταν και εάν η πρωτεύουσα διαδρομή της F1 παρουσιάσει κάποιο πρόβλημα, οπότε η τελευταία θα ακολουθήσει απ' ευθείας το εφεδρικό αυτό μονοπάτι.

Θα μπορούσε, ωστόσο, για το διάστημα που η F1 δεν αντιμετωπίζει κάποιο πρόβλημα, το εφεδρικό της μονοπάτι να χρησιμοποιείται «δανεικά» από την F2. Έτσι, θα εξυπηρετούνταν και οι δύο ροές, αυξάνοντας κατά 10Mbps τη χρησιμοποίηση του δικτύου. Όταν η F1 αντιμετωπίσει πρόβλημα, όντας υψηλότερου επιπέδου προτεραιότητας από την F2, θα πάρει υπό την κατοχή της την εφεδρική διαδρομή, αφήνοντας την F2 «απ' έξω». Δεδομένου, όμως, ότι το να αχρηστευτεί ολόκληρο μονοπάτι ώστε να χρειαστεί το εφεδρικό, δεν είναι ό,τι πιο πιθανό, κατά μεγάλη πιθανότητα και οι δύο ροές θα μπορούσαν να τυγχάνουν εξυπηρέτησης. Αργότερα, έχοντας μείνει εκτός δικτύου, η F2 θα μπορούσε να «ξαναδανειστεί» το εφεδρικό μονοπάτι κάποιας άλλης ροής η οποία διασχίζει το δίκτυο από το A στο Z με επαρκή χωρητικότητα και έχει υψηλότερο βαθμό προτεραιότητας από την F2.

Έτσι, αρκετό ενδιαφέρον ως περαιτέρω μελέτη, θα είχε η τροποποίηση του αλγορίθμου με τρόπο τέτοιο ώστε να κατανέμει και τα εφεδρικά μονοπάτια σε ροές οι οποίες δεν έτυχαν εξυπηρέτησης καθώς και η μελέτη των τελικών αποτελεσμάτων σε σύγκριση με τα τρέχοντα, τόσο από άποψη χρησιμοποίησης του δικτύου και εξυπηρέτησης των ροών, όσο και από τη σκοπιά της τελικής πολυπλοκότητας του αλγορίθμου.

Επιπρόσθετα, επειδή κατά την εκπόνηση της εργασίας όλο το Implementation του αλγορίθμου έγινε σε γλώσσα C, με το traffic profile (SLAs) να βρίσκεται καταχωρημένο σε μία βάση δεδομένων, σκόπιμη θεωρείται η δοκιμή του TiReD σε ένα πραγματικό σύστημα (ή έστω σε κάποιο Simulator) για να φανεί πλέον υπό πιο πραγματικές συνθήκες η λειτουργία του και το κατά πόσο κάτι χρειάζεται διόρθωση, επανασχεδιασμό και τα λοιπά.

