Π Α Ν Ε Π Ι Σ Τ Η Μ Ι Ο Π Α Τ Ρ Ω Ν

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΗΛΕΚΤΡΟΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ & ΠΛΗΡΟΦΟΡΙΚΗΣ

Δ Ι Π Λ Ω Μ Α Τ Ι Κ Η Ε Ρ Γ Α Σ Ι Α

ΜΕΛΕΤΗ BANDWIDTH BROKERS ΟΣΟΝ ΑΦΟΡΑ ΤΗΝ ΚΑΤΑΝΕΜΗΜΕΝΗ ΛΕΙΤΟΥΡΓΙΑ ΚΑΙ ΤΗ ΣΥΝΕΡΓΑΣΙΑ ΜΕ ΓΕΙΤΟΝΙΚΑ DOMAINS

ΙΩΑΝΝΗΣ ΠΑΠΠΑΣ

Α.Μ. 2508

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ:

Χρήστος Μπούρας, Αναπληρωτής Καθηγητής

ΕΠΙΒΛΕΠΟΝΤΕΣ

Δημήτρης Πρίμπας, Κώστας Στάμος

ΠΑΤΡΑ 2006

ΠΕΡΙΛΗΨΗ

Σε αυτήν τη διπλωματική εργασία θα ασχοληθούμε με το θέμα παροχής Ποιότητας Υπηρεσίας σε IP δίκτυα. Συγκεκριμένα, θα παρουσιαστούν μερικοί από τους μηχανισμούς που υλοποιούν την DiffServ αρχιτεκτονική για παροχή Ποιότητα Υπηρεσίας. Στη συνέχεια θα επικεντρωθούμε στους bandwidth brokers.

Μετά την αναφορά στη λειτουργικότητά τους, παρουσιάζουμε και συγκρίνουμε τέσσερα διαφορετικά μοντέλα που υλοποιούν έναν Bandwidth Broker σε ένα DiffServ δίκτυο. Παρουσιάζουμε τα σχετικά θέματα υλοποίησης στον ns-2 εξομοιωτή και παρουσιάζουμε έναν αριθμό από πειράματα που έγιναν με σκοπό να επαληθεύσουν την υλοποίησή μας, να αναλύσουν τα χαρακτηριστικά απόδοσης των διαφορετικών εναλλακτικών και να εκτιμήσουν τη χρησιμότητα του ns-2 περιβάλλοντος εξομοίωσης για παρόμοιους σκοπούς. Παρουσιάζουμε τα πειράματά μας που στοχεύουν στην διερεύνηση ενός αριθμού από θέματα σχετικά με τη λειτουργία των Bandwidth Broker modules, και συγκεκριμένα πως το ποσοστό αποδοχής, ο φόρτος δικτύου και ο χρόνος απόκρισης του κάθε Bandwidth Broker μοντέλου επηρεάζονται από την τοπολογία δικτύου, ποια πρέπει να είναι τα διαθέσιμα buffers για τα εισερχόμενα αιτήματα και ποιο το χρονικό διάστημα που πρέπει να υπάρχει από την στιγμή που ένα αίτημα υποβάλλεται μέχρι τη στιγμή που η δέσμευση bandwidth λαμβάνει χώρα.

Η διπλωματική αυτή εργασία οδήγησε επίσης στην δημιουργία της παρακάτω δημοσίευσης σε διεθνές συνέδριο που περιγράφεται παρακάτω:

1. **“Using the ns-2 simulation environment to implement and evaluate Bandwidth Broker Models”***2nd Conference on Next Generation Internet Design and Engineering (NGI 2006), Valencia*, C. Bouras, I. Pappas, D. Primpas, K. Stamos, 3 - 5 April 2006

ΠΡΟΛΟΓΟΣ

«Όταν κάποιος βρίσκει το δρόμο του, πρέπει να έχει τη σοφία να αντιλαμβάνεται τα λάθος βήματα. Οι απογοητεύσεις, οι ήττες και η αποκαρδίωση είναι εργαλεία που χρησιμοποιεί ο Θεός για να μας δείξει το δρόμο.»

Μπρίντα, Paulo Coelho

Θα ήθελα να ευχαριστήσω θερμά τον αναπληρωτή καθηγητή του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής κύριο Χρήστο Μπούρα για την ανάθεση της διπλωματικής, την αμέριστη βοήθειά του και την καθοδήγηση που μου παρείχε καθ' όλη τη διάρκεια της εκπόνησής της αλλά και διότι μου έδωσε τη δυνατότητα να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα. Επίσης θα ήθελα να ευχαριστήσω θερμά τους φίλους μου Δημήτρη Πρίμπα και Κώστα Στάμο, υποψήφιους διδάκτορες του τμήματος Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, οι οποίοι με καθοδηγούσαν και με συμβούλευαν σε όλη τη διάρκεια πραγμάτωσης αυτής της διπλωματικής. Τέλος, ευχαριστώ όλα τα παιδιά της ερευνητικής ομάδας 6 του ΙΤΥ για τη βοήθεια που μου προσέφεραν και την υπομονή που έδειξαν στο πρόσωπό μου.

Ιωάννης Παππάς Πάτρα Ιούλιος 2006

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ ..... **III**

ΠΡΟΛΟΓΟΣ ..... **V**

ΠΕΡΙΕΧΟΜΕΝΑ ..... **VII**

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ ..... **XI**

ΛΙΣΤΑ ΠΙΝΑΚΩΝ ..... **XIII**

ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ **15**

ΚΕΦΑΛΑΙΟ 2: QOS ΚΑΙ ΜΗΧΑΝΙΣΜΟΙ **19**

2.1 ΕΙΣΑΓΩΓΗ **21**
2.2 ΜΕΤΡΙΚΕΣ ΠΟΙΟΤΗΤΑΣ **21**
2.3 ΤΥΠΟΙ QOS **23**
2.3.1 INTEGRATED SERVICE (INTSERV) 24
2.3.2 DIFFERENTIATED SERVICE (DIFFSERV) 24
2.3.2.1 Εισαγωγή 24
2.3.2.2 Μηχανισμοί DiffServ 25
2.3.2.3 Υπηρεσίες DiffServ 39
2.3.2.4 Υπηρεσία Less Than Best Effort 43

ΚΕΦΑΛΑΙΟ 3: ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ BANDWIDTH BROKERS **45**

3.1 ΕΙΣΑΓΩΓΗ **47**
3.2 ΣΥΜΒΟΛΑΙΑ ΥΠΗΡΕΣΙΑΣ **48**
3.3 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ BANDWIDTH BROKER **49**
3.3.1 USER/APPLICATION INTERFACE 50
3.3.2 INTER-DOMAIN INTERFACE 50

3.3.2.1 Τοπικό configuration 51
3.3.2.2 Αρχικοποίηση Σύνδεσης 51
3.3.2.3 Διαχείριση Υπηρεσιών 52
3.3.2.4 Κατανομή Πόρων 54
3.3.2.5 Διατήρηση της σύνδεσης 55
3.3.3 INTRA-DOMAIN INTERFACE 56
3.3.3.1 Μηχανισμοί υλοποίησης του interface 56
3.3.3.2 Οι δυνατότητες του intra-domain interface 59
3.3.4 ROUTING INTERFACE 59
3.3.5 ΑΠΟΘΗΚΗ ΔΕΔΟΜΕΝΩΝ, POLICY MANAGER INTERFACE, NETWORK MANAGER INTERFACE 59
3.4 ΠΡΩΤΟΚΟΛΛΑ ΧΡΗΣΗΣ ΤΩΝ BANDWIDTH BROKER 59
3.4.1 ΠΡΩΤΟΚΟΛΛΟ COPS 60
3.4.1.1 Το βασικό μοντέλο COPS 60
3.4.1.2 Το πρωτόκολλο COPS 62
3.4.1.3 Ακεραιότητα μηνυμάτων 65
3.4.1.4 Πρωτόκολλο COPS για Policy Provisioning(COPS-PR) 65
3.4.1.5 Υλοποιήσεις COPS 66
3.4.1.6 Σύνοψη 66

ΚΕΦΑΛΑΙΟ 4: Ο ΕΞΟΜΟΙΩΤΗΣ NS 69

4.1 Ο ΕΞΟΜΟΙΩΤΗΣ NS 71

ΚΕΦΑΛΑΙΟ 5: ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ BANDWIDTH BROKERS 75

5.1 Η ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΩΝ BANDWIDTH BROKER 77
5.2 ΟΙ AGENTS BBEDGE, BBBASE, CENTRALBBBASE, DISTRIBUTEDBBBASE 80
5.3 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ BANDWIDTH BROKER 81
5.3.1 BB INTERFACE 82
5.3.2 ROUTING INFORMATION 84
5.3.3 ADJACENT NODES INTERFACE 85
5.3.4 USER/APPLICATION INTERFACE 86
5.3.5 INTRA DOMAIN INTERFACE 87
5.4 ΖΗΤΗΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ 89
5.4.1 ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΥΛΟΠΟΙΗΜΕΝΩΝ ΜΟΝΤΕΛΩΝ 90

ΚΕΦΑΛΑΙΟ 6: ΠΕΙΡΑΜΑΤΑ 93

6.1 ΕΙΣΑΓΩΓΗ 95
6.2 ΔΟΜΗ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ 95
6.3 ΜΕΛΕΤΩΝΤΑΣ ΤΙΣ ΕΠΙΡΡΟΕΣ ΤΟΥ ΜΕΓΕΘΟΥΣ ΤΟΥ BUFFER 97
6.4 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΔΙΑΦΟΡΕΤΙΚΩΝ ΤΟΠΟΛΟΓΙΩΝ 99
6.4.1 OVERHEAD ΔΙΚΤΥΟΥ 100
6.4.2 ΡΥΘΜΟΣ ΑΠΟΔΟΧΗΣ ΑΙΤΗΜΑΤΩΝ 100
6.4.3 Η ΑΠΟΚΡΙΣΗ ΧΡΟΝΟΥ ΤΩΝ ΜΟΝΤΕΛΩΝ 101

ΚΕΦΑΛΑΙΟ 7: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ.....103

ΠΑΡΑΡΤΗΜΑ Ι: ΑΝΑΦΟΡΕΣ.....107

ΠΑΡΑΡΤΗΜΑ ΙΙ: ΕΓΚΑΤΑΣΤΑΣΗ ΚΩΔΙΚΑ 113

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

ΕΙΚΟΝΑ 1 17
ΕΙΚΟΝΑ 2 ΤΑ ΕΙΔΗ ΤΗΣ ΚΑΘΥΣΤΕΡΗΣΗΣ 22
ΕΙΚΟΝΑ 3 Η ΜΕΤΡΙΚΗ ΠΟΙΟΤΗΤΑΣ JITTER 22
ΕΙΚΟΝΑ 4 ΟΙ ΒΑΣΙΚΟΙ ΜΗΧΑΝΙΣΜΟΙ DATA PATH ΚΑΙ Η ΣΕΙΡΑ ΜΕ ΤΗΝ ΟΠΟΙΑ ΕΚΤΕΛΟΥΝΤΑΙ 26
ΕΙΚΟΝΑ 5 ΤΟ TOS OCTET ΤΗΣ IPV4 ΕΠΙΚΕΦΑΛΙΔΑΣ 27
ΕΙΚΟΝΑ 6 Η IPV4 ΕΠΙΚΕΦΑΛΙΔΑ ΣΥΜΦΩΝΑ ΜΕ ΤΗΝ DIFFSERV ΑΡΧΙΤΕΚΤΟΝΙΚΗ 27
ΕΙΚΟΝΑ 7 Η ΒΑΣΙΚΗ IPV6 ΕΠΙΚΕΦΑΛΙΔΑ 28
ΕΙΚΟΝΑ 8 Η MPLS ΕΠΙΚΕΦΑΛΙΔΑ 29
ΕΙΚΟΝΑ 9 Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ TOKEN BUCKET 30
ΕΙΚΟΝΑ 10 Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ LEAKY BUCKET 31
ΕΙΚΟΝΑ 11 ΜΗΧΑΝΙΣΜΟΣ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ ΤΗΣ ΚΙΝΗΣΗΣ ΣΕ 3 ΕΠΙΠΕΔΑ 31
ΕΙΚΟΝΑ 12 Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ RED 35
ΕΙΚΟΝΑ 13 Η ΛΕΙΤΟΥΡΓΙΑ ΤΟΥ ΜΗΧΑΝΙΣΜΟΥ WEIGHTED RED 36
ΕΙΚΟΝΑ 14 ΔΕΙΓΜΑ ΤΗΣ ΚΑΤΑΣΤΑΣΗΣ ΔΙΚΤΥΟΥ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΕΙ BANDWIDTH BROKERS ΓΙΑ ΤΗ ΔΙΑΧΕΙΡΙΣΗ ΤΟΥ BANDWIDTH 48
ΕΙΚΟΝΑ 15 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΕΝΟΣ BANDWIDTH BROKER 49
ΕΙΚΟΝΑ 16 DOMAINS ΜΕ ΠΟΛΛΑΠΛΕΣ ΔΙΑΣΥΝΔΕΣΕΙΣ 52
ΕΙΚΟΝΑ 17 ΤΟ SNMP ΔΙΕΥΚΟΛΥΝΕΙ ΤΗΝ ΑΝΤΑΛΛΑΓΗ ΤΗΣ ΔΙΚΤΥΑΚΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΑΝΑΜΕΣΑ ΣΤΙΣ ΣΥΣΚΕΥΕΣ 57
ΕΙΚΟΝΑ 18 ΈΝΑ SNMP ΔΙΟΙΚΟΥΜΕΝΟ ΔΙΚΤΥΟ ΑΠΟΤΕΛΕΙΤΑΙ ΑΠΟ ΔΙΑΧΕΡΙΖΟΜΕΝΕΣ ΣΥΣΚΕΥΕΣ ,AGENTS ΚΑΙ NMS 58
ΕΙΚΟΝΑ 19 ΤΟ ΜΟΝΤΕΛΟ COPS 61
ΕΙΚΟΝΑ 20 ΔΙΑΔΙΚΑΣΙΑ ΑΠΟΔΟΧΗΣ ΑΙΤΗΜΑΤΟΣ ΣΤΟ COPS 62
ΕΙΚΟΝΑ 21 Η ΕΠΙΚΕΦΑΛΙΔΑ COPS 63
ΕΙΚΟΝΑ 22 COPS POLICY ELEMENTS 63
ΕΙΚΟΝΑ 23 64
ΕΙΚΟΝΑ 24 ΤΟ ΔΙΚΤΥΟ ΑΠΟΤΕΛΕΙΤΑΙ ΑΠΟ ΤΡΕΙΣ ΚΟΜΒΟΥΣ, ΣΕ ΚΑΘΕ ΕΝΑΝ ΑΝΑΤΙΘΕΤΑΙ ΚΑΙ ΕΝΑΣ EDGE BANDWIDTH BROKER ΚΑΙ ΣΕ ΟΠΟΙΟΔΗΠΟΤΕ ΑΠΟ ΑΥΤΟΥΣ ΤΟΥΣ ΚΟΜΒΟΥΣ ΜΠΟΡΕΙ ΝΑ ΤΟΠΟΘΕΤΗΘΕΙ ΚΑΙ Ο BASE BANDWIDTH BROKER 77
ΕΙΚΟΝΑ 25 ΣΕ ΚΑΘΕ ΚΟΜΒΟ ΤΡΕΧΕΙ ΕΝΑΣ EDGE BANDWIDTH BROKER ΕΝΩ ΕΧΕΙ ΠΡΟΣΤΕΘΕΙ ΚΑΙ ΕΝΑΣ ΕΠΙΠΛΕΟΝ ΚΟΜΒΟΣ (Ο ΑΝΩΤΕΡΟΣ ΚΟΜΒΟΣ) ΣΤΟΝ ΟΠΟΙΟ ΤΡΕΧΕΙ Ο BASE BANDWIDTH BROKER. . 78
ΕΙΚΟΝΑ 26 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ BANDWIDTH BROKER ΠΟΥ ΥΛΟΠΟΙΗΘΗΚΕ 82
ΕΙΚΟΝΑ 27 ΣΕΙΡΙΑΚΗ ΤΟΠΟΛΟΓΙΑ 98
ΕΙΚΟΝΑ 28 98
ΕΙΚΟΝΑ 29 ΠΟΣΟΣΤΟ ΑΠΟΔΟΧΗΣ VS ΜΕΓΕΘΟΣ ΤΟΥ BUFFER(1MS ΚΑΘΥΣΤΕΡΗΣΗ ΓΡΑΜΜΗΣ) 98
ΕΙΚΟΝΑ 30 ΕΠΙΠΛΕΟΝ ΤΟΠΟΛΟΓΙΕΣ ΠΟΥ ΧΡΗΣΙΜΟΠΟΙΗΘΗΚΑΝ ΓΙΑ ΤΑ ΠΕΙΡΑΜΑΤΑ 99

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΠΙΝΑΚΑΣ 1 OVERHEAD ΔΙΚΤΥΟΥ(ΜΕΣΟΣ ΟΡΟΣ ΑΡΙΘΜΩΝ ΠΑΚΕΤΟΥ ΑΝΑ ΑΙΤΗΜΑ) 100
ΠΙΝΑΚΑΣ 2 ΡΥΘΜΟΣ ΑΠΟΔΟΧΗΣ (ΡΥΘΜΟΣ ΑΠΟΔΟΧΗΣ ΤΩΝ ΑΙΤΗΜΑΤΩΝ ΠΟΥ ΥΠΟΒΛΗΘΗΚΑΝ) 100
ΠΙΝΑΚΑΣ 3 ΡΥΘΜΟΣ ΑΠΟΔΟΧΗΣ ΓΙΑ LATENCY 10MS 101
ΠΙΝΑΚΑΣ 4 ΧΡΟΝΟΣ ΑΠΟΚΡΙΣΗΣ(Ο ΜΕΣΟΣ ΧΡΟΝΟΣ ΠΟΥ ΠΕΡΑΣΕ ΜΕΧΡΙ ΤΗ ΣΤΙΓΜΗ ΠΟΥ Η ΑΠΑΝΤΗΣΗ ΕΠΙΣΤΡΕΦΕΙ ΣΤΟ ΑΠΟΣΤΟΛΕΑ ΑΙΤΗΜΑΤΟΣ)(MSEC) 102
ΠΙΝΑΚΑΣ 5 ΤΥΠΙΚΗ ΑΠΟΚΛΙΣΗ ΤΟΥ ΧΡΟΝΟΥ ΑΠΟΚΡΙΣΗΣ(10^{-3}) 102

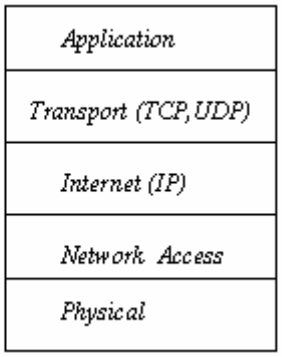
ΚΕΦΑΛΑΙΟ 1: ΕΙΣΑΓΩΓΗ

ΕΙΣΑΓΩΓΗ

Στη σημερινή εποχή τα δίκτυα υπολογιστών διαδραματίζουν πολύ σημαντικό ρόλο. Με την έννοια δίκτυο υπολογιστών ονομάζουμε ένα σύνολο από υπολογιστές που είναι διασυνδεδεμένοι μεταξύ τους με μέσα μετάδοσης. Τα δίκτυα συνεπώς αποτελούνται από υπολογιστές και μέσα μετάδοσης. Και οι δυο αυτοί παράγοντες θεωρούνται κρίσιμοι για την λειτουργία των δικτύων, για το λόγο αυτό τα τελευταία χρόνια έχει συντελεσθεί πολύ σημαντική πρόοδος, που οφείλεται στην ισόρροπη εξέλιξη και των δύο.

