

An Improvement to OSPF for avoiding overloaded nodes

ΧΑΡΧΑΛΑΚΗΣ ΣΤΕΦΑΝΟΣ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων Καθηγητής Σαμαράς Νικόλαος

Εξεταστές Ευαγγελίδης Γεώργιος

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πανεπιστήμιο Μακεδονίας

Θεσσαλονίκη

25 Σεπτεμβρίου 2006

Copyright © ΧΑΡΧΑΛΑΚΗΣ ΣΤΕΦΑΝΟΣ, 2006
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής εργασίας από το Τμήμα Εφαρμοσμένης Πληροφορικής του
Πανεπιστημίου Μακεδονίας δεν υποδηλώνει απαραιτήτως και αποδοχή των απόψεων του
συγγραφέα εκ μέρους του Τμήματος.

Abstract

This document examines a new subject in the areas of routing and computer networks. We begin with a comprehensive introduction to networks, TCP/IP and Internet technologies. Next we examine in more detail the IP routing and particularly the OSPF routing algorithm. We discuss in depth the routing procedure analyzing the problems that occur when a router becomes overloaded and can no longer handle the network load. An algorithm is being proposed as a solution to this problem. It is based on the Link State routing protocols operation and tries to find and use alternative routes to route network traffic without creating routing loops. This leads to a decrease of the overloaded routers' load and gives a boost to network performance. Detailed examples of the algorithm operation are being presented. The document also includes the complete pseudocode of the algorithm which was implemented in C/C++. The implementation was tested against more than 800.000 graphs and it resulted in the experimental proof of the algorithm functionality.

Περίληψη

Η εργασία αυτή μελετά ένα νέο θέμα στο αντικείμενο των δικτύων και της δρομολόγησης. Αρχικά γίνεται μια εκτενής εισαγωγή στα δίκτυα, στο TCP/IP και στις τεχνολογίες του Internet. Στη συνέχεια γίνεται εμβάθυνση στο τομέα της δρομολόγησης και πιο συγκεκριμένα στο δυναμικό πρωτόκολλο δρομολόγησης OSPF. Ακολουθεί μια μελέτη της διαδικασίας δρομολόγησης και αμέσως μετά παρουσιάζονται και εξηγούνται τα προβλήματα που προκύπτουν όταν κάποιος δρομολογητής υπερφορτώθει και δεν μπορεί να αντεπεξέλθει πλέον στις ανάγκες του δικτύου. Σαν λύση των προβλημάτων αυτών προτείνεται ένας αλγόριθμος ο οποίος στηρίζεται στη λειτουργία των Link State πρωτοκόλλων δρομολόγησης και προσπαθεί να χρησιμοποιήσει εναλλακτικούς δρόμους για τη δρομολόγηση της κίνησης χωρίς να προκαλέσει routing loops. Αυτό έχει σαν αποτέλεσμα να μειώνεται ο φόρτος του δρομολογητή που παρουσιάζει πρόβλημα δρομολόγησης και να αυξάνονται οι επιδόσεις του δικτύου. Παρουσιάζονται αναλυτικά παραδείγματα της λειτουργίας του αλγορίθμου καθώς και πλήρης ψευδοκώδικας. Έχει γίνει υλοποίηση του αλγορίθμου σε γλώσσα προγραμματισμού C/C++ και έγινε πειραματικός έλεγχος της λειτουργίας του σε περισσότερους από 800.000 γραφους.

Περιεχόμενα

| | |
|---|-----------|
| 1 Δίκτυα | 7 |
| 1.1 Εισαγωγή | 7 |
| 1.2 Τεχνολογίες Δικτύων | 9 |
| 1.2.1 OSI Reference Model | 9 |
| 1.2.2 TCP/IP Protocol Suite | 11 |
| 1.2.2.1 Διευθυνσιοδότηση | 12 |
| 1.2.2.2 Μορφή του IP πακέτου | 14 |
| 1.3 Δρομολόγηση | 16 |
| 1.3.1 Παράδειγμα πίνακα δρομολόγησης | 17 |
| 1.3.2 Τεχνικές βελτιστοποίησης | 21 |
| 1.3.3 Είδη δρομολόγησης | 22 |
| 1.3.4 Είδη πρωτοκόλλων δρομολόγησης | 24 |
| 1.3.4.1 Distance Vector Routing Protocols | 24 |
| 1.3.4.2 Link State Routing Protocols | 25 |
| 2 Open Shortest Path First | 27 |
| 2.1 Γενικά | 27 |
| 2.1.1 Περιοχές στο OSPF | 28 |
| 2.1.2 Δίκτυα στο OSPF | 30 |
| 2.2 Ο γράφος | 31 |
| 2.2.1 Ένα παράδειγμα | 33 |
| 2.3 Λειτουργία | 36 |
| 2.4 Πακέτα του OSPF | 37 |
| 2.5 Ο αλγόριθμος του Dijkstra | 38 |

| | |
|--|-----------|
| 3 Συμφόρηση κόμβων | 41 |
| 3.1 Εισαγωγή | 41 |
| 3.1.1 Αδυναμία δρομολόγησης | 42 |
| 3.1.2 Απώλεια πακέτων στο TCP | 43 |
| 3.1.3 Παραδείγματα | 46 |
| 3.2 Μια νέα πρόταση | 46 |
| 3.3 Ο αλγόριθμος με παράδειγμα | 49 |
| 3.3.1 Επαναληπτηκότητα | 52 |
| 3.3.2 Πολλαπλές διαδρομές | 52 |
| 3.3.3 Ένα πλήρες παράδειγμα | 53 |
| 3.4 Ψευδοκώδικας | 56 |
| 3.4.1 find_paths_to | 58 |
| 3.4.2 find_intermediate | 59 |
| 3.4.3 get_next_ignore | 59 |
| 3.4.4 get_via_interm | 60 |
| 3.4.5 update_rtable | 60 |
| 3.4.6 Χρήση από OSPF | 62 |
| 3.5 Υλοποίηση | 64 |
| 4 Συμπεράσματα | 66 |
| α' Αποτελέσματα εκτέλεσης | 67 |

Κεφάλαιο 1

Διάτυπα

1.1 Εισαγωγή

To Internet αποτελεί ένα από τα μεγαλύτερα επιτεύγματα της ανθρωπότητας στη σύγχρονη εποχή (και όχι μόνο). Οι λόγοι που έχουν συμβάλει στο να γίνει αυτό που είναι σήμερα συμπεριλαμβάνουν την απλότητα, την ευελιξία, την προσαρμοστικότητα, την επεκτασιμότητα, την προτυποποίηση (standardization), την ελευθερία καθώς και άλλους παράγοντες. Έχει καταφέρει να αλλάξει το τρόπο ζωής, λειτουργίας και εργασίας στη σύγχρονη κοινωνία και είναι πρακτικά αδύνατο να υπολογιστούν οι περαιτέρω επιδράσεις που θα έχει στο βραχυπρόθεσμο και μακροπρόθεσμο μέλλον.

Κατά τη πορεία της ιστορίας της ανθρωπότητας, το τηλέφωνο, το ραδιόφωνο και η τηλεόραση κατάφεραν να ‘ενώσουν’ τους ανθρώπους του πλανήτη και τους επιμέρους πολιτισμούς, με το κάθε ένα από αυτά να έχει μεγαλύτερη επίδραση από τα προηγούμενά του. To Internet ήρθε να συμπληρώσει τα παραπάνω μέσα επικοινωνίας και να καταφέρει ακόμα περισσότερα, παρέχοντας πολλές παραπάνω δυνατότητες και αποτελώντας υπερσύνολο αυτών.

Σήμερα το Internet αποτελεί το μεγαλύτερο μέσο επικοινωνίας και διάδοσης νέων και πληροφορίας μετά από το τηλέφωνο και τη τηλεόραση, έχοντας ξεπεράσει τα αντίστοιχα έντυπα μέσα. Ακολουθόντας την ίδια πορεία είναι πολύ πιθανό να καταφέρει να ξεπεράσει ή να ενσωματώσει στο βραχυπρόθεσμο μέλλον και τα πρωτοπόρα αυτά μέσα.

Τηλέφωνο

Ήδη έχει γίνει η απαραίτητη έρευνα ώστε να μπορεί να μεταφέρεται ικανοποιητικά η φωνή μέσω του Internet. Δεν είναι τυχαίο πως πολλοί πάροχοι σταθερής τηλεφωνίας χρησιμοποιούν δίκτυα δεδομένων ακριβώς ίδια με το Internet για να μεταφέρουν τη φωνή των πελατών τους

σε ψηφιακή μορφή. Ενδεικτικά μπορούμε να αναφέρουμε πως στην Ελλάδα όλοι οι πάροχοι σταθερή τηλεφωνίας πλην του ΟΤΕ χρησιμοποιούν τεχνολογίες Voice over IP (VoIP). Επίσης έχει αρχίσει και αναπτύσσεται μια αγορά τηλεφωνίας βασισμένη στο Internet η οποία δεν έχει τους περιορισμούς και τις χρεώσεις της συμβατικής τηλεφωνίας. Τέτοιοι φορείς είναι η Altec με το i-call (<http://www.i-call.gr/>), το Skype (<http://www.skype.com/>) κ.α. Μια αρκετά μεγάλη λίστα υπάρχει στο <http://www.voipproviderslist.com/>

Τηλεόραση

Η έρευνα για μεταφορά φωνής μέσω του Internet συμπεριλαμβάνει και τη μεταφορά εικόνας σε πραγματικό χρόνο. Αυτή τη στιγμή υπάρχει η απαραίτητη τεχνογνωσία για τη μαζική μεταφορά εικόνας και ήχου αλλά δεν έχει αναπτυχθεί ακόμη η απαραίτητη υποδομή και θεωρείται πολύ πιθανό να υπάρξει απότομη στροφή στο μέσο μεταφορά της εικόνας στο κοντινό μέλλον. Το ραδιόφωνο μπορεί να θεωρηθεί ο πρόδρομος της τηλεόρασης και ήδη υπάρχει πληθώρα ραδιοφωνικών σταθμών, τόσο Ελληνικών όσο και ξένων, που εκπέμπουν σε πραγματικό χρόνο μέσω του Internet. Οι Έλληνες φοιτητές του εξωτερικού είναι οι κυριότεροι χρήστες των Ελληνικών αυτών σταθμών.

Έντυπα

Το Internet μπορεί να θεωρείται πλέον σαν η μεγαλύτερη πηγή και ο μεγαλύτερος φορέας πληροφορίας στον κόσμο, έχοντας πιθανότατα ξεπεράσει τις επιμέρους εναλλακτικές λύσεις. Τα νέα και η γνώση μεταφέρονται μέσω αυτού σε τεράστιες ποσότητες καθημερινά. Οι περισσότεροι άνθρωποι με πρόσβαση στο Internet το χρησιμοποιούν και σαν μέσο για την ενημέρωσή τους και για εύρεση πληροφοριών που τους ενδιαφέρουν. Τα μεγαλύτερα έντυπα περιοδικά έχουν αρχίσει να προσφέρουν την ύλη τους και σε ηλεκτρονική μορφή, τόσο παράλληλα, όσο και σαν εναλλακτική μορφή των εντύπων τους. Ορισμένοι εκδοτικοί οίκοι πουλάνε πλέον τα βιβλία τους και σε ηλεκτρονική μορφή μέσω του Internet σε όσους ενδιαφέρονται, ενώ υπάρχει ένα πολύ μεγάλο πλήθος βιβλίων και ηλεκτρονικών εντύπων τα οποία διανέμονται ελεύθερα. Τρανό παράδειγμα απότελεί το Wikipedia (<http://www.wikipedia.org/>) το οποίο συγκρίνεται από πολλούς με την εγκυκλοπαίδεια Britannica.

Πέρα από τα παραπάνω υπάρχουν πλέον υπηρεσίες του χράτους όπως το Taxis, οι οποίες είναι διαθέσιμες αποκλειστικά και μόνο μέσω του Internet. Αν μπορούσαμε να φανταστούμε τον κόσμο χωρίς το Internet για μία ημέρα θα λέγαμε ότι μάλλον δεν θα λειτουργούσε τίποτα. Αυτό οφείλεται στο ότι υπάρχει ένα σύνολο από υπηρεσίες στις οποίες βασιζόμαστε για να υλοποιήσουμε καθημερινές μας ανάγκες, εργασιακές και μη. Οι περισσότερες εταιρίες σήμερα στηρίζονται στην ανταλλαγή e-mail για να μεταφέρουν κείμενα και άλλες ηλεκτρονικές πληροφορίες μεταξύ των εργαζομένων τους, στους πελάτες τους και στους συνεργάτες τους.

Οι τεχνολογίες του Internet χρησιμοποιούνται και σε κλειστά δίκτυα για την εσωτερική διασύνδεση οργανισμών, εταιριών, τραπεζών, του στρατού και ίσως και άλλων, στα οποία δεν υπάρχει εξωτερική πρόσβαση. Αναλογιζόμενοι όλα αυτά αποκτάμε μια ιδέα της σημαντικότητας των τεχνολογιών των δικτύων που χρησιμοποιούνται σήμερα. Πράγματι, η ομαλή λειτουργία πολλών πραγμάτων όπως των τραπεζικών συναλλαγών είναι άρρητα δεμένη με την ομαλή και αδιάκοπη λειτουργία του εκάστοτε δικτύου. Για το λόγο αυτό έχουν ερευνηθεί και υλοποιηθεί διάφορες λύσεις οι οποίες κάνουν πραγματικότητα τη πολύ μεγάλη διαθεσιμότητα των δικτύων που συναντάμε σήμερα.

1.2 Τεχνολογίες Δικτύων

Όπως αναφέρθηκε και στην εισαγωγή, η επιτυχία του Internet οφείλεται μερικώς και στις δυνατότητες των τεχνολογιών που το δομούν. Η καρδιά όλων αυτών είναι η σουίτα πρωτοκόλλων TCP/IP και πιο συγκεκριμένα το Internet Protocol ή IP. Το TCP/IP με τη σειρά του έχει βασιστεί στο μοντέλο αναφοράς του OSI¹.

1.2.1 OSI Reference Model

Το μοντέλο αναφοράς του OSI γνωστό και σαν “τα 7 επίπεδα του OSI” είναι ένα μοντέλο για τη σωστή δόμηση πρωτοκόλλων δικτύων χρησιμοποιώντας τη λογική των επιπέδων. Τα επίπεδα που το αποτελούν φαίνονται στο σχήμα 1.2.1. Κάθε ένα από τα επίπεδα έχει προκαθορισμένες και συγκεκριμένες λειτουργίες και δυνατότητες. Σκοπός του κάθε επιπέδου είναι η παροχή υπηρεσιών στο παραπάνω επίπεδο με εξαίρεση το επίπεδο 7 το οποίο είναι αυτό που παρέχει τις περισσότερες υπηρεσίες στο χρήστη. Ακολουθώντας τη λογική αυτή, κάθε ένα πρωτόκολλο

¹Open Systems Interconnection

ανήκει σε ένα μόνο επίπεδο και μπορεί να χρησιμοποιήσει τις υπηρεσίες του ιδίου και των παρακάτω επιπέδων μόνο.

| |
|-----------------|
| 7. Application |
| 6. Presentation |
| 5. Session |
| 4. Transport |
| 3. Network |
| 2. Data Link |
| 1. Physical |

Σχήμα 1.2.1 Τα 7 επίπεδα του OSI

Ο σκοπός του κάθε επιπέδου είναι διαχριτός από τους υπόλοιπους:

1. Physical Layer

Αποστέλλει τα δεδομένα με μορφή σήματος στο φυσικό μέσο επικοινωνίας, π.χ. ρεύμα στο καλώδιο, ραδιοκύματα στον αέρα και φως στην οπτική ίνα.

2. Data Link Layer

Κάνει δυνατή την επικοινωνία μεταξύ δύο ή περισσοτέρων άμεσα συνδεδεμένων σταθμών. Π.χ. δύο υπολογιστές που έχουν μια ασύρματη κάρτα δίκτυου μπορούν και ανταλλάσσουν δεδομένα χάρη στο επίπεδο αυτό το οποίο καθορίζει πως ακριβώς θα στείλει ο κάθε σταθμός τα δεδομένα που έχει, πως θα τα ομαδοποιήσει, πως θα καταλάβει ο παραλήπτης ότι προορίζονται για αυτόν χλπ. Πρέπει να τονίσουμε ότι η επικοινωνία εδώ γίνεται ακόμη σε τοπικό επίπεδο. Στο επίπεδο αυτό ανήκουν τα πρωτόκολλα Ethernet (802.3), Frame Relay, PPP, 802.1a/b/g κ.α. και τα οποία μπορούν να χωριστούν σε δύο μεγάλες κατηγορίες:

- Στα Point-to-Point πρωτόκολλα όπου μόνο δύο σταθμοί είναι άμεσα συνδεδεμένοι μεταξύ τους, π.χ. οι dial-up συνδέσεις με modem όπου ο τοπικός υπολογιστής συνδέεται απευθείας και μόνο με τον απομακρυσμένο σταθμό.
- Στα πρωτόκολλα πολλαπλής πρόσβασης (Multiple Access) όπου υπάρχουν περισσότεροι από ένας σταθμοί που επικοινωνούν μεταξύ τους, π.χ. τα ασύρματα δίκτυα όπου όταν ένας σταθμός εκπέμπει δεδομένα πρέπει να τοποθετεί και μία “ταμπέλα” σε αυτά η οποία να λέει για ποιον σταθμό προορίζονται.

3. Network Layer

Εδώ γίνεται δυνατή η διαδικτυακή επικοινωνία, δηλαδή η επικοινωνία μεταξύ δικτύων. Η όλη επικοινωνία στο Internet οφείλεται στο αντίστοιχο επίπεδο του TCP/IP το οποίο επιτρέπει τα επιμέρους τοπικά δίκτυα να επικοινωνούν μεταξύ τους παρόλο που μπορεί σε κάθε ένα από αυτά να χρησιμοποιείται διαφορετική τεχνολογία. Το γεγονός αυτό επιτυγχάνεται χάρη στη δρομολόγηση.

4. Transport Layer

Επιτρέπει την αξιόπιστη επικοινωνία μεταξύ δύο τελικών σταθμών και εφαρμογών. Χάρη σε αυτό είναι δυνατή οι ταυτόχρονη μεταφορά δεδομένων πολλών εφαρμογών στον ίδιο υπολογιστή. Το αντίστοιχο επίπεδο του TCP/IP έχει ελαφρώς διαφορετική λειτουργία μιας και δεν απαιτεί η επικοινωνία να είναι αξιόπιστη.

5. Session Layer

Επιτρέπει τη δημιουργία, διακοπή και επαναλειτουργία sessions, ανεξαρτητοποιώντας τα από την επικοινωνία των παρακάτω επιπέδων.

6. Presentation Layer

Καθορίζει τη μορφή των δεδομένων που μεταφέρονται. Εδώ ανήκουν λειτουργίες χρυστογραφίας, μέθοδοι μεταφοράς δεδομένων όπως η XML κλπ.

7. Application Layer

Περιέχει τις εφαρμογές οι οποίες χρησιμοποιούνται από τον χρήστη.

1.2.2 TCP/IP Protocol Suite

Παρόλο που το μοντέλο του OSI αποτελεί τη βάση κάθε διδασκαλίας δικτύων, το Internet έχει στηριχθεί στη σουίτα πρωτοκόλλων που ονομάζεται TCP/IP. Και πάλι ακολουθείται η λογική των επιπέδων αλλά υπάρχει σαφής επικέντρωση στο διαδικτυακό κομμάτι. Το TCP/IP ασχολείται κυρίως με τα επίπεδα 3 και 4 του μοντέλου του OSI και συμπληρώνει άλλα δύο επίπεδα στα οποία συμπεριλαμβάνονται τα υπόλοιπα επίπεδα του OSI, ένα για τα επίπεδα 1-2 και ένα για τα επίπεδα 5-7, όπως φαίνεται στο σχήμα 1.2.2.

Οι λειτουργίες του κάθε επιπέδου είναι όμοιες με αυτές των αντίστοιχων επιπέδων του OSI. Τα πρωτόκολλα τα οποία καθορίζουν τη λειτουργία του TCP/IP αφορούν τα επίπεδα 2 και 3

| TCP/IP | OSI |
|-------------------|---|
| 4. Application | 7. Application 6. Presentation 5. Session |
| 3. Transport | 4. Transport |
| 2. Internet | 3. Network |
| 1. Network Access | 2. Data Link 1. Physical |

Σχήμα 1.2.2 Τα 4 επίπεδα του TCP/IP σε αντιστοιχία με τα 7 επίπεδα του OSI

(Internet και Transport). Για λόγους σαφήνειας θα αναφερόμαστε στα επίπεδα χρησιμοποιώντας το μοντέλο του OSI.

Η λειτουργία του TCP/IP και επομένως και του Internet βασίζεται στην δυνατότητα μεταφοράς δεδομένων μεταξύ δύο σταθμών, ασχέτως απόστασης και επιμέρους τεχνολογιών. Αυτό επιτυγχάνεται με το πρωτόκολλο IP, το οποίο είναι ένα best effort, μη αξιόπιστο² πρωτόκολλο 3ου επιπέδου.

1.2.2.1 Διευθυνσιοδότηση

Μια από τις σημαντικότερες λειτουργίες του IP είναι η δυνατότητα μοναδικής διευθυνσιοδότησης οποιουδήποτε διασυνδεδεμένου σταθμού στο Internet. Με άλλα λόγια κάθε ένας σταθμός έχει μία διεύθυνση με την οποία αναγνωρίζεται μοναδικά ανά τον κόσμο³

Οι διευθύνσεις που υποστηρίζει το IP είναι αριθμοί των 32bit και επομένως μπορούν να πάρουν τις τιμές 0–4294967295. Για να μπορούν να αναφερθούν, να διαβασθούν και να απομημονευθούν ευκολότερα, οι αριθμοί αυτοί χωρίζονται σε 4 “οκτάδες” των 8bit (octets) και αναγράφονται με τη μορφή A.B.C.D όπου A,B,C,D είναι οι δεκαδικές αναπαραστάσεις των τεσσάρων οκτάδων. Η γραφή αυτή είναι γνωστή με το όνομα Dotted Decimal Notation ή DDN.

Όπως αναφέρθηκε ήδη, το Internet αποτελείται από επιμέρους τοπικά δίκτυα συνδεδεμένα μεταξύ τους. Η διαδικασία της διευθυνσιοδότησης καλύπτει τόσο την αναφορά στους επιμέρους

²Ειρωνικά αναφέρεται πολλές φορές ότι ένα πρωτόκολλο που δεν μεταφέρει ποτέ δεδομένα θα μπορούσε να αποτελεί πρωτόκολλο τρίτου επιπέδου, αντίστοιχο με το IP.