Τέλος, επειδή από παραδοχή, ο αλγόριθμος υπάγεται στους Offline, σημαίνει ότι αυτό που κάνει, τελικά, είναι το Network Planning. Έχει, σαν δεδομένα εισόδου, όλα τα SLAs συγκεντρωτικά και τα κατανέμει σε ένα δίκτυο στο οποίο ακόμα δεν υπάρχει κίνηση. Αρκετό ενδιαφέρον, λοιπόν, θα είχε η δοκιμή του σε ένα ήδη operational δίκτυο στο οποίο εκτός από τα δοθέντα SLAs προστίθενται συνέχεια νέα. Στο σημείο αυτό, ενδεχομένως να πρέπει να γίνει κάποιος επανασχεδιασμός για την κατανομή των νέων ροών ή για την προηγούμενη (κατά το Network Planning) δέσμευση μονοπατιών για τις ροές που μπορεί να εμφανιστούν αργότερα, προκειμένου ο αλγόριθμος να μην κατανέμει μόνο τις προσυμφωνημένες ροές κατά την Offline λειτουργία του δικτύου, αλλά να μπορεί να εισάγει και νέα κίνηση σε ένα δίκτυο το οποίο ήδη λειτουργεί.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1]A. S. Tanenbaum, «Δίκτυα Υπολογιστών», Τέταρτη Αμερικάνικη Έκδοση, *Εκδόσεις Κλειδάριθμος* (2003)
- [2] B. G. Jozsa et al, «An Efficient Algorithm for Global Path Optimization in MPLS Networks», *Optimization and Engineering*, 2, 321–347 (2001)
- [3]C. L. Liu, «Elements of Discrete Mathematics, 2nd Edition», *Πανεπιστημιακές Εκδόσεις Κρήτης* (2005)
- [4]D. Awduche et al, «RSVP-TE: Extensions to RSVP for LSP Tunnels», *RFC 3209* (2001)
- [5]D. Awduche et al, «Requirements for Traffic Engineering Over MPLS», *RFC 2702, UUNET MCI Worldcom* (1999)
- [6]D. Fedyk et al, «Multiple Metrics for Traffic Engineering with IS-IS and OSPF», <http://tools.ietf.org/html/draft-fedyk-isis-ospf-te-metrics-01> (2001)
- [7]H. Wang, K. Zhang, W. Du, L. Wang, «Implementation of IPV6 Network QoS Based on IntServ and SNMP», *Proceedings of the 2009 International Symposium on Information Processing* (2009)
- [8]I. Mahadevan and K.M. Sivalingam, «Quality Of Service Architectures for Wireless Networks: IntServ and DiffServ Models», *In Mobile Computing Workshop at Intl. Symposium on Parallel Architectures, Algorithms and Networks* (1999)
- [9]J. de Oliveira et al, «Label Switched Path (LSP) Preemption Policies for MPLS Traffic Engineering», *RFC 4829*, (2007)

- [10] J. Bartlett and R. Wetzel, «QoS over MPLS – The Complete Story», *BCR Volume 36, Number 2* (2006)
- [11] K. Banglore, «A Minimum Interference Hybrid Algorithm for MPLS Networks»
- [12] K. Kar, M. Kodialam and T.V. Lakshman «Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications», *IEEE Journal on Selected Areas in Communications, Vol. 18, No. 12* (2000)
- [13] K. Manayya, «Constrained Shortest Path First», <http://www.ietf.org/id/draft-manayya-cspf-00.txt> (2010)
- [14] K. Walkowiak, «Survivable Online Routing for MPLS Traffic Engineering», *Lecture Notes in Computer Science* (2004)
- [15] R. Boutaba, W. Szeto and Y. Iraqi, «DORA: Efficient Routing for MPLS Traffic Engineering», *Journal of Network and Systems Management, Vol. 10, No. 3* (2002)
- [16] R.N. Pise, S.A. Kulkarni and R.V. Pawar, «Packet Forwarding with Multiprotocol Label Switching», *World Academy of Science, Engineering and Technology* (2005)
- [17] S. Norden and J. Turner, «DRES: Network Resource Management using Deferred Reservations», *Department of Computer Science, Washington University*
- [18] T. C. Hung et al, «Advanced Routing Algorithms and Load Balancing on MPLS», *ICACT* (2007)
- [19] Α. Κανάτας, Φ. Κωνσταντίνου, Γ. Πάντος, «Συστήματα Κινητών Επικοινωνιών», *Εκδόσεις Παπασωτηρίου* (2008)

[20] Α. Χ. Παναγιωτόπουλος, «Στοιχεία Μαθηματικού Προγραμματισμού», Εκδόσεις Σταμούλης (1991)

[21] Π. Σαμπατάκος, «Προσαρμοζόμενη διαχείριση πόρων διαδικτύου για υποστήριξη ποιότητας υπηρεσίας», Διδακτορική Διατριβή, Εθνικό Μετσόβιο Πολυτεχνείο (2006)

[22] «Constructing High Quality IP Core Network», Huawei-3Com

[23] «Multiprotocol Label Switching», http://en.wikipedia.org/wiki/Multiprotocol_Label_Switching

[24] http://en.wikipedia.org/wiki/Ford-Fulkerson_algorithm

[25] http://en.wikipedia.org/wiki/Edmonds-Karp_algorithm

[26] http://en.wikipedia.org/wiki/Dinitz_blocking_flow_algorithm

[27] http://en.wikipedia.org/wiki/Relabel-to-front_algorithm

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

A

AF	Assured Forwarding
ATM	Asynchronous Transfer Mode

B

BB	Bandwidth Broker
BGP	Border Gateway Protocol

C

CRM	Customer Relationship Management
CSPF	Constraint – Based Shortest Path First

D

DiffServ	Differential Services
DLCI	Datalink Connection Identifier
DLL	Data Link Layer
DORA	Dynamic Online Routing Algorithm
DSCP	Differentiated Service Code Point