Η ανάπτυξη σύγχρονων και αξιόπιστων δικτύων υπολογιστών οδήγησε σήμερα πολλές πτυχές της οικονομίας και κατ' επέκταση και της καθημερινής μας ζωής να βασίζονται σε αυτά. Αυτά έχουν κυριαρχήσει σε όλες τις εταιρίες και έχουν εισχωρήσει τα τελευταία χρόνια και στην προσωπική ζωή του ανθρώπου μέσω του Διαδικτύου (Internet).

Η σχεδίαση των δικτύων έγινε σε επίπεδα όπου το καθένα «χτίζεται» πάνω στο κατώτερό του. Ο λόγος είναι ότι με το τρόπο αυτό μειώνεται η πολυπλοκότητα στη σχεδίαση των δικτύων. Χαρακτηριστικό μοντέλο σχεδίασης ενός δικτύου αποτελεί το μοντέλο TCP/IP όπου είναι και αυτό που χρησιμοποιείται σήμερα στο Διαδίκτυο. Αυτό αποτελείται από 5 επίπεδα, όπως παρουσιάζεται και στην Εικόνα 1.



Εικόνα 1

Κύριος στόχος του μοντέλου TCP/IP είναι να μπορεί να συνδέει με διάφανο τρόπο πολλαπλά δίκτυα, τα οποία πιθανώς θα χρησιμοποιούν διαφορετικές τεχνολογίες. Αυτό επιτυγχάνεται με την ανεξαρτησία και τη διάφανη λειτουργία των επιπέδων. Το βασικότερο επίπεδο του μοντέλου είναι το επίπεδο διαδικτύου (internet layer) που η δουλειά του είναι να επιτρέπει στους πελάτες να εισάγουν πακέτα στο δίκτυο σε οποιοδήποτε σημείο και στη συνέχεια αυτό αναλαμβάνει να τα αποστείλει στον προορισμό. Ενδεχομένως ο προορισμός να βρίσκεται σε άλλο δίκτυο από αυτό που βρίσκεται η πηγή, και επίσης πρέπει να σημειωθεί ότι ο χειρισμός κάθε πακέτου γίνεται ξεχωριστά. Συνεπώς αν ένας πομπός εισάγει πολλά πακέτα στη σειρά μπορεί αυτά να ληφθούν από τον παραλήπτη με διαφορετική σειρά και να έχουν ακολουθήσει διαφορετικές διαδρομές προκειμένου να φτάσουν σ' αυτόν. Ουσιαστικά το επίπεδο διαδικτύου καθορίζει μια προτυποποιημένη μορφή πακέτου (IP packet) και πρωτοκόλλου που λέγεται πρωτόκολλο διαδικτύου IP (Internet Protocol) και έχει ως στόχο να μεταδίδει στον παραλήπτη τα IP πακέτα.

Το μεγαλύτερο IP δίκτυο είναι φυσικά το Internet. Σε ένα τέτοιο πραγματικό IP δίκτυο, η βασική υπηρεσία που προσφέρεται είναι η υπηρεσία best effort (καλύτερης προσπάθειας). Σύμφωνα με αυτή, κάθε πακέτο που φτάνει σε ένα δρομολογητή δέχεται την ακόλουθη επεξεργασία:

- Αρχικά γίνεται έλεγχος για το που θα σταλεί το πακέτο που μόλις έφτασε.
- Στη συνέχεια το πακέτο στέλνεται στη γραμμή εξόδου για το επόμενο hop. Εάν δεν είναι δυνατό το πακέτο να σταλεί άμεσα αυτό αποθηκεύεται προσωρινά σε μια ουρά εξόδου.
- Εάν η ουρά αυτή είναι γεμάτη, το πακέτο απορρίπτεται. Σε περίπτωση που όταν φτάσει το πακέτο η ουρά περιέχει ήδη άλλα πακέτα, τότε το πακέτο αυτό δέχεται επιπλέον καθυστέρηση σύμφωνα με το χρόνο που απαιτείται ώστε τα παλιότερα πακέτα να φύγουν από την ουρά.

Ουσιαστικά στην best effort υπηρεσία όλα τα πακέτα αντιμετωπίζονται όμοια και δεν υπάρχουν εγγυήσεις, διαφοροποιήσεις ή προσπάθεια επιβολής δικαιοσύνης. Εντούτοις το δίκτυο προσπαθεί να προωθήσει όσο περισσότερη κίνηση μπορεί με «λογική» ποιότητα. Στο δίκτυο πολλές φορές παρουσιάζεται το φαινόμενο της συμφόρησης, που ουσιαστικά συμβαίνει όταν ένας δρομολογητής αποθηκεύει πακέτα σε μια ουρά εξόδου, γεγονός που συμβαίνει όταν λαμβάνει περισσότερα πακέτα από αυτά που μπορεί να μεταδώσει. Στη διάρκεια της περιόδου συμφόρησης είναι λογικό τα πακέτα να δέχονται μεγαλύτερη καθυστέρηση ενώ όταν η ουρά εξόδου γεμίσει, τότε αυτά απορρίπτονται.

Πολλές εφαρμογές, όπως εφαρμογές Voice over IP (VoIP) και video-conference, δεν μπορούν να λειτουργήσουν ικανοποιητικά όταν τα πακέτα τους παρουσιάζουν μεγάλη καθυστέρηση ή όταν χάνεται μεγάλος αριθμός πακέτων τους. Τέτοιες εφαρμογές απαιτούν εγγυήσεις υπηρεσίας τις οποίες η best-effort φύση του Internet δε μπορεί να προσφέρει.

Ένας τρόπος προκειμένου να υπάρξει παροχή εγγυήσεων σε κάποια κίνηση είναι η διαχείριση ορισμένων πακέτων διαφορετικά έναντι των υπολοίπων. Στο σημείο αυτό ουσιαστικά εισέρχεται η έννοια της ποιότητας υπηρεσίας (Quality of Service)[1]. Ένας ορισμός της είναι: **«η ικανότητα ενός στοιχείου του δικτύου να παρέχει ένα επίπεδο διαβεβαίωσης (εγγύησης) σε ένα υποσύνολο κίνησης ότι οι απαιτήσεις υπηρεσίας της μπορεί να επιτευχθούν με συγκεκριμένη (πολύ μεγάλη) πιθανότητα»**. Ουσιαστικά οι μηχανισμοί του Quality of Service δεν παρέχουν μεγαλύτερη χωρητικότητα στο δίκτυο ή κάτι παρόμοιο, αλλά απλώς κάνουν καλύτερη διαχείριση του δικτύου ώστε να χρησιμοποιείται πιο αποδοτικά και σύμφωνα με τις απαιτήσεις των εφαρμογών

ΚΕΦΑΛΑΙΟ 2: QoS ΚΑΙ ΜΗΧΑΝΙΣΜΟΙ

QoS ΚΑΙ ΜΗΧΑΝΙΣΜΟΙ

2.1 ΕΙΣΑΓΩΓΗ

Η ποιότητα υπηρεσίας(QoS)[1] είναι ένα σύνολο από προϋποθέσεις υπηρεσίας που πληρούνται σε ένα δίκτυο κατά τη μετάδοση μιας ροής. Οι λειτουργίες Ποιότητας Υπηρεσίας σε ένα IP δίκτυο σκοπεύουν στο να παρέχουν εγγυημένες καθώς και διαφοροποιημένες υπηρεσίες δίνοντας τον έλεγχο των πόρων του δικτύου στο διαχειριστή. Ένα δίκτυο που παρέχει Ποιότητα Υπηρεσίας μπορεί να παρέχει εγγυήσεις υπηρεσίας για ορισμένες μετρικές απόδοσης όπως κατανομή πόρων, δρομολόγηση, χρονοδρομολόγηση πακέτων και μηχανισμούς απόρριψης πακέτων.

2.2 ΜΕΤΡΙΚΕΣ ΠΟΙΟΤΗΤΑΣ

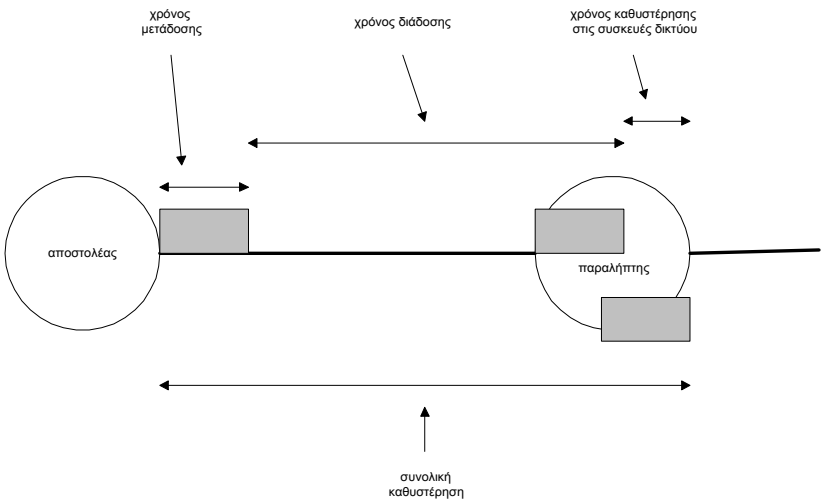
Οι μετρικές που ενδιαφέρουν τις εφαρμογές που ζητούν ποιότητα υπηρεσίας στην εξυπηρέτηση τους είναι γενικά 4 και περιγράφονται αμέσως παρακάτω. Η σημαντικότερη μετρική που ενδιαφέρει και επηρεάζει τις εφαρμογές είναι η **χωρητικότητα (bandwidth)**, που ορίζεται ως το πλήθος των δεδομένων, σε bits per second, που μεταδίδονται από ένα χρήστη στον άλλο. Το bandwidth χαρακτηρίζεται από 4 μεγέθη που είναι:

- Το μέγιστο μέγεθος καταιγισμού (maximum burst size), δηλαδή ο μέγιστος αριθμός πακέτων που μπορούν να βρεθούν στην ουρά του δρομολογητή χωρίς να απορριφθούν. Μία εφαρμογή που κατά τα άλλα συμπεριφέρεται μέσα στα προκαθορισμένα όρια, μπορεί για διάφορους λόγους να στείλει κάποια χρονική στιγμή δεδομένα με ρυθμό καταιγισμού.
- Η μέγιστη χωρητικότητα (peak bandwidth), δηλαδή η ανώτατη επιτρεπόμενη τιμή της χωρητικότητας που επιτρέπεται μία ροή να διατηρήσει σταθερή.
- Η ελάχιστη εγγυημένη χωρητικότητα (minimum guaranteed bandwidth)
- Η μέση χωρητικότητα (average bandwidth), δηλαδή η μέση τιμή της χωρητικότητας που υπολογίζεται διαιρώντας τον αριθμό των bytes που μεταδόθηκαν προς το συγκεκριμένο χρονικό διάστημα.

Η δεύτερη μετρική που ενδιαφέρει σχεδόν όλες τις εφαρμογές είναι η **καθυστέρηση (delay)**, που ορίζεται ως ο χρόνος μεταξύ της μετάδοσης του πρώτου bit ενός IP πακέτου και της λήψης του τελευταίου bit αυτού του πακέτου από τον παραλήπτη(Εικόνα 2). Ουσιαστικά η συνολική αυτή καθυστέρηση ισούται με το άθροισμα των καθυστερήσεων σε κάθε τμήμα του δικτύου. Η καθυστέρηση σε κάθε τμήμα του δικτύου είναι 3 ειδών και συγκεκριμένα:

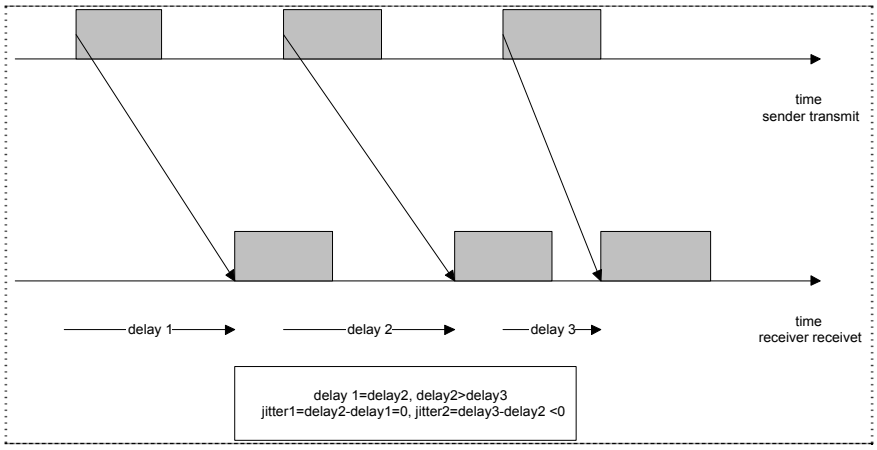
- Χρόνος μετάδοσης, είναι δηλαδή ο χρόνος που απαιτείται για την τοποθέτηση πάνω σε μια γραμμή μετάδοσης όλων των bit του πακέτου και είναι ανάλογος της ταχύτητας της γραμμής
- Χρόνος διάδοσης, είναι ο χρόνος από τη μετάδοση του πρώτου (ή του τελευταίου bit του πακέτου) και τη λήψη αυτού του bit από τον παραλήπτη. Ο χρόνος αυτός εξαρτάται από την τεχνολογία μετάδοσης και την απόσταση.

- Χρόνος καθυστέρησης στις συσκευές του δικτύου. Η καθυστέρηση αυτή εισάγεται στα σημεία που λαμβάνουν πληροφορία και είναι ο χρόνος από τη λήψη τους μέχρι η πληροφορία να μεταδοθεί στην επόμενη συσκευή. Ο χρόνος αυτός αποτελείται από το χρόνο επεξεργασίας και το χρόνο που η πληροφορία παραμένει στην ουρά.



Εικόνα 2 Τα είδη της καθυστέρησης

Η επόμενη μετρική που χαρακτηρίζει την ποιότητα υπηρεσίας είναι το **jitter (IP packet delay variation)**(Εικόνα 3). Ουσιαστικά, το jitter αναφέρεται σε ζεύγη πακέτων και είναι η διαφορά μεταξύ της καθυστέρησης του πρώτου πακέτου από το δεύτερο. Όπως φαίνεται και από το παρακάτω σχήμα, το jitter μεταξύ των πακέτων 1 και 2 ισούται με delay2-delay1 και μεταξύ των πακέτων 2 και 3 με delay3-delay2 αντίστοιχα. Πολλές εφαρμογές απαιτούν να έχουν ένα άνω όριο για το jitter προκειμένου η απόδοσή τους να είναι καλή.



Εικόνα 3 Η μετρική ποιότητας jitter

Τέλος, μια μετρική που ενδιαφέρει πολλές εφαρμογές είναι η **απώλεια πακέτων (packet loss)** και είναι ουσιαστικά το ποσοστό των πακέτων που μεταδόθηκαν από την πηγή και δεν λήφθηκαν από τον παραλήπτη ή παραλήφθηκαν με λάθη. Η απώλεια πακέτων προκαλείται είτε από απώλεια κάποιου link (συνδέσμου του

δικτύου), είτε εξαιτίας προβλημάτων στη ρύθμιση των συσκευών του δικτύου είτε τέλος από συμφόρηση στο δίκτυο. Γενικά η επίδραση της απώλειας πακέτων στις εφαρμογές μπορεί να είναι καταστροφική υποβαθμίζοντας την απόδοσή τους. Επίσης σε πολλές εφαρμογές ενδεχόμενη αποστολή ξανά ενός χαμένου πακέτου δεν έχει καμιά απολύτως σημασία και αντιθέτως δυσχεραίνει την λειτουργία της εφαρμογής παρά την βοηθά. Ένα παράδειγμα τέτοιας εφαρμογής είναι η τηλεδιάσκεψη.

2.3 ΤΥΠΟΙ QoS

Η ικανότητα του δικτύου να παρέχει υπηρεσία στις δικτυακές εφαρμογές κατηγοριοποιείται σε τρία επίπεδα υπηρεσίας:

- **Best-effort υπηρεσία**. Υπάρχει η βασική συνδεσιμότητα χωρίς καμία εγγύηση για το αν και πότε θα φτάσει το κάθε πακέτο στον προορισμό. Η best-effort υπηρεσία δεν είναι μέρος της Ποιότητας Υπηρεσίας διότι δεν δίνεται καμία εγγύηση για την υπηρεσία κατά την προώθηση των πακέτων. Αυτή είναι η υπηρεσία που παρέχει το Internet σήμερα. Οι περισσότερες εφαρμογές λειτουργούν ορθά με την best-effort υπηρεσία αν και μπορεί να έχουν μειωμένη απόδοση.
- **Διαφοροποιούμενη υπηρεσία (Differentiated service)**. Στην differentiated service, η κίνηση ομαδοποιείται σε κλάσεις με βάση τις απαιτήσεις υπηρεσίας. Κάθε κίνηση διαφοροποιείται από το δίκτυο και εξυπηρετείται με διαφορετικό τρόπο από τους μηχανισμούς Ποιότητας υπηρεσίας για κάθε κλάση. Η differentiated service δεν παρέχει αυστηρές εγγυήσεις υπηρεσίας, απλά επιτρέπει την προνομιακή μεταχείριση ορισμένων κλάσεων έναντι άλλων. Για αυτό το λόγο λέγεται και *soft QoS*.
- **Εγγυημένη Υπηρεσία (Integrated service)**. Η integrated service είναι υπηρεσία η οποία υλοποιεί κατακράτηση δικτυακών πόρων ώστε να διασφαλίσει το δίκτυο ότι το πραγματικό προφίλ κάθε ροής συμφωνεί με αυτό που έχει προσυμφωνηθεί μεταξύ του διαχειριστή δικτύου και του πελάτη. Επειδή η integrated service απαιτεί ρητές εγγυήσεις από το δίκτυο λέγεται και *hard QoS*.

Τα κριτήρια για την επιλογή του κατάλληλου τύπου Ποιότητας Υπηρεσίας μπορούν να είναι:

- Η εφαρμογή του πελάτη. Κάθε ένας από τους τρεις τύπους υπηρεσίας είναι κατάλληλος για συγκεκριμένες εφαρμογές.
- Η συχνότητα με την οποία μπορούν οι πελάτες να ανανεώσουν τις υποδομές τους. Κάποιος πελάτης που εξυπηρετείται από differentiated service χρειάζεται να ανανεώσει τον εξοπλισμό του ώστε να μπορεί να λάβει integrated service.
- Το κόστος υλοποίησης και ανάπτυξης της integrated service είναι μεγαλύτερο από αυτό της differentiated service.
- Η integrated service δεν έχει υλοποιηθεί ικανοποιητικά.

Στις επόμενες παραγράφους παρουσιάζεται η integrated service και η differentiated service.