³Η τεχνική του NAT (Network Address Translation) ουσιαστικά ομαδοποιεί ένα σύνολο από σταθμούς δημιουργώντας κάποιους περαιτέρω περιορισμούς και μπορεί με ασφάλεια να παραληφθεί χωρίς να δημιουργηθεί κάποια ασάφεια ή κάποιο πρόβλημα στους συλλογισμούς αυτής της εργασίας.

σταθμούς, όσο και την αναφορά στα δίκτυα αυτά. Ουσιαστικά γίνεται μια απεικόνιση της εικόνας του Internet στη διευθυνσιοδότηση. Κάθε ένα τοπικό δίκτυο προσδιορίζεται μοναδικά με έναν αριθμό ο οποίος ονομάζεται αριθμός δικτύου (N) και κάθε σταθμός μέσα σε αυτό προσδιορίζεται μοναδικά με ένα δεύτερο αριθμό (H). Έτσι, σύμφωνα με τη λογική αυτή μπορούμε να αναφερθούμε μοναδικά σε κάθε σταθμό χρησιμοποιώντας τον συμβολισμό N.H. Οι διευθύνσεις του IP χρησιμοποιούν αυτό ακριβώς το συμβολισμό (N.H) χρησιμοποιώντας κάποια bit τους σαν αριθμό δικτύου και κάποια άλλα σαν αριθμό σταθμού μέσα στο δίκτυο. Οποιοδήποτε πλήθος bit μπορεί να αφιερωθεί για κάθε ένα από τους δύο σκοπούς με μοναδικό περιορισμό το άθροισμα τους το οποίο πρέπει να είναι 32. Στο σχήμα 1.2.3 φαίνεται ένα αφηρημένο παράδειγμα στο οποίο χρησιμοποιούνται 20 bit ως αναγνωριστικό δικτύου και 12 bit ως αναγνωριστικό σταθμού.

| | |
|----------------------------|-----------------|
| NNNNNNNNNNNNNNNNNNNN | HHHHHHHHHHHH |
| Network | Host |
| 20 bit | 12 bit |
| NNNNNNNN . NNNNNNNN . NNNN | HHHH . HHHHHHHH |

Σχήμα 1.2.3: Παράδειγμα IP διεύθυνσης με 20 bit Network κομμάτι και 12 bit Host κομμάτι

Η αναφορά στα δίκτυα γίνεται χρησιμοποιώντας τον συμβολισμό A.B.C.D/X⁴, όπου A.B.C.D είναι ο αριθμός του δικτύου έχοντας 0 στα bit που προσδιορίζουν τους σταθμούς μέσα σε αυτό και X είναι ο αριθμός των συνεχόμενων από αριστερά bit που χρησιμοποιούνται για το προσδιορισμό του δικτύου. Τα bit τα οποία προσδιορίζουν το δίκτυο πολλές φορές αναφέρονται και σαν ένας αριθμός των 32 bit ο οποίος αναπαρίσταται με τη DDN του μορφή και σχηματίζεται θέτοντας όλα τα bit τα οποία προσδιορίζουν τον αριθμό του δικτύου ως 1 και όλα τα υπόλοιπα ως 0 και ονομάζεται Subnet Mask. Όλα αυτά φαίνονται στο παράδειγμα του σχήματος 1.2.4.

Σύμφωνα με τη λογική του CIDR η οποία είναι αυτή που χρησιμοποιείται σήμερα, η ανάθεση των διευθύνσεων στο Internet γίνεται ιεραρχικά με ανάθεση block διευθύνσεων σε providers και εφαρμόζοντας τη διαδικασία του subnetting. Έτσι, ένας μεγάλος provider μπορεί να λάβει το εύρος διευθύνσεων 120.120.0.0/15 το οποίο συμπεριλαμβάνει τις διευθύνσεις από 120.120.0.0 έως 120.121.255.255. Αυτός με τη σειρά του μπορεί να αναθέσει σε κάποιον μικρότερο provider, ο οποίος είναι “πελάτης” του, το εύρος διευθύνσεων 120.120.128.0/17 κ.ο.κ. Παρόλα αυτά στο

⁴Ο συμβολισμός A.B.C.D/X ονομάζεται CIDR (Common InterDomain Routing) μορφή και οφείλει την ονομασία του στη μετάβαση της διευθυνσιοδότησης στο Internet από classfull σε classless λογική.

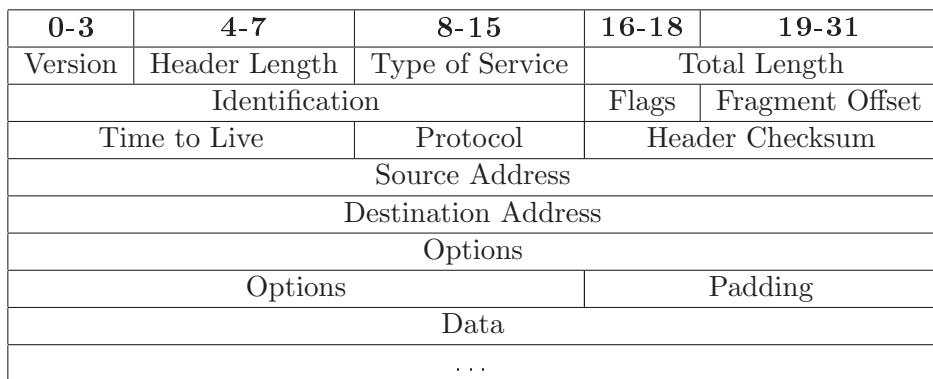
| | | |
|----------------------------|--------------------------------|------------------|
| | Network (20 bit) | Host (12 bit) |
| Aριθμός δικτύου - σταθμού: | 10110000110011000011 | 0000000000000000 |
| DDN ξεχωριστά: | 10110000 . 11001100 . 0011 | 0000 . 00000000 |
| DDN ενωμένα: | 10110000 . 11001100 . 00110000 | . 00000000 |
| Subnet Mask: | 11111111 . 11111111 . 11110000 | . 00000000 |
| Network Number: | 176 . 204 . 48 . 0 | 0 |
| CIDR: | 176 . 204 . 48 . 0 / 20 | |

Σχήμα 1.2.4: Παράδειγμα Αριθμού δικτύου με 20 bit Network κομμάτι και 12 bit Host κομμάτι

τέλος θα καταλήξουμε και πάλι να έχουμε ένα τοπικό δίκτυο το οποίο θα αναγνωρίζεται μοναδικά από τον αριθμό δικτύου του και θα έχει κάποια bit με τα οποία θα μπορεί να αριθμοδοτήσει τους σταθμούς που βρίσκονται μέσα σε αυτό. Π.χ. το 120.120.129.192/26 το οποίο είναι ένα δίκτυο που μπορεί να αριθμοδοτήσει μέχρι 64 σταθμούς εργασίας.

1.2.2.2 Μορφή του IP πακέτου

Το πρωτόκολλο IP (Postel 1981a) ορίζει τη μορφή που πρέπει να έχουν τα δεδομένα όταν αποστέλλονται στο δίκτυο, ή, με άλλα λόγια, τη μορφή του IP πακέτου, όπως αυτή φαίνεται στο σχήμα 1.2.5.



Σχήμα 1.2.5 Μορφή του IP πακέτου

Τα πεδία που το αποτελούν έχουν ως εξής:

Version

Περιέχει τη δυαδική τιμή “0100” η οποία υποδηλώνει πως πρόκειται για IPv4.

Header Length

Το μέγεθος σε πολλαπλάσια των 32 bit της επικεφαλίδας του IP. Το μέγεθος δεν είναι

σταυρό μιας και μπορούν να υπάρχουν μεταβλητού μεγέθους και πλήθους options μέσα στην επικεφαλίδα.

Type of Service

Ένα πεδίο το οποίο μπορεί να χρησιμοποιηθεί για το καθορισμό πολιτικής όταν υπάρχει συμφόρηση.

Total Length

Το συνολικό μέγεθος του IP πακέτου σε bytes. Ένα IP πακέτο μπορεί να έχει μέγεθος μέχρι 65536 bytes (64K).

Identification

Ένας αριθμός ο οποίος χρησιμοποιείται όταν γίνεται κατακερματισμός (fragmentation) του πακέτου από ενδιάμεσους σταυρούς. Η περιγραφή της διαδικασίας αυτής ξεφεύγει από τους στόχους της εργασίας.

Flags

Ορισμένα Flags τα οποία καθορίζουν παραμέτρους για τη συμπεριφορά του IP πακέτου.

Fragment Offset

Το byte της αρχικής ακολουθίας από το οποίο αρχίζει το συγκεκριμένο fragment. Χρησιμοποιείται σε συνδυασμό με το Identification και κάποια flags.

Time to Live

Ένας αριθμός ο οποίος παίρνει αρχική τιμή από τον αποστολέα και μειώνεται κατά ένα σε κάθε ενδιάμεσο σταυρό που παραλαμβάνει το IP πακέτο. Αν ο αριθμός αυτός γίνει 0 το πακέτο απορρίπτεται. Με τη τεχνική αυτή είναι σίγουρο ότι ένα IP πακέτο δεν πρόκειται ποτέ να ταξιδεύει επ'άπειρον στο Internet.

Protocol

Ένας αριθμός ο οποίος προσδιορίζει το πρωτόκολλο του παραπάνω επίπεδου το οποίο εμπεριέχεται μέσα στο IP πακέτο (π.χ. 6:TCP, 17:UDP κλπ).

Header Checksum

Ένα checksum από την επικεφαλίδα του IP πακέτου για να είναι δυνατή η ανίχνευση λαθών. Σε περίπτωση που υπάρχει κάποιο λάθος το IP πακέτο απορρίπτεται.

Source Address

Η IP διεύθυνση του αποστολέα.

Destination Address

Η IP διεύθυνση του παραλήπτη.

Options

Μια ακολουθία μεταβλητού μήκους η οποία περιέχει περαιτέρω πληροφορίες για τη συμπεριφορά του IP πακέτου.

Padding

Μια ακολουθία από μηδενικά, ώστε να γίνει το μέγεθος των Options πολλαπλάσιο των 32 bit.

Data

Τα δεδομένα που μεταφέρει και τα οποία μεταφράζονται σύμφωνα με τη τιμή του πεδίου Protocol.

1.3 Δρομολόγηση

Η δρομολόγηση είναι μια διαδικασία η οποία αποτελεί χαρακτηριστικό του τρίτου επιπέδου και είναι ο κύριος υπεύθυνος για την παράδοση (η όχι) των πακέτων των δεδομένων στον τελικό παραλήπτη. Η δρομολόγηση είναι η κεντρική λειτουργία εξειδικευμένων συσκευών οι οποίες ονομάζονται δρομολογητές και έχουν σαν σκοπό να λαμβάνουν IP πακέτα και να τα αποστέλλουν όπου χρειάζεται.

Μπορούμε να πούμε πως σε γενικές γραμμές ένας δρομολογητής αποτελείται από:

- Κάποιο κεντρικό επεξεργαστή.
- Ένα σύνολο από interfaces στα οποία συνδέονται τα επιμέρους δίκτυα που αναλαμβάνει να ενώσει.
- Κάποιο λειτουργικό σύστημα.
- Μνήμη.

Ένας δρομολογητής λαμβάνει IP πακέτα από ένα Interface και αναλαμβάνει να τα στείλει σε κάποιο άλλο ώστε αυτά να φτάσουν στον προορισμό τους. Η απόφαση για το αν και που

Θα τα στείλει ονομάζεται δρομολόγηση και λαμβάνεται με βάση τις καταχωρήσεις του πίνακα δρομολόγησης. Ο πίνακας δρομολόγησης είναι ένας πίνακας με εγγραφές οι οποίες περιέχουν τις παρακάτω πληροφορίες:

- Δίκτυο προορισμού
- Επόμενος κόμβος
- Interface
- Τον τρόπο με τον οποίο αποκτήθηκε η γνώση της συγκεκριμένης εγγραφής
- Ένα ενδεικτικό προτίμησης της συγκεκριμένης εγγραφής (Metric ή Administrative Distance)

Οι πληροφορίες αυτές έχουν τη μορφή του σχήματος **1.3.6**.

| Network | Next-Hop | Interface | Routing Protocol | A/D |
|---------------|-----------------|---------------------|------------------|-----|
| 195.251.123.0 | 195.251.240.23 | GigabitEthernet 0/0 | Static | 1 |
| 195.251.120.0 | 195.251.240.122 | FastEthernet 1/0 | OSPF | 110 |
| ... | | | | |

Σχήμα 1.3.6 Ένας πίνακας δρομολόγησης

1.3.1 Παράδειγμα πίνακα δρομολόγησης

Ένα απόσπασμα ενός πραγματικού πίνακα δρομολόγησης φαίνεται στο σχήμα 1.3.7. Κάθε γραμμή της εικόνας αυτής έχει μία από τις παρακάτω μορφές:

- *<P> <Network> is directly connected, <Interface>*
- *<P> <Network> [<AD>/<Metric>] via <Next-hop>*
- *<P> <Network> [<AD>/<Metric>] via <Next-hop>, <Age>, <Interface>*
- *<Network> is subnetted, <XX> subnets*

όπου:

P: Ένα γράμμα το οποίο υποδηλώνει το τρόπο με τον οποίο αποκτήθηκε η γνώση της συγκεκριμένης εγγραφής. Πιθανές τιμές είναι οι:

C: Directly Connected - Για δίκτυα στα οποία είναι άμεσα συνδεδεμένος ο δρομολογητής.

```

S      195.251.120.192/26 [1/0] via 195.251.240.245
D      195.251.120.128/26 [90/130816] via 195.251.240.227, 1d01h, GigabitEthernet2/1
C      192.168.12.0/24 is directly connected, Vlan2
      192.168.192.0/32 is subnetted, 11 subnets
D      192.168.192.8 [90/2298368] via 195.251.240.34, 2d05h, GigabitEthernet1/1
O E2    192.168.192.9 [110/20] via 195.251.240.62, 2d21h, GigabitEthernet4/4
D      192.168.192.10 [90/130816] via 195.251.240.34, 4d21h, GigabitEthernet1/1
D      192.168.192.11 [90/131072] via 195.251.240.34, 4d21h, GigabitEthernet1/1
C      192.168.192.2 is directly connected, Loopback1
D      192.168.192.3 [90/131072] via 195.251.240.34, 4d21h, GigabitEthernet1/1
O E2    192.168.192.5 [110/20] via 195.251.240.62, 2d21h, GigabitEthernet4/4
O E2    192.168.192.6 [110/20] via 195.251.240.62, 2d21h, GigabitEthernet4/4
D      192.168.192.7 [90/2298368] via 195.251.240.34, 4d21h, GigabitEthernet1/1

```

Σχήμα 1.3.7 Απόσπασμα πίνακα δρομολόγησης από δρομολογητή Cisco

S: Static Route - Εγγραφές οι οποίες έχουν καθοριστεί από το διαχειριστή με κατάλληλο configuration.

O: OSPF - Από το πρωτόκολλο δρομολόγησης OSPF (Open Shortest Path First). Οι συγκεκριμένες εγγραφές μπορούν να συνοδεύονται και από τα ενδεικτικά E1 και E2 τα οποία αποτελούν ιδιαιτερότητα του συγκεκριμένου πρωτοκόλλου.

D: EIGRP - Από το πρωτόκολλο δρομολόγησης EIGRP (Enhanced Interior Gateway Routing Protocol).

R: RIP - Από το πρωτόκολλο δρομολόγησης RIP (Routing Information Protocol).

Network: Ο αριθμός του δικτύου της εγγραφής.

AD: Administrative Distance - Είναι ένα νούμερο το οποίο καθορίζει τη σειρά προτίμησης αν υπάρχουν δύο ίδιες η αλληλοεπικαλυπτώμενες εγγραφές στον πίνακα δρομολόγησης. Οι εγγραφές με τη μικρότερη τιμή έχει μεγαλύτερη προτεραιότητα. Η τιμή αυτή ορίζεται από τον διαχειριστή αν πρόκειται για στατική εγγραφή (S), ή προκύπτει από το πρωτόκολλο δρομολόγησης από το οποίο προέρχεται η συγκεκριμένη εγγραφή, σύμφωνα με τον πίνακα 1.1.⁵

Metric: Είναι ένας αριθμός ο οποίος υπάρχει στο πίνακα δρομολόγησης για πληροφοριακούς και μόνο σκοπούς. Η τιμές που παίρνει είναι συγκρίσιμες μόνο μεταξύ εγγραφών οι οποίες προέρχονται από το ίδιο πρωτόκολλο δρομολόγησης.

⁵To Administrative Distance είναι χαρακτηριστικό των δρομολογητών Cisco και δεν συναντάται σε άλλους δρομολογητές. Παρόλα αυτά, παντού υπάρχει ένα αντίστοιχο ενδεικτικό στοιχείο το οποίο συνήθως ονομάζεται metric. Η προσέγγιση της Cisco στο συγκεκριμένο θέμα θεωρείται καλύτερη μιας και διαφοροποιεί το “εσωτερικό” metric του κάθε πρωτοκόλλου δρομολόγησης με τη σειρά προτίμησης του πρωτοκόλλου.

| A/D | Routing Protocol |
|-----|--------------------|
| 0 | Directly Connected |
| 1 | Static * |
| 5 | EIGRP Summary |
| 20 | External BGP |
| 90 | EIGRP |
| 100 | IGRP |
| 110 | OSPF |
| 120 | RIP |
| 170 | External EIGRP |
| 200 | Internal BGP |
| 255 | Unknown ** |

* Στα static routes η τιμή αυτή μπορεί να καθοριστεί από το διαχειριστή. Αν όχι τότε ισχύει το 1 όταν ορίζεται Next-Hop και το 0 όταν ορίζεται εξερχόμενο Interface.

** Η τιμή 255 καθορίζει πως η εγγραφή αυτή είναι σαν να μην υπάρχει. Δεν πρόκειται να εμφανιστεί ποτέ στον πίνακα δρομολόγησης μιας και ο δρομολογητής θα θεωρεί ότι η πηγή από την οποία προέρχεται η εγγραφή αυτή δεν είναι έγκυρη.

Πίνακας 1.1: Πίνακας αντιστοίχησης πρωτοκόλλων δρομολόγησης και Administrative Distance για Cisco δρομολογητές

Next-Hop: Η διεύθυνση του αμέσως επόμενου δρομολογητή, δηλαδή του δρομολογητή ο οποίος πιστεύεται πως είναι πιο κοντά στο προορισμό από οποιοδήποτε άλλο γειτονικό δρομολογητή. Αυτός είναι ο δρομολογητής στον οποίο στέλνονται τα πακέτα τα οποία προορίζονται για το συγκεκριμένο δίκτυο για περαιτέρω δρομολόγηση.

Interface: Η συγκεκριμένη τιμή έχει δύο μεταφράσεις:

1. Αν συνοδεύεται από Next-hop τότε είναι το Interface στο οποίο βρίσκεται ο δρομολογητής που αναφέρεται σαν Next-hop.
2. Αν δεν συνοδεύεται από Next-Hop τότε θεωρείται πως το δίκτυο βρίσκεται άμεσα συνδεδεμένο στο συγκεκριμένο Interface. Η δρομολόγηση προς το συγκεκριμένο δίκτυο δεν γίνεται με βάση το Next-hop, αλλά με απευθείας ARP request για τη destination address του IP πακέτου. Οι εγγραφές αυτές αναγράφουν “is directly connected”.

Age: Άλλη μία εντελώς πληροφοριακή ένδειξη η οποία δείχνει το χρονικό διάστημα κατά το οποίο η συγκεκριμένη εγγραφή βρίσκεται στο πίνακα δρομολόγησης.

Κάθε μία εγγραφή του πίνακα δρομολόγησης ονομάζεται “route” και πιο συγκεκριμένα “route προς το δίκτυο N”, όπου N το δίκτυο το οποίο αναγράφεται στη συγκεκριμένη εγγραφή.

Όταν ο δρομολογητής λάβει ένα IP πακέτο τότε εκτελεί κάποιες διεργασίες με τη παρακάτω σειρά:

1. Βλέπει τη διεύθυνση για τη οποία προορίζεται το πακέτο (destination address) (π.χ. 192.168.2.5)
 - Λαμβάνεται ο αριθμός του δικτύου μαζί με το network prefix που αναγράφεται και διαχωρίζονται αυτά τα δύο (π.χ. 192.168.1.0/24 → [192.168.1.0, 24]).
 - Σχηματίζεται η subnet mask από το network prefix όπως είχε αναφερθεί προηγουμένως (π.χ. 24 → 11111111 11111111 11111111 00000000 → 255.255.255.0).
 - Εφαρμόζεται λογικό KAI (AND) μεταξύ της διεύθυνσης προορισμού του IP πακέτου και της subnet mask (π.χ. 192.168.2.5 AND 255.255.255.0 → 192.168.2.0).
 - Ελέγχεται αν ο αριθμός που προέκυψε από το τελευταίο βήμα είναι ο αριθμός του δικτύου (π.χ. 192.168.2.0 == 192.168.1.0).
 - Αν είναι τότε η εγγραφή αυτή είναι η κατάλληλη, αλλιώς ελέγχεται η αμέσως επόμενη.
2. Ελέγχεται η πρώτη εγγραφή του πίνακα δρομολόγησης ως εξής:
 - Λαμβάνεται ο αριθμός του δικτύου μαζί με το network prefix που αναγράφεται και διαχωρίζονται αυτά τα δύο (π.χ. 192.168.1.0/24 → [192.168.1.0, 24]).
 - Σχηματίζεται η subnet mask από το network prefix όπως είχε αναφερθεί προηγουμένως (π.χ. 24 → 11111111 11111111 11111111 00000000 → 255.255.255.0).
 - Εφαρμόζεται λογικό KAI (AND) μεταξύ της διεύθυνσης προορισμού του IP πακέτου και της subnet mask (π.χ. 192.168.2.5 AND 255.255.255.0 → 192.168.2.0).
 - Ελέγχεται αν ο αριθμός που προέκυψε από το τελευταίο βήμα είναι ο αριθμός του δικτύου (π.χ. 192.168.2.0 == 192.168.1.0).
 - Αν είναι τότε η εγγραφή αυτή είναι η κατάλληλη, αλλιώς ελέγχεται η αμέσως επόμενη.
3. Εφόσον κάποια εγγραφή έχει ταιριάζει, ελέγχει αν αυτή δηλώνει το δίκτυο σαν άμεσα συνδεδεμένο, ή περιέχει κάποιο Next-hop:
 - Αν είναι άμεσα συνδεδεμένο ο δρομολογητής ελέγχει αν ξέρει την MAC address του υπολογιστή για τον οποίο προορίζεται το πακέτο. Αν δεν τη γνωρίζει κάνει ένα ARP request για να τη βρει. Στη συνέχεια στέλνει το πακέτο στο κατάλληλο Interface προς το σταθμό για τον οποίο προορίζεται.
 - Αν δεν είναι άμεσα συνδεδεμένο, τότε ελέγχει αν ξέρει τη MAC address του Next-hop. Αν όχι κάνει ένα ARP request για να τη βρει. Στη συνέχεια στέλνει το πακέτο στο κατάλληλο Interface προς το Next-hop.
4. Αν δεν έχει ταιριάζει καμιά εγγραφή ελέγχεται το default route αν υπάρχει. Αυτό είναι μια ειδική καταχώρηση με αριθμό δικτύου 0.0.0.0/0, η οποία χρησιμοποιείται όταν δεν έχει βρεθεί κάτι καλύτερο και εξετάζεται πάντα τελευταία. Και πάλι ισχύουν οι κανόνες του προηγούμενου βήματος.
5. Αν δεν υπάρχει κάποια εγγραφή που να ταιριάζει και δεν έχει καθοριστεί κάποιο default route τότε το πακέτο απορρίπτεται.