E

EF	Expedited Forwarding
EGP	Exterior Gateway Protocol
EIGPR	Enhanced Interior Gateway Routing Protocol
ERP	Enterprise Resource Planning
ESRN	Extended Sub Residual Network

F

FEC	Forwarding Equivalence Class
FTP	File Transfer Protocol

I

IETF	Internet Engineering Task Force
IGP	Interior Gateway Protocol
IntServ	Integrated Services
IP	Internet Protocol
IS – IS	Intermediate System to Intermediate System
ISP	Internet Service Provider

L

LDP	Label Distribution Protocol
LER	Label Edge Router
LFIB	LSP Forwarding Information Base
LIB	Label Information Base
LIFO	Last In First Out
LSP	Label Switched Path
LSR	Label Switching Router

M

MIB	Management Information Base
MIH	Minimum Interference Hybrid
MIRA	Minimum Interference Routing Algorithm
MPLS	Multiprotocol Label Switching

N

NL	Network Layer
NP	Non Polynomial

O

OSPF	Open Shortest Path First
------	--------------------------

P

PHB	Per Hop Behavior
PL	Priority Level

Q

QoS Quality Of Service

R

RED Random Early Detection
RFC Request for Comments
RIP Routing Information Protocol
RSVP Resource ReserVation Protocol

S

SCM Supply Chain Management
SLA Service Level Agreement
SONET Synchronous optical networking
SPF Shortest Path First
SRN Sub Residual Network

T

TCP Transmission Control Protocol
TE Traffic Engineering
TiReD Traffic Resource Distribution
ToS Type of Service
TTL Time to Live

V

VCI Virtual Circuit Identifier
VPI Virtual Path Identifier
VPN Virtual Private Network

ΠΑΡΑΡΤΗΜΑ Ι

NETWORK.H

```
#pragma once
#include <iostream>
#include <list>
#include "RoutedDemands.h"
#include "Demands.h"
#include "Memo.h"
#include "tree.hh"

class Network
{
private:

    struct Link;

    typedef struct Node{
        int Name;
        int Height;
        bool Visited;
        struct Node *Next;
        struct Link *Adj;
        int LinkNumber;
    } Node;

    struct Link{
        int Capacity;
        int Flow;
        struct Node *Next;
        struct Link *Adj;
        RoutedDemands routedDemands;
    };

    Node *NodeHead;
    Link *LinkHead;

    int NodesCount;
    int Type;

    char *Server;
    char *User;
    char *Password;
    char *DatabaseName;

    char *NetworkID;
    char *DemandsListID;

    bool DebugMode;

    std::string IntToStr(int AnInt);

    //Network Creation
    bool CreateNetwork();
    int AddNode(int Name);
    int AddLink(int From, int To, int Capacity, int Flow);
    Network ResidualNetwork(int Type, Demands::Demand *CurrentDemand);
    void DeleteLinks(Link **head );
    void DeleteNetwork(Node **head);
    Network& operator =(Network const& CopiedNetwork);
```



```

//Demands serving
void ServeDemand(Demands::Demand *DemandToServe);
void TrashDemand(Demands::Demand *DemandToTrash);
void SaveServedDemands();

//Path extraction
bool GetPaths(tree<int> Paths, tree<int>::iterator SourceIterator, int
Source, int Destination);
void AddDemandPath(std::list<int> DemandPath, int DemandID, int
Cut,int Bandwidth, bool IsPrimary);
int FeasiblePath(std::list<int>&Path, Demands::Demand *CurrentDemand,
int Source, int Destination);
int FeasiblePathBandwidth(std::list<int> APath, Demands::Demand
*CurrentDemand, int *ThisMaxHops, int *ThisLevelID, int
*ThisMaxBand);

//Entity Existence Checks
bool NodeExists(int Name);
bool LinkExists(int From, int To);
bool NeighboursExist(std::list<int> Path, int FromNode, int ToNode);

//Useful Functions
void SetToUnvisited();
bool LayerNetwork(int Source, int Destination);
int LinkCapacity(int From, int To);
int ResidualCapacity(int From, int To);
void SetPathValue(Memo& OldValue, Memo NewValue);
void SaveAlgorithmCalculation(bool Beginning);

//Displays
void DisplayNetwork(int what);
void DisplayDemandsResults();

public:
Network(void);
virtual ~Network(void);
void SetConnectionParams(char *server, char *user,
char *password, char *database);
int SetTopology(char *demandsSetID);
void SetCalculationMode(bool debugMode);
void TrafficResourceDistributionAlgorithm();
};