2.3.1 Integrated Service (IntServ)

Η IntServ αρχιτεκτονική [2] δημιουργήθηκε το 1994 ώστε να επεκτείνει το μοντέλο υπηρεσίας του Internet για να εξυπηρετεί καλύτερα εφαρμογές πραγματικού χρόνου όπως εφαρμογές φωνής ή video. Έτσι, ορίστηκε ένα νέο μοντέλο εξυπηρέτησης του Internet με περισσότερες δυνατότητες το οποίο παρέχει τη δυνατότητα να παρέχει εγγυημένους πόρους σε εφαρμογές από άκρο σε άκρο. Αυτό επιτυγχάνεται με το διαχωρισμό των ροών και την ξεχωριστή διαχείριση κάθε ροής. Το πρωτόκολλο που χρησιμοποιείται για τη σηματοδότηση είναι το Resource Reservation Protocol (RSVP)[14].

Το **RSVP** παρέχει δύο τύπους υπηρεσίας:

- Controlled Load: Στην controlled load υπηρεσία το δίκτυο εγγυάται ότι η ροή που επιθυμεί να κατακρατήσει πόρους θα φτάνει στον προορισμό με μία ελάχιστη αλληλεπίδραση από την best-effort κίνηση. Η ροή δηλαδή δέχεται καλύτερη υπηρεσία από την best-effort αλλά δεν παρέχονται αυστηρές εγγυήσεις.
- Guaranteed: Στην guaranteed υπηρεσία υπάρχουν αυστηρές εγγυήσεις ποιότητας υπηρεσίας, φραγμένη καθυστέρηση και μηδενικές απώλειες πακέτων στις ουρές. Η υπηρεσία εξομοιώνει ένα ιδεατό κύκλωμα.

Το IntServ μοντέλο προϋποθέτει την εγγύηση για ποιότητα υπηρεσίας ανά ροή. Σήμερα που υπάρχουν εκατοντάδες χιλιάδες ροές στο Internet, η πληροφορία την οποία θα έπρεπε να γνωρίζουν οι δρομολογητές ώστε να υλοποιούν το IntServ μοντέλο θα ήταν τεράστια. Επίσης η φύση του RSVP είναι προβληματική και μπορεί να δημιουργεί προβλήματα. Αυτά τα πολύ σημαντικά προβλήματα καθιστούν μη εφικτή τη χρήση του IntServ μοντέλου στο Internet για παροχή εγγυήσεων ποιότητας υπηρεσίας και δε θα μας απασχολήσει άλλο σε αυτή τη διπλωματική.

2.3.2 Differentiated Service (DiffServ)

2.3.2.1 Εισαγωγή

Η DiffServ αρχιτεκτονική [3] παρέχει ένα πλαίσιο με το οποίο αυτοί που παρέχουν υπηρεσίες μπορούν να προσφέρουν έναν αριθμό δικτυακών υπηρεσιών όπου κάθε μία έχει διαφορετική απόδοση. Ένας πελάτης μπορεί να επιλέξει το επίπεδο υπηρεσίας που θέλει απλά μαρκάροντας το Differentiated Services Code Point (DSCP) της IP επικεφαλίδας σε μια συγκεκριμένη τιμή αλλά το μαρκάρισμα μπορεί να γίνεται και από το ίδιο το δίκτυο. Αυτή η τιμή καθορίζει την Per Hop Behavior που δίνεται σε κάθε πακέτο. Τυπικά, ο διαχειριστής του δικτύου διαπραγματεύεται με τον πελάτη για ένα προφίλ περιγράφοντας τη συχνότητα με την οποία θα εισέρχεται η κίνηση σε κάθε επίπεδο υπηρεσίας. Επίσης καθορίζεται η συμπεριφορά που θα έχει το δίκτυο αν ένας πελάτης το τροφοδοτήσει με περισσότερα πακέτα από αυτά που αναφέρονται στο προφίλ. Εφαρμόζεται δηλαδή αστυνόμευση στην κίνηση ώστε να αποφεύγονται επιθέσεις τύπου Denial of Service(DoS).

Η DiffServ αρχιτεκτονική καθορίζει μόνο τους βασικούς μηχανισμούς με τους οποίους γίνεται η διαχείριση των πακέτων. Χρησιμοποιώντας αυτούς τους μηχανισμούς ως δομικά στοιχεία, ο διαχειριστής του δικτύου μπορεί να δημιουργήσει μια ποικιλία από υπηρεσίες. Μια υπηρεσία ορίζει κάποια χαρακτηριστικά της μετάδοσης όπως το throughput, την καθυστέρηση, το jitter και την απώλεια πακέτων

σε ένα μονοπάτι του δικτύου. Επίσης, μια υπηρεσία μπορεί να κατηγοριοποιηθεί σύμφωνα με τη σχετική προτεραιότητα στην πρόσβαση της υπηρεσίας στους πόρους του δικτύου. Μόλις οριστεί μια υπηρεσία, καθορίζεται η PHB σε όλους τους κόμβους του δικτύου που την παρέχουν και ανατίθεται ένα DSCP στην PHB. Η PHB (Per Hop Behavior) είναι η συμπεριφορά προώθησης που έχει ένας κόμβος του δικτύου στα πακέτα που έχουν μια συγκεκριμένη τιμή στο DSCP. Όλα τα πακέτα της κίνησης που λαμβάνει την υπηρεσία έχουν αυτήν την τιμή στο πεδίο DSCP της IP επικεφαλίδας.

Όλοι οι κόμβοι σε ένα diffserv domain παρέχουν PHB με βάση το DSCP. Επιπρόσθετα, οι κόμβοι που βρίσκονται στα όρια του δικτύου εκτελούν λειτουργίες ελέγχου της κίνησης που εισέρχεται στο domain. Αυτές οι λειτουργίες περιλαμβάνουν λειτουργίες όπως κατηγοριοποίηση των πακέτων και αστυνόμευση της κίνησης.

Στη επόμενη παράγραφο παρουσιάζονται μηχανισμοί που υλοποιούν την DiffServ αρχιτεκτονική.

2.3.2.2 Μηχανισμοί DiffServ

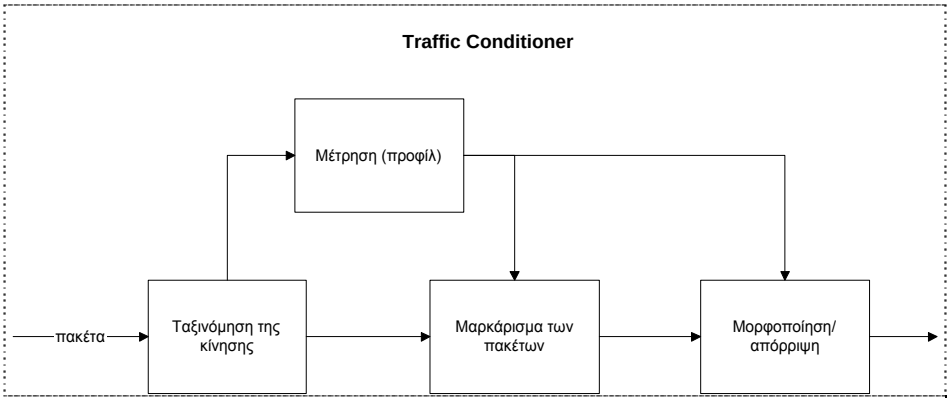
2.3.2.2.1 Εισαγωγή

Προκειμένου να υλοποιηθεί μια υπηρεσία παροχής ποιότητας υπηρεσίας στο επίπεδο δικτύου και ειδικότερα ποιότητα υπηρεσίας με τη μέθοδο DiffServ[3], απαιτείται να λειτουργήσουν στο δίκτυο μια σειρά από μηχανισμούς. Αυτοί ενεργούν πάνω στις ροές και αναλυτικά είναι οι ακόλουθοι.

- Ταξινόμηση των πακέτων (packet classification). Ο μηχανισμός αυτός ταξινομεί τα πακέτα που φτάνουν σε ένα κόμβο σε ροές ή συνενώσεις ροών ώστε στη συνέχεια αυτά να εξυπηρετηθούν κατάλληλα.
- Μαρκάρισμα (marking) των πακέτων. Με το μηχανισμό αυτό τα πακέτα μαρκάρονται ανάλογα με την κλάση στην οποία ανήκουν (προέκυψε από τον προηγούμενο μηχανισμό) είτε με βάση άλλα κριτήρια όπως τα χαρακτηριστικά της κίνησης που παρουσιάζουν κλπ.
- Μέτρηση (metering) της κίνησης. Στην προκειμένη περίπτωση ο μηχανισμός αυτός ελέγχει το προφίλ της κίνησης που δέχεται και το συγκρίνει με το προσυμφωνημένο προφίλ κίνησης όπως προκύπτει από το SLA που έχει υπογράψει με τον διαχειριστή του δικτύου. Στη συνέχεια ο μηχανισμός αυτός διαχωρίζει τα πακέτα σε έναν αριθμό κατηγοριών (ανάλογα αν βρίσκονται στα νόμιμα πλαίσια ή όχι). Ο αριθμός των κατηγοριών αυτών εξαρτάται από τη συμφωνία που έχει γίνει με τον πάροχο όπου επίσης καθορίζεται η μεταχείριση που θα έχουν τα πακέτα όλων των κατηγοριών.
- Μηχανισμός μορφοποίησης (shaping) της κίνησης όπου τροποποιούνται τα χαρακτηριστικά της κίνησης που έλαβε ο κόμβος (δρομολογητής). Επίσης αντί του μηχανισμού αυτού μπορεί να υπάρχει μηχανισμός απόρριψης (dropping) των πακέτων.

Γενικά η σειρά με τη οποία συνήθως αυτοί χρησιμοποιούνται είναι και η σειρά με την οποία παρουσιάστηκαν. Πρέπει στο σημείο αυτό να αναφέρουμε ότι είναι επίσης δυνατό οι μηχανισμοί μαρκαρίσματος και μέτρησης του προφίλ της κίνησης να εμφανίζονται αντίστροφα, δηλαδή πρώτα μέτρηση του προφίλ της κίνησης και ύστερα με βάση αυτό το κριτήριο μαρκάρισμα των πακέτων. Επίσης μετά από τη

διαδικασία μέτρησης του προφίλ, σε ορισμένες «κατηγορίες» πακέτων (και κυρίως στα νόμιμα πακέτα) συνήθως δεν εφαρμόζεται κανένας περαιτέρω μηχανισμός και εισάγονται έτσι στο δίκτυο. Οι μηχανισμοί αυτοί και η σειρά με την οποία συνήθως εφαρμόζονται παρουσιάζεται σχηματικά στην Εικόνα 4.



Εικόνα 4 Οι βασικοί μηχανισμοί data path και η σειρά με την οποία εκτελούνται

Στο σημείο αυτό είναι αναγκαίο να τονιστεί πως όλοι οι παραπάνω μηχανισμοί και λειτουργικότητες εφαρμόζονται στους συνοριακούς κόμβους (edge routers) σε ένα DiffServ enabled domain. Αντίθετα, στους ενδιάμεσους κόμβους (core routers) η DiffServ αρχιτεκτονική προσδιορίζει πως οι παραπάνω μηχανισμοί δεν έχουν καμία εφαρμογή.

2.3.2.2.2 Ταξινόμηση της κίνησης

Η ταξινόμηση της κίνησης είναι το πρώτο σημείο στο οποίο βασίζεται η παροχή ποιότητας υπηρεσίας στην εξυπηρέτηση των πακέτων και για το λόγο αυτό αποτελεί ιδιαίτερα σημαντικό παράγοντα. Η ταξινόμηση των πακέτων προκειμένου να εξυπηρετηθούν κατάλληλα σε ένα δίκτυο που υποστηρίζει QoS γίνεται είτε σε επίπεδο ροών, είτε σε επίπεδο συνενώσεων ροών (aggregates). Η διαδικασία αυτή γίνεται κυρίως με τον έλεγχο της επικεφαλίδας κάθε πακέτου και την άντληση από εκεί κάποιας πληροφορίας με βάση την οποία γίνεται η ταξινόμηση. Γενικά ο μηχανισμός αυτός απαιτείται να είναι πολύ γρήγορος, ακολουθώντας το ρυθμό άφιξης των πακέτων, και ιδιαίτερα ακριβής.

Θεωρητικά οι ροές χαρακτηρίζονται από μια πεντάδα που αποτελείται από:

- Την IP διεύθυνση του αποστολέα
- Τον αριθμό port του αποστολέα
- Την IP διεύθυνση του παραλήπτη
- Τον αριθμό port του παραλήπτη
- Το πρωτόκολλο που χρησιμοποιείται

Κάνοντας τελικά ταξινόμηση ανά ροή με βάση αυτή την πεντάδα, είναι μια διαδικασία αρκετά δύσκολη παρότι όλα αυτά τα πεδία υπάρχουν στην IP επικεφαλίδα (στο σημείο αυτό να σημειώσουμε ότι όλοι αυτοί οι μηχανισμοί λειτουργούν στο επίπεδο δικτύου του OSI μοντέλου). Η δυσκολία έγκειται στο γεγονός ότι απαιτείται η διαδικασία της ταξινόμησης να γίνεται άμεσα και συνεπώς ο έλεγχος τόσων πεδίων απαιτεί μεγάλη επεξεργαστική ισχύ. Η μέθοδος αυτή της ταξινόμησης εφαρμόζεται

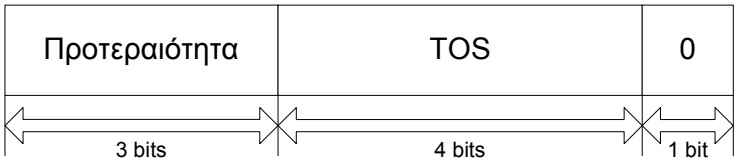
μόνο όταν θέλουμε οπωσδήποτε να κάνουμε ταξινόμηση με βάση ξεχωριστές ροές, γεγονός που συμβαίνει πολλές φορές στην DiffServ αρχιτεκτονική στα σημεία εισόδου της κίνησης σε DiffServ enabled domains. Τέλος αυτή η μέθοδος ονομάζεται Multifield classification.

Αντίθετα στην περίπτωση όπου επιθυμούμε να κάνουμε ταξινόμηση σε συνενώσεις ροών τότε αρκεί να χρησιμοποιηθεί ένας συνδυασμός των παραπάνω πεδίων της πεντάδας που χαρακτηρίζει μια ροή, ή ακόμη και ένα μόνο πεδίο. Η περίπτωση αυτή είναι πιο εύκολη να γίνει και μπορεί τελικά να πραγματοποιείται ταχύτατα σε σύγκριση με τον έλεγχο όλης της πεντάδας.

Στην πραγματικότητα ισχύει πως η ταξινόμηση των πακέτων είναι επιθυμητό να γίνεται σε έναν περιορισμένο αριθμό κατηγοριών (κλάσεων) και συνεπώς αρκεί να χρησιμοποιηθεί ένα σταθερό πεδίο στην επικεφαλίδα των πακέτων. Η μέθοδος αυτή είναι σαφώς απλούστερη και πιο αποδοτική. Στην περίπτωση της DiffServ αρχιτεκτονικής ονομάζεται behaviour aggregate classification και πρέπει να παρατηρήσουμε ότι η ταξινόμηση που επιτυγχάνει είναι σε επίπεδο συνενώσεων ροών (aggregates).

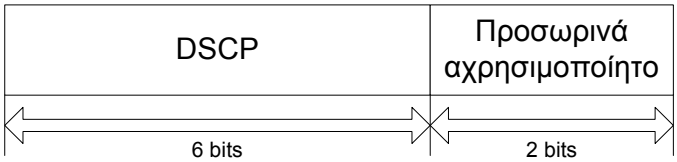
2.3.2.2.2.1 Ταξινόμηση με βάση την IPv4 επικεφαλίδα

Η behaviour aggregate ταξινόμηση πραγματοποιείται χρησιμοποιώντας μια οκτάδα από bits που υπάρχει την επικεφαλίδα των IPv4 πακέτων και η οποία ονομάζεται TOS octet(Εικόνα 5). Σε αυτή τα τρία πρώτα πακέτα δηλώνουν την προτεραιότητα κάθε πακέτου και συνεπώς υπάρχουν 8 διαφορετικές κλάσεις προτεραιότητας. Τα επόμενα 4 bits χαρακτηρίζουν το είδος της υπηρεσίας που επιθυμεί η εφαρμογή, δηλαδή ελαχιστοποίηση της καθυστέρησης, η ελαχιστοποίηση της απώλειας πακέτων κλπ.



Εικόνα 5 Το TOS octet της IPv4 επικεφαλίδας

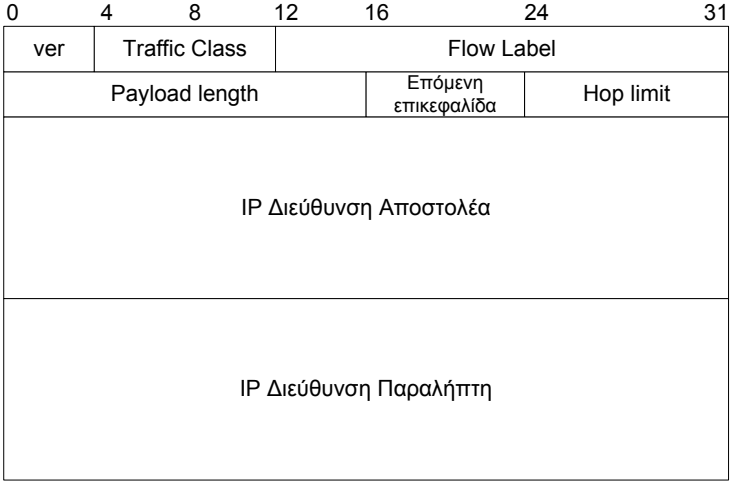
Ακόμη, πρόσφατα καθορίστηκε στο TOS octet τα 6 πιο σημαντικά bits να αναπαριστούν το DiffServ Code Point το οποίο ουσιαστικά δημιουργεί 64 δυνατές συνδυασμούς για τη διαχείριση ουρών και χρονοδρομολόγησης των IP πακέτων(Εικόνα 6). Τέλος να αναφέρουμε πως αντίστοιχο πεδίο έχει οριστεί και για το πρωτόκολλο IPv6.



Εικόνα 6 Η IPv4 επικεφαλίδα σύμφωνα με την DiffServ αρχιτεκτονική

2.3.2.2.2.2 Ταξινόμηση με βάση την IPv6 επικεφαλίδα

Ομοίως στο IPv6 έχει οριστεί αντίστοιχο πεδίο για μαρκάρισμα κίνησης προκειμένου να δεχτεί ποιότητα εξυπηρέτησης. Η Εικόνα 7 παρουσιάζει τη δομή της βασικής επικεφαλίδας του IPv6.