1.3.2 Τεχνικές βελτιστοποίησης

Η διαδικασία της εύρεσης της κατάλληλης εγγραφής μέσα στον πίνακα δρομολόγησης είναι μια σειριακή διαδικασία, η οποία πραγματοποιείται σε γραμμικό χρόνο ($O(x)$) εκτός και αν υλοποιηθεί κάποια τεχνική βελτιστοποίησης του αλγορίθμου. Οι τεχνικές βελτιστοποίησης δεν μπορούν να χρησιμοποιήσουν αλγόριθμους όπως hashing και δέντρα γιατί το πρόβλημα δεν είναι η απλή αναζήτηση, αλλά η “αναζήτηση της εγγραφής με το μεγαλύτερο πρόθεμα” (Best Matching Prefix (BMP)). Για το λόγο αυτό υπάρχουν βελτιώσεις (Marcel Waldvogel and George Varghese and Jon Turner and Bernhard Plattner 1997) όπως:

Τροποποιήσεις των κοινών αλγορίθμων αναζήτησης

Πρόκειται για λύσεις που στηρίζονται στους κλασσικούς αλγόριθμους οι οποίες απαιτούν λιγότερο χρόνο από τη σειριακή αναζήτηση:

- Ο Butler Lampson έχει προτείνει μια λύση η οποία ελαττώνει τις επαναλήψεις σε $\log_2(2N)$ (Radia Perlman 1992), όπου N είναι το μέγεθος του πίνακα δρομολόγησης σε εγγραφές, η οποία απαιτεί μία πρόσβαση στη μνήμη ανά επανάληψη.
- Υπάρχει άλλη μία λύση κατά την οποία γίνονται πολλαπλές αναζητήσεις, μία για κάθε πιθανό prefix (Keith Sklower 1993). Εδώ πραγματοποιείται πλήθος επαναλήψεων ίσο με το μέγεθος της διεύθυνσης σε bit και σε κάθε μία επανάληψη γίνεται αναζήτηση με κάποιο κλασσικό αλγόριθμο.

Χρήση Tries

Μια από τις πιο διαδεδομένες λύσεις είναι αυτή που χρησιμοποιείται στον kernel του BSD (Keith Sklower 1993). Στηρίζεται σε ένα radix trie στο οποίο αποθηκεύονται οι εγγραφές και για την αναζήτηση σε αυτό χρειάζονται το πολύ $\Omega(W)$ επαναλήψεις, όπου W είναι το μέγεθος της διεύθυνσης σε bit (32 για IPv4). Ο αλγόριθμος κάνει μία προσπέλαση στη μνήμη για κάθε επανάληψη και έχει σχετικά υψηλές απαιτήσεις σε μνήμη.

Λύσεις στηριγμένες στο Hardware - Content Addressable Memory (CAM)

Η CAM είναι ένα είδος μνήμης το οποίο μπορεί να κάνει αναζήτηση σύμφωνα με τα δεδομένα και όχι με τη διεύθυνση μνήμης. Ένα ακριβές ταίριαγμα (χωρίς prefix) απαιτεί μία μόνο επανάληψη με μία μόνο προσπέλαση στη μνήμη. Η CAM είναι ακριβή και η χρήση της σε συνδυασμό με τις προηγούμενες τεχνικές δεν έχει καλά αποτελέσματα μιας

και η αναζήτηση σε αυτή μπορεί να είναι πιο αργή από τη συμβατική μνήμη (συγχρίνοντας μία αναζήτηση με βάση τα περιεχόμενα και με βάση τη διεύθυνση).

Route cache

Η πιο διαδεδομένη τεχνική σε δρομολογητές είναι η χρήση μιας cache. Η route cache κρατάει μέσα τα αποτελέσματα των αναζητήσεων στους πίνακες δρομολόγησης. Αυτό έχει σαν αποτέλεσμα να υπάρχει μια δεύτερη μνήμη η οποία αποθηκεύει εγγραφές της μορφής Destination IP \Rightarrow Routing entry. Η αναζήτηση σε αυτή τη μνήμη μπορεί πλέον να γίνει με όλους τους συμβατικούς αλγόριθμους σε μικρότερους χρόνους. Το καλύτερο δυνατό αποτέλεσμα της τεχνικής αυτής επιτυγχάνεται με τη χρήση Route cache και την αποθήκευσή της σε CAM. Η τεχνική αυτή μπορεί να εφαρμοστεί μόνο σε hardware λύσεις, όπως είναι οι δρομολογητές, γιατί απαιτεί την ύπαρξη CAM η οποία είναι αρκετά εξειδικευμένη και ακριβή.

1.3.3 Είδη δρομολόγησης

Για να λειτουργήσει επιτυχώς ένα οποιοδήποτε δίκτυο, οποιασδήποτε κλίμακας, είτε πρόκειται για ένα δίκτυο με ένα δρομολογητή, είτε πρόκειται για το Internet είναι απαραίτητο να υπάρχει σωστή δρομολόγηση. Γενικά, υπάρχουν δύο τρόποι για να γίνει κάτι τέτοιο:

1. Οι διαχειριστές του δικτύου να κάνουν το απαραίτητο configuration σε όλους τους δρομολογητές του δικτύου, βάζοντας εκατοντάδες ή χιλιάδες εγγραφές σε κάθε δρομολογητή και προσέχοντας μην κάνουν κάποιο τυπογραφικό λάθος κατά την εισαγωγή των δεδομένων. Η λύση αυτή είναι γνωστή και σαν “στατική δρομολόγηση”.
2. Να χρησιμοποιηθεί κάποιο δυναμικό πρωτόκολλο δρομολόγησης. Η αντιμετώπιση αυτή ονομάζεται “δυναμική δρομολόγηση”.

Από το πρώτο καιρό της ύπαρξης του Internet, η πρώτη λύση αποτέλεσε την εξαίρεση και η δεύτερη τον κανόνα⁶ γιατί είναι πρακτικά αδύνατο να μπορέσει κάποιος να υλοποιήσει 100% στατική δρομολόγηση σε ένα δίκτυο μεσαίου μεγέθους.

Εδώ πρέπει να αναφέρουμε ότι το Internet είναι χωρισμένο σε μεγάλες αυτόνομες μονάδες οι οποίες ονομάζονται “Αυτόνομα Συστήματα”, “Autonomous Systems” ή “AS”, κάθε μία από τις οποίες είναι ένα δίκτυο μεγάλου μεγέθους κάτω από κοινή διαχείριση. Κάθε Αυτόνομο

⁶Ο αλγόριθμος των Bellman-Ford, ο οποίος χρησιμοποιείται από το RIP, χρησιμοποιήθηκε το 1969 σαν ο αρχικός αλγόριθμος δρομολόγησης του ARPANET, το οποίο αποτελεί το πρόγονο του Internet

Σύστημα εφαρμόζει εσωτερικά τους δικούς του κανόνες και επικοινωνεί με άλλα Αυτόνομα Συστήματα σύμφωνα με τη πολιτική που ακολουθεί.

Εξαιτίας της αδυναμίας της χρήσης της στατικής δρομολόγησης, εφευρέθηκαν και χρησιμοποιούνται μια πληθώρα δυναμικών πρωτοκόλλων δρομολόγησης, τα οποία με τη σειρά τους χωρίζονται σε δύο μεγάλες κατηγορίες.

EGPs: Exterior Gateway Protocols - Πρωτόκολλα τα οποία αναλαμβάνουν να σχηματίσουν τη δρομολόγηση μεταξύ Αυτόνομων Συστημάτων. Τα κριτήρια για την επιλογή των καλύτερων διαδρομών καθορίζονται από τους διαχειριστές σύμφωνα με την πολιτική που θέλουν να ακολουθήσουν. Για το λόγο αυτό τα EGPs δίνουν τη δυνατότητα στους διαχειριστές να παρέμβουν στις αποφάσεις που παίρνουν, βάζοντας παραμέτρους σύμφωνα με κάποια πολιτική. Η ανάγκη για ύπαρξη πολιτικής οφείλεται στην ύπαρξη οικονομικών συμφωνιών και στο ότι τα διάφορα Αυτόνομα Συστήματα δεν βρίσκονται κάτω από κοινή διαχείριση. Στα EGPs υεωρείται πως τα δεδομένα που λαμβάνει ένας δρομολογητής για τις ανάγκες της δρομολόγησης δεν είναι αξιόπιστα και επομένως υπάρχει και η ανάγκη προσθήκης περιορισμών και φίλτρων πέρα από τους κανόνες πολιτικής. Το Internet αυτή τη στιγμή χρησιμοποιεί στο σύνολό του το BGP (Border Gateway Protocol) ως EGP, το οποίο βρίσκεται στην έκδοση 4 και είναι ένα ανοιχτό και πάρα πολύ ισχυρό πρωτόκολλο δυναμικής εξωτερικής δρομολόγησης.

IGPs: Interior Gateway Protocols - Πρωτόκολλα τα οποία αναλαμβάνουν να σχηματίσουν τη δρομολόγηση στο εσωτερικό ενός Αυτόνομου Συστήματος. Θεωρείται πως ένα Αυτόνομο Σύστημα βρίσκεται κάτω από κοινή διαχείριση και συνεπώς δεν τίθενται θέματα εμπιστοσύνης. Επακόλουθο αυτού είναι το να μην υπάρχει η ανάγκη εφαρμογής κανόνων πολιτικής και πλέον δίνεται βάση στην ευκολία εγκατάστασης και συντήρησης, στις δυνατότητες και στη ταχύτητα λειτουργίας. Υπάρχει μια πληθώρα από IGP με επικρατέστερα τα RIP, OSPF και EIGRP⁷.

⁷Το EIGRP είναι αποκλειστικότητα της Cisco και δεν αποτελεί κάποιο ανοιχτό πρωτόκολλο. Συναντάται μόνο σε συσκευές της συγκεκριμένης εταιρίας και για αυτό το λόγο δεν έχει ευρεία διάδοση στο Internet

1.3.4 Είδη πρωτοκόλλων δρομολόγησης

Στο Internet μπορούμε να συναντήσουμε μια πληθώρα από IGP τα οποία χωρίζονται σε δύο κατηγορίες⁸, με κάθε μία να έχει πλεονεκτήματα και μειονεκτήματα. Οι κατηγορίες αυτές είναι γενικότερες στη λογική της δρομολόγησης και συναντώνται και σε πρωτόκολλα δρομολόγησης πέρα από το IP και το Internet⁹.

1.3.4.1 Distance Vector Routing Protocols

Τα Distance Vector πρωτόκολλα είναι τα πρώτα που χρησιμοποιήθηκαν στο Internet με κυριότερο εκπρόσωπο τους το RIP. Η λογική των Distance Vector πρωτοκόλλων λέει ότι κάθε δρομολογητής συλλέγει όσες πληροφορίες μπορεί για το δίκτυο και σχηματίζει τον πίνακα δρομολόγησής του. Στη συνέχεια στέλνει τα δεδομένα του πίνακα δρομολόγησής του (σχεδόν ολόκληρο τον πίνακα δρομολόγησης) σε όλους τους γειτονικούς του δρομολογητές. Αυτοί με τη σειρά τους συλλέγουν τα νέα στοιχεία, τα ενώνουν με τα ήδη υπάρχοντα και τα στέλνουν σε όλους τους δικούς τους γείτονες, συμπεριλαμβανομένου και του δρομολογητή που τους τα έστειλε. Μετά από μερικές επαναλήψεις όλοι οι δρομολογητές θα έχουν αποκτήσει “κοινή¹⁰” εικόνα του δικτύου και τότε λέμε πως το δίκτυο (και οι δρομολογητές) έχουν συγκλίνει (converged).

Τα Distance Vector πρωτόκολλα έχουν πολύ λίγες απαιτήσεις σε μνήμη και υπολογιστική ισχύ αλλά προκαλούν μια πληθώρα από προβλήματα τα οποία πολλές φορές δεν είναι εύκολο να λυθούν. Η καρδιά όλων των προβλημάτων τους είναι το ότι οι δρομολογητές αφού λάβουν τη πληροφορία, την επεξεργάζονται και στη συνέχεια στέλνουν στους γείτονές τους τα συμπεράσματα που οι ίδιου βγάζουν και όχι την αρχέγονη πληροφορία. Αυτό συνεπάγεται πως η πληροφορία που λαμβάνει ένας δρομολογητής δεν είναι πάντα σωστή μιας και υπάρχουν αρκετοί παράγοντες που μπορούν να οδηγήσουν ένα δρομολογητή στο να έχει βγάλει λάθος συμπεράσματα σε μια δεδομένη χρονική στιγμή.

Τα λάθη στα Distance Vector πρωτόκολλα προκύπτουν όταν υπάρχουν αναταραχές στο δίκτυο και πρέπει να ενημερωθούν όλοι οι δρομολογητές για τα νέα δεδομένα. Στη περίπτω-

⁸ Υπάρχει και μια τρίτη κατηγορία η οποία προκαλεί δρομολόγηση on-demand αλλά δεν συναντάται στον Internet μιας και οι εφαρμογές της είναι σε δίκτυα τα οποία αλλάζουν τη δομή τους συνεχώς

⁹ Πρωτόκολλα δρομολόγησης χρησιμοποιούνται και για την επιτυχή μεταφορά δεδομένων στο δεύτερο επίπεδο σε Ad-Hoc δίκτυα, στο 7ο επίπεδο σε Peer-to-peer δίκτυα και αλλού.

¹⁰ Κοινή εικόνα δεν σημαίνει ίδια. Όταν λέμε κοινή εικόνα σημαίνει πως οι δρομολογητές μπορούν πλέον να λειτουργήσουν όπως χρειάζεται ώστε να μεταφέρονται τα δεδομένα με το καλύτερο δυνατό τρόπο μέσα στο δίκτυο.

ση αυτή, εφόσον δεν μπορεί να υπάρξει ακαριαία ενημέρωση των πάντων, υπάρχουν χρονικά διαστήματα κατά τα οποία οι δρομολογητές έχουν διαφορετική εικόνα για το δίκτυο. Όταν κάποιος δρομολογητής έχει “λάθος” πίνακα δρομολόγησης είναι πολύ πιθανό να μην προλάβει να ενημερωθεί και επομένως να μεταφέρει στους γείτονές του τη λανθασμένη γνώση που ο ίδιος κατέχει. Χαρακτηριστικό παράδειγμα είναι ότι το RIP μπορεί να χρειαστεί μέχρι και 7 λεπτά¹¹ για να οδηγήσει ένα δίκτυο σε σύγκλιση μετά από κάποια αναταραχή.

Για την αντιμετώπιση των λαθών έχουν ανακαλυφθεί τεχνικές όπως οι:

- Split-Horizon
- Poison Reverse
- Route Poisoning
- Holddown timers

οι οποίες όταν εφαρμόζονται σε Distance Vector πρωτόκολλα τα οποία λειτουργούν με triggered updates, δηλαδή σε πρωτόκολλα στα οποία οι δρομολογητές στέλνουν ενημερώσεις αμέσως μόλις καταλάβουν κάποια αλλαγή στο δίκτυο, μπορούν ουσιαστικά να εξαλείψουν τα παραπάνω προβλήματα. Παρόλα αυτά υπάρχουν περιπτώσεις όπου η εφαρμογή των τεχνικών αυτών δεν είναι δυνατή ή επιθυμητή λόγω των παρενεργειών που έχουν.

1.3.4.2 Link State Routing Protocols

Τα Link State πρωτόκολλα δρομολόγησης έκαναν την εμφάνισή τους μετά από τα Distance Vector για να αντιμετωπίσουν τα προβλήματα που αναφέραμε. Γνωστότερος εκπρόσωπός τους είναι το OSPF. Τα Link State πρωτόκολλα στηρίζουν τη λειτουργία τους στη καθολική γνώση του δικτύου και στον Shortest Path αλγόριθμο, γνωστό και σαν “αλγόριθμος του Dijkstra”.

Στα Link State πρωτόκολλα ο κάθε δρομολογητής είναι υπεύθυνος για να ενημερώνει τους υπόλοιπους δρομολογητές με τα δικά του στοιχεία. Μόλις συμβιούν αλλαγές στη κατάσταση ενός δρομολογητή, αυτός αναλαμβάνει να ενημερώσει όλους τους άλλους για τις αλλαγές αυτές. Η ενημέρωση γίνεται με κάποια τεχνική flooding και καταλήγει σε όλους τους δρομολογητές. Με το τρόπο αυτό κάθε ένας δρομολογητής αποκτά καθολική γνώση του δικτύου και ίδια με όλους τους υπόλοιπους. Στη συνέχεια εφαρμόζεται ο αλγόριθμος του Dijkstra από όλους

¹¹Ο χρόνος αυτός αποτελεί τον κανόνα και όχι κάποιο ακραίο παράδειγμα και προκύπτει από θεωρητική μελέτη των πιο συνηθισμένων τοπολογιών δικτύου

τους δρομολογητές και εφόσον όλοι έχουν την ίδια ακριβώς εικόνα του δικτύου προκύπτει το επιθυμητό αποτέλεσμα. Τα Link State πρωτόκολλα ονομάζονται έτσι επειδή οι δρομολογητές στέλνουν σε κάθε αλλαγή τη νέα κατάστασή τους, η οποία περιλαμβάνει κυρίως την κατάσταση των συνδέσεών τους με άλλους δρομολογητές και δίκτυα. Οι ενημερώσεις αυτές στέλνονται σε πακέτα δεδομένων τα οποία ονομάζονται Link State Advertisements (LSAs).

Τα Link State πρωτόκολλα απαιτούν περισσότερη μνήμη από τα Distance Vector γιατί ο κάθε ένας δρομολογητής πρέπει να κρατάει όλα τα Link State Advertisements που έχει λάβει. Επίσης έχουν μεγαλύτερες ανάγκες σε επεξεργαστική ισχύ γιατί κάθε φορά που συμβαίνει κάποια αλλαγή στο δίκτυο οι δρομολογητές εφαρμόζουν ξανά τον αλγόριθμο του Dijkstra στα νέα δεδομένα. Παρόλα αυτά έχουν καλύτερα αποτελέσματα μιας και δεν εμφανίζουν τα προβλήματα των Distance Vector πρωτοκόλλων και καταναλώνουν λιγότερους δικτυακούς πόρους μιας και όταν υπάρχουν αλλαγές στο δίκτυο ανακοινώνουν μόνο τα νέα δεδομένα και όχι ολόκληρους τους πίνακες δρομολόγησης, όπως κάνουν τα Distance Vector πρωτόκολλα.

Κεφάλαιο 2

Open Shortest Path First

2.1 Γενικά

To Open Shortest Path First (OSPF) είναι ένα ανοιχτό, Link State πρωτόκολλο δρομολόγησης, το οποίο χρησιμοποιείται κατά κόρον στο Internet. Ορίστηκε αρχικά με το RFC 1131 (Moy 1989) και στη συνέχεια αναθεωρήθηκε και επεκτάθηκε αρκετές φορές¹. Σήμερα χρησιμοποιείται το OSPFv2 το οποίο ορίζεται με το RFC 2328 (Moy 1998) και το OSPFv3 (Coltun, Ferguson, and Moy 1999), το οποίο αποτελεί επέκταση του OSPFv2 για να υποστηρίζει το IPv6. Πρόκειται για ένα σύγχρονο πρωτόκολλο με πρακτικά απεριόριστες δυνατότητες επέκτασης. Τα σημαντικότερα πλεονεκτήματα του είναι:

- Είναι Link State.
- Είναι ανοιχτό πρωτόκολλο.
- Υποστηρίζει Classless Routing.
- Υποστηρίζει Variable Length Subnet Masks.
- Είναι ασφαλές χρησιμοποιώντας αυθεντικοποίηση.
- Κάνει δυνατή την εραρχική σχεδίαση των δικτύων και συνεπώς της δρομολόγησης.
- Είναι επεκτάσιμο.
- Υποστηρίζει όλα τα είδη δικτύων (Point-to-point, Broadcast, Non Broadcast Multi-access (NBMA)).

¹<http://www.ietf.org/html.charters/ospf-charter.html>

- Μπορεί να χρησιμοποιηθεί και σαν πρωτόκολλο δρομολόγησης για multicast πακέτα (Moy 1994).
- Υποστηρίζει πολλαπλές διαδρομές ιδίου χόστους.