```

DEMANDS.H

```
#pragma once
class Demands
{
private:
    char *Server;
    char *User;
    char *Password;
    char *DatabaseName;
    char *DemandsListID;

public:

    struct Demand{
        int ID;
        int Source;
        int Destination;
        int PriorityLevel;
        int Bandwidth;
        int MaxHops;
        int HopsDeviation;
        int MaxCuts;
        bool NeedsBackup;
        bool HasLinksRequirements;
        int MaxLinkRequirementsLevel;

        bool ExtSubResCalculated;
        bool CalculateExtSubRes;
        bool IsBackup;

        int Cut;

        Demand *Next;
    };

    Demand *DemandHead;

    Demands(void);
    virtual ~Demands(void);

    bool CreateDemands();
    void Append(Demand *NewDemand);
    void Push(Demand *NewDemand);
    void DisplayDemands();
    bool Empty();
    void Delete(int DemandID, bool IsBackup, bool All);
    int Count();
    Demand *GetFirst();
    void DeleteDemands(Demand **head);
    void SetConnectionParams(char *server, char *user,
                            char *password, char *database);
    void SetDemandsList(char *demandsListID);
};
```

ROUTEDDEMANDS.H

```
#pragma once
class RoutedDemands
{
public:

    struct RoutedDemand{
        int ID;
        int Cut;
        int Bandwidth;
        bool IsPrimary;
        RoutedDemand *Next;
    };

    RoutedDemand *RoutedDemandHead;

    RoutedDemands(void);
    virtual ~RoutedDemands(void);
    void Append(int NewID, int Cut, int NewBandwidth, bool IsPrimary);
    int GetBandwidth(int ID);
    void Delete(int ID);
    bool ExistsPrimaryDemand(int DemandID);
    void DeleteRoutedDemands(RoutedDemand **head);
    void Display();
    void SaveServed(int LinkFrom, int LinkTo, char *Server, char
    *User, char *Password, char *DatabaseName);
};
```

MEMO.H

```
#pragma once

class Memo
{
public:
    struct MemoRecord{
        int Source;
        int Destination;
        int Bandwidth;
        MemoRecord *Next;
    };

    MemoRecord *MemoRecordHead;
    Memo(void);
    Memo(const Memo& NewMemo);
    Memo& operator =(Memo const& CopiedMemo);
    bool Exists(int Source, int Destination, int Bandwidth);
    void Update(int Source, int Destination, int Bandwidth);
    void Append(int Source, int Destination, int Bandwidth);
    void Push(int VisitedNode);
    void Insert(int VisitedNode, int Index);
    int GetVisitedNode(int Index); void Display();
    void DeleteTail();
    int Count();
    bool ExistNeighbours(int From, int To);
    bool Empty();
    void DeleteMemo (MemoRecord **head);
    virtual ~Memo(void);
};
```

TRAFFICRESOUCEDISTRIBUTIONALGORITHM()