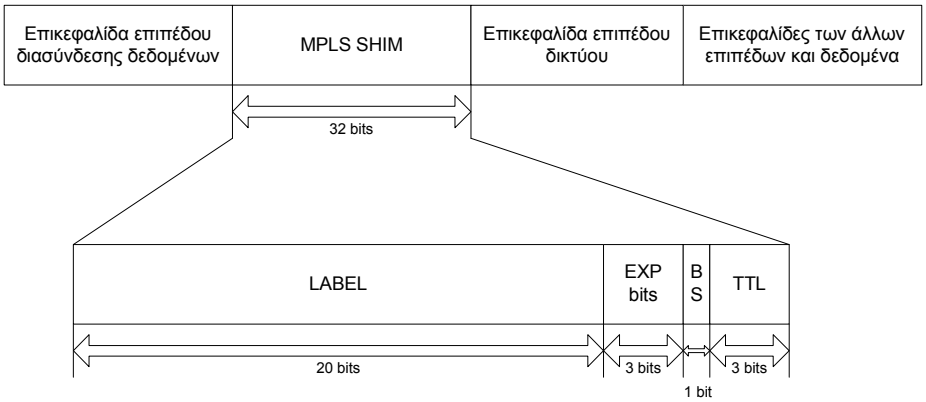


Εικόνα 7 Η βασική IPv6 επικεφαλίδα

Ένα από τα πεδία, όπως φαίνεται παραπάνω ονομάζεται **Traffic Class (Τάξη Κυκλοφορίας)** και έχει μήκος 8 bits. Το πεδίο αυτό περιέχει το πεδίο DSCP (Differentiated Code Point) όπως το αντίστοιχο type of service του IPv4 και χρησιμοποιείται για μαρκάρισμα πακέτων προκειμένου να ανήκουν σε κάποια κλάση υπηρεσίας. Επίσης, το IPv6 πρωτόκολλο έχει εισάγει και ένα νέο πεδίο που ονομάζεται **Flow Label (Ετικέτα Ροής)** και έχει μήκος 20 bit. Αυτό χρησιμοποιείται για να γνωστοποιεί ποια πακέτα ανήκουν σε μια συγκεκριμένη ροή. Ένας κόμβος μπορεί να είναι η αφετηρία για πάνω από μια ροές ταυτόχρονα. Γι' αυτό η ετικέτα ροής σε συνδυασμό με τη διεύθυνση της αφετηρίας μπορούν να αναγνωρίσουν μονοσήμαντα μια ροή. Γενικά πάντως μέχρι σήμερα δεν έχει μοντελοποιηθεί πλήρως η χρήση του.

2.3.2.2.2.3 Ταξινόμηση με βάση την MPLS επικεφαλίδα

Το MPLS αποτελεί ένα σύγχρονο και δυναμικό πρωτόκολλο που ανήκει μεταξύ του επιπέδου 2 και του επιπέδου 3 του ISO/OSI μοντέλου. Το πρωτόκολλο αυτό τοποθετεί μια δική του επικεφαλίδα στα πακέτα (κάτω από την IP επικεφαλίδα) και προωθεί τα πακέτα με βάση τις πληροφορίες της ετικέτας αυτής. Συνεπώς, για να υποστηριχτεί Ποιότητα Υπηρεσίας σε MPLS δίκτυα είναι απαραίτητο το μαρκάρισμα (classification) να γίνεται στην MPLS επικεφαλίδα. Η διάταξη της MPLS ετικέτας φαίνεται στην Εικόνα 8.



Εικόνα 8 Η MPLS επικεφαλίδα

Η MPLS επικεφαλίδα περιλαμβάνει το πεδίο EXP (experimental), με μήκος 3 bits που χρησιμοποιείται για να καθοριστεί ο τύπος της μεταχείρισης. Ιδιαίτερη σημασία βέβαια πρέπει να δίνεται σε περιπτώσεις όπου πακέτα έχουν μαρκαρισμένο το DSCP πεδίο στο IP επίπεδο και εισέλθουν σε ένα MPLS domain όπου η μεταγωγή γίνεται με έλεγχο της MPLS επικεφαλίδας αγνοώντας την IP επικεφαλίδα. Στο σημείο αυτό σχεδόν όλες οι εταιρίες παροχής δικτυακού εξοπλισμού έχουν προβλέψει και όταν ένα IP πακέτο εισέρχεται σε ένα MPLS domain αντιγράφουν στο πεδίο EXP τα IP Precedence bits, δηλαδή τα 3 πιο σημαντικά bits του DSCP πεδίου. Βέβαια υπάρχει και η πρόβλεψη να μπορεί ο διαχειριστής του δικτύου να μαρκάρει το πεδίο αυτό με άλλη τιμή (άσχετη με την τιμή του IP Precedence). Τέλος πρέπει να σημειωθεί ότι επειδή το EXP πεδίο έχει μήκος 3 bits, μόνο 8 διαφορετικές κλάσεις μπορούν να υποστηριχτούν σε ένα MPLS δίκτυο.

2.3.2.2.3 Μηχανισμοί μαρκαρίσματος, μέτρησης της κίνησης, μορφοποίησης και απόρριψης πακέτων

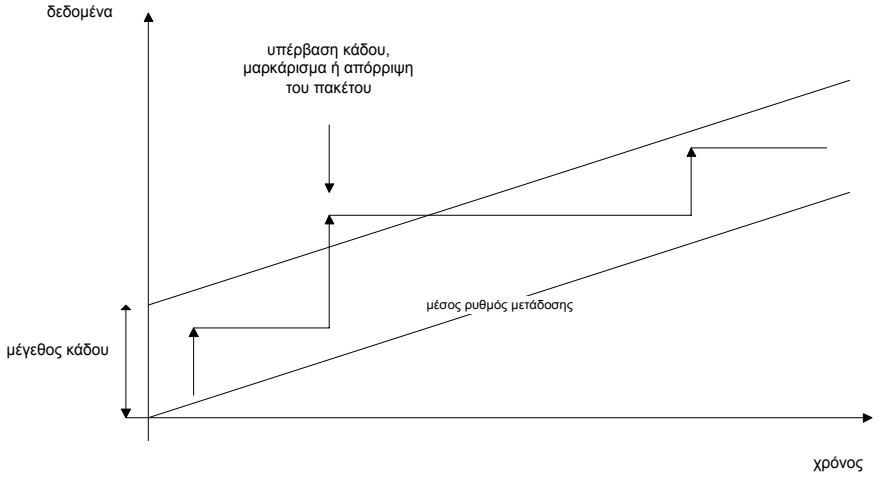
Γενικά οι μηχανισμοί μαρκαρίσματος των πακέτων, μέτρησης του προφίλ της κίνησης και μορφοποίησης ή απόρριψης της κίνησης ονομάζονται όλοι μαζί μηχανισμοί ελέγχου της κίνησης (traffic conditioning). Συνήθως οι μηχανισμοί αυτοί εφαρμόζονται στον αποστολέα, στα σημεία εισόδου της κίνησης σε κάποιο domain. Εντούτοις, έχει αναφερθεί πως μπορεί ο μηχανισμός μέτρησης να βρίσκεται στον παραλήπτη με ορισμένες βέβαια προϋποθέσεις. Για να είναι αυτό εφικτό απαιτείται από το δίκτυο να υποστηρίζει σε όλους τους δρομολογητές του την λειτουργικότητα του ECN (Explicit Congestion Notification) που είναι μια λειτουργία ελέγχου συμφόρησης. Στην πράξη το ECN είναι ένα bit στην επικεφαλίδα των πακέτων που τίθεται στην τιμή 1 όταν ανιχνεύσει στο δίκτυο συμφόρηση. Με τον τρόπο αυτό ενημερώνονται οι υπόλοιποι κόμβοι από τους οποίους περνά το συγκεκριμένο πακέτο πως σε κάποιο σημείο στο δίκτυο παρατηρήθηκε συμφόρηση. Αναλυτικότερα η λειτουργικότητα του ECN περιγράφεται σε παρακάτω ενότητα.

Οι μηχανισμοί ελέγχου της κίνησης που παρουσιάζουμε στις επόμενες παραγράφους της ενότητας υποθέτουν πως το μαρκάρισμα και η μέτρηση των πακέτων γίνεται στα σημεία εισόδου στο δίκτυο.

2.3.2.2.3.1 Αλγόριθμοι Token Bucket και Leaky Bucket

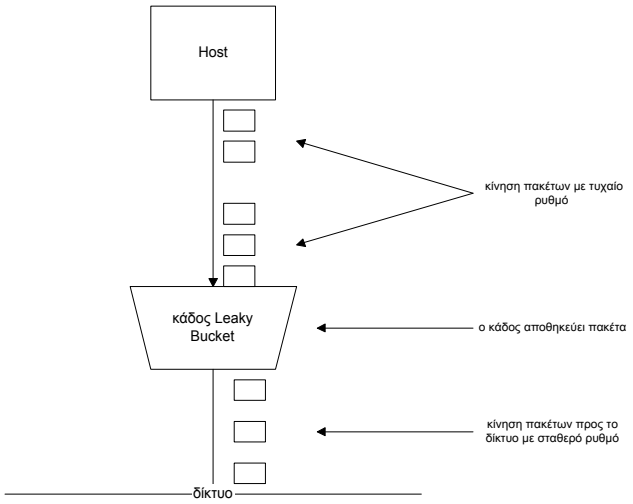
Ένας απλός μηχανισμός για τον έλεγχο της κίνησης, που διαχωρίζει τα πακέτα σε 2 κατηγορίες, σε αυτά που είναι εντός προφίλ και αντίστροφα σε όσα είναι εκτός προφίλ είναι η εφαρμογή κάποιου από τους αλγορίθμους token ή leaky bucket. Η λειτουργία τους βασίζεται στην ίδια λογική αλλά επιτυγχάνουν διαφορετικά αποτελέσματα όπως θα παρουσιαστούν αμέσως.

Ο αλγόριθμος token bucket καθορίζει 2 μεταβλητές, το μέσο ρυθμό αποστολής πακέτων r και το μέγιστο μέγεθος του κάδου b. Σε αυτόν παράγονται tokens με ρυθμό ίσο με το μέσο ρυθμό που καθορίστηκε, και αν αυτά δεν χρησιμοποιούνται, συσσωρεύονται στο κάδο μέχρι το πολύ b. Όταν φτάσει ένα πακέτο, αν υπάρχει ελεύθερο token, τότε θεωρείται ότι το token ανατίθεται στο πακέτο αυτό και το πακέτο χαρακτηρίζεται σαν εντός προφίλ. Αντίθετα αν φτάσει ένα πακέτο και δεν υπάρχει ελεύθερο token, τότε το πακέτο μαρκάρεται ως εκτός προφίλ έτσι ώστε αργότερα να δεχτεί ανάλογη μεταχείριση, όπως εξυπηρέτηση με ελάχιστη ποιότητα ή ακόμη και απόρριψη ανάλογα με το SLA που έχει υπογραφεί. Συμπερασματικά λοιπόν ο αλγόριθμος token bucket καθορίζει το μέσο ρυθμό μετάδοσης και επομένως επιτρέπει διακυμάνσεις του στιγμιαίου ρυθμού. Επίσης ο ρόλος του κάδου, που έχει μέγιστο μέγεθος b, είναι ιδιαίτερα σημαντικός αφού επιτρέπει να μαρκάρονται σαν κίνηση εντός προφίλ, εκρήξεις που δεν ξεπερνούν όμως την τιμή b.



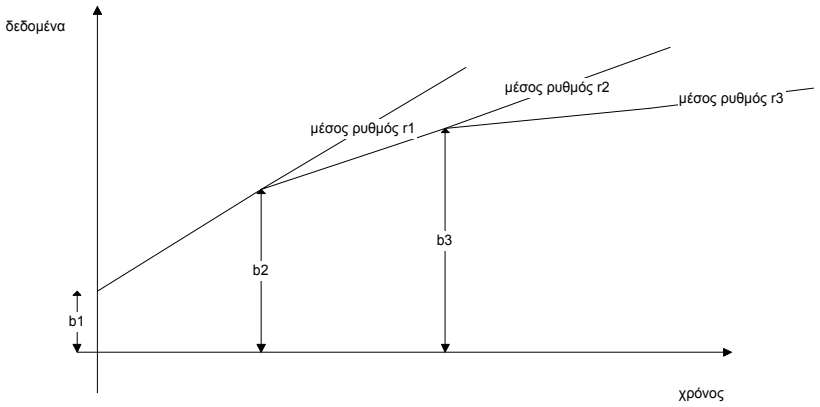
Εικόνα 9 Η λειτουργία του μηχανισμού token bucket

Παρόμοιος σε φιλοσοφία είναι και ο αλγόριθμος leaky bucket που καθορίζει αυστηρά το ρυθμό εξόδου των πακέτων από τον κάδο και είσοδό τους στο δίκτυο. Αν ο ρυθμός με τον οποίο καταφθάνουν τα πακέτα στον κάδο είναι μεγαλύτερος από τον ρυθμό με τον οποίο αυτά εξέρχονται από αυτόν, τότε συσσωρεύονται, μέχρι όμως μια τιμή που αποτελεί και το μέγιστο μέγεθος του κάδου. Ουσιαστικά λοιπόν ο αλγόριθμος αυτός οδηγεί τα πακέτα να εξέρχονται με σταθερό ρυθμό από τον κάδο και επιτρέπει αυτά να συσσωρεύονται στον κάδο εφόσον βέβαια υπάρχει διαθέσιμος χώρος. Τα πακέτα αυτά θεωρούνται ως νόμιμα (εντός προφίλ), ενώ αντίθετα όσα δεν εισέρχονται στον κάδο μαρκάρονται ως εκτός προφίλ και χειρίζονται σύμφωνα με ότι προβλέπει η συμφωνία που έχει υπογραφεί. Συμπερασματικά ο αλγόριθμος leaky bucket αποτελεί έναν πολύ καλό αλγόριθμο μορφοποίησης της κίνησης αφού σταθεροποιεί το ρυθμό μετάδοσης των πακέτων εξαλείφοντας εκρήξεις.



Εικόνα 10 Η λειτουργία του μηχανισμού leaky bucket

Επίσης προκειμένου να επιτευχθεί ο έλεγχος της κίνησης σε περισσότερα από 2 επίπεδα τότε μπορεί να χρησιμοποιηθεί κάποιος από τους παραπάνω αλγορίθμους διαδοχικά[6][7]. Έτσι τα πακέτα κατηγοριοποιούνται σε περισσότερα επίπεδα και μπορούν στη συνέχεια τα πακέτα κάθε επιπέδου να μαρκάρονται και να εξυπηρετούνται ξεχωριστά.



Εικόνα 11 μηχανισμός κατηγοριοποίησης της κίνησης σε 3 επίπεδα

2.3.2.2.4 Αστυνόμευση (policing) της κίνησης

Η λειτουργία αυτή πραγματοποιείται επίσης στα σημεία εισόδου της κίνησης σε ένα DiffServ domain. Η αστυνόμευση έχει την έννοια του ελέγχου της κίνησης με βάση ένα συγκεκριμένο προφίλ που έχει συμφωνηθεί και στη συνέχεια τη λήψη συγκεκριμένων αποφάσεων για τον χειρισμό της κίνησης που ξεφεύγει από το συμφωνηθέν προφίλ. Οι αποφάσεις αυτές μπορεί να είναι είτε μαρκάρισμα των πακέτων σε μικρότερη κλάση εξυπηρέτησης, να εξυπηρετηθούν χωρίς εγγυημένη ποιότητα ή τέλος στη χειρότερη περίπτωση να απορριφθούν. Στο σημείο αυτό πρέπει να σημειωθεί πως οι αποφάσεις αυτές επίσης έχουν συμφωνηθεί εκ των προτέρων μεταξύ του πελάτη και του διαχειριστή του δικτύου (SLA). Τα κριτήρια αστυνόμευσης που χρησιμοποιούνται μπορεί να είναι με βάση τη χρονική στιγμή στη

διάρκεια της μέρας, βάση της πηγής και του προορισμού ή γενικότερα με βάση κάθε δεδομένο της κίνησης.

Παράλληλα η λειτουργία της μορφοποίησης της κίνησης που περιγράφηκε προηγουμένως επιτυγχάνει να διαμορφώνει την κίνηση εξαλείφοντας εκρήξεις. Επίσης μπορεί να προβλεφθεί τα πακέτα που κανονικά απορρίπτονται (πακέτα εκτός προφίλ) να αποθηκεύονται προσωρινά και να διοχετεύονται αργότερα στο δίκτυο αφού πλέον έχει εξομαλυνθεί η εκρηκτικότητα της μετάδοσής τους. Επομένως είναι δυνατό οι μηχανισμοί αστυνόμευσης και μορφοποίησης της κίνησης να χρησιμοποιηθούν συνδυασμένα ώστε ένα μέρος των πακέτων που θεωρούνται εκτός προφίλ από τον μηχανισμό αστυνόμευσης να μορφοποιείται και να μεταδίδεται. Γενικά το πεδίο αυτό είναι ανοικτό και μπορούν να παρουσιαστούν διάφοροι μηχανισμοί που συνδυάζουν μηχανισμούς αστυνόμευσης και μορφοποίησης.

2.3.2.2.5 Διαχείριση ουρών (Queue management)

Το θέμα της διαχείρισης των ουρών αποτελεί ένα σημαντικό και κρίσιμο ζήτημα για το διαχειριστή του δικτύου προκειμένου να είναι σε θέση να προσφέρει ποιότητα υπηρεσίας στις διάφορες ροές όπως έχει συμφωνήσει. Επίσης η διαχείριση των ουρών είναι μια βασική προϋπόθεση για τη λειτουργία του μηχανισμού της χρονοδρομολόγησης που θα περιγράψουμε στην επόμενη ενότητα. Προκειμένου το δίκτυο να ικανοποιήσει όλες τις εγγυήσεις παροχής ποιότητας υπηρεσίας πρέπει να χειρίζεται τα πακέτα κάθε κλάσης ποιότητας σε ξεχωριστή ουρά ώστε να μπορεί να εφαρμόζει τον κατάλληλο μηχανισμό χρονοδρομολόγησης. Σε αντίθετη περίπτωση δεν είναι δυνατό ο μηχανισμός χρονοδρομολόγησης να διαχωρίσει τις διαφορετικές κλάσεις ποιότητας και να προσφέρει επομένως τις κατάλληλες εγγυήσεις στις αντίστοιχες ροές. Πιο αναλυτικά, αναφέροντας ένα παράδειγμα, αν δεν γίνει διαχωρισμός των κλάσεων ποιότητας σε διαφορετικές ουρές, θα συσσωρεύονται στην ίδια ουρά ροές με διαφορετικές απαιτήσεις με αποτέλεσμα είτε πακέτα να απορρίπτονται (αν γεμίσει η ουρά) είτε να παρουσιάζεται μεγάλη καθυστέρηση. Συνέπεια όλων αυτών είναι το δίκτυο να μην μπορεί να παρέχει τις καλύτερες εγγυήσεις και αντίστοιχα η απόδοση που επιτυγχάνουν οι εφαρμογές των πελατών να υποβαθμίζεται σημαντικά.

Οι κατεξοχήν λειτουργίες του διαχειριστή των ουρών παρουσιάζονται αμέσως παρακάτω και επιγραμματικά συνοψίζονται στην σωστή λειτουργία των ουρών και στη χρήση μηχανισμών για τον έλεγχο τους.