2.1.1 Περιοχές στο OSPF

Ένα από τα σημαντικότερα πλεονεκτήματα του OSPF είναι το ότι παρέχει τη δυνατότητα στους διαχειριστές να χωρίσουν ένα δίκτυο (ή καλύτερα ένα Αυτόνομο Σύστημα (AS)) σε μικρότερες περιοχές (areas). Κάθε μία περιοχή λειτουργεί εν μέρη σαν ένα αυτόνομο κομμάτι, μερικώς ανεξάρτητο από το υπόλοιπο δίκτυο, με αποτέλεσμα:

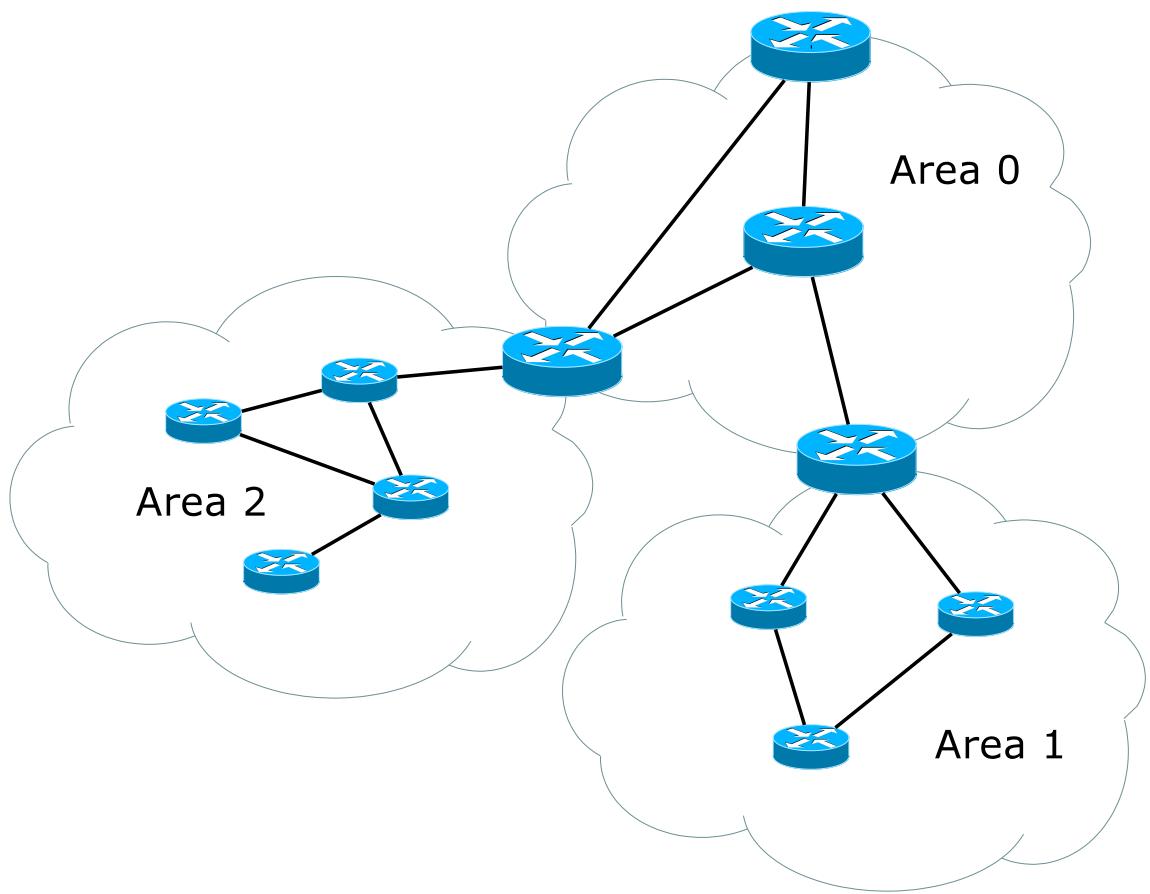
- Οι αλλαγές σε μία περιοχή να μην προκαλούν αστάθεια στις υπόλοιπες.
- Οι δρομολογητές μιας περιοχής να χρειάζεται να κατέχουν μόνο τις απαραίτητες πληροφορίες της κάθε περιοχής και όχι του συνόλου του δικτύου, μειώνοντας κατά πολύ τις απαιτήσεις σε μνήμη και υπολογιστική ισχύ.
- Η δρομολόγηση, η σχεδίαση και η διαχείριση να μπορούν να γίνονται ιεραρχικά και από διαφορετικές ομάδες, χωρίς να επηρεάζει η μία την άλλη.

Στο σχήμα 2.1.1 φαίνεται ένα παράδειγμα δικτύου με τρεις περιοχές. Η κάθε μία περιοχή έχει έναν αριθμό που τη χαρακτηρίζει και ο οποίος μπορεί να είναι ένας από τους:

1–65535: Υποδηλώνει μια κανονική περιοχή στην οποία λειτουργεί το OSPF σχεδόν ανεξάρτητα από τις υπόλοιπες περιοχές

0: Αποτελεί ειδική περιοχή η οποία ονομάζεται backbone. Είναι αυτή μέσω της οποίας επικοινωνούν όλες οι υπόλοιπες περιοχές. Π.χ. για να μεταφερθούν πληροφορίες από τη περιοχή 1 στη περιοχή 2 είναι απαραίτητο να περάσουν από τη περιοχή 0.

Όπως φαίνεται και στο σχήμα, υπάρχουν δρομολογητές οι οποίοι είναι συνδεδεμένοι σε περισσότερες από μία περιοχής. Κάθε interface ενός δρομολογητή μπορεί να ανήκει σε διαφορετική περιοχή και μέσω αυτού να πραγματοποιείται η επικοινωνία μεταξύ των δύο. Οι δρομολογητές που βρίσκονται στα “άκρα” μιας περιοχής και την ενώνουν με το backbone ονομάζονται Area Border Routers (ABRs) και εκτελούν περισσότερες λειτουργίες από τους υπόλοιπους. Όλοι οι ABRs είναι συνδεδεμένοι μέσω της περιοχής 0 και επικοινωνούν μέσω αυτής.



Σχήμα 2.1.1 Παράδειγμα OSPF δικτύου με περιοχές

Πέρα από τους ABRs υπάρχουν και οι δρομολογητές οι οποίοι αναλαμβάνουν να ενώσουν το παρόν Αυτόνομο Σύστημα με κάποιο άλλο Αυτόνομο Σύστημα (π.χ. με κάποιο πάροχο για πρόσβαση στο Internet). Οι δρομολογητές αυτοί λέμε ότι βρίσκονται στα “όρια” του Αυτόνομου Συστήματος και ονομάζονται AS Boundary Routers (ASBRs). Τέλος, οι δρομολογητές που βρίσκονται μέσα στο backbone ονομάζονται backbone routers και έχουν σαν βασικό σκοπό τη δρομολόγηση κίνησης μεταξύ των υπολοίπων περιοχών.

Σε απλές περιπτώσεις το OSPF λειτουργεί με μία μόνο περιοχή (το backbone) στην οποία εμπεριέχεται ολόκληρο το δίκτυο. Για τις ανάγκες αυτής της εργασίας δεν θα ασχοληθούμε καθόλου με περιπτώσεις περισσότερων από μία περιοχή γιατί κάτι τέτοιο δεν είναι απαραίτητο. Λόγω του τρόπου λειτουργίας του OSPF, όσα θα αναφέρουμε μπορούν να εφαρμοστούν και σε περισσότερες από μία περιοχής χωρίς καμιά τροποποίηση.

2.1.2 Δίκτυα στο OSPF

Το OSPF αναγνωρίζει τα παρακάτω είδη δικτύων²:

Point-to-point

Πρόκειται για την πιο απλή περίπτωση σύνδεσης δύο κόμβων όπου συνδέονται απευθείας.

Λέγοντας Point-to-point δίκτυα εννοούμε κάθε δίκτυο το οποίο υποστηρίζει μόνο point-to-point σύνδεση και αποκλείει την περίπτωση να υπάρξει και τρίτος σταθμός. Π.χ. ένα Ethernet link μεταξύ δύο δρομολογητών δεν θεωρείται point-to-point γιατί το Ethernet δεν είναι Point-to-point τεχνολογία. Στη κατηγορία αυτή ανήκουν χυρίως οι WAN συνδέσεις με PPP, Frame Relay με Point-to-Point subinterfaces κλπ.

Broadcast

Εδώ ανήκουν τα περισσότερα δίκτυα ανά τον κόσμο μιας και αυτά χρησιμοποιούν την τεχνολογία του Ethernet. Γενικά, οποιοδήποτε δίκτυο υποστηρίζει περισσότερους από έναν κόμβους και παρέχει τη δυνατότητα broadcast³, ανήκει σε αυτή τη κατηγορία.

Non-Broadcast Multi-Access (NBMA)

Πρόκειται για την πιο σπάνια κατηγορία μιας και βρίσκει εφαρμογή μόνο στο Frame Relay.

Η κατηγορία αυτή συνεπάγεται πως ένας σταθμός μπορεί να βρίσκεται στο ίδιο δίκτυο με

²Στη πραγματικότητα υπάρχουν και μερικές ακόμη περιπτώσεις αλλά η περιγραφή τους είναι πέρα από τους σκοπούς της εργασίας αυτής.

³Για να είναι broadcast ένα δίκτυο πρέπει η τεχνολογία του να παρέχει ένα τρόπο ώστε να μπορεί ο αποστολέας να στέλνει κάτι και να το λαμβάνουν όλοι οι κόμβοι που βρίσκονται μέσα σε αυτό.

άλλους, αλλά είτε δεν μπορεί να στείλει broadcasts, είτε δεν μπορεί να επικοινωνήσει με όλους.

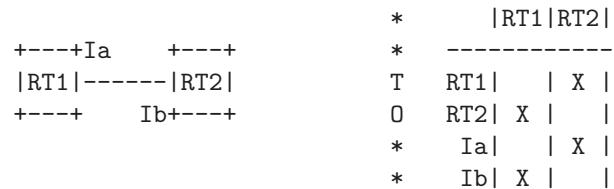
2.2 Ο γράφος

Μετά από την ανταλλαγή των πληροφοριών κάθε δρομολογητής σχηματίζει ένα γράφο που αναπαριστά το δίκτυο. Στο γράφο αυτό κάθε δρομολογητής είναι και ένας κόμβος. Πέρα από τους δρομολογητές υπάρχει και ένας κόμβος για κάθε ένα Broadcast δίκτυο. Μετά από τους κόμβους τοποθετούνται και οι συνδέσεις σαν ακμές ως εξής:

- Κάθε μία αμφιδρομη σύνδεση τοποθετείται σαν δύο μονόδρομες. Αυτό έχει σαν αποτέλεσμα να υποστηρίζονται και συνδέσεις με μη συμμετρικές ταχύτητες.
- Κάθε μία point-to-point σύνδεση τοποθετείται σαν μία σύνδεση των δύο δρομολογητών με το απέναντι interface τους.
- Κάθε μία σύνδεση σε τελικό δίκτυο τοποθετείται σαν μια μονόδρομη σύνδεση από το δρομολογητή προς το δίκτυο αυτό.
- Κάθε μία σύνδεση ενός δρομολογητή με ένα broadcast δίκτυο τοποθετείται σαν σύνδεση του δρομολογητή με το δίκτυο και του δικτύου με το δρομολογητή.
- τα NBMA δίκτυα ανάγονται σε point-to-point με τη τεχνική point-to-multipoint ή σε broadcasts και αντιμετωπίζονται ανάλογα.

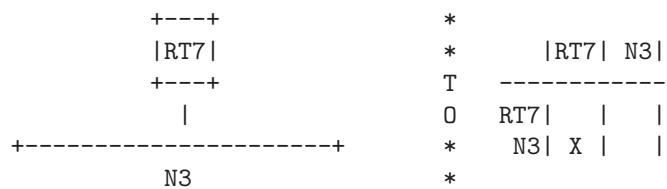
Στο σχήμα 2.2.2 φαίνονται τα τρία είδη δικτύων και οι πίνακες οι οποίοι δείχνουν τις συνδέσεις που θα σχηματιστούν. Η παραπάνω διαδικασία έχει σαν αποτέλεσμα να κατασκευαστεί ένας κατευθυντικός γράφος με κόμβους τους δρομολογητές και τα broadcast δίκτυα και ακμές τις συνδέσεις που υπάρχουν. Στο γράφο αυτό κάθε σύνδεση έχει και ένα “βάρος” το οποίο ονομάζεται “κόστος” (cost) της σύνδεσης και είναι αυτό το οποίο καθορίζει το τελικό αποτέλεσμα της δρομολόγησης. Το RFC του OSPF δεν ορίζει κάποια μέθοδο για αυτόματο καθορισμό του κόστους μιας και κάτι τέτοιο δεν θα μπορούσε να λάβει υπόψη πιθανές μελλοντικές ταχύτητες και συνίσταται ο καθορισμός από τους διαχειριστές του συστήματος.

FROM



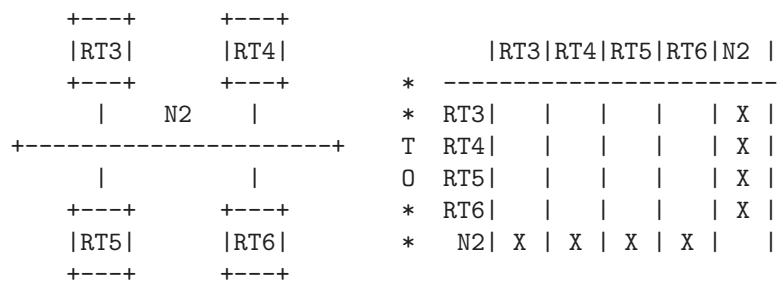
Physical point-to-point networks

FROM



Stub networks

FROM



Broadcast networks

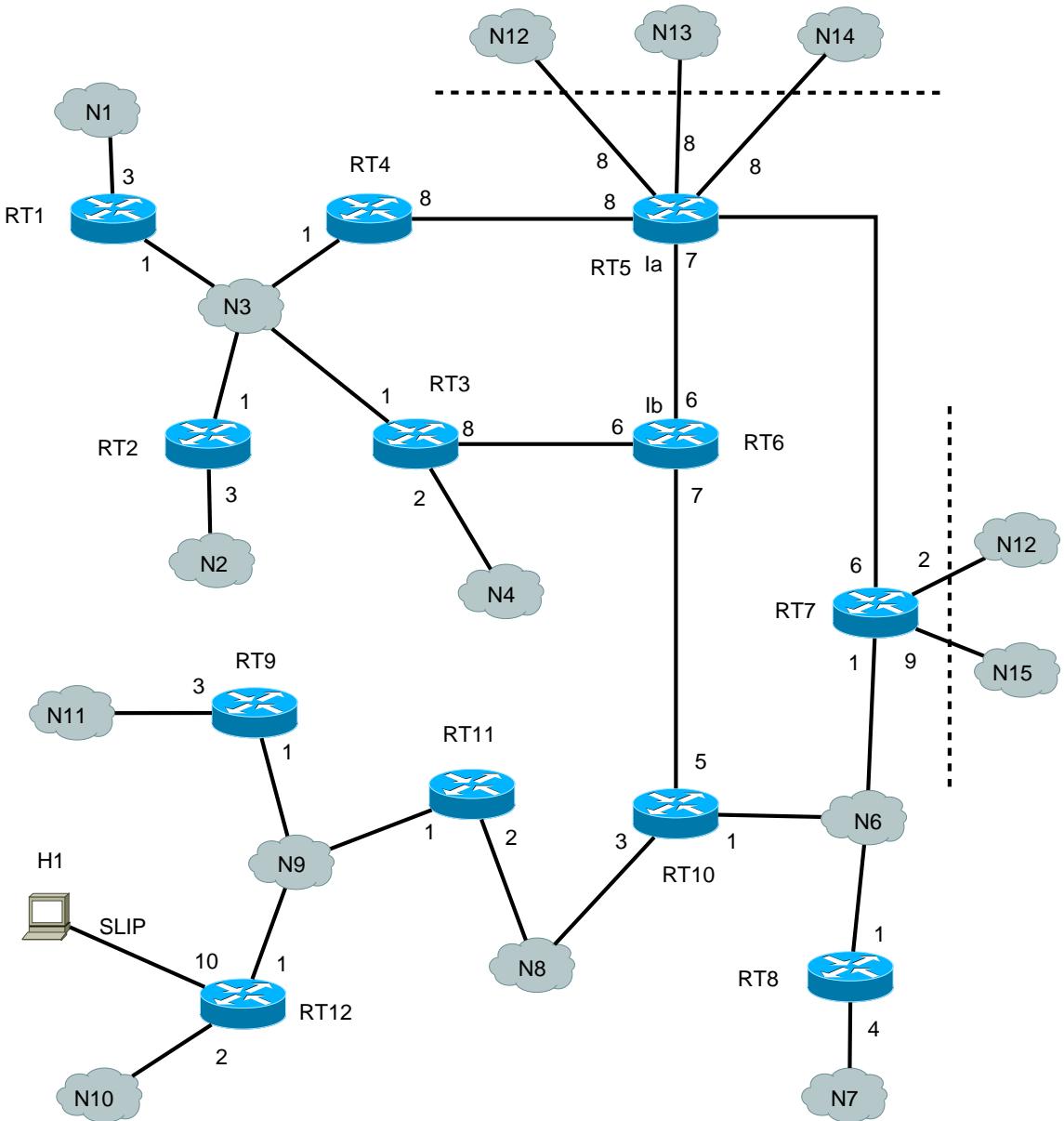
Σχήμα 2.2.2 Αναπαράσταση δικτύων στο OSPF

2.2.1 Ένα παράδειγμα

Στο σχήμα 2.2.3 φαίνεται ένα παράδειγμα ενός δικτύου. Το παράδειγμα αυτό προέρχεται από το RFC 2328 και είναι αντιπροσωπευτικό του OSPF μιας και περιέχει όλες τις πιθανές συνδέσεις. Παρατηρούμε ότι υπάρχουν:

- Point-to-point συνδέσεις (RT6—RT7).
- Broadcast δίκτυα (N3, N6, N8, N9) στα οποία συμμετέχουν πολλοί δρομολογητές.
- Stub δίκτυα (N1, N2, N4, N7, N10, N11), δηλαδή δίκτυα στα οποία υπάρχει συνδεδεμένος μόνο ένα δρομολογητής.
- Point-to-point συνδέσεις σε μη δρομολογητές (H1—RT12).
- Δίκτυα εκτός AS (N12, N13, N14, N15) και με ένα από αυτά (N12) να συνδέεται σε δύο δρομολογητές (RT5, RT7).

Από το σχήμα προκύπτει ο γράφος που περιγράφεται στο σχήμα 2.2.4. Στο πάνω μέρος του πίνακα υπάρχει το “από” και στο αριστερό το “του” της κάθε σύνδεσης, ενώ μέσα στον πίνακα υπάρχει το κόστος της. Όπου δεν αναγράφεται κάποια τιμή σημαίνει πως δεν υπάρχει φυσική σύνδεση. Όπου υπάρχει ο αριθμός 0 σημαίνει πως πρόκειται για σύνδεση από δίκτυο σε δρομολογητή.



Σχήμα 2.2.3 Παράδειγμα Αυτόνομου Συστήματος

| | * * FROM * | | | | | | | | | | | | | | | | |
|------|------------|-----|-----|-----|-----|-----|-----|-----|-----|------|------|------|----|----|----|----|--|
| | RT1 | RT2 | RT3 | RT4 | RT5 | RT6 | RT7 | RT8 | RT9 | RT10 | RT11 | RT12 | N3 | N6 | N8 | N9 | |
| RT1 | | | | | | | | | | | | | 0 | | | | |
| RT2 | | | | | | | | | | | | | 0 | | | | |
| RT3 | | | | | 6 | | | | | | | | 0 | | | | |
| RT4 | | | | | 8 | | | | | | | | 0 | | | | |
| RT5 | | | | 8 | | 6 | | | | | | | | | | | |
| RT6 | | | | 8 | | 7 | | | | | | | 5 | | | | |
| RT7 | | | | | | | | | | | | | 0 | | | | |
| RT8 | | | | | | 6 | | | | | | | 0 | | | | |
| RT9 | | | | | | | | | | | | | 0 | | | | |
| RT10 | | | | | | | 7 | | | | | | 0 | | | | |
| RT11 | | | | | | | | | | | | | 0 | | | | |
| RT12 | | | | | | | | | | | | | 0 | | | | |
| N1 | 3 | | | | | | | | | | | | | | | 0 | |
| N2 | | 3 | | | | | | | | | | | | | | | |
| N3 | 1 | 1 | 1 | 1 | | | | | | | | | | | | | |
| N4 | | | 2 | | | | | | | | | | | | | | |
| N6 | | | | | | | 1 | 1 | | | 1 | | | | | | |
| N7 | | | | | | | | 4 | | | | | | | | | |
| N8 | | | | | | | | | | | | | 3 | 2 | | | |
| N9 | | | | | | | | | | | 1 | | 1 | 1 | | | |
| N10 | | | | | | | | | | | | | | 2 | | | |
| N11 | | | | | | | | | | | | 3 | | | | | |
| N12 | | | | | | | | 2 | | | | | | | | | |
| N13 | | | | | | | | 8 | | | | | | | | | |
| N14 | | | | | | | | 8 | | | | | | | | | |
| N15 | | | | | | | | | | | 9 | | | | | | |
| H1 | | | | | | | | | | | | | 10 | | | | |

Σχήμα 2.2.4 Ο γράφος του σχήματος 2.2.3

2.3 Λειτουργία

Το OSPF λειτουργεί με ελαφρώς διαφορετικό τρόπο ανά είδος σύνδεσης. Στις point-to-point συνδέσεις τα πράγματα είναι απλά και ο κάθε ένας δρομολογητής γνωρίζει πως οτιδήποτε σταλεί στο φυσικό μέσο όταν το παραλάβει ο απέναντι και μόνο. Αυτό συνεπάγεται πως δεν χρειάζεται κάποια ιδιαίτερη πρόβλεψη για την ανταλλαγή πληροφοριών μεταξύ των δύο. Μόλις δύο δρομολογητές που βρίσκονται στα δύο άκρα ενός point-to-point link εντοπίσουν ότι ένας τον άλλον, “εκλέγουν” ένα από τους δύο σαν “master” και αρχίζουν την ανταλλαγή πληροφοριών η οποία θα διαρκέσει μέχρι να τερματίσει η λειτουργία του OSPF ή να πάψει να υπάρχει επικοινωνία. Η εκλογή του ενός δρομολογητή σαν “master” χρειάζεται για να καθοριστεί ποιος από τους δύο θα ζητήσει πρώτος δεδομένα από τον απέναντι του.

Σε broadcast δίκτυα η επικοινωνία είναι ελαφρώς πιο πολύπλοκη. Το OSPF λειτουργεί αμέσως πάνω από το IP με αποτέλεσμα να πρέπει το ίδιο το πρωτόκολλο να διασφαλίσει την αξιόπιστη μεταφορά δεδομένων. Για να μπορέσει να γίνει αυτό, κάθε ένας δρομολογητής που λαμβάνει νέα στοιχεία επιβεβαιώνει τη λήψη τους στον αποστολέα. Η διαδικασία της επιβεβαίωσης ακολουθείται και στα point-to-point δίκτυα και απαιτεί από κάθε ένα δρομολογητή να έχει ότι ένας αμφίδρομης και αξιόπιστης επικοινωνίας με τους δρομολογητές με τους οποίους ανταλλάσσει δεδομένα. Για να είναι δυνατή αυτή η επικοινωνία οι δρομολογητές σχηματίζουν μεταξύ τους γειτονίες (adjacencies) με αποτέλεσμα να κρατάνε πληροφορίες ο ένας για τον άλλον. Ακολουθώντας αυτή τη λογική, σε ένα broadcast δίκτυο με N δρομολογητές, ο κάθε ένας δρομολογητής θα έπρεπε να σχηματίσει $N-1$ γειτονίες και συνολικά να υπάρχουν:

$$\sum_{n=1}^{N-1} n$$

Πέρα από το θέμα των γειτονιών υπάρχουν και άλλα προβλήματα τα οποία κάνουν ότι τέτοιο σχήμα λειτουργίας μη αποδοτικό και μη θεμιτό.