//NETWORK.C

```
void Network::TrafficResourceDistributionAlgorithm() {
    printf("Creating Network...\n");
    if (CreateNetwork()){
        DisplayNetwork(0);

        //First, set the parameters of the current calculation
        DemandsList.SetDemandsList(DemandsListID);
        DemandsList.SetConnectionParams(Server, User, Password, DatabaseName);

        //Create the demands list
        printf("Creating Demands List...\n");
        if (DemandsList.CreateDemands()){
            if (DebugMode)
                DemandsList.DisplayDemands();
            SaveAlgorithmCalculation(true);

            //Check whether the demands list is empty
            while(!DemandsList.Empty()){
                Demands::Demand *CurrentDemand = DemandsList.GetFirst();
                printf(" Running algorithm for Demand <%d>\n", CurrentDemand->ID);

                if (CurrentDemand->ExtSubResCalculated){
                    printf(" Already calculated the Extended SubResidual Network
                        for <%d>\n", CurrentDemand->ID);
                    printf(" Trashing Demand <%d> ... \n", CurrentDemand->ID);
                    TrashDemand(CurrentDemand);
                }
                else{
                    bool BuildExtSubRes =
                        (
                            (NetworkMemo.Exists(CurrentDemand->Source,
                                CurrentDemand->Destination,
                                CurrentDemand->Bandwidth))
                            ||
                            (CurrentDemand->CalculateExtSubRes)
                        );
                    Network *Residual;
                    if (BuildExtSubRes){
                        Residual = new Network(ResidualNetwork(EXTENDED_SUBRESIDUAL,
                            CurrentDemand));
                        CurrentDemand->ExtSubResCalculated = true;
                        printf(" Building an Extended SubResidual Network...\n");
                        (*Residual).DisplayNetwork(1);
                        (*Residual).DisplayNetwork(2);
                    }
                    else{
                        Residual = new Network(ResidualNetwork(SUBRESIDUAL,
                            CurrentDemand));
                        printf(" Building SubResidual Network...\n");
                        (*Residual).DisplayNetwork(1);
                        (*Residual).DisplayNetwork(2);
                    }

                    //If you didn't manage to layer the residual,
                    then you failed to reach Destination
                    //Thus, either get out and calculate the extended subresidual,
                    or trash the demand
                    bool ResidualLayered =
                        (*Residual).LayerNetwork(CurrentDemand->Source,
                            CurrentDemand->Destination);
                }
            }
        }
    }
}
```

```

if(ResidualLayered){
    printf(" Layered the Residual Network.\n");

    (*Residual).SetConnectionParams(this->Server, this->User,
                                    this->Password, this->DatabaseName);
    (*Residual).DisplayNetwork(3);
    Memo Path;
    Memo *PathPtr;
    PathPtr = &Path;
    std::list<int> FoundPath;

    int Bandwidth =
        (*Residual).FeasiblePath(FoundPath,
                                CurrentDemand,
                                CurrentDemand->Source,
                                CurrentDemand->Destination);

    //If the capacity is adequate, serve the demand
    if(Bandwidth >= CurrentDemand->Bandwidth){
        AddDemandPath(FoundPath, CurrentDemand->ID,
                      CurrentDemand->Cut, CurrentDemand->Bandwidth,
                      !CurrentDemand->IsBackup);
        printf(" Demand <%d> is served. \n", CurrentDemand->ID);
        ServeDemand(CurrentDemand);
    }

    //If there is a path but the capacity is inadequate
    else if (Bandwidth > 0){
        printf(" Found a path with inadequate capacity.
              Updating Memo...\n");

        NetworkMemo.Update(CurrentDemand->Source,
                           CurrentDemand->Destination, Bandwidth);

        //See if you can cut the demand.
        If you cannot cut it anymore, trash it
        if(CurrentDemand->Cut == CurrentDemand->MaxCuts){
            printf(" Cannot cut Demand <%d>. Trashing it... \n",
                  CurrentDemand->ID);

            TrashDemand(CurrentDemand);
        }
        //If you can cut it, then cut it
        else{
            printf(" Cutting Demand <%d>... \n", CurrentDemand->ID);
            Demands::Demand *FirstDemandCut;
            FirstDemandCut = new Demands::Demand(*CurrentDemand);
            Demands::Demand *SecondDemandCut;
            SecondDemandCut = new Demands::Demand(*CurrentDemand);
            FirstDemandCut->Bandwidth = Bandwidth;

            AddDemandPath(FoundPath, FirstDemandCut->ID,
                          FirstDemandCut->Cut, Bandwidth,
                          !FirstDemandCut->IsBackup);
            printf(" A bandwidth of %d units is served for Demand <%d>.
                  \n", Bandwidth, CurrentDemand->ID);

            ServeDemand(FirstDemandCut);

            SecondDemandCut->Cut = SecondDemandCut->Cut + 1;
            SecondDemandCut->Bandwidth =
                SecondDemandCut->Bandwidth - Bandwidth;
            SecondDemandCut->ExtSubResCalculated = false;

            DemandsList.Push(SecondDemandCut);
        }
    }
}