- Είσοδος ενός πακέτου στη σωστή ουρά με βάση τη κατηγοριοποίηση του πακέτου από τον αντίστοιχο μηχανισμό.
- Απόρριψη ενός πακέτου στην περίπτωση που η ουρά που πρέπει να εισαχθεί είναι γεμάτη.
- Απομάκρυνση ενός πακέτου από την κορυφή της ουράς όταν το ζητήσει ο χρονοδρομολογητής προκειμένου να μεταδοθεί στον επόμενο κόμβο.
- Έλεγχος της κατάστασης της ουράς, δηλαδή της μέσης πληρότητάς της και ανάληψη πρωτοβουλιών ανάλογα με αυτή την τιμή, με στόχο τη διατήρηση της μέσης πληρότητας σε χαμηλά επίπεδα. Οι πρωτοβουλίες που μπορεί να αναλάβει είναι οι ακόλουθες:
- Αφαίρεση ενός πακέτου από την ουρά και απόρριψή του στην περίπτωση που η ουρά έχει αρχίσει να γεμίζει.

- Μαρκάρισμα ενός πακέτου όταν η ουρά παρουσιάζει μεγάλη πληρότητα (ECN).

Γενικά λοιπόν παρατηρείται ότι εκτός από τις κλασικές λειτουργίες υποδοχής και αποχώρησης ενός πακέτου, ο διαχειριστής μιας ουράς ενδιαφέρεται και για την αποδοτική λειτουργίας της που εξασφαλίζεται κυρίως μέσα από τη διατήρηση σε χαμηλά επίπεδα της μέσης πληρότητάς της. Αυτό δικαιολογείται από το γεγονός ότι διατηρώντας χαμηλά τη μέση πληρότητα τότε οι ουρές μπορούν να απορροφούν εύκολα εκρήξεις της κίνησης. Αντίθετα, αν η μέση πληρότητα ήταν υψηλή τότε πλήθος πακέτων κατά τη διάρκεια εκρήξεων θα απορρίπτονταν. Επίσης η μικρή πληρότητα μιας ουράς συνεπάγεται πως η μέση καθυστέρηση εξυπηρέτησης θα παραμένει χαμηλή, γεγονός που είναι ιδιαίτερα επιθυμητό.

Το θέμα της διαχείρισης των ουρών γίνεται ακόμα επιτακτικότερο και πιο κρίσιμο ειδικά σε καταστάσεις συμφόρησης του δικτύου όπου πρέπει πλέον οι ουρές να αντιδράσουν σωστά και άμεσα. Το κυριότερο πρόβλημα είναι πως να προσδιοριστούν συγκεκριμένες στρατηγικές αποφάσεις για την ανάληψη δράσεων. Μια από αυτές τις αποφάσεις είναι η χρονική στιγμή απόρριψης πακέτων, δηλαδή αν απορρίπτονται πακέτα μόλις φτάσουν στην ουρά ή επιτρέπεται να απορρίπτονται πακέτα που βρίσκονται μέσα στην ουρά προκειμένου να εξυπηρετηθούν άλλα μεγαλύτερης προτεραιότητας. Επίσης κρίσιμη απόφαση είναι με βάση ποια κριτήρια και πληροφορίες απορρίπτονται τα πακέτα, αφού μπορεί να κρατούνται γενικές πληροφορίες για όλη την κίνηση ή αντίθετα για κάθε είδος κίνησης ξεχωριστά.

Συμπερασματικά, οι παραπάνω στρατηγικές αποφάσεις για τη λειτουργία της διαχείρισης ουρών επηρεάζουν άμεσα την απόδοση των ίδιων των ουρών και κατ' επέκταση όλου του δικτύου. Γενικό στόχο αποτελεί η δίκαια διαχείριση των ουρών για όλες τις κλάσεις ποιότητας χωρίς σε καμία περίπτωση να παραβούμε τις συμφωνίες που έχουν υπογραφεί με τους πελάτες του δικτύου.

Επιστρέφοντας στο θέμα της συμφόρησης πρέπει να τονιστεί πως η ύπαρξη συμφόρησης στο δίκτυο συνεπάγεται και αύξηση του μέσου μεγέθους των ουρών. Για την αποφυγή της συμφόρησης υπάρχουν συγκεκριμένοι μηχανισμοί από το πρωτόκολλο TCP στο επίπεδο μεταφοράς σύμφωνα με το OSI μοντέλο. Παράλληλα με αυτούς, οι διαχειριστές έχουν αναπτύξει και δικούς τους μηχανισμούς που θα παρουσιαστούν παρακάτω. Γενικά οι μηχανισμοί αυτοί προσπαθούν να βελτιώσουν το πρόβλημα της συμφόρησης που παρατηρείται στο δίκτυο και οφείλεται σε αποστολή πακέτων με ρυθμό υψηλότερο από αυτό που μπορεί να αντιμετωπίσει το δίκτυο και δεν οφείλεται σε μικροεκρήξεις παροδικού χαρακτήρα. Οι μηχανισμοί που μπορεί να χρησιμοποιήσει ο διαχειριστής του δικτύου είναι:

- Απόρριψη των πακέτων. Ο μηχανισμός αυτός έχει διπλό αποτέλεσμα καθώς αφενός μειώνει άμεσα το φόρτο του δικτύου και αφετέρου ενημερώνει άμεσα το πρωτόκολλο TCP για συμφόρηση. Αυτό επιτυγχάνεται αφού το TCP θεωρεί ότι κάθε απώλεια πακέτου οφείλεται σε συμφόρηση και στη συνέχεια ενεργοποιεί αυτόματα το μηχανισμό του για την αποφυγή συμφόρησης.
- Μαρκάρισμα των πακέτων. Η δεύτερη αυτή μέθοδος είναι λιγότερο καταστροφική από την πρώτη αφού δεν απορρίπτει πακέτα αλλά και λιγότερο άμεση αφού το δίκτυο δεν «αποφορτίζετε» άμεσα.

Στη συνέχεια περιγράφονται τεχνικές και μηχανισμοί που ανήκουν στη δεύτερη κατηγορία.

2.3.2.2.5.1 Explicit Congestion Notification (ECN)

Η μέθοδος αυτή στηρίζεται στα 2 αχρησιμοποίητα bits του πεδίου DSCP (DiffServ Code Point), τα οποία πλέον ονομάζονται ECN Capable Transport (ECT) και Congestion Experienced (CE) αντίστοιχα.[15] Αυτός ο μηχανισμός ελέγχεται από τα πρωτόκολλα του επιπέδου μεταφοράς και η λειτουργία του είναι απλή. Τα 2 αυτά bits επιτελούν συγκεκριμένες λειτουργίες:

- Το bit ECT τίθεται στην τιμή 1 αν τα άκρα μιας ροής που μεταδίδεται κατανοούν την λειτουργία του bit CE και κατ' επέκταση του όλου αυτού μηχανισμού.
- Το bit CE τίθεται στην τιμή 1 όταν κάποιος δρομολογητής επιθυμεί να ειδοποιήσει για συμφόρηση και το bit ECT είναι ενεργοποιημένο.

Συνεπώς σε κάθε πακέτο το ECT είναι 1 όταν και οι 2 άκρες της ροής κατανοούν τη λειτουργία του μηχανισμού. Κάθε δρομολογητής αν θέλει να ειδοποιήσει για συμφόρηση θέτει το CE στην τιμή 1 αν το ECT είναι ενεργοποιημένο αλλιώς απορρίπτει το πακέτο. Ουσιαστικά με τη μέθοδο αυτή ειδοποιούνται τα πρωτόκολλα με μη καταστροφικό τρόπο αν κατανοούν τη μέθοδο αυτή και σε αντίθετη περίπτωση (δεν καταλαβαίνουν τη μέθοδο) κατανοούν τη συμφόρηση από την απόρριψη του πακέτου. Ένα κρίσιμο σημείο στη μέθοδο αυτή είναι πότε ο δρομολογητής αποφασίζει να ειδοποιήσει για συμφόρηση, αφού σε περιπτώσεις παροδικής συμφόρησης λόγω μικροεκρήξεων της κίνησης, δεν είναι αποδοτικό να ειδοποιείται το πρωτόκολλο καθώς τότε θα υποβαθμιστεί η απόδοση του δικτύου χωρίς λόγο.

2.3.2.2.5.2 Μηχανισμός RED (Random Early Detection)

Ένας δεύτερος μηχανισμός που ειδοποιεί τα πρωτόκολλα για ενδεχόμενη συμφόρηση είναι ο RED. Στους μηχανισμούς αποφυγής συμφόρησης σημαντικό πρόβλημα αποτελεί ο καθορισμός πότε θα αποστέλλεται ειδοποίηση για συμφόρηση και πόσο έντονη αυτή θα είναι. Παράλληλα το θέμα αυτό σχετίζεται και με την διαμόρφωση των ουρών, πως έχουν δηλαδή οριστεί ώστε να είναι ξεχωριστές για κάθε ροή ή επιτρέπεται aggregates να περνούν από την ίδια ουρά.

Με το θέμα αυτό ασχολήθηκε για αρκετά χρόνια η IRTF (Internet Research Task Force) που κατέληξε να προτείνει τον μηχανισμό RED ο οποίος στέλνει ειδοποιήσεις για συμφόρηση τυχαία και η συχνότητα με την αυτές στέλνονται εξαρτάται από τη μέση πληρότητα της ουράς. Βασική παράμετρος με βάση την οποία αποφασίζει ο μηχανισμός αυτός είναι η μέση πληρότητα της ουράς. Ο τρόπος με τον οποίο ειδοποιεί τα πρωτόκολλα για συμφόρηση είναι έμμεσος καθώς αυτό γίνεται με απόρριψη πακέτων. Σε κάθε ουρά που εφαρμόζεται ο RED ορίζονται τρία μεγέθη:

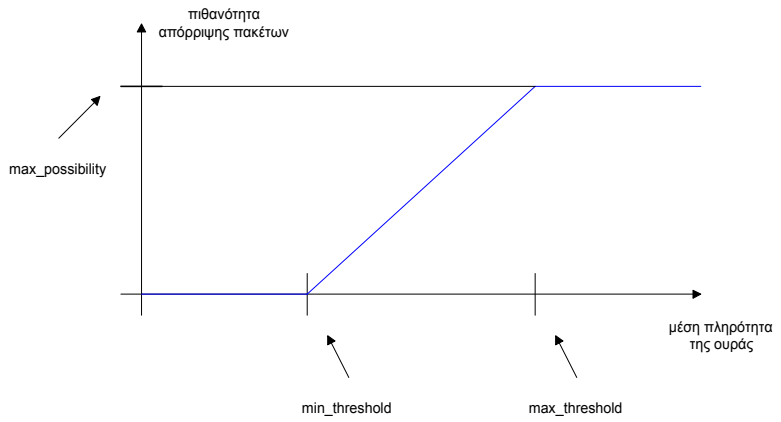
- Το min threshold
- Το max threshold
- Το max possibility

Έτσι ο μηχανισμός λειτουργεί ως εξής:

- Εάν η μέση πληρότητα είναι μικρότερη από την τιμή min_threshold τότε όλα τα πακέτα διέρχονται κανονικά και δεν έχουμε καμία απόρριψη.

- Εάν η μέση πληρότητα είναι μεγαλύτερη από την τιμή min_threshold και μικρότερη από την τιμή max_threshold τότε η πιθανότητα απόρριψης αυξάνει γραμμικά από 0 έως την τιμή max_possibility.
- Τέλος αν η μέση πληρότητα ξεπερνά την τιμή του max_threshold, τότε όλα τα πακέτα απορρίπτονται.

Οι τρεις αυτές καταστάσεις στις οποίες μπορεί να βρίσκεται μια ουρά ονομάζονται αντίστοιχα κανονική, αποφυγής συμφόρησης και ελέγχου συμφόρησης. Κρίσιμο παράγοντα για τη λειτουργία του αλγορίθμου αποτελεί η σωστή εκτίμηση της μέσης πληρότητας της ουράς. Αυτή υπολογίζεται κάθε φορά που ένα πακέτο εισέρχεται στην ουρά, και ο υπολογισμός της γίνεται με τη χρήση ενός κατωπερατού φίλτρου. Επίσης ο RED έχει προβλέψει και την περίπτωση όπου να μεσολαβήσει μεγάλο χρονικό διάστημα μεταξύ 2 απορρίψεων πακέτων και εν τω μεταξύ να έχει παρουσιαστεί συμφόρηση. Αυτό μπορεί να παρατηρηθεί εξαιτίας του γεγονότος ότι η απόρριψη πακέτων γίνεται πιθανοτικά. Λύση σε αυτό το θέμα δίνει ένας μετρητής που χρησιμοποιείται και μετρά τον αριθμό των πακέτων που πέρασαν από την ουρά χωρίς απόρριψη. Έτσι η πιθανότητα απόρριψης πολλαπλασιάζεται τώρα και με την ποσότητα 1/1-c όπου c ο μετρητής αυτός.



Εικόνα 12 Η λειτουργία του μηχανισμού RED

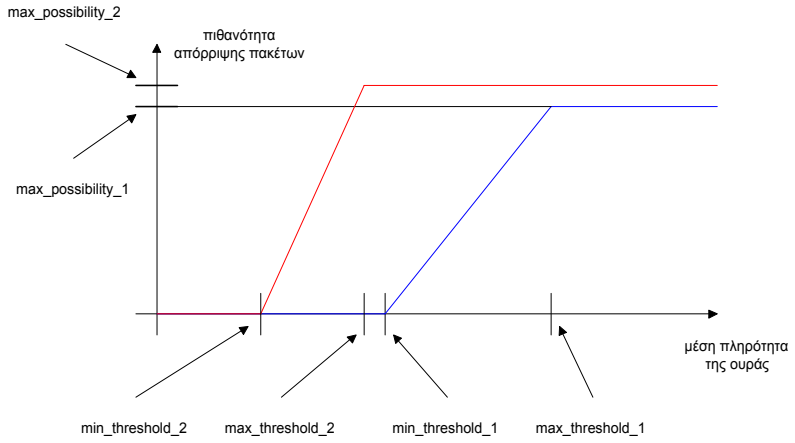
Γενικά ο μηχανισμός αυτός είναι ιδιαίτερα αποδοτικός, αλλά παρουσιάζει σημαντική δυσκολία στη ρύθμιση των παραμέτρων του. Το σημείο που πρέπει να τονιστεί είναι ότι απαιτείται να επιτρέπει να περνούν μικροεκρήξεις χωρίς απόρριψη πακέτων, ενώ αντίθετα θα πρέπει να αντιδρά άμεσα σε περιπτώσεις παρατεταμένης αύξησης της μέσης πληρότητας της ουράς. Τέλος προκειμένου να είναι πραγματικά αποδοτικός ο αλγόριθμος πρέπει πραγματικά να απορρίπτει πακέτα (άρα και να ειδοποιεί για συμφόρηση) από τις ροές που δημιουργούν το πρόβλημα.

Στη συνέχεια παρουσιάστηκαν διάφορες παραλλαγές του RED που προσπαθούσαν να εξαλείψουν ορισμένα μειονεκτήματα του. Τέτοιες ήταν ο Adaptive και ο Flow RED. Πάντως κυριότερη παραλλαγή του αποτελεί ο Weighted RED που περιγράφεται στην επόμενη ενότητα.

2.3.2.2.5.3 Μηχανισμός Weighted RED

Ο Weighted RED αποτελεί μια σημαντική παραλλαγή του κλασσικού RED μηχανισμού αφού επιτρέπει να συμπεριφέρεται η ουρά με διαφορετικό τρόπο

(εφαρμόζοντας διαφορετικά κριτήρια) στα πακέτα μιας ροής. Στην πραγματικότητα δίνει τη δυνατότητα να μεταχειρίζεται τα πακέτα της ίδιας ροής με διαφορετικό τρόπο μεταξύ τους ανάλογα βέβαια με κάποιο κριτήριο.



Εικόνα 13 Η λειτουργία του μηχανισμού Weighted RED

Ο μηχανισμός αυτός λειτουργεί όπως ο απλός RED με τη διαφορά ότι σε αυτόν ορίζονται περισσότερες τριάδες μεταβλητών (Min_threshold, max_threshold, max_possibility), ίσες σε αριθμό με τον αριθμό των διαφορετικών τρόπων χειρισμού πακέτων που ζητείται. Ένα παράδειγμα αποτελεί να οριστούν 2 επίπεδα τιμών, με το δεύτερο αυστηρότερο από το πρώτο και να διαχειρίζονται με βάση το πρώτο τα πακέτα που καταφθάνουν κανονικά στην ουρά, ενώ αντίθετα με το δεύτερο επίπεδο τιμών τα πακέτα που είχαν μαρκαριστεί εκτός προφίλ σε προηγούμενο δρομολογητή.

Ολοκληρώνοντας την παρουσίαση του Weighted RED να επισημανθεί ότι είναι ιδιαίτερα χρήσιμος μηχανισμός που έχει ευρεία χρήση σήμερα.

2.3.2.2.6 Χρονοδρομολόγηση

Το επόμενο σημαντικό ζήτημα στη προσπάθεια ενός δικτύου να παρέχει εγγυήσεις ποιότητας είναι το θέμα της χρονοδρομολόγησης. Αναλύοντας την έννοια αυτή, χρονοδρομολόγηση σημαίνει ο τρόπος με τον οποίο χειρίζεται το δίκτυο τις ουρές, δηλαδή ποια ουρά στέλνει δεδομένα και για πόσο χρόνο. Ουσιαστικά ο μηχανισμός αυτός έχει στην διάθεσή του το σύνολο των ουρών που έχει ένας δρομολογητής και αποφασίζει με ποια σειρά θα μεταδώσουν πακέτα και για πόσο διάστημα η καθεμία.

Ο ρόλος του χρονοδρομολογητή είναι ιδιαίτερα κρίσιμος για ένα δίκτυο όταν επιθυμεί να προσφέρει και να παρέχει εγγυήσεις ποιότητας. Αυτό δικαιολογείται από το γεγονός ότι η λειτουργία του δρομολογητή καθορίζει την καθυστέρηση σε κάθε ουρά και τον τρόπο που διαμοιράζεται η γραμμή μετάδοσης μεταξύ των ουρών. Στην πραγματικότητα ο μηχανισμός του χρονοδρομολογητή είναι αυτός που καθορίζει το είδος της ποιότητας που παρέχει το δίκτυο. Οι παράμετροι αναλυτικά που μπορεί και επηρεάζει είναι:

- Η χωρητικότητα κάθε ροής, αφού μπορεί και ελέγχει κάθε πότε η ροή αυτή θα μεταδίδει.

- Την καθυστέρηση κάθε ροής, αφού ελέγχει όπως αναφέρθηκε παραπάνω το ρυθμό με τον οποίο κάθε ροή μεταδίδει, άρα καθορίζει και το χρονικό διάστημα που τα πακέτα παραμένουν στην ουρά.
- Επίσης ο χρονοδρομολογητής καθορίζει και το jitter, που είναι η διαφορά στην καθυστέρηση μεταξύ 2 διαδοχικών πακέτων.

Γενικά λοιπόν αυτοί είναι οι κύριοι παράμετροι που μπορεί να επηρεάσει ο χρονοδρομολογητής και συνεπώς προδιαγράφει την ποιότητα υπηρεσίας που μπορεί να παρέχει το δίκτυο. Επομένως εξαιτίας της σπουδαιότητας του μηχανισμού αυτού και του γεγονότος ότι οι μηχανισμοί χρονοδρομολόγησης που υπάρχουν είναι αρκετοί, είναι αναγκαίο η επιλογή του καταλληλότερου να γίνεται προσεκτικά και με βάση ορισμένα κριτήρια. Είναι απαραίτητο στην επιλογή του μηχανισμού χρονοδρομολόγησης να ελέγχεται πρώτα το είδος των εγγυήσεων που παρέχει και ο βαθμός επιτυχίας του, έτσι ώστε να ταιριάζει απόλυτα στη φύση των εφαρμογών που θα υποστηρίξει.