Για να αποφευχθεί το παραπάνω πρόβλημα, στα broadcast δίκτυα εκλέγεται ένας δρομολογητής ο οποίος αναλαμβάνει να επικοινωνεί με όλους τους άλλους και ο οποίος ονομάζεται Designated Router (DR). Για να είναι εξασφαλισμένη η συνεχής λειτουργία του δικτύου και να μην υπάρχει κάποιο single point of failure, πέρα από τον DR εκλέγεται και ένας Backup DR, ο οποίος πραγματοποιεί τις ίδιες λειτουργίες με τον DR και παίρνει τη θέση του όταν χαθεί η επικοινωνία μαζί του. Σύμφωνα με αυτή τη λογική, ο DR συλλέγει πληροφορίες από όλους τους άλλους δρομολογητές του δικτύου και τις στέλνει ξανά στους ίδιους. Πέρα από αυτό,

ο DR αναλαμβάνει να κάνει τις απαραίτητες ανακοινώσεις στην περιοχή του OSPF εκ μέρους του δικτύου το οποίο αντιπροσωπεύει, ώστε να μπορέσει να υπάρξει το δίκτυο σαν κόμβος του γραφήματος.

Η εκλογή του master, η εκλογή του DR, η δημιουργία των adjacencies και ο έλεγχος για νέους δρομολογητές ή για δρομολογητές που έχουν πάψει να επικοινωνούν γίνονται με τη χρήση των Hello πακέτων. Αυτά αποστέλλονται από κάθε δρομολογητή και συμπεριλαμβάνουν μια λίστα με όλους τους δρομολογητές του αντίστοιχου δικτύου τους οποίους γνωρίζει ο αποστόλεας. Η λίστα με τους δρομολογητές του δικτύου τους οποίους γνωρίζει ο κάθε δρομολογητής είναι αποθηκευμένη τοπικά σε έναν πίνακα ο οποίος ονομάζεται Neighbor Table.

Αφού οι δρομολογητές σχηματίσουν τις απαραίτητες γειτονίες, αρχίζει η επικοινωνία για ανταλλαγή πληροφοριών, όπως ακριβώς και στα point-to-point δίκτυα, με την εκλογή ενός master, ο οποίος είναι συνήθως ο DR⁴. Αφού γίνει αυτό οι δρομολογητές ανταλλάσσουν μια σειρά από Database Description Packets (DDPs), τα οποία περιέχουν συνοπτικές περιγραφές των πληροφοριών που γνωρίζει ο κάθε ένας. Στη συνέχεια συγχρίνουν τα δεδομένα που γνωρίζει ο απέναντί τους με αυτά που γνωρίζουν οι ίδιοι και αν χρειαστεί ζητάνε τις επιπλέον πληροφορίες που υπάρχουν. Η τοπική βάση η οποία κρατάει όλες τις πληροφορίες που γνωρίζει ο κάθε δρομολογητής δεν είναι τίποτε παραπάνω από το σύνολο των LSA τα οποία έχει λάβει και ονομάζεται Link State Database.

Μόλις οι δρομολογητές αποκτήσουν τις πληροφορίες που χρειάζονται και σχηματίσουν την εικόνα του γράφου που αναφέρθηκε, εφαρμόζουν τον αλγόριθμο του Dijkstra και βρίσκουν τις συντομότερες διαδρομές προς όλους τους πιθανούς προορισμούς.

2.4 Πακέτα του OSPF

Για την λειτουργία του το OSPF χρησιμοποιεί 5 ειδών πακέτα:

Type 1 - Hello

Πρόκειται για τα Hello πακέτα τα οποία αναφέρθηκαν προηγουμένως.

Type 2 - Database Description (DDP)

Περιέχουν συνοπτικές περιγραφές των πληροφοριών που έχει ο δρομολογητής που τα στέλνει.

⁴ Αυτό οφείλεται στο τρόπο με τον οποίο εκλέγονται ο DR και ο master

Type 3 - Link State Request (LSR)

Χρησιμοποιούνται από τους δρομολογητές για να ζητήσουν πληροφορίες από άλλους δρομολογητές

Type 4 - Link State Update (LSU)

Περιέχουν τις πληροφορίες που ζητήθηκαν με τα LSR.

Type 5 - Link State Ack (LSACK)

Επιβεβαιώνουν τη λήψη δεδομένων από τους απέναντι δρομολογητές.

Από αυτά, τα Link State Updates περιέχουν ένα η παραπάνω Link State Advertisements (LSAs). Η βασική λειτουργία του OSPF χρησιμοποιεί 5 τύπους LSA:

Type 1 - Router LSAs

Στέλνονται από όλους τους δρομολογητές και περιέχουν τη κατάσταση των Interface του κάθε δρομολογητή.

Type 2 - Network LSAs

Στέλνονται μόνο από τους DR και περιέχουν πληροφορίες για το ποιοι δρομολογητές είναι συνδεδεμένοι σε κάθε broadcast δίκτυο. Χάρη σε αυτά δημιουργούνται στο γράφο οι κόμβοι που αντιπροσωπεύουν τα Broadcast δίκτυα.

Type 3,4 - Summary LSAs

Στέλνονται από τους ABRs και η περιγραφή της λειτουργίας τους ξεφεύγει από τους στόχους αυτής της εργασίας.

Type 5 - AS-external-LSAs

Στέλνονται από τους ASBRs και όπως και πριν, η περιγραφή της λειτουργίας τους δεν είναι αντικείμενο αυτής της εργασίας.

2.5 Ο αλγόριθμος του Dijkstra

Η λειτουργία του αλγόριθμου του Dijkstra στηρίζεται στη εύρεση του επόμενου κοντινότερου κόμβου σε κάθε βήμα εκτέλεσης του. Πρόκειται για έναν πολύ γρήγορο και αποδοτικό αλγόριθμο με γραμμική πολυπλοκότητα. Ο αλγόριθμος συνοψίζεται στο παρακάτω ψευδοκώδικα:

Δεδομένα:

G: Γράφος

w: Βάρη

s: Κόμβος ρίζας

Μεταβλητές:

E[G]: Οι ακμές του γράφου

V[G]: Οι κόμβοι του γράφου

d: Πίνακας με τις αποστάσεις/κόστη που έχουν καθοριστεί μέχρι το τρέχον loop

p: Πίνακας με το προηγούμενος του κάθε κόμβου σύμφωνα με το Shortest Path

Αρχικοποίηση:

for $n \in V[G]$

$d[n] \leftarrow +\infty$

$p[n] \leftarrow \emptyset$

$d[s] \leftarrow 0$

$Q \leftarrow V[G]$

Main loop:

while Q not empty **do**

$u \leftarrow \text{find_min_du}(d)$

$Q \leftarrow Q - u$

for each $(u,v) \in E[G]$

if $(d[u] + w(u,v)) < d[v]$ **then**

$d[v] = d[u] + w(u,v)$

$p[v] = u$

Με λόγια μπορούμε να αναφέρουμε τη λειτουργία του αλγορίθμου ως εξής:

1. Αρχικοποιούμε τα δεδομένα:

- Θεωρούμε πως η απόσταση του κάθε κόμβου X είναι άπειρο ($d(X) = +\infty$), εκτός από τον αρχικό κόμβο s , ο οποίος απέχει 0 ($d(s) = 0$).

- Για κάθε κόμβο X είναι απαραίτητο να κρατάμε την απόστασή του $d(X)$ (σύμφωνα με το shortest path) και το προηγούμενο κόμβο της διαδρομής από την οποία φτάνουμε με αυτή την απόσταση ($p(X)$). Αρχικά ορίζουμε ότι δεν υπάρχει κανένας προηγούμενος σε όλους τους κόμβους.

2. Όσο υπάρχει κόμβος που δεν τον έχουμε επισκεφθεί:

- Επιλέγουμε έναν κόμβο N τον οποίο δεν έχουμε επισκεφθεί, έτσι ώστε να είναι αυτός που απέχει λιγότερο από οποιοδήποτε άλλο σύμφωνα με τον πίνακα $d()$. Αυτό θεωρείται επίσκεψη του κόμβου. Αν υπάρχει ισοδυναμία επιλέγουμε έναν τυχαίο κόμβο μεταξύ των ισοδύναμων.
- Για κάθε γείτονα K του N υπολογίζουμε την απόστασή $d_{new} = d(N) + w(N, K)$ μέσω του N . Αν η απόσταση d_{new} είναι μικρότερη από την ήδη υπάρχουσα ($d(K)$), τότε κρατάμε την νέα ($d(K) = d_{new}$) και ορίζουμε σαν προηγούμενο κόμβο του K τον N ($p(K) = N$).

Κεφάλαιο 3

Συμφόρηση κόμβων

3.1 Εισαγωγή

Όπως αναφέρθηκε στη παράγραφο 1.3.2, οι δρομολογητές χρησιμοποιούν τεχνικές βελτιστοποίησης της διαδικασίας της αναζήτησης στον πίνακα δρομολόγησης για να μπορέσουν να επιτύχουν τους επιθυμητούς ρυθμούς μεταφοράς δεδομένων. Στη πράξη χρησιμοποιούνται αρκετά πιο πολύπλοκες και εξειδικευμένες τεχνικές στηριζόμενες καθαρά σε hardware λύσεις. Ένας από τους περιορισμούς των δρομολογητών είναι το ότι ο επεξεργαστής τους αποτελεί συνήθως το πιο αδύναμο σημείο τους. Για το λόγο αυτό οι κατασκευαστές κάνουν όσα περισσότερα μπορούν ώστε να μην χρειάζεται να απασχολείται καθόλου, όπου αυτό είναι δυνατόν. Για να επιτευχθεί κάτι τέτοιο γίνεται εκτενής χρήση εξειδικευμένων κυκλωμάτων, αφιερωμένων σε συγκεκριμένες (και σχετικά απλές) διαδικασίες, τα οποία ονομάζονται “Application Specific Integrated Circuits” (ASICs). Κάθε Interface ενός δρομολογητή περιέχει μια πληθώρα από ASICs τα οποία εκτελούν λειτουργίες όπως:

- Αποστολή/λήψη δεδομένων προς/από το φυσικό μέσο.
- Έλεγχος της ακεραιότητας των δεδομένων.
- Έλεγχος λαθών.
- Διαχείριση της ουράς λήψης/αποστολής.
- Αναζήτηση στην CAM, η οποία μπορεί να είναι τοπική στο Interface.
- Επικοινωνία με το/τα bus.

Με το τρόπο αυτό επιτυγχάνεται η αύξηση των δυνατοτήτων του δρομολογητή, οι οποίες μετριούνται σε packets per second (pps) τα οποία μπορεί να χειριστεί και να δρομολογήσει.

Όπως προκύπτει από τα προηγούμενα, η ισχύς ενός δρομολογητή είναι άμεσα εξαρτώμενη από το τρόπο εσωτερικής λειτουργίας του και δεν μπορεί να μετρηθεί παρά μόνο κάτω από συνθήκες. Πράγματι, ένας δρομολογητής ο οποίος εξυπηρετεί λίγους σταθμούς εργασίας δεν απαιτεί να είναι τόσο τεχνολογικά προηγμένος όσο ένας ο οποίος εξυπηρετεί πολλούς. Αυτό είναι έμμεσο επακόλουθο της χρήσης της Route cache και της ταχύτητας αναζήτησης μέσα στον πίνακα δρομολόγησης. Παράγοντες όπως:

- Πολλοί χρήστες του τοπικού δικτύου.
- Χρήση προγραμμάτων όπως peer-to-peer, ύπαρξη ιών κλπ τα οποία προκαλούν την επικοινωνία με πολλούς διαφορετικούς κόμβους.
- Ύπαρξη μεγάλου Routing Table.

είναι αυτοί οι οποίοι επιβαρύνουν τους δρομολογητές περισσότερο από τη μεταφορά μεγάλου όγκου δεδομένων.

3.1.1 Αδυναμία δρομολόγησης

Παρόλη τη τεχνολογική εξέλιξή τους, υπάρχει πάντα η περίπτωση κάποιοι δρομολογητές να ξεπεράσουν τις ικανότητές τους, χυρίως αν πρόκειται για σχετικά μικρά μοντέλα. Τέτοια παραδείγματα μπορούν να αναφέρουν οι περισσότεροι διαχειριστές δικτύων και πολλές φορές ευθύνονται σε απλή χρήση peer-to-peer προγραμμάτων και στην ύπαρξη ιομορφικού λογισμικού το οποίο προσπαθεί να μεταδώσει τον εαυτό του (worms).

Παράδειγμα όλων αυτών αποτελεί ο ιός SQL Slammer (αναφέρεται και ως Saphire), ο οποίος στις 25 Ιανουαρίου 2003, στις 05:30 UTC κυκλοφόρησε στο Internet και σε μισή ώρα είχε μολύνει περισσότερους από 70.000 υπολογιστές ανά τον κόσμο (David Moore and Vern Paxson and Stefan Savage and Colleen Shannon and Stuart Staniford and Nicholas Weaver 2003). Η “επιτυχία” του οφείλεται στο ότι διαδιδόταν με UDP¹ πακέτα προς τυχαίες διευθύνσεις. Σύμφωνα με αυτά που έχουμε περιγράψει, το είδος αυτής της κίνησης προκαλεί συνεχόμενες αναζητήσεις στον πίνακα δρομολόγησης και υπερχείλιση της route cache, το οποίο με τη σειρά του δημιουργεί ακόμα μεγαλύτερες ανάγκες για αναζητήσεις στον πίνακα δρομολόγησης.

¹User Datagram Protocol. Η μεταφορά δεδομένων με UDP είναι αναξιόπιστη αλλά δεν δημιουργεί overhead και αποτελεί το τρόπο επικοινωνίας με το μικρότερο δυνατό latency στο Internet

Πέρα από τους ιούς, αδυναμία δρομολόγησης λόγο φόρτου μπορεί να δημιουργηθεί και από οποιαδήποτε διαδικασία του δρομολογητή η οποία μπορεί να τον επιβαρύνει, καταναλώνοντας επεξεργαστική ισχύ. Όταν ένας δρομολογητής ξεπεράσει τα όριά του και δεν μπορεί πλέον να δρομολογήσει IP πακέτα με το ρυθμό που του έρχονται, αναγκάζεται να τα απορρίψει. Οι επιπτώσεις που έχει η απόρριψη πακέτων στην ομαλή λειτουργία ενός δικτύου είναι αρκετά μεγάλες και αποτελούν από μόνες τους αντικείμενο προς μελέτη.

3.1.2 Απώλεια πακέτων στο TCP

Το ποιο συχνά χρησιμοποιούμενο πρωτόκολλο 4ου επιπέδου του Internet είναι το Transmission Control Protocol (TCP) (Postel 1981b). Το TCP παρέχει αξιόπιστη μεταφορά δεδομένων με επαναποστολή και επιβεβαίωση λήψης και χρησιμοποιείται οπουδήποτε απαιτείται αξιόπιστη μεταφορά δεδομένων στο Internet. Παρόλο που το TCP παρέχει αυτόματη επαναποστολή των δεδομένων, φαινόμενα απώλειας πακέτων λειτουργούν αρνητικά στις επιδόσεις του. Ανάλογα με τον αλγόριθμο καθορισμού του Window Size που χρησιμοποιείται υπάρχουν και διαφορετικές επιπτώσεις. Στις συνήθεις υλοποίησεις του, το TCP μειώνει στα δύο το window size, το οποίο μπορεί να οδηγήσει στη στιγμιαία μείωση έως και 50% του ρυθμού αποστολής δεδομένων. Αυτό οφείλεται στο ότι η αδυναμία παράδοσης πακέτων μεταφράζεται από το πρωτόκολλο σαν μη-ύπαρξη αρκετού διαθέσιμου bandwidth στο δίκτυο.

Η τακτική απώλεια πακέτων ουσιαστικά μπορεί να οδηγήσει το TCP σε αδυναμία λειτουργίας, κάτι το οποίο παρατηρείται πάντα σε αυτές τις περιπτώσεις. Απώλεια πακέτων της τάξης του 5–10% οδηγούν σε συμαντική μείωση της ταχύτητας μεταφοράς δεδομένων, ενώ οποιοδήποτε μεγαλύτερο ποσοστό μπορεί να οδηγήσει σε σοβαρή δυσλειτουργία του πρωτοκόλλου.

Για την ομαλή λειτουργία του TCP στο Internet έχουν προταθεί διάφοροι αλγόριθμοι για την αποφυγή συμφόρησης. Η βασικές τους αρχές είναι:

- Η αργή εκκίνηση (Slow Start), δηλαδή η μικρή αρχική ταχύτητα μετάδοσης και η σταδιακή αύξηση της (π.χ. γραμμικά).
- Η προσπάθεια αποφυγής της συμφόρησης με τη μείωση του Window Size και επομένως και της μέγιστης δυνατής ταχύτητας μετάδοσης, όταν εντοπιστεί συμφόρηση.
- Η γρήγορη επαναποστολή των χαμένων πακέτων χωρίς κάποια ενδιάμεση καθυστέρηση.
- Η γρήγορη επανα-αύξηση της ταχύτητας μετάδοσης μετά από την απότομη μείωση αυτής

(π.χ. εκθετικά).

Οι αλγόριθμοι αυτοί είναι γνωστοί σαν “Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms” (Stevens 1997). Ο ποιο διαδεδομένος από αυτούς είναι ο TCP Reno (Van Jacobson 1988) και η ταχύτητα μετάδοσης δεδομένων με τη χρήση του (J. Padhye and V. Firoiu and D. Towsley and J. Krusoe 1998) δίνεται από τον τύπο:

$$B(p) = \begin{cases} \frac{\frac{1-p}{p} + E[W] + \hat{Q}(E[W]) \frac{1}{1-p}}{RTT \left(\frac{b}{2} E[W] + 1 \right) + \hat{Q}(E[W]) T_0 \frac{f(p)}{1-p}} & E[W] < W_{max} \\ \frac{\frac{1-p}{p} + W_{max} + \hat{Q}(W_{max}) \frac{1}{1-p}}{RTT \left(\frac{b}{8} W_{max} + \frac{1-p}{p W_{max}} + 2 \right) + \hat{Q}(W_{max}) T_0 \frac{f(p)}{1-p}} & E[W_u] \geq W_{max} \end{cases} \quad (1)$$

όπου W_{max} είναι το μέγιστο δυνατό Window Size, RTT είναι το Round Trip Time, δηλαδή ο χρόνος που χρειάζεται ένα πακέτο για να φτάσει από τον ένα σταθμό στον άλλον και να επιστρέψει, p είναι η πιθανότητα απώλειας πακέτων, b είναι το πλήθος των πακέτων που γίνονται Acknowledge με ένα ACK και:

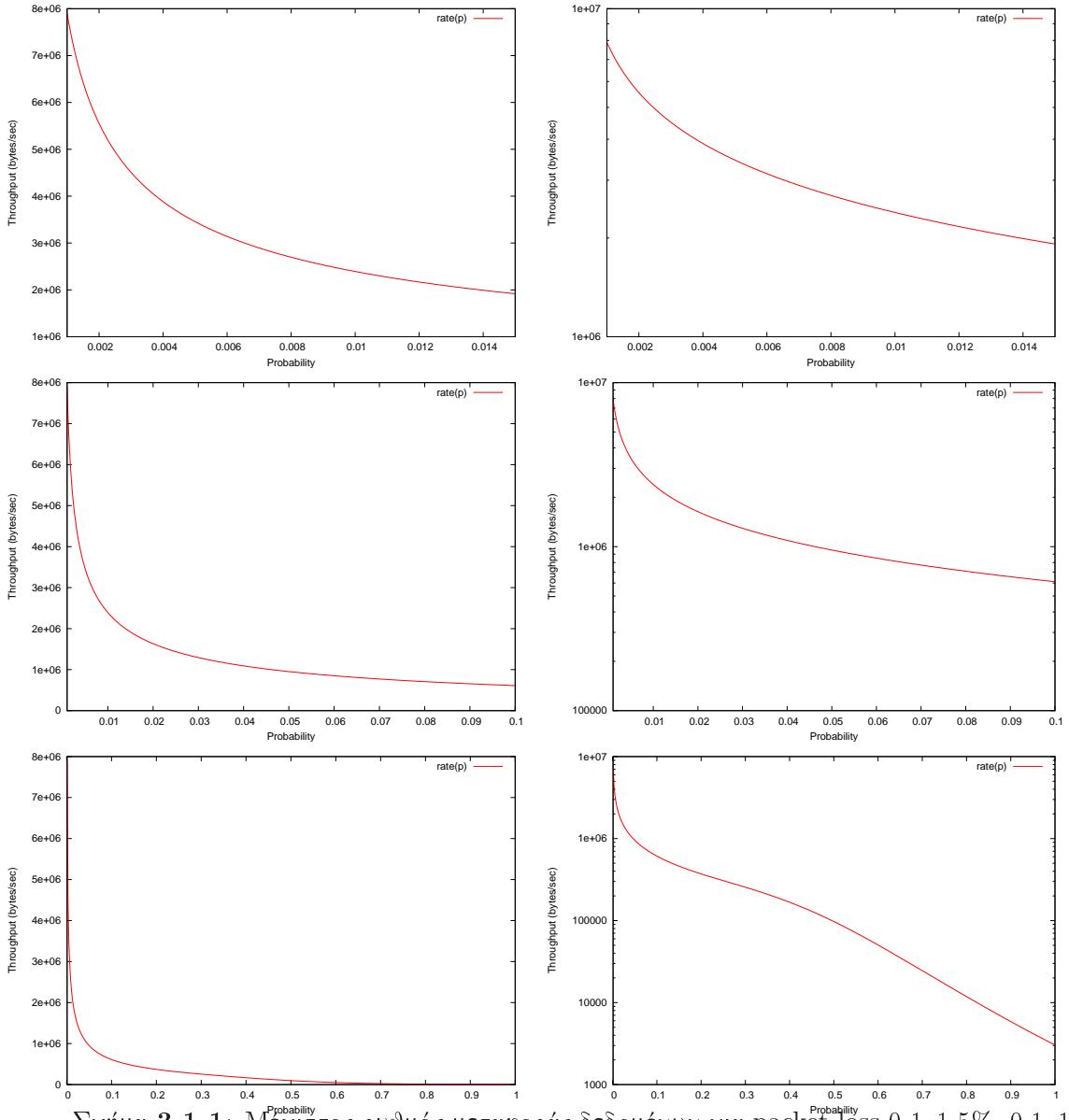
$$E[W] = \frac{2+b}{3b} + \sqrt{\frac{8(1-p)}{3bp} + \left(\frac{2+b}{3b}\right)^2} \quad (2)$$

$$\hat{Q}(w) = \min \left(1, \frac{(1 - (1-p)^3)(1 + (1-p)^3(1 - (1-p)^{w-3}))}{1 - (1-p)^w} \right) \quad (3)$$

Για να μπορέσουμε να δούμε τη μέγιστη θεωρητική ταχύτητα, θεωρούμε ότι το window size μπορεί να μεγαλώσει απεριόριστα και επομένως χρησιμοποιούμε το 1ο σκέλος της σχέσης 1. Πέρα από αυτό θεωρούμε ότι:

- $RTT = 5ms$ το οποίο είναι μια ιδιαίτερα χαμηλή τιμή.
 - $MSS = 1460$ το οποίο είναι η πιο συνηθισμένη τιμή που συναντάται στο Internet.
 - $T_0 = 1.5 * RTT$ το οποίο είναι ο χρόνος μετά από τον οποίο επαναποστέλεται ένα πακέτο.
- Στην πράξη εφαρμόζεται ο τύπος $T_0 = \min(UBOUND, \max(LBOUND, (\beta * SRTT)))$ (Postel 1981b) όπου UBOUND και LBOUND είναι τα πάνω και κάτω όρια (π.χ. 1 λεπτό και 1 δευτερόλεπτο), $\beta \in [1.3, 2.0]$ και $SRTT = (\alpha * SRTT) + ((1 - \alpha) * RTT)$ (το οποίο ονομάζεται Smooth Roundtrip Time και ενημερώνεται συνεχώς).