```


CREATEDEMANDS() //DEMANDS.C

```
bool Demands::CreateDemands() {
    try{
        MYSQL *Connection;
        MYSQL_RES *ResultSet;
        MYSQL_ROW Row;

        Connection = mysql_init(NULL);

        //Connect to the Database
        if (!mysql_real_connect(Connection, Server, User, Password,
                                DatabaseName, 0, NULL, 0)) {
            fprintf(stderr, " ***%s\n", mysql_error(Connection));
            return false;
        }

        //Now let's get all the information we need for the Demands
        //The demands will be ordered by :
        // Priority Level, bandwidth (*2 if needs backup),
        // LinkRequirements length, MaxHops, HopsDeviation and MaxCuts
        char Query[1500];
        char *SelectDemands =
            "select
            "      D.DemandID, "
            "      D.Source, "
            "      D.Destination, "
            "      D.PriorityLevel, "
            "      D.Bandwidth, "
            "      ifnull(D.MaxHops, -1), "
            "      ifnull(D.HopDeviation, 0), "
            "      ifnull(D.NeedsBackup, 0), "
            "      ifnull "
            "      (
            "          ( select distinct 1 "
            "            from LinkRequirements L "
            "            where L.DemandID = D.DemandID) "
            "      , 0) HasLinksRequirements, "
            "      ifnull "
            "      (
            "          ( select max(LevelID) "
            "            from LinkRequirements L "
            "            where L.DemandID = D.DemandID) "
            "      , 0) MaxLinkRequirementsLevel, "
            "      ifnull(D.MaxCuts, 3) MaxCuts "
            "from Demands D "
            "where D.DemandsListID = '%s' "
            "order by "
            "  D.PriorityLevel, "
            "  D.Bandwidth*Power(2, ifnull(NeedsBackup, 0)) desc, "
            "  ifnull((select count(1) from LinkRequirements L where "
            "              L.DemandID = D.DemandID), 0) desc, "
            "  ifnull(D.MaxHops, 0) asc, "
            "  ifnull(D.HopDeviation, 0) asc, "
            "  ifnull(D.MaxCuts, 3) asc; ";

        sprintf(Query, SelectDemands, DemandsListID);
        if (mysql_query(Connection, Query)) {
            fprintf(stderr, " ***%s\n", mysql_error(Connection));
            return false;
        }

        ResultSet = mysql_store_result(Connection);
        while((Row = mysql_fetch_row(ResultSet)) != NULL){
            string DemandID = Row[0];
        }
    }
}
```

```

string Source = Row[1];
string Destination = Row[2];
string PriorityLevel = Row[3];
string Bandwidth = Row[4];
string MaxHops = Row[5];
string HopsDeviation = Row[6];
string NeedsBackup = Row[7];
string HasLinksRequirements = Row[8];
string MaxLinksRequirementsLevel = Row[9];
string MaxCuts = Row[10];

Demand *NewDemand = new Demand;
NewDemand->Next = NULL;
NewDemand->ID = atoi(DemandID.c_str());
NewDemand->Source = atoi(Source.c_str());
NewDemand->Destination = atoi(Destination.c_str());
NewDemand->PriorityLevel = atoi(PriorityLevel.c_str());
NewDemand->Bandwidth = atoi(Bandwidth.c_str());
NewDemand->MaxHops = atoi(MaxHops.c_str());
NewDemand->HopsDeviation = atoi(HopsDeviation.c_str());
NewDemand->NeedsBackup = atoi(NeedsBackup.c_str());
NewDemand->HasLinksRequirements = atoi(HasLinksRequirements.c_str());
NewDemand->MaxLinkRequirementsLevel =
    atoi(MaxLinksRequirementsLevel.c_str());
NewDemand->MaxCuts = atoi(MaxCuts.c_str());

NewDemand->ExtSubResCalculated = false;
NewDemand->CalculateExtSubRes = false;
NewDemand->IsBackup = false;
NewDemand->Cut = 1;

//Now that you've created the demand, add it to the list
if (NewDemand!=NULL) {
    Append(NewDemand);

    //If you need a backup, create a clone of the primary demand,
    //change it to be a "backup" and append it too.
    if (NewDemand->NeedsBackup){
        Demand *NewBackUpDemand;
        NewBackUpDemand = new Demand(*NewDemand);
        NewBackUpDemand->NeedsBackup = false;
        NewBackUpDemand->IsBackup = true;
        Append(NewBackUpDemand);
    }
}
}
mysql_free_result(ResultSet);

//Close the connection
mysql_close(Connection);

return true;
}
catch(exception e){
    printf("    ***Encountered exception in CreateDemands()");
    return false;
}
return false;
} //end CreateDemands()