Στη συνέχεια της ενότητας παρουσιάζονται ορισμένοι τέτοιοι μηχανισμοί, ο τρόπος λειτουργίας τους, το είδος των εγγυήσεων που παρέχουν και τα μειονεκτήματά τους.

2.3.2.2.6.1 FIFO

Ο πρώτος μηχανισμός χρονοδρομολόγησης είναι ο λεγόμενος FIFO[16] και είναι ο παλαιότερος που υπάρχει. Ο μηχανισμός αυτός υποθέτει ότι υπάρχει μόνο 1 ουρά και η λογική του είναι ότι εξέρχεται από την ουρά το πρώτο πακέτο που μπήκε, δηλαδή κάθε φορά το παλαιότερο πακέτο μέσα στην ουρά. Ο μηχανισμός αυτός όπως γίνεται σαφές αντιμετωπίζει όλα τα πακέτα όμοια και δεν χρησιμοποιεί καμία έννοια προτεραιότητας.

Ο μηχανισμός αυτός έχει ως πλεονέκτημά του μόνο την απλότητα του και μπορεί να βρει εφαρμογή σε περιπτώσεις γραμμών μετάδοσης πολύ μεγάλης ταχύτητας όπου δεν υπάρχει καθόλου συμφόρηση, και ειδικότερα όταν η χρησιμοποίηση της γραμμής είναι πολύ χαμηλή. Αντιθέτως σε περιπτώσεις συμφόρησης έχει πολύ κακή απόδοση. Επίσης αντίστοιχα κακή απόδοση παρουσιάζει και στις περιπτώσεις όπου υπάρχουν εφαρμογές με καταιγισμούς που μπορεί να καταλαμβάνουν όλη την ουρά και να απορρίπτονται πακέτα άλλων εφαρμογών. Γενικά ο μηχανισμός αυτός είναι απλός και δεν ενδείκνυται για δρομολογητές που πρέπει να παρέχουν εγγυήσεις ποιότητας.

2.3.2.2.6.2 Priority Queueing (PQ)

Ένας δεύτερος μηχανισμός χρονοδρομολόγησης είναι ο Priority Queueing[1] ο οποίος σε αντίθεση με το μηχανισμό FIFO επιτρέπει διαφορετικές προτεραιότητες και μπορεί να χειριστεί πλήθος ουρών. Η λογική του είναι ότι μια ουρά έχει αυστηρή προτεραιότητα και πάντοτε εξυπηρετείται όταν έχει πακέτα σε βάρος των υπολοίπων ουρών. Ουσιαστικά λοιπόν ο μηχανισμός αυτός συμπεριφέρεται προνομιακά στην ουρά υψηλής προτεραιότητας αφού πάντοτε εκείνη μεταδίδει άμεσα και υποβαθμίζει τις υπόλοιπες.

Στην πράξη, τα πακέτα ανάλογα με την ταξινόμηση που έχουν δεχτεί εισέρχονται στην αντίστοιχη ουρά. Τότε ο PQ ελέγχει πάντοτε τις ουρές με τη σειρά, αρχίζοντας από την ουρά μεγαλύτερης προτεραιότητας και προχωρώντας προς την ουρά μικρότερης προτεραιότητας έως ότου βρει μια ουρά που έχει πακέτο και το μεταδίδει. Στη συνέχεια επαναλαμβάνει ξανά την ίδια διαδικασία.

Ο μηχανισμός αυτός εισάγει στο δίκτυο ένα είδος αδικίας καθώς ανάλογα με το ρυθμό που καταφτάνουν πακέτα στην ουρά υψηλής προτεραιότητας, μπορεί οι άλλες ουρές είτε να εξυπηρετούνται ελάχιστα είτε ακόμη και καθόλου. Το τελευταίο μπορεί να συμβεί αν ο ρυθμός με τον οποίο εισέρχονται πακέτα στην ουρά υψηλής προτεραιότητας ισούται με το ρυθμό μετάδοσης πάνω στο link. Αυτό μπορεί να διορθωθεί είτε με αστυνόμευση, είτε εφαρμόζοντας μορφοποίηση στην κίνηση υψηλής προτεραιότητας σε κάποιο προηγούμενο σημείο της διαδρομής.

Επικεντρώνοντας τώρα στα θετικά σημεία αυτού του μηχανισμού πρέπει να αναφερθεί ότι μπορεί και προσφέρει πολύ χαμηλή καθυστέρηση στα πακέτα που ανήκουν στην ουρά υψηλής προτεραιότητας. Σε αυτή την περίπτωση αν η ουρά αυτή δεν έχει πακέτα, μόλις θα φτάσει ένα, τότε θα περιμένει μέχρι να μεταδοθεί το πακέτο που μεταδίδεται εκείνη τη στιγμή. Στη περίπτωση που η ουρά έχει πακέτα θα περιμένει μέχρι να σταλούν τα προηγούμενα, κάτι που εξαρτάται από το μέγεθος αυτών και το bandwidth του link.

2.3.2.2.6.3 Modified Deficit Round Robin (M-DRR)

Στη συνέχεια ένας άλλος μηχανισμός χρονοδρομολόγησης, με πολλές επιλογές χρήσης και ιδιαίτερα αποδοτικός είναι ο M-DRR[17]. Αυτός στηρίζεται στη λογική του Deficit Round Robin (DRR) και του Round Robin (RR), όπου κρίνεται σκόπιμο να περιγραφούν εν συντομία. Ο Round Robin αλγόριθμος χειρίζεται όλες τις ουρές όμοια και τις ελέγχει κυκλικά. Σε όποια βρει πακέτο στην αναμονή το μεταδίδει και συνεχίζει τον κυκλικό έλεγχο. Η μέθοδος συνεπώς αυτή έχει το μειονέκτημα ότι δεν μπορεί να παρέχει εγγυήσεις για την καθυστέρηση. Η μέθοδος DRR αποτελεί μια παραλλαγή της απλής μεθόδου Round Robin, όπου τώρα οι ουρές ελέγχονται ξανά κυκλικά αλλά προσπαθούν να διατηρούν σταθερό το μέσο ρυθμό μετάδοσης. Αυτό επιτυγχάνεται με την ακόλουθη τεχνική, σε κάθε ουρά ορίζονται 2 ποσότητες, το κβάντο Q και το έλλειμμα D. Το κβάντο είναι ο μέγιστος αριθμός bytes που μπορεί να μεταδώσει η ουρά κάθε φορά. Αν μεταδώσει λιγότερα τότε η διαφορά αποθηκεύεται στο έλλειμμα και προστίθεται στην μέγιστη τιμή bytes που θα μεταδώσει την αμέσως επόμενη φορά.

Ο μηχανισμός M-DRR λειτουργεί ουσιαστικά όπως ο DRR με τη διαφορά ότι εισάγει και μια ουρά προτεραιότητας ώστε να επιτυγχάνει χαμηλή καθυστέρηση. Οι υπόλοιπες ουρές εξυπηρετούνται με τη σειρά σύμφωνα με τον μηχανισμό DRR και η ουρά προτεραιότητας είτε εξυπηρετείται εναλλάξ με τις άλλες είτε κατά απόλυτη προτεραιότητα. Υπάρχουν 2 παραλλαγές του M-DRR όπου το σημείο που διαφέρουν είναι πόσο συχνά εξυπηρετείται η ουρά προτεραιότητας. Αναλυτικότερα αυτές είναι:

- Alternate Priority. Στη μέθοδο αυτή η ουρά προτεραιότητας εξυπηρετείται εκ περιτροπής με τις υπόλοιπες ουρές, οι οποίες εξυπηρετούνται με τη σειρά. Για παράδειγμα, εξυπηρετείται η ουρά προτεραιότητας, ύστερα η πρώτη από τις άλλες, μετά πάλι η ουρά προτεραιότητας, ύστερα η δεύτερη κοκ.
- Strict Priority. Αντίθετα στη μέθοδο αυτή η ουρά προτεραιότητας εξυπηρετείται κατά απόλυτη προτεραιότητα για όσο διάστημα έχει πακέτα προς μετάδοση. Όταν δεν έχει, εξυπηρετούνται οι υπόλοιπες σύμφωνα με τον DRR μηχανισμό.

Γενικά λοιπόν η μέθοδος Modified-Deficit Round Robin είναι ευέλικτη και αποδοτική μόνο όμως σε περιπτώσεις όπου δεν υπάρχει μεγάλη συμφόρηση. Γι' αυτό ακριβώς το λόγο προτείνεται να χρησιμοποιείται παράλληλα με ένα μηχανισμό

διαχείρισης ουρών που προλαμβάνει τη συμφόρηση, όπως αυτοί που περιγράφηκαν στην προηγούμενη ενότητα.

2.3.2.2.6.4 Fair Queueing – Weighted Fair Queueing

Η μέθοδος Fair Queueing[1][17] θεωρείται ως μια καλή παραλλαγή της μεθόδου Round Robin, όπου πλέον ο στόχος είναι να εξυπηρετεί όλες τις ουρές με τέτοιο τρόπο ώστε να ικανοποιείται μακροπρόθεσμα ο στόχος για ίσο μέσο εύρος ζώνης. Το ιδανικό για να ικανοποιηθεί ο στόχος είναι αν γινόταν η διαμοίραση σε επίπεδο bit (θα έστελνε η κάθε ουρά για σταθερό χρόνο 1/N, ιδανική διαμοίραση χρόνου), οπότε μπορούσε να μετέδιδε από συγκεκριμένο αριθμό bits από κάθε ουρά, επιτυγχάνοντας πλέον ίσο μέσο εύρος ζώνης. Όμως επειδή η χρονοδρομολόγηση γίνεται σε επίπεδο πακέτου, το αποτέλεσμα είναι πλέον προσεγγιστικό και μακροπρόθεσμο. Ουσιαστικά, η μέθοδος αυτή χρονοδρομολογεί τις μεταδόσεις πακέτων από τις ουρές με την σειρά με την οποία θα είχαν καταφτάσει στο άλλο άκρο της γραμμής μετάδοσης αν είχε χρησιμοποιηθεί ο ιδανικός χρονοδρομολογητής διαμοίρασης χρόνου. Η μέθοδος Fair Queueing έχει 2 συγκεκριμένα μειονεκτήματα:

- Για να επιτύχει το στόχο της αυτή η μέθοδος χρονοδρομολόγησης πρέπει να σημαδεύει τα πακέτα και να υπολογίζει τον αναμενόμενο χρόνο αναχώρησής τους. Όμως, οι υπολογισμοί αυτοί είναι αρκετοί και χρονοβόροι με αποτέλεσμα τελικά να γίνονται προσεγγιστικά, γεγονός που οδηγεί σε βραχυπρόθεσμες αδικίες μερικές φορές.
- Η μέθοδος Fair Queueing είναι σχεδιασμένη να εφαρμόζεται στη περίπτωση ξεχωριστών ουρών για κάθε ροή προκειμένου να επιτυγχάνει να μην επηρεάζεται κάθε ροή από την ύπαρξη της άλλης. Η λογική αυτή όμως δεν μπορεί να λειτουργήσει με το μοντέλο της συγκέντρωσης ροών (aggregate classes)

Μια σημαντική παραλλαγή της Fair Queueing μεθόδου αποτελεί η μέθοδος Weighted Fair Queueing. Στόχος της μεθόδου αυτής είναι να αποδίδει βάρη σε κάθε ουρά επιτρέποντας να διαφοροποιείται το μέσο εύρος ζώνης που κάθε μια αντιλαμβάνεται. Αν για παράδειγμα υπάρχουν N ουρές με πακέτα προς μετάδοση τότε κάθε ουρά M εξυπηρετείται έτσι ώστε να λαμβάνει ποσοστό W_M από το συνολικό ρυθμό της γραμμής. Σε περίπτωση που κάποιες από τις ουρές είναι άδειες τότε το επιπλέον εύρος ζώνης (που θα έπαιρναν οι ουρές αυτές) μοιράζεται αναλογικά στις υπόλοιπες ουρές με βάση πάντοτε τα βάρη τους.

2.3.2.3 Υπηρεσίες DiffServ

2.3.2.3.1 EF-based υπηρεσίες

Οι υπηρεσίες EF-based[18] έχουν σαν στόχο να προσομοιώσουν τις εικονικές μισθωμένες γραμμές για την εξυπηρέτηση κίνησης που προέρχεται από συνένωση ροών, όπως προκύπτει από την εφαρμογή των μηχανισμών ταξινόμησης και μαρκαρίσματος. Οι υπηρεσίες αυτές απευθύνονται σε εφαρμογές πραγματικού χρόνου ευαίσθητες στο jitter και στην απώλεια πακέτων που απαιτούν εγγυήσεις χωρητικότητας. Ο ορισμός τους βασίζεται στα ακόλουθα σημεία:

- Στον καθορισμό της διεπαφής μεταξύ DiffServ domains έτσι ώστε και αυτό να εμφανίζει συμπεριφορά EF-based.

- Στον περιορισμό της χωρητικότητας που αφιερώνεται στην υπηρεσία για να αποφευχθεί αποκλεισμός της best effort κίνησης
- Σε μια αρχική στατική ρύθμιση των δικτυακών συσκευών για την παροχή της υπηρεσίας.
- Στην ελαχιστοποίηση των μηχανισμών που χρησιμοποιούνται ανά κόμβο.

Τα παραπάνω σημεία αποτελούν ένα συνοπτικό ορισμό των EF-based υπηρεσιών. Μια από αυτές είναι η IP Premium[19][20][21][22] και παρέχεται σε συνενώσεις IP κίνησης. Οι μηχανισμοί που απαιτούνται για την υλοποίηση της είναι αρχικά ένας μηχανισμός αποδοχής κλήσης, όπου συνήθως η αποδοχή κλήσης γίνεται με βάση τα SLAs. Επίσης απαιτείται ένας μηχανισμός ταξινόμησης σε συνενώσεις όπου γίνεται με βάση τις διευθύνσεις πηγής και προορισμού. Σύμφωνα με τον ορισμό της IP Premium υπηρεσίας αν τα πακέτα περιέχουν διευθύνσεις που εξυπηρετούνται από την υπηρεσία αυτή εντάσσονται στο aggregate της κίνησης στο interface εισόδου του router. Στην συνέχεια και αφού απομακρυνόμαστε από την πηγή η ταξινόμηση γίνεται αποκλειστικά με βάση το DSCP στην επικεφαλίδα του πακέτου. Η τιμή για το DSCP που συνιστάται για την υπηρεσία IP Premium είναι η τιμή 101110. Τέλος αν εμφανίζονται κοντά στην πηγή τους πακέτα να περιέχουν το IP premium DSCP αλλά μη συμβατό ζεύγος πηγής προορισμού απορρίπτονται σαν μη νόμιμα.

Η συνένωση IP Premium ροής που μπαίνει σε ένα DiffServ domain αρχικά υπόκειται σε μηχανισμό αστυνόμευσης. Αν η ροή βρίσκεται στο domain του χρήστη, τότε η αστυνόμευση γίνεται με βάση ένα token bucket και τις παραμέτρους του SLA μεταξύ του χρήστη και του δικτύου του πρόσβασης. Επίσης σε αυτό το σημείο ο χρήστης μπορεί να μορφοποιήσει και την κίνηση του αφού η υπηρεσία αυτή δεν επιτρέπει μορφοποίηση σε άλλο σημείο του δικτύου. Η διαδικασία της μορφοποίησης θεωρείται σημαντική καθώς οι buffers στις δικτυακές συσκευές κατά μήκος του μονοπατιού που ακολουθείται είναι περιορισμένοι με αποτέλεσμα αν δεν έχει μορφοποιηθεί επαρκώς η κίνηση, τότε αυτοί να γεμίζουν και πακέτα να απορρίπτονται. Μια πρόταση που έχει παρουσιαστεί είναι να γίνεται μορφοποίηση από την ίδια την πηγή.

Επίσης, στον ορισμό της IP Premium υπηρεσίας αναφέρεται ότι ο μηχανισμός αστυνόμευσης πρέπει να στηρίζεται σε ένα token bucket με βάθος μερικά πακέτα και χωρητικότητα ελαφρά μεγαλύτερη από αυτή που εγγυάται η υπηρεσία. Αυτό έχει σαν αποτέλεσμα να μπορούν να απορροφώνται και κάποιες μικρές αλλοιώσεις στα χαρακτηριστικά της κίνησης. Επειδή αποτελεί στόχο η ελαχιστοποίηση των μηχανισμών που χρησιμοποιούνται ανά κόμβο έχει προταθεί η αστυνόμευση να γίνεται μόνο στους κόμβους εισόδου σε DiffServ domains. Επίσης αυτό μπορεί να αποφευχθεί αν η κίνηση προέρχεται από κάποιο άλλο «έμπιστο» domain.

Για την υλοποίηση της IP Premium υπηρεσίας είναι απαραίτητο να χρησιμοποιείται αυστηρή κατά προτεραιότητα χρονοδρομολόγηση προκειμένου να επιτυγχάνεται η ελάχιστη δυνατή καθυστέρηση των πακέτων των IP συνενώσεων. Επίσης η υπηρεσία αυτή δεν απαιτεί μηχανισμούς διαχείρισης ουρών αφού προφανώς έχουν πολύ μικρό μήκος. Τέλος απαραίτητοι θεωρούνται οι μηχανισμοί διάδοσης των κανόνων εφαρμογής της υπηρεσίας και οι μηχανισμοί παρακολούθησης και καταγραφής της λειτουργίας της υπηρεσίας.

Μια δεύτερη υπηρεσία EF-based είναι η λεγόμενη QBone Premium η οποία ορίστηκε από τη ομάδα QBone του Internet2. Η υπηρεσία αυτή έχει σαν στόχο την παροχή εγγυήσεων χωρητικότητας χωρίς απώλειες πακέτων, με την ελάχιστη δυνατή

καθυστέρηση και jitter σε κίνηση με περιορισμένο μέγιστο ρυθμό που διατρέχει πολλαπλά domains. Σε αυτή την υπηρεσία κρίσιμος παράγοντας αποτελεί επίσης η χρονοδρομολόγηση των πακέτων όπου πρέπει να είναι αυστηρής προτεραιότητας. Η πηγή καθορίζει 2 παραμέτρους, το μέγιστο ρυθμό και το μέγιστο μέγεθος έκρηξης και όταν η κίνηση ξεπερνά αυτό το προφίλ τότε τα πακέτα απορρίπτονται στο σημείο εισόδου που διαπιστώνεται η παράβαση.

Επικεντρώνοντας το ενδιαφέρον στις διαφορές της QBone Premium υπηρεσίας σε σχέση με την IP Premium διαπιστώνεται ότι η κυριότερη διαφορά τους είναι το γεγονός ότι η QBone Premium απαιτεί οι συνοριακοί κόμβοι ενός domain να υποστηρίζουν την μορφοποίηση της συνένωσης ροών που εξέρχονται από αυτούς.