Εφαρμόζοντας τους τύπους αυτούς προκύπτουν τα γραφήματα της εκόνας 3.1.1, στα οποία παρατηρούμε ότι πράγματι το TCP Reno είναι ιδιαίτερα ευαίσθητο στις απώλειες πακέτων.



Σχήμα 3.1.1: Μέγιστος ρυθμός μεταφοράς δεδομένων για packet loss 0.1–1.5%, 0.1–10% και 0.1–100% σε απλή και λογαριθμική (\log_{10}) κλίμακα

3.1.3 Παραδείγματα

Όπως είδαμε, ένας δρομολογητής ο οποίος αντιμετωπίζει προβλήματα υπερφόρτωσης, ασχέτως λόγου, μπορεί να προκαλέσει σοβαρή δυσλειτουργία του δικτύου, ανάλογα με το πόσο κεντρικός είναι. Γνωστότερο τέτοιο παράδειγμα στην Εκπαιδευτική Κοινότητα της Ελλάδας αποτελούν ίσως οι ταχτικές δυσλειτουργίες του Εθνικού Δικτύου Έρευνας και Τεχνολογίας (ΕΔΕΤ) οι οποίες συνέβαιναν έως και το 2005. Οι δυσλειτουργίες αυτές οφείλονταν σε ιούς, οι οποίοι προκαλούσαν προβλήματα ίδια ή αντίστοιχα με αυτά που περιγράφηκαν παραπάνω. Παρόμοια τέτοια παραδείγματα υπάρχουν σχεδόν σε όλα τα επιμέρους δίκτυα ανά τον κόσμο. Στο TEI Θεσσαλονίκης κατά τα προηγούμενα έτη αντιμετωπίζονταν παρόμοια προβλήματα λόγω ιών και έχουν ληφθεί δραστικά μέτρα τα οποία συνήθως οδηγούν στη διακοπή της πρόσβασης στο Internet σε όσους υπολογιστές αντιμετωπίζουν προβλήματα ιών.

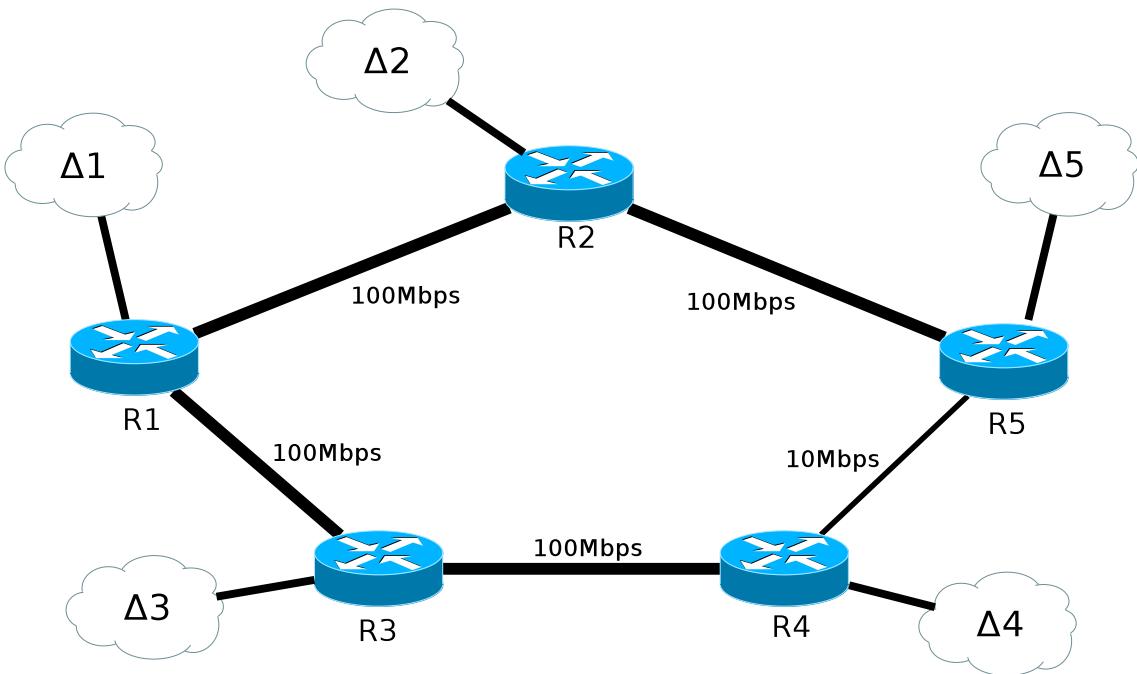
3.2 Μια νέα πρόταση

Αν ψεωρήσουμε ένα δίκτυο όπως το Δίκτυο A' του σχήματος **3.2.2** και υποθέσουμε πως λειτουργεί το πρωτόκολλο δρομολόγησης OSPF, τότε εύκολα συμπεραίνουμε πως η επικοινωνία των δικτύων προς το δίκτυο Δ5 θα γίνεται όπως φαίνεται στον παρακάτω πίνακα:

| Από → Προς | Διαδρομή |
|------------|------------------------|
| Δ1 → Δ5 | R1 → R2 → R5 |
| Δ2 → Δ5 | R2 → R5 |
| Δ3 → Δ5 | R3 → R1 → R2 → R5 |
| Δ4 → Δ5 | R4 → R3 → R1 → R2 → R5 |

Για τη συνέχεια θα ψεωρούμε πως:

- Υπάρχουν 5 δρομολογητές.
- Υπάρχουν μόνο οι συνδέσεις που αναγράφονται στον σχήμα, με τις αντίστοιχες ταχύτητες.
- Μας ενδιαφέρει η επικοινωνία με το δίκτυο Δ5.
- Λειτουργεί το πρωτόκολλο δρομολόγησης OSPF.
- Κάθε ένα από τα δίκτυα Δ1, Δ2, Δ3, Δ4, Δ5 συμβάλει ισοδύναμα στο δικτυακό φόρτο του Δικτύου A και επομένως όλων των δρομολογητών.

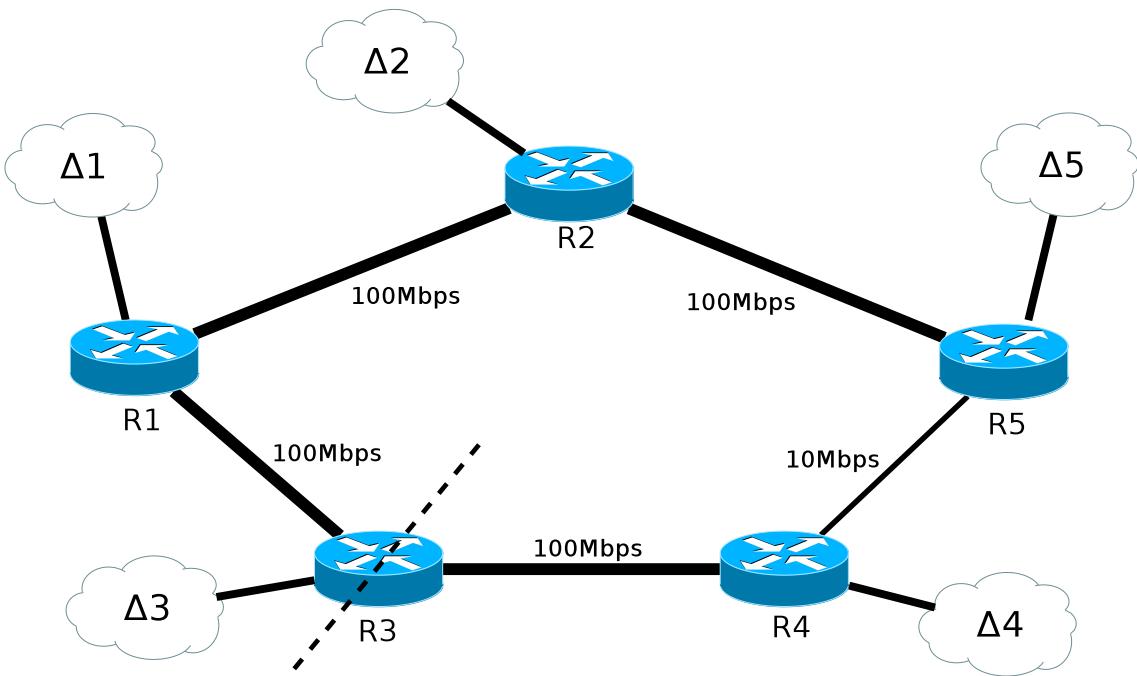


Σχήμα 3.2.2 Δίκτυο Α'

- Παρουσιάζει πρόβλημα υπερφόρτωσης ο δρομολογητής R2 .

Αν ο δρομολογητής R2 παρουσιάσει επεξεργαστικό φόρτο της τάξης του 90%, είναι αρκετά πιθανό να παρουσιαστεί απώλεια πακέτων της τάξης του 50% (και παραπάνω). Αυτό, σύμφωνα με όσα ειπαμε στην παράγραφο 3.1.2, θα οδηγήσει το δίκτυο σε σοβαρή δυσλειτουργία και θα επιφρεάσει όλα τα δίκτυα τα οποία δρομολογούνται μέσω αυτού. Στο παράδειγμα κάτι τέτοιο θα δημιουργήσει πρόβλημα στα δίκτυα Δ1, Δ2, Δ3 και Δ4. Αυτό οφείλεται στο ότι το OSPF, σαν ένα state-of-the-art πρωτόκολλο δρομολόγησης, θα λειτουργήσει όπως αναμένεται και θα οδηγήσει το δίκτυο στο να χρησιμοποιεί τις γρύγορες συνδέσεις επικοινωνίας. Παρόλα αυτά εμείς γνωρίζουμε πως όταν συμβεί ένα τέτοιο περιστατικό, είναι προτιμότερο το να δρομολογείται η κίνηση μέσω της αργής σύνδεσης R4–R5 και όχι μέσω του δρομολογητή R2.

Αν τροποποιήσουμε το OSPF έτσι ώστε σε μία τέτοια περίπτωση να θεωρεί πως δεν μπορεί να δρομολογήσει μέσω του R2 και να χρησιμοποιεί τον εναλλακτικό δρόμο, δημιουργούμε το φαινόμενο του ping-pong. Αυτό οφείλεται στο ότι αν αλλάξει η δρομολόγηση, είναι πιθανό κάποιος άλλος δρομολογητής από τον εναλλακτικό δρόμο (π.χ. ο R4) να εμφανίσει τα ίδια συμπτώματα, ενώ παράλληλα ο R2 να πάψει να αντιμετωπίζει πρόβλημα. Αυτό θα έχει σαν αποτέλεσμα τις αλλεπάλληλες αλλαγές στο δίκτυο με ακόμη χειρότερες συνέπειες.



Σχήμα 3.2.3 Δίκτυο A' – χωρισμένο στα δύο

Η ιδανική λύση στο δίκτυο αυτό είναι η ισοκατανομή της κίνησης στις δύο πιθανές διαδρομές. Εφόσον έχουμε ψεωρήσει πως κάθε ένας δρομολογητής συμβάλει ισόποσα στο συνολικό φόρτο του δικτύου, αρκεί να λειτουργήσουμε ως εξής:

- Να εξετάσουμε κάθε ένα από τα 5 δίκτυα του σχήματος ως πιθανά δίκτυα προορισμού της δικτυακής κίνησης.
- Για κάθε ένα από αυτά να χωρίσουμε το δίκτυο κατάλληλα, έτσι ώστε: οι μισοί δρομολογητές να δρομολογούν μέσω της αρχικής διαδρομής και οι άλλοι μισοί μέσω της εναλλακτικής. Αν υπήρχαν περισσότερες από μία εναλλακτικές θα προτιμούσαμε να χρησιμοποιήσουμε τη καλύτερη δυνατή, σύμφωνα πάντα με τα συμπεράσματα του OSPF.

Η υλοποίηση των παραπάνω φαίνεται στο σχήμα 3.2.3. Πλέον η δρομολόγηση προς το δίκτυο Δ5 θα γίνεται ως εξής:

| Από → Προς | Διαδρομή |
|---------------------------------|---|
| $\Delta 1 \rightarrow \Delta 5$ | $R1 \rightarrow R2 \rightarrow R5$ |
| $\Delta 2 \rightarrow \Delta 5$ | $R2 \rightarrow R5$ |
| $\Delta 3 \rightarrow \Delta 5$ | $R3 \rightarrow R1 \rightarrow R2 \rightarrow R5$ |
| | $R3 \rightarrow R4 \rightarrow R5$ |
| $\Delta 4 \rightarrow \Delta 5$ | $R4 \rightarrow R5$ |

Παρατηρούμε ότι ενώ προηγουμένως οι δρομολογητές ο οποίοι δρομολογούσαν πακέτα προς το δίκτυο $\Delta 5$ μέσω του $R2$ ήταν 3 (εξαιρουμένου του $R2$), τώρα έγιναν οι μισοί (1.5^2). Αυτό θα έχει σαν αποτέλεσμα ο $R2$ να απελευθερωθεί από ένα αρκετά σημαντικό ποσοστό κίνησης κάτι που ίσως να οδηγήσει στην ομαλή λειτουργία του. Επιλέχθηκε το να μην υπολογίζεται ο δρομολογητής ο οποίος αντιμετωπίζει προβλήματα υπερφόρτωσης στη διαδικασία του χωρισμού του δικτύου στα δύο γιατί δημιουργεί λογικά προβλήματα σε πιο πολύπλοκες τοπολογίες.

Εδώ είναι σημαντικό να τονίσουμε πως ο διαχωρισμός αυτός είναι μια διαδικασία, η οποία πρέπει να πραγματοποιείται σε κάθε δρομολογητή (όπως και η διαδικασία της δρομολόγησης) για κάθε πιθανό προορισμό.

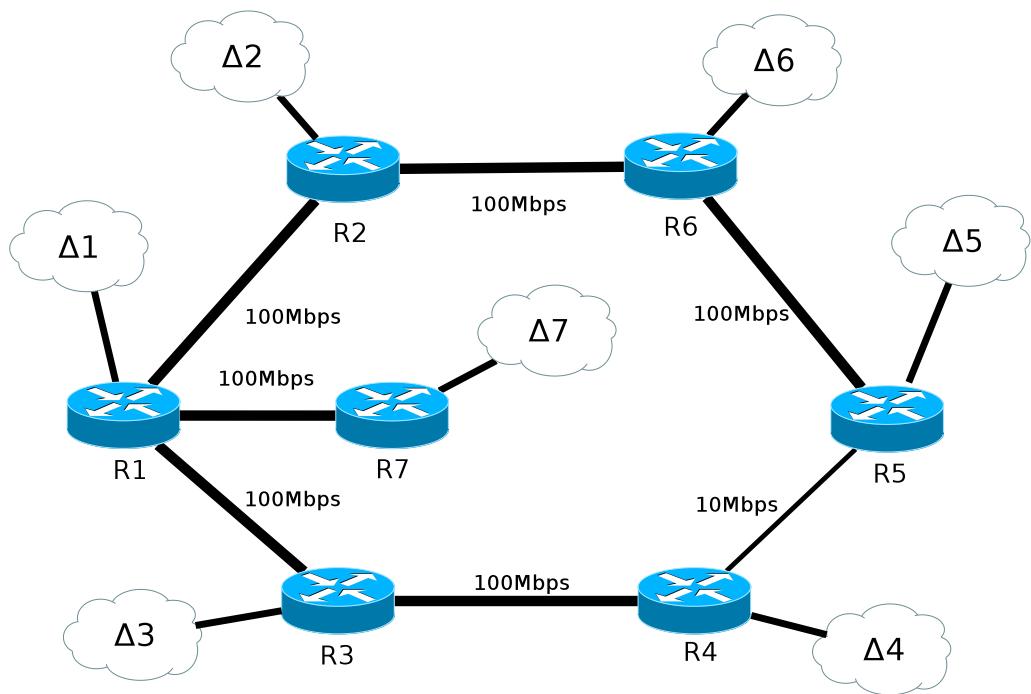
Ένα δεύτερο παράδειγμα

Στο σχήμα 3.2.4 βλέπουμε το Δίκτυο β', το οποίο είναι ένα παρόμοιο δίκτυο με έναν επιπλέον δρομολογητή μεταξύ $R2$ και $R5$ και έναν συνδεδεμένο στον $R1$. Αν εφαρμόσουμε τη λογική που περιγράψαμε πριν όταν θέλουμε να προκύψει το “δίκτυο” του σχήματος 3.2.5. Παρατηρούμε ότι ο δρομολογητής $R6$ δεν θέλουμε να επηρεάσει την συγκεκριμένη απόφαση, γιατί ο ίδιος δεν συμβάλει στο φόρτο του $R2$ όσον αφορά την δρομολόγηση προς το δίκτυο $\Delta 5$. Επίσης πρέπει να επισημάνουμε πως ο δρομολογητής $R7$ δεν έχει κάποια εναλλακτική διαδρομή προς το δίκτυο $\Delta 5$ και επομένως δεν χρειάζεται να αλλάξει τη δρομολόγησή του.

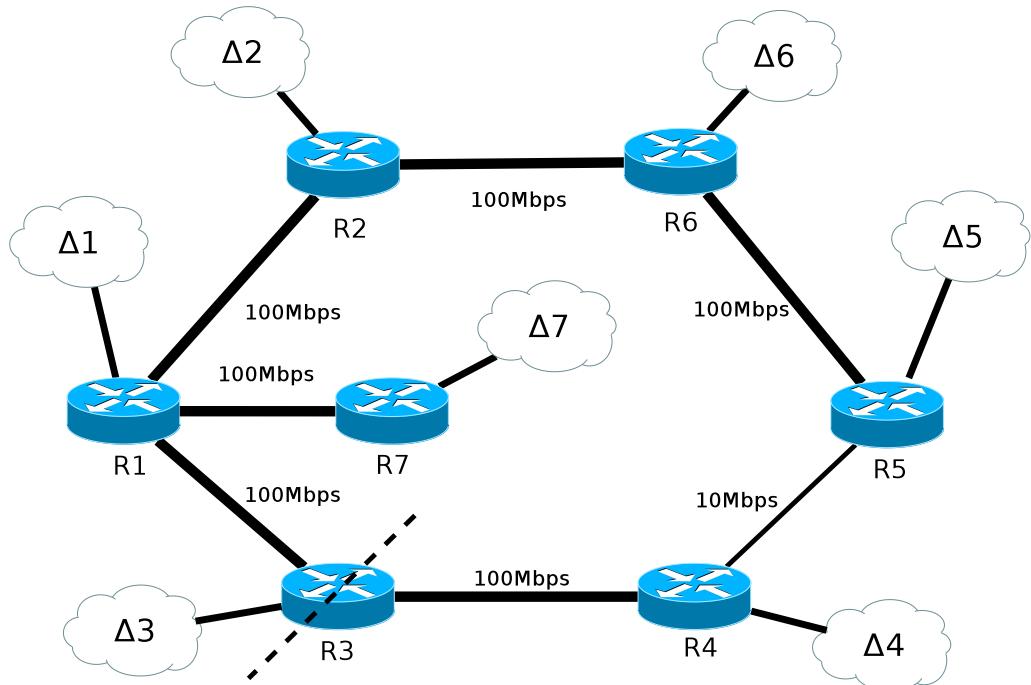
3.3 Ο αλγόριθμος με παράδειγμα

Η λειτουργία του αλγορίθμου μπορεί να παρουσιαστεί καλύτερα αν δούμε τις λειτουργίες που θα εκτελέσει ένας δρομολογητής. Παρακάτω παρουσιάζονται τα βήματα που ακολουθεί ο δρομολογητής $R4$ για να μεταβάλει τον πίνακα δρομολόγησής του:

²Το 0.5 σημαίνει πως ένας δρομολογητής δρομολογεί τη μισή κίνηση από τη μία διαδρομή και την υπόλοιπη από την άλλη.



$\Sigma\chi\gamma\mu\alpha \text{ 3.2.4 } \Delta\text{ίκτυο } B'$



$\Sigma\chi\gamma\mu\alpha \text{ 3.2.5 } \Delta\text{ίκτυο } B' - \chi\omega\rho\iota\sigma\mu\acute{\epsilon}\nu\text{o } \sigma\tau\alpha \delta\acute{\nu}\text{o}$

- Αντιλαμβάνεται ότι ο δρομολογητής R2 αντιμετωπίζει πρόβλημα υπερφόρτωσης Επακόλουθο αυτού είναι να “ερευνά” όλες του τις διαδρομές προς κάθε δίκτυο. Θεωρούμε και πάλι ότι λειτουργεί για το δίκτυο Δ5.
- Αναγνωρίζει πως ο ίδιος δρομολογεί μέσω του R2 προς το δίκτυο Δ5.
- Ελέγχει την εναλλακτική διαδρομή και βρίσκει ότι αυτή έχει διαφορετικό next-hop από την παλιά. Αν δεν συνέβαινε αυτό (όπως π.χ. στον δρομολογητή R7), δεν θα χρειαζόταν να κάνει κάτι παραπάνω.
- Υπολογίζει το κόστος KΔ5 προς το δίκτυο Δ5.
- Υπολογίζει το κόστος KΔ5' προς το δίκτυο Δ5 μέσω της νέας διαδρομής.
- Μετράει τους δρομολογητές που:
 1. Βρίσκονται μεταξύ αυτού και του R2 (παλιά διαδρομή) (έστω N1=2).
 2. Βρίσκονται μεταξύ αυτού και του R5 (νέα διαδρομή) και οι οποίοι δρομολογούν μέσω αυτού (του R4) (έστω N2=0). Αυτό γίνεται ως εξής:
 - Υπολογίζει το κόστος KX προς τον κάθε ενδιάμεσο δρομολογητή RX μεταξύ αυτού (R4) και τού Δ5.
 - Υπολογίζει το κόστος του RX προς το Δ5, μέσω της παλιάς διαδρομής (αν υποθέσουμε πως δρομολογούσε μέσω του R4), το οποίο είναι KΔ5+KX.
 - Υπολογίζει το κόστος του RX προς το Δ5 μέσω της νέας διαδρομής, το οποίο είναι KΔ5'-KX.
 - Ελέγχει αν $(KΔ5'-KX) > (KΔ5+KX)$. Αυτό σημαίνει πως με τη παλιά διαδικασία, ο συγκεκριμένος δρομολογητής προτιμούσε τη διαδρομή μέσω αυτού.
- Συγκρίνει τα N1 και N2 και επιλέγει τη διαδρομή με το μικρότερο N, η οποία είναι η νέα διαδρομή γιατί N2=0 και N1=2.