```


ΠΑΡΑΡΤΗΜΑ ΙΙ

ΟΔΗΓΙΕΣ ΕΓΚΑΤΑΣΤΑΣΗΣ

ΕΓΚΑΤΑΣΤΑΣΗ MYSQLSERVER

1. Στην περίπτωση που δεν έχετε ήδη εγκαταστημένο στο σύστημά σας έναν MySQL Server, επισκεφθείτε την Official ιστοσελίδα της MySQL και κατεβάστε την τελευταία έκδοση του MySQL Community Server (<http://dev.mysql.com/downloads/>)

2. Εκτελέστε τον Installer που κατεβάσατε και εγκαταστήστε το MySQL Server στο σύστημά σας, ακολουθώντας τις οδηγίες. Μπορείτε να επιλέξετε όποιο password επιθυμείτε. Σημειώστε επίσης ότι η τοποθεσία εγκατάστασης του Server δεν έχει σημασία για την εκτέλεση του προγράμματος : μπορείτε να εγκαταστήσετε τον Server τόσο τοπικά (localhost) όσο και απομακρυσμένα.

3. Αφού εγκαταστήσετε τον Server, βεβαιωθείτε ότι τρέχει :

```
Open services.msc
```

Το Service με το όνομα «MySQL» πρέπει να έχει status «Started». Εάν όχι, κάντε δεξί κλικ στο Service και επιλέξτε «Start»

ΕΓΚΑΤΑΣΤΑΣΗ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ

1. Ανοίξτε μία κονσόλα και συνδεθείτε στον MySQL Server σας με τα στοιχεία που επιλέξατε κατά την εγκατάσταση. Έστω ότι συνδέεστε ως root, ο server σας βρίσκεται στο τοπικό σας σύστημα και έχετε εισάγει κωδικό πρόσβασης :

```
> mysql -u root -h localhost -p
```

2. Δημιουργήστε μία νέα βάση δεδομένων με όποιο όνομα επιθυμείτε. Η default βάση Δεδομένων του προγράμματος ονομάζεται «TiReDDB», ωστόσο μπορείτε να δώσετε όποια άλλη ονομασία θέλετε εσείς.

```
> mysql -u root -h localhost -p -e "create database TiredDB"
```

3. Στο συνοδευτικό CD, μέσα στο φάκελο «Databases» θα βρείτε δύο αρχεία:

- TiredDB.sql : Περιέχει τον MySQL κώδικα για τη δημιουργία των πινάκων και των σχέσεων της Βάσης Δεδομένων του προγράμματος.
- TiredDBTestData.sql : Περιέχει τον MySQL κώδικα για τη δημιουργία των πινάκων, των σχέσεων καθώς και των test/ demo δεδομένων όπως αυτά εισέρχονταν κατά τις ποικίλες εκτελέσεις του αλγορίθμου για διάφορες τοπολογίες και ομάδες ροών.

Επιλέξτε ένα από αυτά και κάντε το import της βάσης που μόλις δημιουργήσατε (έστω ότι επιλέγετε να δημιουργήσετε κενούς πίνακες, μέσα στη βάση με όνομα «TiredDB»)

```
> mysql -u root -h localhost -p TiredDB < "...\\TiReD Algorithm\\Databases\\TiredDB.sql"
```

4. Βεβαιωθείτε ότι όλα πήγαν καλά. Κάντε login στον server σας :

```
> mysql -u root -h localhost -p
```

Επιλέξτε τη βάση που δημιουργήσατε (έστω TiredDB)

```
mysql > use TiredDB;
```

Και εμφανίστε τους πίνακες της Βάσης. Το αποτέλεσμα του Query θα πρέπει να φέρει 9 εγγραφές με τα ονόματα των πινάκων.

```
mysql > show tables;
```

ΟΔΗΓΙΕΣ ΕΚΤΕΛΕΣΗΣ

Για να εκτελέσετε το πρόγραμμα, εκτός από τη Βάση Δεδομένων που δημιουργήσατε στα παραπάνω βήματα, θα χρειαστείτε και έναν MySQL Connector. Μπορείτε να τον κατεβάσετε από την Official ιστοσελίδα της MySQL επιλέγοντας την τελευταία έκδοση ενός MySQL Connector για C/C++ προγράμματα. Εναλλακτικά, μπορείτε να βρείτε το αντίστοιχο DLL στον φάκελο «...\TiReD Algorithm\Executable» του συνοδευτικού CD.