Γενικά όλες οι υπηρεσίες που βασίζονται στο μοντέλο DiffServ και επομένως και η IP Premium παρουσιάζουν μια σημαντική δυσκολία που συνίσταται στη μελέτη και ανάλυση της ποιότητας υπηρεσίας που παρέχουν. Όπως είναι γνωστό αφορούν συνενώσεις ροών που η μορφή και τα χαρακτηριστικά τους δεν μπορούν να προσεγγιστούν εύκολα, ενώ επίσης δέχονται και αλλοίωση από τα IP δίκτυα. Συνεπώς για να υλοποιηθεί μια τέτοια υπηρεσία πρέπει εκτός από τον καθορισμό των παραμέτρων που αναφέρθηκαν παραπάνω, να ακολουθηθεί και μια πειραματική διαδικασία σε συνθήκες πραγματικής λειτουργίας προκειμένου να ρυθμιστούν όλες οι παράμετροι του δικτύου και να προκύψει το αναμενόμενο αποτέλεσμα.

Συνολικά, έχουν παρουσιαστεί διάφορες έρευνες για την απόδοση των παραπάνω υπηρεσιών. Τα σημεία τα οποία εξετάζονται είναι η επίδραση του μεγέθους της ουράς στην οποία εισέρχονται τα EF πακέτα, του αλγόριθμου δρομολόγησης και του μεγέθους των πακέτων. Παράλληλα ελέγχεται η επίδραση της ύπαρξης ή όχι best–effort κίνησης στην εγγυημένη, από την EF-based υπηρεσία, ποιότητα. Από τη διερεύνηση προέκυψε ότι όσο αφορά τη χρονοδρομολόγηση, η χρήση ουράς προτεραιότητας δίνει για κάθε μέγεθος πακέτου την ελάχιστη καθυστέρηση. Επίσης οι αλγόριθμοι Priority Queueing (PQ) και Weighted Fair Queueing (WFQ) παρουσιάζουν παρόμοιες μέσες τιμές για το jitter ενώ ο PQ ενδεχομένως να ελαχιστοποιεί το jitter σε σχέση με τον WFQ μόνο στην περίπτωση που ο WFQ έχει αριθμό ουρών μεγαλύτερο από 2. Τα συμπεράσματα αυτά επαληθεύτηκαν και στην περίπτωση όπου υπήρχε και best-effort κίνηση. Ορισμένες επιπλέον παρατηρήσεις είναι ότι η μέση καθυστέρηση φαίνεται να αυξάνεται γραμμικά με την αύξηση του μεγέθους του πακέτου, ενώ η αύξηση παρουσιάζεται περισσότερο απότομη με την απουσία best –effort κίνησης. Στη συνέχεια επικεντρώνοντας στο jitter παρουσιάζεται κατά την απουσία best effort κίνησης να αυξάνεται γραμμικά ενώ με την παρουσία της παρουσιάζει μεταβολές χωρίς να ακολουθεί συγκεκριμένη μορφή.

Τέλος, έχουν πραγματοποιηθεί μελέτες με μια σειρά πειραμάτων για την απόδοση που αντιλαμβάνεται η EF κίνηση κατά μήκος ενός DiffServ domain όπου υπάρχουν πολλά σημεία συγκέντρωσης ροών και επομένως πιθανά σημεία συμφόρησης. Τα βασικά συμπεράσματα είναι ότι η καθυστέρηση από άκρο σε άκρο είναι ανάλογη του αριθμού των σημείων που παρουσιάζουν συμφόρηση στη διαδρομή των πακέτων από την πηγή στον προορισμό. Παράλληλα οι καθυστερήσεις των EF πακέτων οφείλονται κυρίως στην καθυστέρηση στις ουρές. Επίσης η συνένωση πολλών ροών οδηγεί στην αύξηση της καθυστέρησης και του jitter και ιδίως στη δεύτερη περίπτωση η επίδραση του βαθμού συνένωσης είναι αντιστρόφως ανάλογη του μεγέθους των EF πακέτων. Παράλληλα jitter εμφανίζεται ακόμη και στις περιπτώσεις όπου δεν έχουμε best effort κίνηση αλλά υπάρχουν συνενώσεις EF κίνησης. Το γεγονός αυτό μάλλον οφείλεται

στη συγκέντρωση πακέτων στις ουρές της EF κίνησης καθώς απομακρυνόμαστε από την είσοδο του DiffServ domain.

Ανακεφαλαιώνοντας λοιπόν, στην υλοποίηση EF-based υπηρεσιών πρέπει να δοθεί ιδιαίτερη προσοχή για την εξάλειψη των παρενεργειών στην απόδοση εξαιτίας της δημιουργίας συνενώσεων καθώς η EF κίνηση διατρέχει ένα DiffServ enabled domain. Για να επιτευχθεί αυτό πρέπει να διατηρείται το φορτίο σε χαμηλά επίπεδα προκειμένου να μπορεί το δίκτυο να απορροφά τις εκρήξεις χωρίς επιβάρυνση της απόδοσης. Συνεπώς η σωστή υλοποίηση μιας EF-based υπηρεσίας πρέπει να επιτυγχάνει 2 πράγματα:

- Να έχουν ρυθμιστεί οι buffers να έχουν τόσες θέσεις ώστε να απορροφούν τις εκρήξεις και συνάμα
- Το μέγεθος των buffers να μην προκαλεί αύξηση της καθυστέρησης από άκρο σε άκρο λόγω παρατεταμένης παραμονής σε αυτούς πακέτων.

Επίσης σημαντικό παράγοντα αποτελεί και ο αλγόριθμος χρονοδρομολόγησης που θα επιλεγεί, όπου φαίνεται να υπερτερεί ο αλγόριθμος PQ έναντι του WFQ με βάση τις παραμέτρους ποιότητας. Όμως παρατηρήθηκε ότι σε μετρήσεις σε σχέση με την διασύνδεση πολλών κόμβων τους οποίους διατρέχουν οι συνενώσεις EF ροών, ο WFQ επιτυγχάνει μικρότερες εκρήξεις για τις διάφορες συνενώσεις. Μια πρόταση που έχει διατυπωθεί είναι η χρησιμοποίηση priority queueing σε συνδυασμό με μορφοποίηση της κίνησης στα κατάλληλα σημεία του δικτύου.

2.3.2.3.2 AF based Υπηρεσίες

Οι υπηρεσίες αυτές βασίζονται στην Assured Forwarding PHB[23] και στόχο τους αποτελεί να εγγυηθούν την εξυπηρέτηση μιας ροής που εναρμονίζεται με το συμφωνηθέν προφίλ με πολύ μεγάλη πιθανότητα και αντίστοιχα κίνηση που βρίσκεται εκτός προφίλ με μικρότερη πιθανότητα. Γενικά οι υπηρεσίες αυτές βρίσκονται σε πρώιμο στάδιο και αποτελεί θέμα διερεύνησης ο ακριβής καθορισμός τους και η λεπτομερειακή μελέτη της απόδοσης που μπορεί να επιτύχουν. Οι γενικοί κανόνες του AF PHB που χρησιμοποιούνται προκειμένου να οριστούν οι υπηρεσίες αυτές είναι οι ακόλουθοι:

- Δεν πρέπει να γίνεται συνένωση των ροών 2 διαφορετικών AF κλάσεων.
- Σε κάθε κλάση πρέπει να παρέχεται συγκεκριμένη ποσότητα πόρων και η κλάση πρέπει να επιτυγχάνει το ρυθμό της τόσο βραχυπρόθεσμα όσο και μακροπρόθεσμα.
- Η πιθανότητα εξυπηρέτησης ενός πακέτου πρέπει να είναι αντιστρόφως ανάλογη της προτεραιότητας εξυπηρέτησής του.
- Κάθε κόμβος πρέπει να δέχεται πακέτα και των όλων των επιπέδων προτεραιότητας και πρέπει να τα εξυπηρετεί με τουλάχιστον 2 διαφορετικά επίπεδα προτεραιότητας.
- Κάθε κόμβος πρέπει να διατηρεί αναλλοίωτη τη σειρά των πακέτων που ανήκουν στην ίδια ροή και στην ίδια κλάση AF.

Για να εφαρμοστεί μια AF-based υπηρεσία, στο σημείο εισόδου του domain γίνεται πρώτα σύγκριση του πραγματικού aggregate κίνησης με τα προκαθορισμένα και συμφωνημένα χαρακτηριστικά που θα έπρεπε να έχει. Έτσι η κίνηση εντός προφίλ

μαρκάρεται με προτεραιότητα 1 (πράσινα πακέτα), κίνηση εκτός προφίλ αλλά εντός ενός ορίου σαν προτεραιότητας 2 (κίτρινα πακέτα) και τέλος η υπόλοιπη σαν προτεραιότητας 3 (κόκκινα πακέτα). Η διαδικασία αυτή συνήθως γίνεται με τη χρήση 2 συνεχόμενων leaky buckets που λειτουργούν όμως και σαν μορφοποιητές. Στη συνέχεια μέσα στο δίκτυο κάθε AF κλάση διαχειρίζεται διαφορετικά όσο αφορά το θέμα της απόρριψης πακέτων από τους μηχανισμούς διαχείρισης ουρών.

Μια πρώτη παρατήρηση για τις AF-based υπηρεσίες είναι ότι παρουσιάζουν ορατή βελτίωση στην ποιότητα εξυπηρέτησης μόνο στην περίπτωση που το δίκτυο βρίσκεται σε συμφόρηση. Επίσης ένα μειονέκτημα τους είναι το γεγονός ότι είναι προσανατολισμένες στους αποστολείς και όχι στους παραλήπτες. Έτσι, στην ειδική περίπτωση που ο παραλήπτης πρέπει να λαμβάνει πληροφορία με συγκεκριμένο ρυθμό, αυτό δεν είναι εφικτό διότι η συνένωση ροών που φτάνουν στον παραλήπτη μπορεί να προέρχονται από διαφορετικά σημεία εισόδου στο DiffServ domain.

Γενικά οι AF-based υπηρεσίες δεν παρέχουν συγκεκριμένες εγγυήσεις για την καθυστέρηση και το jitter, αφού κάτι τέτοιο αναιρεί τη δυνατότητα που έχουν οι συνενώσεις ροών να καταλαμβάνουν περισσότερους πόρους από όσους συμφωνήθηκαν, αν αυτοί είναι διαθέσιμοι. Συγκεκριμένα για την παροχή εγγυήσεων από τις AF-based υπηρεσίες ισχύουν:

- Για την εξασφάλιση χωρητικότητας σε over-provisioned δίκτυα (δίκτυα όπου έχουν γίνει κρατήσεις για περισσότερους πόρους από όσους είναι πραγματικά διαθέσιμοι, με την εκτίμηση ότι δεν θα χρησιμοποιηθεί η υπηρεσία από όλους τους πελάτες ταυτόχρονα στο όριο) είναι εφικτή η εγγύηση ρυθμού που αφορά όμως μόνο την βασική πρόσβαση στους πόρους του δικτύου και όχι στην παροχή εγγυήσεων για χωρητικότητα που μπορεί να είναι αδιάθετη.
- Η πιθανότητα απόρριψης πακέτων μπορεί να είναι εγγυημένη μόνο όταν πρόκειται για SLAs μεταξύ 2 domains ενώ είναι δύσκολο και χρειάζεται ιδιαίτερη προσοχή η εγγύηση της από άκρο σε άκρο πιθανότητας απόρριψης.
- Τέλος για την καθυστέρηση ισχύουν ακριβώς τα ίδια όπως και στην περίπτωση της απόρριψης πακέτων.

Μια παραλλαγή των AF-based υπηρεσιών είναι οι Assured Data όπου χρησιμοποιούνται AF PHB και PDB όχι για τη διασφάλιση του ρυθμού αλλά για την διασφάλιση μετάδοσης συγκεκριμένων πακέτων. Για παράδειγμα ο μηδενισμός της πιθανότητας απόρριψης πράσινων πακέτων.

2.3.2.4 Υπηρεσία Less Than Best Effort

Παράλληλα με την υπηρεσία IP Premium που έχει υλοποιηθεί και δοκιμαστεί πειραματικά, έχει προταθεί και μια υπηρεσία που παρέχει εξυπηρέτηση πακέτων χειρότερη από Best effort. Η υπηρεσία αυτή ονομάζεται less than best effort[20] και προσανατολίζεται για χρήση από εφαρμογές που δεν ενδιαφέρονται για βέβαια μετάδοση πακέτων καθώς δεν είναι κρίσιμες. Τέτοια παραδείγματα είναι εφαρμογές ftp mirroring, GRID data transfer, network backups ή experimental data transfer. Ο στόχος της υπηρεσίας αυτής είναι να μεταφέρει πακέτα που προέρχονται από εφαρμογές που δεν ενδιαφέρονται άμεσα για την αποστολή και αυτά μεταδίδονται όταν υπάρχει διαθέσιμο εύρους ζώνης (δεν υπάρχει συμφόρηση στο δίκτυο).

Η υπηρεσία Less than best effort χρησιμοποιεί το Differentiated Code Point 001000 και για την ομαλή λειτουργία έχουν προκύψει ορισμένοι κανόνες:

- Τα πακέτα που ανήκουν στην κλάση αυτή πρέπει να χρησιμοποιούν ξεχωριστή ουρά, χωρίς να μπλέκονται με πακέτα άλλων κλάσεων.
- Στους δρομολογητές που υλοποιούν την υπηρεσία πρέπει να χρησιμοποιείται ένας μηχανισμός χρονοδρομολόγησης (WFQ, WRR) όπου θα δίνουν απειροελάχιστο εύρος ζώνης σε αυτά.
- Σε δρομολογητές όπου δεν υλοποιούν την υπηρεσία πρέπει η μετάδοση των πακέτων που ανήκουν στην κλάση αυτή να γίνεται διάφανα, δηλαδή σαν να ανήκουν στην best effort υπηρεσία και δεν επιτρέπεται σε κανένα σημείο να τους αλλάζουν την τιμή του DSCP πεδίου.

Η υπηρεσία έχει ήδη δοκιμαστεί σε αρκετές περιπτώσεις και πλέον θα αποτελεί υπηρεσία παραγωγής των σύγχρονων δικτύων

ΚΕΦΑΛΑΙΟ 3: ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ BANDWIDTH BROKERS

ΕΙΣΑΓΩΓΗ ΣΤΟΥΣ BANDWIDTH BROKERS

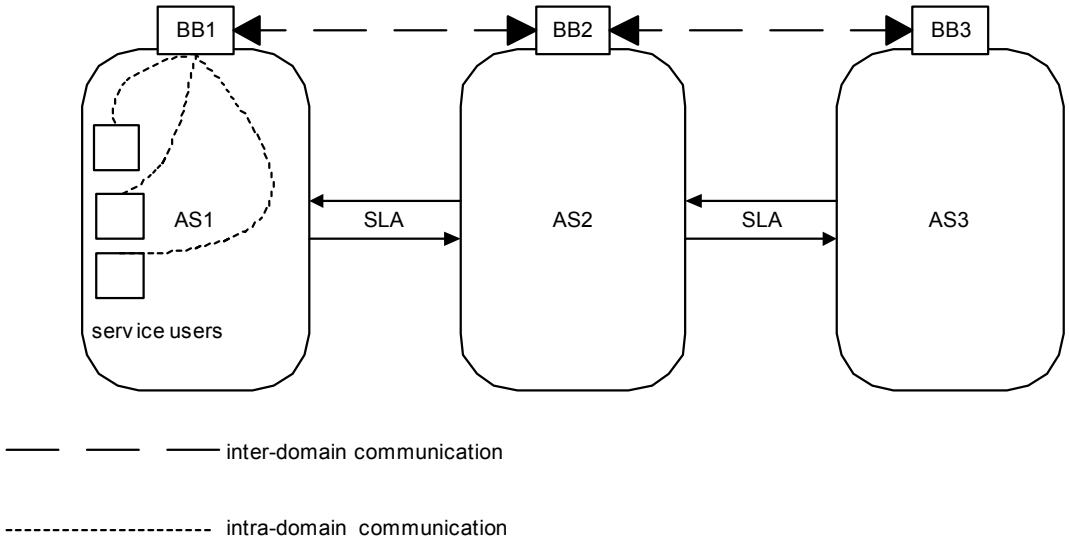
3.1 ΕΙΣΑΓΩΓΗ

Ένας Bandwidth Broker[24] διαχειρίζεται τους πόρους που βρίσκονται μέσα σε ένα συγκεκριμένο domain σύμφωνα με τις προδιαγραφές υπηρεσίας (Service Level Specifications-SLS) που έχουν συμφωνηθεί μέσα σε αυτό το domain. Οι προδιαγραφές υπηρεσίας είναι μια μετάφραση του συμβολαίου υπηρεσίας (Service Level Agreement-SLA)[25] στην κατάλληλη πληροφορία η οποία είναι αναγκαία για την κατανομή και την παροχή των πόρων των δικτυακών συσκευών και ειδικά στα άκρα του domain, σε links που το συνδέουν με γειτονικά domains. Ο Bandwidth Broker είναι επίσης υπεύθυνος για τη διαχείριση της inter-domain επικοινωνίας, με Bandwidth Brokers γειτονικών δικτύων, ώστε να συντονίζονται οι προδιαγραφές υπηρεσίας κατά μήκος των ορίων όλων των domains.

Ο Bandwidth Broker συγκεντρώνει και παρακολουθεί την κατάσταση των πόρων που βρίσκονται εντός του δικού του domain και στις ακμές που συνδέουν αυτό το domain με τα γειτονικά domains. Αυτή η πληροφορία μαζί με την πληροφορία αστυνόμευσης (από την policy rules data base) χρησιμοποιείται για αποφάσεις αποδοχής κλήσης αιτημάτων για Ποιότητα Υπηρεσίας στο δίκτυο. Ο διαχειριστής αστυνόμευσης (policy manager), αν υπάρχει, επιβεβαιώνει τέτοια αιτήματα χρησιμοποιώντας τους κανόνες αστυνόμευσης, ελέγχοντας αν υπάρχουν conflicts με υπάρχοντα αιτήματα, λαμβάνοντας μέτρα ώστε να προεκχωρούνται πόροι για συγκεκριμένες υπηρεσίες κλπ. Ο Bandwidth Broker χρησιμοποιεί την πληροφορία της κατάστασης του δικτύου και για να επιβεβαιώνει ότι όντως υπάρχουν διαθέσιμοι πόροι στο δίκτυο ώστε να εξυπηρετηθούν αιτήματα. Αυτή η πληροφορία μπορεί να λαμβάνεται απευθείας από τον Bandwidth Broker δια μέσου ενός interface με τον policy manager, ή ο Bandwidth Broker μπορεί να την τοποθετεί σε μια βάση δεδομένων που την χρησιμοποιεί από κοινού με τον policy manager.

Κατά μήκος των ορίων των domains, οι προδιαγραφές υπηρεσίας μπορούν να είναι στη βάση συνενώσεων ροών, όπου οι συνενώσεις ροών γίνονται χρησιμοποιώντας ένα συγκεκριμένο πεδίο μαρκαρίσματος, όπως για παράδειγμα το DiffServ codepoint(DSCP). Μέσα σε ένα domain μπορούν να κατανέμονται πόροι σε ανεξάρτητες ροές χρησιμοποιώντας αιτήματα κατανομής πόρων (Resource Allocation Requests-RARs). Είναι ευθύνη του Bandwidth Broker να συντονίζει την κατανομή και την παροχή των πόρων του δικτύου στις συνενώσεις ροών που εισέρχονται ή εξέρχονται από το δικό του domain όταν δέχεται RARs.