Ο ίδιος αλγόριθμος αν εφαρμοστεί για τον δρομολογητή R3 για διαδρομές προς το δίκτυο Δ5 θα λειτουργήσει ως εξής:

- Υπολογίζει το κόστος KΔ5 ($R3 \rightarrow R1 \rightarrow R2 \rightarrow R5$) και το κόστος KΔ5' ($R3 \rightarrow R4 \rightarrow R5$).

- Μετράει τους δρομολογητές και καταλήγει πως $N1=1$ και $N2=1$.
- Συγκρίνει τα $N1$ και $N2$ και τα βρίσκει ίσα, με αποτέλεσμα να επιλέξει και τις δύο διαδρομές για χρήση.

3.3.1 Επαναληπτηκότητα

Ο αλγόριθμος αυτός στηρίζεται στη λογική πως αν υπάρχει μία διαδρομή A και μία εναλλακτική διαδρομή B προς το ίδιο destination, τότε αυτές οι δύο σχηματίζουν έναν κύκλο, πάνω στην περιφέρεια του οποίου βρίσκονται οι δρομολογητές των δύο διαδρομών. Απομονώνει το κομμάτι μεταξύ του σημείου της υπερφόρτωσης και του τελικού αποδέκτη, πάνω στο οποίο βρίσκεται ο παρόν δρομολογητής και στη συνέχεια το χωρίζει στη μέση. Με το τρόπο αυτό είναι αδύνατο να σχηματιστεί routing loop.

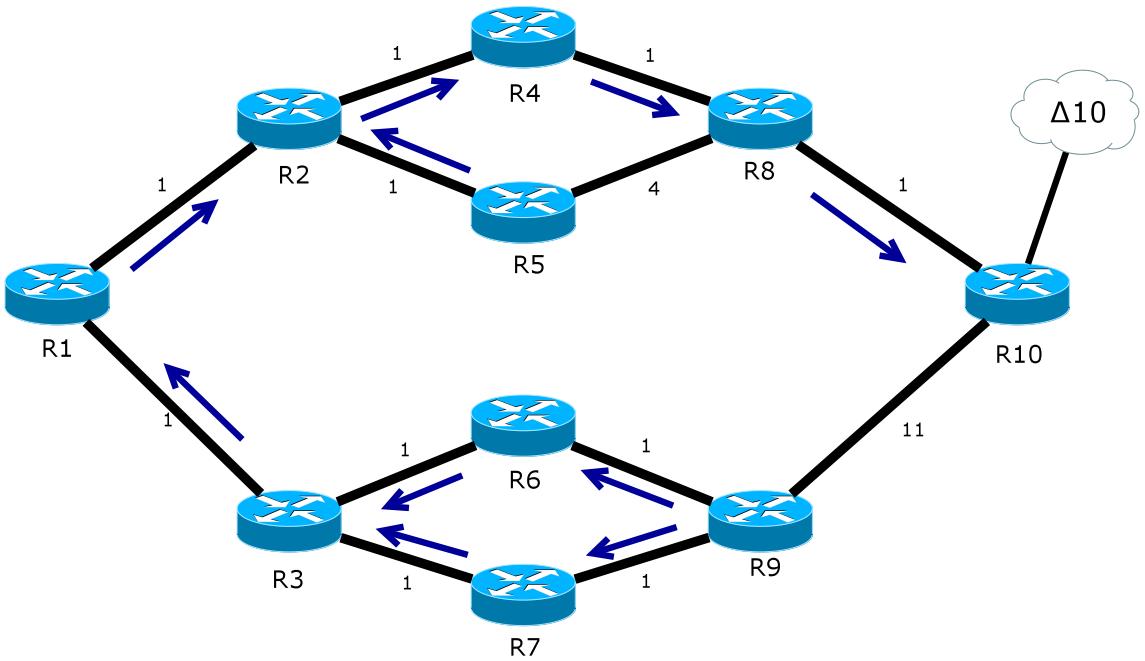
Αν τώρα υπάρξει νέο περιστατικό υπερφόρτωσης, αυτό θα λειτουργήσει πάνω στο νέο σχήμα και θα επηρεάσει τους κατάλληλους δρομολογητές. Ο αλγόριθμος θα τρέξει ξανά, στα νέα δεδομένα, έχοντας σαν βάση την τελευταία του κατάσταση, η οποία μπορεί να έχει προκύψει από κάποιο άλλο προηγούμενο πρόβλημα. Δεδομένης της λογικής του κύκλου, είναι και πάλι αδύνατη η ύπαρξη κύκλου στο δίκτυο μετά από την νέα εφαρμογή του αλγορίθμου.

3.3.2 Πολλαπλές διαδρομές

Υπάρχει η περίπτωση να υπάρχουν πολλαπλές πιθανές διαδρομές ίδιου κόστους προς έναν προορισμό. Στη περίπτωση αυτή:

- Αν υπάρχουν πολλαπλές διαδρομές αρχικά, προτείνεται η χρήση της εναλλακτικής διαδρομής ώστε να μην υπάρξει αναστάτωση στο δίκτυο.
- Αν υπάρξουν πολλαπλές διαδρομές κατά την εκτέλεση του αλγορίθμου, τότε στη καταμέτρηση των κόμβων θα συμπεριληφθούν οι δρομολογητές οι οποίοι δρομολογούν μέσω του υπερφορτωμένου κόμβου από όλες τις πιθανές διαδρομές. Στην συνέχεια θα διαιρεθούν με το πλήθος των πιθανών διαδρομών

Η λογική των πολλαπλών διαδρομών αποτελεί γενίκευση της μεθοδολογίας που παρουσιάστηκε προηγουμένως. Πράγματι, πριν λέγαμε ότι αν υπάρχουν N δρομολογητές οι οποίοι δρομολογούν μέσω του υπερφορτωμένου κόμβου, τότε καθορίζουμε ότι $\frac{N}{2}$ δρομολογητές θα χρησιμοποιούν την αρχική διαδρομή και $\frac{N}{2}$ τη νέα. Πλέον λέμε ότι αν υπάρχουν K νέες διαδρομές, τότε



Σχήμα 3.3.6 Δίκτυο Γ'

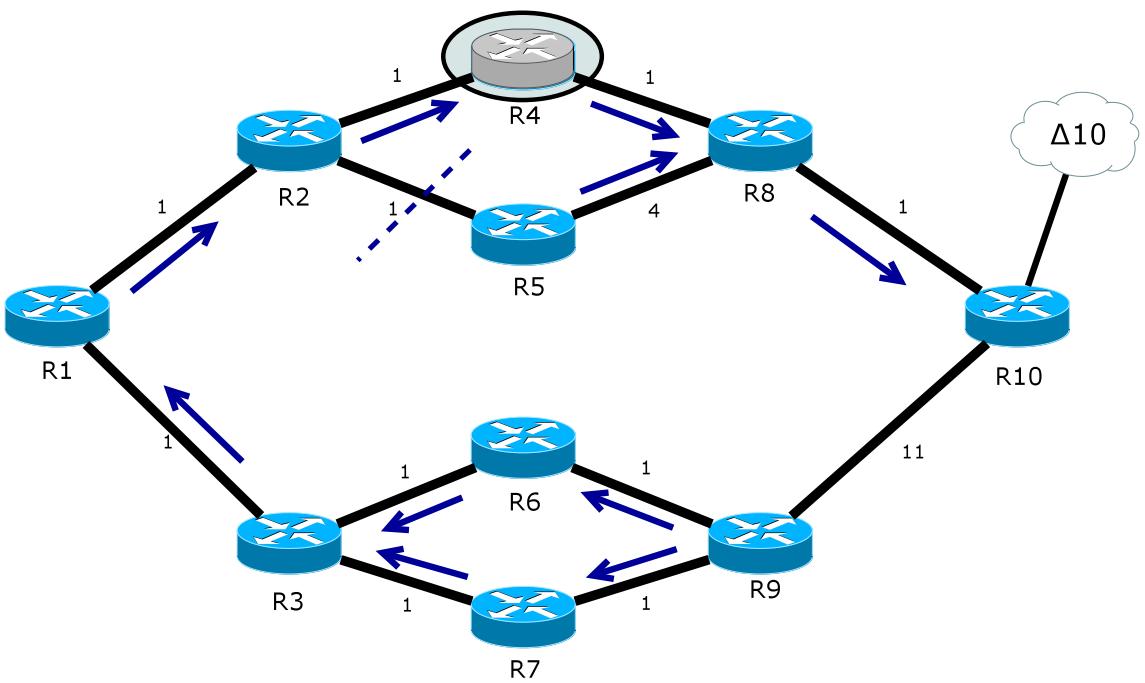
οι N δρομολογητές θα μοιράζονται έτσι ώστε $\frac{N}{K+1}$ δρομολογητές να χρησιμοποιούν τη κάθε διαδρομή.

3.3.3 Ένα πλήρες παράδειγμα

Τη πλήρη λειτουργία του αλγορίθμου μπορούμε να τη δούμε αν θεωρήσουμε ένα πιο πολύπλοκο δίκτυο σαν αυτό του σχήματος 3.3.6. Το δίκτυο έχει 10 δρομολογητές οι οποίοι έχουν πάνω τους από ένα δίκτυο υπολογιστών (Δ_1 - Δ_{10}) παρόλο που στο σχήμα φαίνεται μόνο το δίκτυο Δ_{10} . Όσα θα αναφερθούν παρακάτω αφορούν αποκλειστικά τη δρομολόγηση προς το δίκτυο Δ_{10} . Σε αντίθεση με τα υπόλοιπα σχήματα, στο σχήμα αυτό αναγράφονται τα κόστη της κάθε σύνδεσης.

Παρατηρούμε πως όλοι ανεξαιρέτως οι δρομολογητές δρομολογούν προς το δίκτυο Δ_{10} έτσι ώστε τα δεδομένα να περνάνε από τη σύνδεση R_8 - R_{10} καθώς και από τη σύνδεση R_4 - R_8 . Αυτό οφείλεται στα υψηλά κόστη των συνδέσεων R_5 - R_8 (4) και R_9 - R_{10} (11).

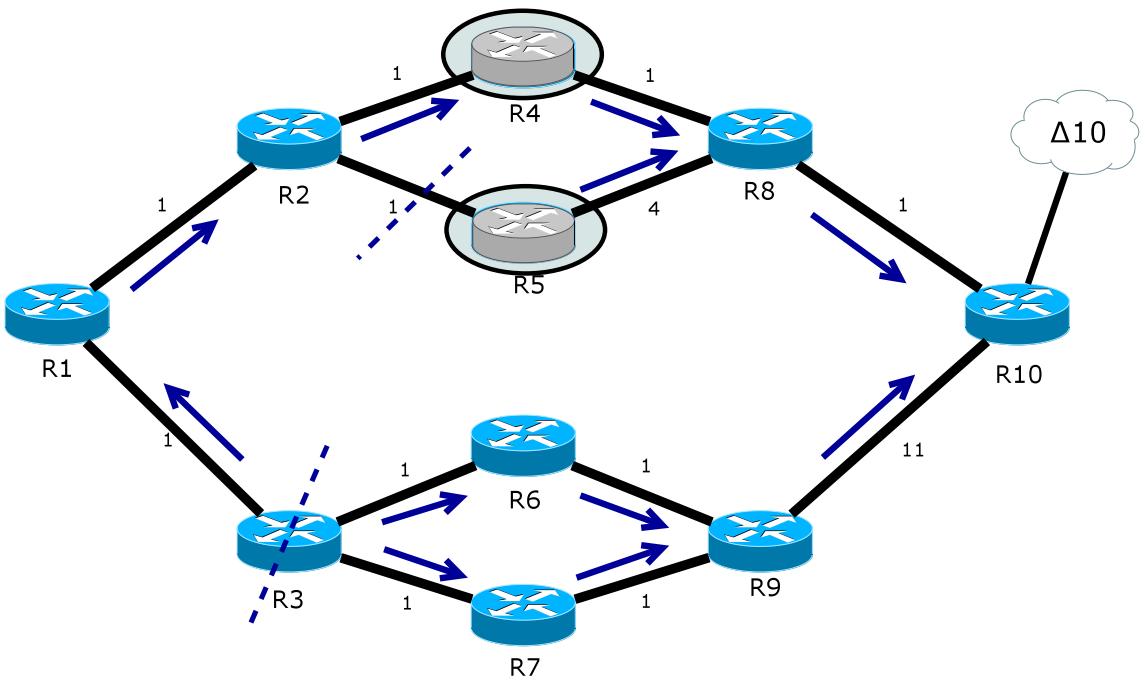
Αρχικά θεωρούμε ότι εμφανίζει πρόβλημα υπερφόρτωσης ο δρομολογητής R_4 . Στην περίπτωση αυτή, όλοι οι δρομολογητές του δικτύου θα εφαρμόσουν τον αλγόριθμο του Dijkstra αφαιρώντας τον R_4 από της Topology Database τους. Αυτό θα οδηγήσει όλους τους δρομολογητές R_1 , R_3 , R_6 , R_7 , R_8 , R_9 , R_{10} στο να συμπεράνουν πως η αμέσως επόμενη καλύτερη



Σχήμα 3.3.7 Δίκτυο Γ' - Χωρισμένο (1η φάση)

διαδρομή είναι η ίδια με αυτή που είχαν και να μην κάνουν κάποια αλλαγή. Με τον τρόπο αυτό βλέπουμε πως το πρόβλημα περιορίζεται σε έκταση στο μικρό δακτύλιο R2–R4–R8–R5 και λύνεται τοπικά (αν το χόστος της σύνδεσης R9–R10 ήταν μικρότερο, ή αν το χόστος της σύνδεσης R5–R8 ήταν μεγαλύτερο, η λειτουργία του αλγορίθμου θα ήταν διαφορετική). Μέσα στο δακτύλιο αυτό οι δρομολογητές οι οποίοι δρομολογούν μέσω του R4 προς το δίκτυο Δ10 είναι οι R2 και R5 και όπως είναι αναμενόμενο, θα αρχίσουν να χρησιμοποιούν ο καθένας τους διαφορετική διαδρομή. Το σχήμα 3.3.7 δείχνει τη νέα κατάσταση. Εδώ παρατηρούμε ότι η πλειοψηφία των δρομολογητών δρομολογεί την κίνησή της μέσω του R4, παρόλο που έχει εφαρμοστεί ο αλγόριθμος. Αυτό γίνεται γιατί η παραδοχή που είχαμε κάνει στη σελίδα 48 (πως κάθε δρομολογητής συμβάλει ισόποσα στο φόρτο του δικτύου) δεν ισχύει εδώ. Θέλουμε να πιστεύουμε ότι στη πράξη κάθε ένας από τους δρομολογητές ενός δικτύου εξυπηρετεί κατά μέσο όρο τον ίδιο όγκο δικτυακής κίνησης τόσο από δικά του δίκτυα, όσο και από άλλους δρομολογητές οι οποίοι δρομολογούν μέσω αυτού.

Για τις ανάγκες του παραδείγματος θεωρούμε ότι αφού έχουν εφαρμοστεί τα παραπάνω, εμφανίζει πρόβλημα και ο δρομολογητής R5. Πλέον, η δρομολόγηση στο Δίκτυο Γ' θα υποστεί αρκετές αλλαγές μιας και σχεδόν όλοι οι δρομολογητές θα αναγκαστούν να εφαρμόσουν τον αλ-



Σχήμα 3.3.8 Δίκτυο Γ' - Χωρισμένο (2η φάση)

γόριθμο. Όπως αναφέρθηκε στη παράγραφο 3.3.1, ο αλγόριθμος θα λειτουργήσει επαναληπτικά και κάθε δρομολογητής θα στηριχθεί στα δεδομένα που ισχύουν μετά τη πρώτη επανάληψη. Το αποτέλεσμα της εκτέλεσης του αλγορίθμου φαίνεται στην εικόνα 3.3.8

Για να καταλάβουμε πως προκύπτει η εικόνα αυτή θα πρέπει να εξετάσουμε τον κάθε ένα δρομολογητή ξεχωριστά:

R1

Ο R1 έχει δύο διαφορετικές διαδρομές προς το Δ10:

- Από τη διαδρομή προς τον R2 (διαδρομή 1): Για να φτάσει τον R5 χρησιμοποιεί τη διαδρομή R1–R2–R5. Αυτό σημαίνει πως μεσολαβεί ένας δρομολογητής και ο οποίος δρομολογεί μέσω του R5³ ($N_1=1$). Το κόστος αυτής της διαδρομής είναι $K_1=4$.

³ Εδώ πρέπει να σημειωθεί ότι ο αλγόριθμος λειτουργεί στοχαστικά σε αυτή τη περίπτωση. Ο δρομολογητής R1 (και κάθε άλλος δρομολογητής πλην του R5), θέλοντας να υπολογίσει πόσοι δρομολογητές από το path R1→R10 δρομολογούν μέσω του R5, ελέγχει τη διαδρομή R1→R5. Είναι πιθανό η διαδρομή αυτή (R1→R5={R1, R2, R5}) να μην εμπεριέχεται στη διαδρομή R1→R10={R1, R2, R4, R8, R10} λόγω προηγούμενης εκτέλεσης του αλγορίθμου. Στην περίπτωση αυτή το δίκτυο μπορεί να οδηγηθεί σε αλλαγή η οποία δεν είναι απαραίτητη και δεν το ωφελεί για το συγκεχριμένο destination. Πράγματι, αν παρατηρήσουμε το σχήμα βλέπουμε πως μετά τη πρώτη εφαρμογή του αλγορίθμου κανένας δρομολογητής δεν δρομολογεί προς το δίκτυο Δ10 μέσω του R5.

- Από τη διαδρομή προς τον R3 (διαδρομή 2): Για να φτάσει τον R10 χρησιμοποιεί τη διαδρομή R1–R3–{R6–R9–R10, R7–R9–R10}, στην οποία υπάρχουν 4 ενδιάμεσοι δρομολογητές. Το κόστος της διαδρομής αυτής προς το Δ10 είναι K2=14. Για κάθε ένα από τους τέσσερις αυτούς δρομολογητές ελέγχει το κόστος προς αυτούς (KX) και αποφασίζει σύμφωνα με την ανισότητα $K1 + KX \leq K2 - KX$. Εφόσον αυτό επαληθεύει και για τους 4 δρομολογητές, ορίζεται N2=4.

Συγκρίνοντας τα N1 και N4, ο R1 αποφασίζει πως πρέπει να χρησιμοποιήσει τη πρώτη διαδρομή μέσω του R2.

R3

Ο R3 έχει τρεις διαφορετικές διαδρομές προς το Δ10. Σύμφωνα με τα προηγούμενα προκύπτει:

- Από τη διαδρομή προς τον R1 (διαδρομή 1): N1=2, K1=3.
- Από τη διαδρομή προς τον R6 (διαδρομή 2): N2=2, K2=13.
- Από τη διαδρομή προς τον R7 (διαδρομή 3): N3=2, K3=13.

Αθροιζόντας έχουμε $N = N1 + N2 + N3 = 6$. Το κριτήριο για να επιλέξει μια διαδρομή για χρήση είναι το NX της διαδρομής X, το οποίο πρέπει να είναι μικρότερο από το $\frac{N}{3}$ (όπου το 3 είναι το πλήθος των πιθανών διαδρομών). Εφόσον όλες οι διαδρομές επαληθεύουν τη συνθήκη αυτή ($NX \leq \frac{N}{3}$) χρησιμοποιούνται στο σύνολό τους.

R2, R6, R7, R9

Εφαρμόζονται τα ίδια βήματα.