Το libmysql.dll που θα κατεβάσετε ή θα αντιγράψετε από το συνοδευτικό CD θα πρέπει να είναι στον ίδιο κατάλογο με το εκτελέσιμο πρόγραμμα «Traffic-Resource Distro Algorithm». Στον φάκελο «...\TiReD Algorithm\Executable» θα βρείτε και το εκτελέσιμο και την αντίστοιχη βιβλιοθήκη που απαιτείται.

Πηγαίνετε στον φάκελο όπου έχετε τοποθετήσει το εκτελέσιμο και τρέξτε το μέσα από μία γραμμή εντολών. Για παράδειγμα :

```
> cd "C:\Users\queen\Desktop\TiReD Algorithm\Executable"
```

```
> "Traffic-Resource Distro Algorithm.exe" 1
```

Για την εκτέλεση του προγράμματος απαιτείται η είσοδος μίας υποχρεωτικής παραμέτρου και 5 προαιρετικών :

```
Usage: TrafficResourceDistro [-u Username]
                               [-p Password]
                               [-h Host]
[-dbDataBase]
[-dm]
DemandsList
```

DemandsList

Είναι η υποχρεωτική παράμετρος και πρέπει πάντα να δηλώνεται στο τέλος της εντολής. Χρησιμοποιείται για να δηλώσει στο πρόγραμμα για ποιο DemandsListID που έχει ήδη δημιουργηθεί στη Βάση Δεδομένων, θα εκτελεστεί ο αλγόριθμος. Δεδομένου ότι η Βάση Δεδομένων στον πίνακα

DemandsList περιλαμβάνει το DemandsListID αλλά και το NetworkID στο οποίο υπάγεται η συγκεκριμένη ομάδα απαιτήσεων, δε χρειάζεται κάποια άλλη πληροφορία όσον αφορά στην τοπολογία και τις ροές για τις οποίες θα εκτελεστεί ο αλγόριθμος.

-u username

Προαιρετική παράμετρος η οποία χρησιμοποιείται για να δηλώσει το username που έχετε θέσει για τη σύνδεσή σας στον MySQL Server. Στην περίπτωση που δε δοθεί, θεωρείται by default το «root»

-p password

Προαιρετική παράμετρος η οποία χρησιμοποιείται για να δηλώσει το password που έχετε θέσει για τη σύνδεσή σας στον MySQL Server. Στην περίπτωση που δε δοθεί, θεωρείται by default το κενό (δηλαδή, κανένα password)

-h host

Προαιρετική παράμετρος η οποία δηλώνει την τοποθεσία στην οποία έχετε εγκαταστήσει τη Βάση Δεδομένων. Default τιμή θεωρείται το «localhost»

-db Database

Προαιρετική παράμετρος η οποία δηλώνει το όνομα που έχετε δώσει στη Βάση Δεδομένων του αλγορίθμου. Εάν δε δοθεί, χρησιμοποιείται ως όνομα το «TiredDB»

-dm

Προαιρετική παράμετρος η οποία δηλώνει τον τρόπο εκτέλεσης του προγράμματος. Εάν δοθεί, χρησιμοποιείται σαν τρόπος το «Debug Mode» και κατά την εκτέλεση του αλγορίθμου εμφανίζονται πολύ περισσότερες

λεπτομέρειες στην κονσόλα (όπως λεπτομερείς τοπολογίες δικτύων, αποφάσεις και άλλα). Εάν δε δοθεί, εμφανίζονται λιγότερες οι οποίες απεικονίζουν τα βήματα του αλγορίθμου σε κάθε στάδιο.

Παραδείγματα

```
> TrafficResourceDistro -h localhost -u devel -p  
develpass -dm 1
```

```
>TrafficResourceDistro -h 192.168.1.10 2
```

```
>TrafficResourceDistro 3
```

```
>TrafficResourceDistro -dm -db MyDatabaseName 4
```