3.4 Ψευδοχώδικας

Συνολικά χρειάζεται να ορίσουμε τις παρακάτω συναρτήσεις:

| | |
|---|--------------|
| Συνάρτηση: | run_spf |
| Δεδομένα: | from, ignore |
| Αποτελέσματα: | prev, dist |
| Περιγραφή: | |
| Ο κόμβος από τον οποίο θα τρέξει ο αλγόριθμος. | |
| from: Μια λίστα από κόμβους τους οποίους θα αγνοήσει από το αρχικό γράφο. | |
| prev: Ένας πίνακας ο οποίος για κάθε στοιχείο (κόμβο) κρατάει μια λίστα με τους προηγούμενους κόμβους του δέντρου. | |
| dist: Ένας πίνακας όπου για κάθε στοιχείο(κόμβο) κρατάει την απόστασή του. | |

| | |
|---|---------------|
| Συνάρτηση: | find_paths_to |
| Δεδομένα: | from, to |
| Αποτελέσματα: | via |
| Περιγραφή: | |
| Επιστρέφει τα next-hops για ένα συγκεκριμένο προορισμό | |
| from: Ο κόμβος από τον οποίο θα ελεγχθούν τα routes. | |
| to: Ο κόμβος προς τον οποίο θα βρεθούν τα next-hops. | |
| via: Μια λίστα με next-hops. | |

| | |
|--|-------------------|
| Συνάρτηση: | find_intermediate |
| Δεδομένα: | from, to |
| Αποτελέσματα: | hl |
| Περιγραφή: | |
| Βρίσκει τους ενδιάμεσους κόμβους σύμφωνα με την τελευταία εκτέλεση του αλγορίθμου του Dijkstra. Αν υπάρχουν περισσότερες από μία ισοδύναμες διαδρομές, τις επισκέπτεται όλες επιστρέφοντας μόνο μία φορά τον κάθε κόμβο. | |
| from, to: Οι κόμβοι μεταξύ των οποίων θα επιστρέψει τους ενδιάμεσους. | |
| hl (hoplist): Μια λίστα με τους ενδιάμεσους κόμβους. Περιλαμβάνει το from αλλά όχι το to. | |

| | |
|--|-----------------|
| Συνάρτηση: | get_next_ignore |
| Δεδομένα: | lastignore |
| Αποτελέσματα: | nextignore |
| Περιγραφή: | |
| Επιστρέφει τον επόμενο κόμβο που παρουσίασε συμφόρηση. | |
| lastignore: Ο τελευταίος κόμβος ο οποίος επιστράφηκε από τη συνάρτηση ή -1 αν είναι η πρώτη φορά. | |
| nextignore: Ο επόμενος κόμβος με πρόβλημα, ή -1 αν δεν υπάρχει άλλος. | |

| | |
|---|----------------|
| Συνάρτηση: | get_via_interm |
| Δεδομένα: | from, to |
| Αποτελέσματα: | via, interm |
| Περιγραφή: | |
| Επιστρέφει τα next-hops από όλα τα διαθέσιμα καλύτερα routes καθώς και τους ενδιάμεσους κόμβους για κάθε route. | |
| from, to: Ο αρχικός και ο τελικός κόμβος για την αναζήτηση. | |
| via: Μια λίστα με next-hop. | |
| interm: Ένας πίνακας όπου για κάθε στοιχείο(next-hop) υπάρχει μια λίστα με ενδιάμεσους κόμβους προς τον προορισμό. | |

| | |
|--|---------------|
| Συνάρτηση: | update_rtable |
| Δεδομένα: | r |
| Αποτελέσματα: | |
| Περιγραφή: | |
| Εφαρμόζει τον αλγόριθμο | |
| r: Ο δρομολογητής για τον οποίο θα εφαρμοστεί ο αλγόριθμος. | |

Ο αλγόριθμος run_spf περιγράφηκε στην παράγραφο 2.5. Οι υπόλοιποι αλγόριθμοι έχουν ως εξής:

3.4.1 find_paths_to

- ¹ via $\leftarrow []$
- ² toecheck $\leftarrow [to]$

```

3  beenthere ← [ ]
4  while tocheck ≠ ∅
5      previd ← pop(tocheck)
6      prevlist ← prev(previd)
7      foreach t ← prevlist
8          if t = from
9              if previd ∉ via
10                 via ← via ∪ previd
11
12          else
13              if t ∉ beenthere
14                  tocheck ← tocheck ∪ t
15                  beenthere ← beenthere ∪ t

```

3.4.2 find_intermediate

```

1  if from = to
2      hl ← [ ]
3      return
4  foreach t ← prev(to)
5      hl2 ← find_intermediate(from, t)
6      hl2 ← hl2 ∪ t
7      foreach t2 ← hl2
8          if t2 ∉ hl
9              hl ← hl ∪ t2

```

3.4.3 get_next_ignore

```

1  if lastignore = -1
2      t_last ← 0
3  else
4      t_last ← timeof(lastignore)
5  t_candidate ← ∞
6  id_candidate ← -1

```

```

7  foreach t  $\leftarrow$  overloaded_nodes
8      if (timeof(t) = t.last and lastignore < t and
9          (t < id_candidate or id_candidate < 0)) or
10         (timeof(t) > t.last and timeof(t) = t.candidate and t < id_candidate) or
11         (timeof(t) > t.last and timeof(t) < t.candidate))
12         id_candidate  $\leftarrow$  t
13         t_candidate  $\leftarrow$  timeof(t)
14     nextignore  $\leftarrow$  id_candidate      hl2  $\leftarrow$  find_intermediate(from, t)
15     hl2  $\leftarrow$  hl2  $\cup$  t
16     foreach t2  $\leftarrow$  hl2
17         if t2  $\notin$  hl
18             hl  $\leftarrow$  hl  $\cup$  t2

```

3.4.4 get_via_interm

```

1  via  $\leftarrow$  find_paths_to(from, to)
2  interm  $\leftarrow$  [ ]
3  foreach t  $\leftarrow$  via
4      interm(t)  $\leftarrow$  find_intermediate(t, to)

```

3.4.5 update_rtable

```

1  rtable  $\leftarrow$  [ ]
2  foreach destid  $\leftarrow$  nodes
3      ignore  $\leftarrow$  [ ]
4      ignoreid  $\leftarrow$  -1
5      newvia  $\leftarrow$  [ ]
6      newinterm  $\leftarrow$  [ ]
7      newdist  $\leftarrow$  [ ]
8      routevia  $\leftarrow$  [ ]
9      routecost  $\leftarrow$  -1
10     loopcount  $\leftarrow$  0
11     repeat
12         loopcount  $\leftarrow$  loopcount +1

```

```

13      oldvia ← newvia
14      olldist ← newdist
15      oldcost ← newcost
16      totalnewhopcount ← 0
17      newhopcount ← [ ]
18      if oldvia ≠ ∅
19          ignoreid ← get_next_ignore(ignoreid)
20          (ignorevia, ignoreinterm) ← get_via_interm(r, ignoreid)
21          foreach i ← ignoreinterm
22              cnt ← count(i)
23              newhopcount(i) ← cnt
24              totalnewhopcount ← cnt
25          if r ≠ ignoreid
26              ignore ← ignore ∪ ignoreid
27          if r = ignoreid
28              continue
29          run_spf(r, ignore)
30          newcost ← dist(destid)
31          newdist ← dist
32          (newvia, newinterm) ← get_via_interm(r, destid)
33          if newvia = ∅
34              routevia ← oldvia
35              routecost ← oldcost
36              break
37          if oldvia = ∅
38              routevia ← oldvia
39              routecost ← oldcost
40              continue
41          nochange ← false
42          foreach t ← oldvia
43              if t ∉ newvia
44                  nochange ← true

```

```

45      if nochange = true
46          routevia ← oldvia
47          routecost ← oldcost
48          continue
49          newcost ← 10000
50          via ← oldvia ∪ newvia
51          foreach i ← newinterm
52              c1 ← olldist(destid)
53              c3 ← newdist(destid)
54              cnt ← 0
55              foreach i2 ← i
56                  c2 ← newdist(i2)
57                  if c1 + c2 ≤ c3 - c2
58                      cnt ← cnt + 1
59                  newhopcount(i) ← cnt
60                  totalnewhopcount ← totalnewhopcount + cnt
61                  newvia ← []
62                  interm ← newinterm
63                  newinterm ← []
64                  t1 ← totalnewhopcount / count(via)
65                  foreach i ← via
66                      if newhopcount(i) ≤ t1
67                          newvia ← newvia ∪ i
68                      routevia ← newvia
69                      routecost ← newcost
70                  until loopcount > count(problematic_nodes)
71                  rtable(destid) ← (routevia, routecost)

```

3.4.6 Χρήση από OSPF

To OSPF αποτελεί το ιδανικό πρωτόκολλο για την εφαρμογή του συγκεκριμένου αλγορίθμου γιατί:

- Είναι Link State πρωτόκολλο και ο αλγόριθμος μπορεί να ενσωματωθεί μόνο σε πρωτόκολλα αυτής της οικογένειας.
- Είναι ανοιχτό πρωτόκολλο με αρκετές υλοποιήσεις, τόσο για κανονική χρήση, όσο και για ακαδημαϊκή.
- Υπάρχει το RFC2370 με τίτλο “The OSPF opaque LSA Option” (Coltun 1998), το οποίο παρέχει ένα γενικευμένο τρόπο επέκτασης του OSPF χωρίς να χρειάζονται τροποποιήσεις στη λειτουργία μεταφοράς των δεδομένων στις υπάρχουσες εγκαταστάσεις.

Για την εφαρμογή του απαιτείται η προσθήκη δύο νέων πληροφοριών. Το RFC2370 ορίζει τρεις τύπους LSA (9, 10, 11) μέσω των οποίων μεταφέρει τις πληροφορίες. Για το συγκεκριμένο αλγόριθμο πρέπει να χρησιμοποιηθεί το LSA Type 10, το οποίο προορίζεται για αποστολή πληροφοριών εντός μιας περιοχής. Πέρα από αυτό απαιτείται η χρήση ενός Opaque Type ID, το οποίο ανατίθεται από το Internet Assigned Numbers Authority (IANA). Το εύρος του Opaque Type ID είναι 8 bit και μπορεί να πάρει τις τιμές 1–255. Από αυτές, οι τιμές 128–255 προορίζονται για δοκιμές και ιδιωτική χρήση και μπορούν να χρησιμοποιηθούν προσωρινά σε κάποια πιθανή υλοποίηση.

Για τη λειτουργία του αλγορίθμου θα πρέπει να προστεθούν μία ή δύο νέες πληροφορίες:

- **Router OK:** Για να αποστέλλεται όταν ένας δρομολογητής επανέρχεται σε κανονική λειτουργία.
- **Router Overloaded:** Για να αποστέλλεται όταν ένας δρομολογητής αντιμετωπίζει προβλήματα υπερφόρτωσης

Ο κάθε ένας δρομολογητής πρέπει να αποφασίζει μόνος του για το πότε θα αποστέλλει την κάθε μία πληροφορία. Αυτό είναι απαραίτητο γιατί δεν μπορεί να υπάρξει κάποιος γενικός τρόπος για τη μέτρηση του φόρτου των δρομολογητών. Η πληροφορία του φόρτου είναι κάτι το οποίο εξαρτάται τόσο από τους κατασκευαστές, όσο και από τα μοντέλα των δρομολογητών. Παρόλα αυτά, σαν γενικές κατευθύνσεις μπορούμε να πούμε πως:

- Ένας δρομολογητής θα πρέπει να υλοποιεί κάποιο timer, ο οποίος να μετρά το χρόνο κατά τον οποίο είναι υπερφορτωμένος. Τα Router Overloaded μηνύματα δεν πρέπει να στέλνονται αμέσως αλλά μετά το πέρας κάποιου χρονικού διαστήματος ώστε να μην προκαλούνται αναταραχές στο δίκτυο από περιστασιακά γεγονότα.

- Κάθε δρομολογητής θα πρέπει να έχει ένα όριο “ανοχής” κατά το οποίο να διαφέρουν τα όρια στα οποία θεωρείται υπερφορτωμένος η όχι. Π.χ., ένας δρομολογητής μπορεί να θεωρείται υπερφορτωμένος όταν αποκτήσει επεξεργαστικό φόρτο μεγαλύτερο από 70%, αλλα στη συνέχεια θα θεωρήσει ότι έπαψε να αντιμετωπίζει πρόβλημα υπερφόρτωσης αν ο φόρτος πέσει κάτω από το 40% για συγκεκριμένο χρονικό διάστημα. Αυτό είναι απαραίτητο ώστε να παραμένει το δίκτυο όσο γίνεται περισσότερη όρα στην ίδια κατάσταση.
- Οι δρομολογητές θα πρέπει να ελέγχουν αν οι αλλαγές καταστάσεων είναι αλυσιδωτές (δηλαδή αν η μία προκαλεί την άλλη), όσον αφορά τον εαυτό τους, και να προσπαθούν να το αποτρέψουν αυτό, μεταβάλλοντας ίσως τα όρια ή τους timers.

3.5 Υλοποίηση

Για τις ανάγκες της εργασίας υλοποιήθηκε ο αλγόριθμος στη γλώσσα προγραμματισμού C/C++. Ταυτόχρονα υλοποιήθηκε και ότι άλλο ήταν απαραίτητο ώστε να μπορεί να λειτουργήσει και να δοκιμαστεί σε προχαθορισμένους και σε τυχαίους γράφους. Η υλοποίηση λαμβάνει ως είσοδο ένα γράφο με τυχαίο αριθμό κόμβων και μια λίστα από κόμβους οι οποίοι εμφανίζουν προβλήματα συμφόρησης. Το πρόγραμμα εφαρμόζει τον αλγόριθμο που περιγράφηκε και εμφανίζει τους πλήρεις πίνακες δρομολόγησης για όλους τους δρομολογητές. Μετά την εκτέλεση του αλγορίθμου πραγματοποιεί εξαντλητικό έλεγχο από όλους τους δρομολογητές προς όλους τους πιθανούς προορισμούς ώστε να εντοπίσει πιθανά routing loops. Η διαδικασία του ελέγχου εμφάνισε κάποια προβλήματα τόσο στις αρχικές ιδέες όσο και στην υλοποίηση του αλγορίθμου, τα οποία λήφθηκαν υπόψη και είναι λυμένα στον αλγόριθμο που παρουσιάστηκε στην εργασία.

Πιστεύεται πως ο αλγόριθμος δεν παρουσιάζει κάποιο πρόβλημα γιατί:

- Ακολουθεί ένα απλό λογικό πλάνο το οποίο καθορίζει με αρκετά ξεκάθαρο τρόπο ότι δεν θα υπάρξουν routing loops.
- Κατά τη διάρκεια του προγραμματισμού του ελέγχθηκε απέναντι σε δίκτυα τα οποία ήταν γνωστό ότι θα τον οδηγούσαν σε ακραίες καταστάσεις, όπως αυτές φαίνονταν κατά τη συγγραφή του κώδικα.
- Υλοποιήθηκε γεννήτρια γράφων η οποία: Από το σύνολο όλων των γράφων με 9 κόμβους επέλεξε τους γράφους όπου όλοι οι κόμβοι είχαν περισσότερες από μία συνδέσεις αλλά λιγότερες από 6 (2–5). Από αυτούς ξεχώρισε τυχαία περίπου 800.000 γράφους, οι οποίοι

προέκυψαν από τυχαία διαλογή με πιθανότητα 0.001 (0.1%) από το αρχικό δείγμα. Ο αλγόριθμός εφαρμόστηκε στους 800.000 γράφους και δεν παρουσίασε κάποιο routing loop. Επείσης έγινε επιτυχής εξαντλητικός έλεγχος σε όλους τους γράφους 7 κόμβων όπου ο κάθε κόμβος είχε 2–5 συνδέσεις.

Ενδεικτικό της εξόδου του προγράμματος είναι το παράρτημα α'. Σε αυτό υπάρχει το αποτέλεσμα της εκτέλεσης του προγράμματος πάνω στο δίκτυο της εικόνας **3.3.6**, το οποίο χρησιμοποιείται για το παράδειγμα της παραγράφου **3.3.3**.

Η υλοποίηση του αλγορίθμου αποτελεί ελεύθερο λογισμικό και διανέμεται κάτω από τη GNU Public License (GPL).

Κεφάλαιο 4

Συμπεράσματα

Αφού μελετήσαμε το θέμα του φόρτου των δρομολογητών και δείξαμε ότι η υπερφόρτωση οδηγεί σε καταστροφικές συνέπειες, βρήκαμε έναν τρόπο με τον οποίο μπορεί να μετριαστεί το πρόβλημα. Ο αλγόριθμος της αποφυγής των υπερφορτωμένων κόμβων αποτελεί κάτι καινούριο μιας και δεν βρέθηκε σχετική δουλειά σε έρευνα που έγινε στον Internet και δείχνει πως ίσως μπορεί να χρησιμοποιηθεί. Φάνηκε ότι δεν αντιμετωπίζει προβλήματα, τόσο λογικά, όσο και πρακτικά και προκαλεί επιβαρύνσεις μόνο όταν υπάρχει υπερφόρτωση σε κάποιο δρομολογητή του δικτύου. Επιλέχθηκε να προταθεί η ενσωμάτωσή του στο πρωτόκολλο δρομολόγησης OSPF γιατί αυτό συγκεντρώνει όλα τα απαραίτητα χαρακτηριστικά.

Η υλοποίηση του αλγορίθμου οδήγησε στην αντιμετώπιση και λύση προβλημάτων τα οποία δεν είχαν ληφθεί αρχικά υπόψη, ενώ η εφαρμογή του επιβεβαίωσε τη λειτουργία του. Για την απόδειξη του αληθούς και για τις ανάγκες της εργασίας έγιναν επιτυχημένοι έλεγχοι σε αρκετές χιλιάδες τυχαίων γράφων καθώς και σε δεκάδες προκαθορισμένων από εμάς.

Παράρτημα α'

Αποτελέσματα εκτέλεσης

Τα παρακάτω είναι η έξοδος του προγράμματος, όταν αυτό εφάρμοσε τον αλγόριθμο στο γράφο του σχήματος **3.3.6**, όπως πριγράφηκε στη παράγραφο 3.3.3.

Routing table of R1 (id: 1)

| Dest | NextHop | Cost |
|------|---------|-------|
| R2 | - | 1 |
| R3 | - | 1 |
| R4 | R2 | 2 |
| R5 | R2 | 2 |
| R6 | R3 | 2 |
| R7 | R3 | 2 |
| R8 | R2 | 10015 |
| R9 | R3 | 3 |
| R10 | R2 | 10014 |

Routing table of R2 (id: 2)

| Dest | NextHop | Cost |
|------|---------|-------|
| R1 | - | 1 |
| R3 | R1 | 2 |
| R4 | - | 1 |
| R5 | - | 1 |
| R6 | R1 | 3 |
| R7 | R1 | 3 |
| R8 | R4 | 10016 |
| R9 | R1 | 4 |
| R10 | R4 | 10015 |

Routing table of R3 (id: 3)

| Dest | NextHop | Cost |
|------|----------------------------------|------|
| R1 | - | 1 |
| R2 | R1 | 2 |
| R4 | R1 | 3 |
| R5 | R1 | 3 |
| R6 | - | 1 |
| R7 | - | 1 |
| R8 | 3 routes exist with cost 10014 : | |
| | R1 | |
| | R7 | |
| | R6 | |
| R9 | 2 routes exist with cost 2 : | |
| | R7 | |
| | R6 | |
| R10 | 3 routes exist with cost 10013 : | |
| | R1 | |
| | R7 | |
| | R6 | |

Routing table of R4 (id: 4)

| Dest | NextHop | Cost |
|------|---------|------|
| R1 | R2 | 2 |
| R2 | - | 1 |
| R3 | R2 | 3 |
| R5 | R2 | 2 |
| R6 | R2 | 4 |
| R7 | R2 | 4 |
| R8 | - | 1 |
| R9 | R2 | 5 |
| R10 | R8 | 2 |

Routing table of R5 (id: 5)

| Dest | NextHop | Cost |
|------|---------|-------|
| R1 | R2 | 2 |
| R2 | - | 1 |
| R3 | R2 | 3 |
| R4 | R2 | 2 |
| R6 | R2 | 4 |
| R7 | R2 | 4 |
| R8 | - | 10004 |
| R9 | R2 | 5 |
| R10 | R8 | 10005 |

Routing table of R6 (id: 6)

| Dest | NextHop | Cost |
|------|------------------------------|-------|
| R1 | R3 | 2 |
| R2 | R3 | 3 |
| R3 | - | 1 |
| R4 | R3 | 4 |
| R5 | R3 | 4 |
| R7 | 2 routes exist with cost 2 : | |
| | R9 | |
| | R3 | |
| R8 | R9 | 10013 |
| R9 | - | 1 |
| R10 | R9 | 10012 |

Routing table of R7 (id: 7)

| Dest | NextHop | Cost |
|------|------------------------------|-------|
| R1 | R3 | 2 |
| R2 | R3 | 3 |
| R3 | - | 1 |
| R4 | R3 | 4 |
| R5 | R3 | 4 |
| R6 | 2 routes exist with cost 2 : | |
| | R9 | |
| | R3 | |
| R8 | R9 | 10013 |
| R9 | - | 1 |
| R10 | R9 | 10012 |

Routing table of R8 (id: 8)

| Dest | NextHop | Cost |
|------|----------------------------------|------|
| R1 | 2 routes exist with cost 10015 : | |
| | R4 | |
| | R5 | |
| R2 | 2 routes exist with cost 10016 : | |
| | R4 | |
| | R5 | |
| R3 | 2 routes exist with cost 10014 : | |
| | R4 | |
| | R5 | |
| R4 | - | 1 |
| R5 | 2 routes exist with cost 10004 : | |
| | R4 | |
| | - | |
| R6 | 2 routes exist with cost 10013 : | |
| | R4 | |
| | R5 | |
| R7 | 2 routes exist with cost 10013 : | |
| | R4 | |
| | R5 | |
| R9 | 2 routes exist with cost 10012 : | |
| | R4 | |
| | R5 | |
| R10 | - | 1 |

Routing table of R9 (id: 9)

| Dest | NextHop | Cost |
|------|------------------------------|-------|
| R1 | 2 routes exist with cost 3 : | |
| | R7 | |
| | R6 | |
| R2 | 2 routes exist with cost 4 : | |
| | R7 | |
| | R6 | |
| R3 | 2 routes exist with cost 2 : | |
| | R7 | |
| | R6 | |
| R4 | 2 routes exist with cost 5 : | |
| | R7 | |
| | R6 | |
| R5 | 2 routes exist with cost 5 : | |
| | R7 | |
| | R6 | |
| R6 | - | 1 |
| R7 | - | 1 |
| R8 | R10 | 10012 |
| R10 | - | 10011 |

Routing table of R10 (id: 10)

| Dest | NextHop | Cost |
|------|---------|-------|
| R1 | R9 | 10014 |
| R2 | R9 | 10015 |
| R3 | R9 | 10013 |
| R4 | R8 | 2 |
| R5 | R8 | 5 |
| R6 | R9 | 10012 |
| R7 | R9 | 10012 |
| R8 | - | 1 |
| R9 | - | 10011 |

Bibliography

- Coltun, R. (1998, July). The OSPF Opaque LSA Option. RFC 2370 (Proposed Standard). Updated by RFC 3630.
- Coltun, R., D. Ferguson, and J. Moy (1999, December). OSPF for IPv6. RFC 2740 (Proposed Standard).
- David Moore and Vern Paxson and Stefan Savage and Colleen Shannon and Stuart Stanford and Nicholas Weaver (2003). The Spread of the Sapphire/Slammer Worm. Technical report, CAIDA, ICSI, Silicon Defense, UC Berkeley EECS and UC San Diego CSE.
- J. Padhye and V. Firoiu and D. Towsley and J. Krusoe (1998). “Modeling TCP Throughput: A Simple Model and its Empirical Validation”, *Proceedings of the ACM SIGCOMM ’98 conference on Applications, technologies, architectures, and protocols for computer communication*, s. 303–314.
- James Kurose and Keith Ross (2005). *Compter Networking: A top-down approach featuring the internet*, Addison Wesley.
- Keith Sklower (1993). A tree-based routing table for Berkeley Unix. Technical report, University of California, Berkeley.
- Marcel Waldvogel and George Varghese and Jon Turner and Bernhard Plattner (1997, September). Scalable High Speed IP Routing Table Lookups. In: *ACM SIGCOMM ’97*, s. 25–36.
- Radia Perlman (1992). *Interconnections: bridges and routers*, Redwood City, CA, USA, Addison Wesley Longman Publishing Co., Inc.
- Van Jacobson (1988, aug). Congestion Avoidance and Control. In: *ACM SIGCOMM ’88*, s. 314–329, Stanford, CA.
- Moy, J. (1989, October). OSPF specification. RFC 1131 (Proposed Standard). Obsoleted by RFC 1247.
- Moy, J. (1994, March). Multicast Extensions to OSPF. RFC 1584 (Proposed Standard).

Moy, J. (1998, April). OSPF Version 2. RFC 2328 (Standard).

Postel, J. (1981a, September). Internet Protocol. RFC 791 (Standard). Updated by RFC 1349.

Postel, J. (1981b, September). Transmission Control Protocol. RFC 793 (Standard). Updated by RFC 3168.

Stevens, W. (1997, January). TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms. RFC 2001 (Proposed Standard). Obsoleted by RFC 2581.