Η Εικόνα 14 παρουσιάζει ένα υποθετικό δίκτυο. Αποτελείται από τρία domains AS1,AS2 και AS3, ενώ τους πόρους του κάθε domain τους διαχειρίζεται ο Bandwidth Broker BB1, BB2 και BB3 αντίστοιχα. Μεταξύ των AS1 και AS2 καθώς και των AS2 και AS3 βρίσκονται τα SLSs. Χρήστες υπηρεσιών που βρίσκονται εντός του AS1 στέλνουν RARs στον BB1



Εικόνα 14 Δείγμα της κατάστασης δικτύου που χρησιμοποιεί bandwidth brokers για τη διαχείριση του bandwidth

Στην υλοποίηση ενός Bandwidth Broker προηγείται ο σχεδιασμός των τύπων υπηρεσίας που θα υποστηρίζονται μέσα στο domain και από τον Bandwidth Broker, η υλοποίηση των υπηρεσιών και στη συνέχεια ο σχεδιασμός της αντιστοίχησής τους με τις υπηρεσίες των γειτονικών domains.

3.2 ΣΥΜΒΟΛΑΙΑ ΥΠΗΡΕΣΙΑΣ

Το συμβόλαιο υπηρεσίας (SLA – Service Level Agreement)[25] είναι ένα συμβόλαιο που καθορίζει την παρεχόμενη υπηρεσία την οποία ένας πελάτης αναμένεται να λάβει από έναν provider. Ένα συμβόλαιο υπηρεσίας μπορεί να καλύπτει μία ή περισσότερες υπηρεσίες τις οποίες ο provider παρέχει στον πελάτη και ένα μέρος της συμφωνίας καθορίζει τον τρόπο με τον οποίο ο provider θα αναγνωρίζει και θα κατηγοριοποιεί τα πακέτα που πρόκειται να λάβουν αυτές τις υπηρεσίες.

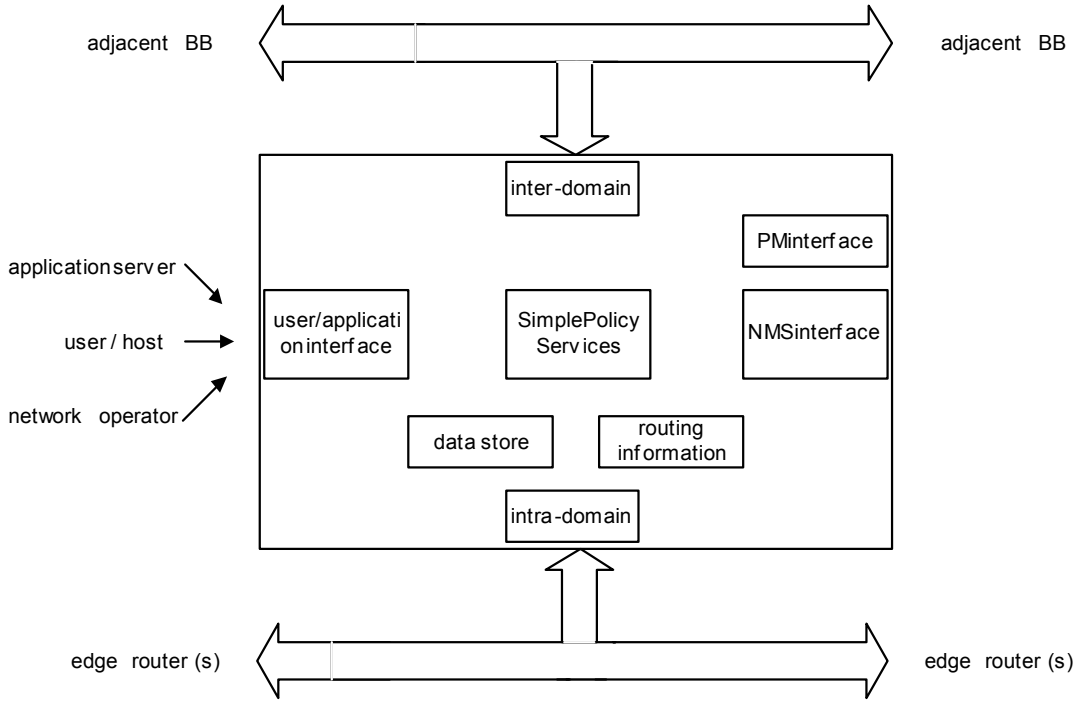
Οι προδιαγραφές υπηρεσίας (SLS – Service Level Specifications) περιγράφουν τους συγκεκριμένους πόρους που παρέχονται από κάθε υπηρεσία όπως αναφέρονται στο συμβόλαιο κίνησης. Οι προδιαγραφές υπηρεσίας είναι προσπελάσιμες από τον Bandwidth Broker. Η πληροφορία σε ένα SLS είναι γενικά στο επίπεδο των συναθροίσεων ροών (flow aggregates) και στους πόρους οι οποίοι παρέχονται σε κάθε aggregate. Ένα SLS τυπικά εφαρμόζεται στα άκρα ενός link που συνδέει γειτονικά domains.

Αιτήματα κατανομής πόρων (RAR – Resource Allocation Requests) στέλνονται στον Bandwidth Broker από τους χρήστες υπηρεσίας ζητώντας τη χρήση των πόρων που παρέχονται μέσα σε ένα domain για μια συγκεκριμένη υπηρεσία. Το RAR περιέχει τον προορισμό της κίνησης για την οποία έγινε το αίτημα και ο Bandwidth Broker ελέγχει για το συγκεκριμένο προορισμό εάν η κίνηση είναι τοπική (εάν και ο προορισμός βρίσκεται εντός του domain που καλύπτει ο Bandwidth Broker) ή αν η

κίνηση είναι προς κάποιο άλλο domain (εάν ο προορισμός βρίσκεται εκτός του domain που καλύπτει ο Bandwidth Broker). Πολλά RARs μπορούν να αντιστοιχίζονται σε ένα μοναδικό SLS. Όταν οι πόροι που καθορίζονται σε ένα SLS έχουν διανεμηθεί πλήρως σε υπάρχοντα RARs και ληφθεί ένα νέο RAR για τον ίδιο προορισμό, το RAR θα πρέπει να απορριφθεί ή το SLS θα πρέπει να επαναδιαπραγματευτεί.

3.3 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ BANDWIDTH BROKER

Η Εικόνα 15 παρουσιάζει την αρχιτεκτονική των λειτουργικών μονάδων ενός Bandwidth Broker. Δεν απαιτούνται όλες οι λειτουργικές μονάδες από όλες τις υλοποιήσεις.



Εικόνα 15 Η αρχιτεκτονική ενός Bandwidth Broker

Ένας Bandwidth Broker μπορεί να έχει τις ακόλουθες λειτουργικές μονάδες:

- Το interface χρήστη / εφαρμογών (user/application interface)
- Το inter-domain interface
- Το intra-domain interface
- Το interface πινάκων δρομολόγησης (routing table interface)
- Το interface διαχείρισης της αστυνόμευσης (policy manager interface)
- Το interface διαχείρισης δικτύου (network management interface)
- Την αποθήκη δεδομένων (data repository)

Στη συνέχεια αναλύουμε κάθε μια από αυτές τις λειτουργικές μονάδες.

3.3.1 User/Application Interface

Το user/application interface αποτελείται από τα παρακάτω τμήματα:

- Το GUI, έτσι ώστε ο χειριστής του δικτύου να μπορεί να:
  - διαχειρίζεται την αποθήκη δεδομένων χρησιμοποιώντας απλές εντολές
  - εισάγει απλούς κανόνες αστυνόμευσης στο σύστημα
  - εισάγει χειροκίνητα στο σύστημα αιτήματα για bandwidth προς τον Bandwidth Broker
  - στέλνει ερωτήματα στον Bandwidth Broker για υπάρχουσες δεσμεύσεις υπηρεσιών, την παροχή πόρων και άλλα.
  - ανταποκρίνεται στον χειριστή με σφάλματα και άλλα.
- Το Host/User interface, το οποίο χρησιμοποιείται ώστε:
  - Οι εφαρμογές του host και των χρηστών να μπορούν απευθείας να στείλουν αιτήματα για bandwidth στον Bandwidth Broker
  - Οι χρήστες να μπορούν να στέλνουν ερωτήματα στον Bandwidth Broker για τις υπάρχουσες δεσμεύσεις υπηρεσιών που τους έχουν χορηγηθεί. Απαιτείται ασφάλεια έτσι ώστε να μην είναι δυνατό κάποιος χρήστης να δει τις δεσμεύσεις εφαρμογών άλλων χρηστών
  - Να υπάρχουν μηχανισμοί που να ανταποκρίνονται στον χρήστη ή στην εφαρμογή
  - Ως προς το interface χρήστη, μπορεί να χρησιμοποιηθεί το ίδιο GUI όπως και αυτό του χειριστή δικτύου. Για τις εφαρμογές απαιτείται η ύπαρξη ενός πρωτοκόλλου
- Το Server/Gateway interface, το οποίο χρησιμοποιείται ώστε να εξυπηρετούνται αιτήματα από servers εφαρμογών, όπως για παράδειγμα το H.323 gateway.

3.3.2 Inter-Domain interface

Το inter-domain interface έχει τη μορφή ενός πρωτοκόλλου αιτήματος-επιβεβαίωσης. Οι Bandwidth Brokers που επιθυμούν να επικοινωνήσουν χρειάζεται αρχικά να εγκαταστήσουν συνδέσεις μεταξύ των ζευγών των γειτονικών Bandwidth Brokers και στη συνέχεια χρησιμοποιώντας αυτές τις συνδέσεις να ανταλλάξουν τα RARs. Οι συνδέσεις μεταξύ των Bandwidth Brokers γίνονται πάντα κατά ζεύγη και είναι και προς τις δύο κατευθύνσεις (και οι δύο Bandwidth Brokers μπορούν να ζητήσουν πόρους ο ένας από τον άλλο). Παρόλα αυτά, τα SLSs είναι προς μια κατεύθυνση. Δηλαδή χρειάζεται να εγκατασταθούν δύο SLSs, ένα προς κάθε κατεύθυνση.

Τα στάδια κατά τη λειτουργία του πρωτοκόλλου είναι:

- Τοπικό Configuration: Το πρωτόκολλο φορτώνει τοπική πληροφορία για το configuration καθώς ξεκινά
- Αρχικοποίηση Σύνδεσης: Μετά το ξεκίνημα, οι Bandwidth Brokers εγκαθιστούν συνδέσεις με τους γειτονικούς Bandwidth Brokers

- Έναρξη Υπηρεσίας: Οι συνδεδεμένοι Bandwidth Brokers ανταλλάσσουν αιτήματα για υπηρεσίες για τις οποίες έχουν συμφωνηθεί στα SLAs
- Κατανομή Πόρων: Οι Bandwidth Brokers ζητούν πόρους και απαντούν σε αιτήματα κατανομής πόρων.

Εκτός από τα παραπάνω μπορεί να υποστηρίζεται επιπρόσθετες δυνατότητες όπως η διαπραγμάτευση, διαχείριση και συντήρηση πόρων. Επίσης είναι σημαντικό να υπάρχει υποδομή τέτοια ώστε ο Bandwidth Broker να ανέχεται σφάλματα του συστήματος όπως το να διακόπτεται η λειτουργία κάποιου ανεξάρτητου Bandwidth Broker ή μια σύνδεση μεταξύ δύο Bandwidth Brokers.

3.3.2.1 Τοπικό configuration

Ο καθορισμός μιας υπηρεσίας μεταξύ δύο domains και των αρχικών διαθέσιμων πόρων που θα υποστηρίζουν αυτήν την υπηρεσία δεν ορίζεται με επικοινωνία μεταξύ των Bandwidth Brokers. Εξάλλου, ο καθορισμός των πόρων, άλλων παραμέτρων όπως τα DS code points, η PHB μεταχείριση και η κίνηση εκτός του προφίλ (out-of-profile traffic) πρέπει να συμφωνηθούν πριν ξεκινήσει η inter-domain επικοινωνία.

Όλοι οι Bandwidth Brokers θα πρέπει να έχουν πρόσβαση στις ακόλουθες πληροφορίες που αφορούν τις inter-domain και τις intra-domain συνδέσεις των routers που βρίσκονται στα όρια του κάθε domain:

a) Μέγιστη διαθεσιμότητα πόρων ανά υπηρεσία
b) Πληροφορία για τη διαστασιολόγηση και το σχεδιασμό κάθε Υπηρεσίας
c) Αντιστοίχηση Υπηρεσίας – PHB
d) Αντιστοίχηση Υπηρεσίας – DSCP
e) SLA ID
f) Μεθόδους σηματοδότησης οι οποίοι είναι επιτρεπτοί από το SLA

Όλες αυτές οι πληροφορίες (a ως f) είναι υποχρεωτικές.

3.3.2.2 Αρχικοποίηση Σύνδεσης

Πριν ξεκινήσουν οποιεσδήποτε υπηρεσίες μεταξύ δύο domains, οι Bandwidth Brokers χρειάζεται να συνδεθούν ώστε να εξασφαλιστεί η επικοινωνία. Κατά τη διάρκεια αυτής της διαδικασίας αρχικοποίησης, ο Bandwidth Broker διαβιβάζει:

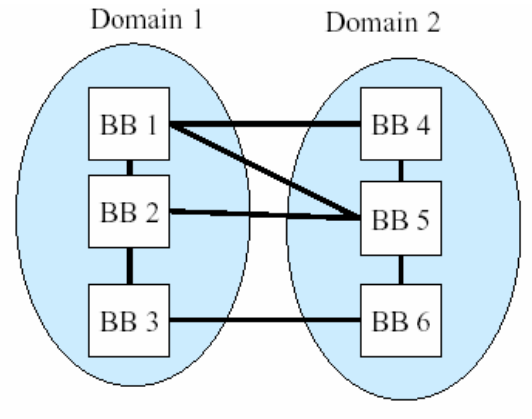
- τα AS που βρίσκονται στο δικό του domain
- τη διεύθυνση – το id του Bandwidth Broker ο οποίος πρόκειται να στείλει αίτημα
- τη λίστα των γειτονικών του Bandwidth Brokers

Όλες αυτές οι πληροφορίες είναι υποχρεωτικές. Σε ένα περιβάλλον όπου υπάρχουν περισσότεροι Bandwidth Brokers από έναν, δύο domains μπορούν να έχουν πολλαπλές συνδέσεις μεταξύ τους. Στην Εικόνα 16 παρουσιάζεται ένα τέτοιο δίκτυο που αποτελείται από δύο domains. Υπάρχουν τρεις συνδέσεις που χρησιμοποιούνται στην inter-domain επικοινωνία. Αυτό προσφέρει πλεονασμό σε περίπτωση κάποιας βλάβης. Η λίστα των Bandwidth Brokers που στέλνει ο BB1 στο domain 2 θα περιλάμβανε τους BB2, BB4 και BB5.

3.3.2.3 Διαχείριση Υπηρεσιών

Οι Bandwidth Brokers είναι υπεύθυνοι για τη διαχείριση των υπηρεσιών οι οποίες είναι διαθέσιμες στους πελάτες μέσω του inter-domain και του intra-domain interface. Οι υπηρεσίες έχουν συμφωνηθεί σε SLAs και έχουν οριστεί οι αντίστοιχες προδιαγραφές υπηρεσίας SLSs. Οι λειτουργίες για τη διαχείριση των SLSs περιλαμβάνουν:

- Έναρξη της Υπηρεσίας (υποχρεωτικό)
- Αναβολή της Υπηρεσίας (υποχρεωτικό)
- Επαναδιαπραγμάτευση της Υπηρεσίας (προαιρετικό)



Εικόνα 16 domains με πολλαπλές διασυνδέσεις

3.3.2.3.1 Έναρξη της Υπηρεσίας

Πριν από την έναρξη μιας υπηρεσίας μεταξύ δύο domains, οι peer Bandwidth Brokers πρέπει να γίνουν configured με ένα συνεπή τρόπο όσον αφορά τη διμερή συμφωνία για αυτή την υπηρεσία. Όλοι οι πόροι που είναι αναγκαίοι για την υποστήριξη της υπηρεσίας θα πρέπει να παρέχονται και να κατακρατούνται σε κάθε τοπικό domain. Όταν οριστούν οι πόροι που διαχειρίζεται ο κάθε Bandwidth Broker, μπορούν να δημιουργηθούν μηνύματα αιτημάτων για την έναρξη υπηρεσίας (Service Setup Requests - SSR). Το SSR μήνυμα είναι ένας απλός μηχανισμός ώστε ένας Bandwidth Broker να επιβεβαιώσει το SLA μεταξύ αυτού και κάποιου άλλου Bandwidth Broker και να επιβεβαιώσει στον άλλο ότι υπάρχουν οι αναγκαίοι πόροι ώστε να τηρηθεί το SLA.

Ένα SSR μήνυμα μπορεί να ξεκινήσει με διάφορους τρόπους:

- Με εντολή του διαχειριστή δικτύου
- Αυτοματοποιημένα με την εκκίνηση του συστήματος ή την επανεκκίνηση (βασισμένο στις κατανεμημένες υπηρεσίες που είχαν πρωτύτερα ζητηθεί και δεν έχουν λήξει ή αναβληθεί)

Ένα SSR μήνυμα περιέχει τα ακόλουθα δεδομένα:

- Το BB_ID του Bandwidth Broker που στέλνει το SSR
- Το SSU_ID (Service Set Up - ID)

- SLA_ID
- PHB
- Τον DiffServ egress router
- Τον γειτονικό DiffServ downstream router
- Το DSCP (μπορεί να είναι περισσότερα από ένα)
- Το DSCP για τους συγκεκριμένους πόρους στο downstream (ένα ανά DSCP)
- Ένα flag που δείχνει τον τύπο σηματοδότησης, από άκρο σε άκρο ή άμεσης απόκρισης

Ο παραλήπτης (downstream) Bandwidth Broker θα ελέγξει τη συνέπεια όσον αφορά το SLS με την πληροφορία που έχει αποθηκεύσει στη δικιά του βάση δεδομένων. Αν υπάρχει συνέπεια, ο Bandwidth Broker απαντά με ένα μήνυμα Απάντησης Εκκίνησης (Set up Answer - SA). Το SA μήνυμα περιέχει:

- Το BB_ID του Bandwidth Broker που απαντά
- Το SSU_ID
- Το SLA_ID
- Ένδειξη έγκρισης ή απόρριψης
- Ένα flag του τύπου σηματοδότησης που χρησιμοποιεί ο Bandwidth Broker που έστειλε το SSR

Κάθε Bandwidth Broker πρέπει να ελέγχει ότι όλοι οι πόροι είναι ενεργοί ή διατεθειμένοι σε υπηρεσίες. Η κατάσταση των πόρων πρέπει να ανανεώνεται κάθε φορά που εγκρίνεται ένα SA. Πρέπει εδώ να σημειωθεί ότι η υπηρεσία δεν είναι συμμετρική. Για παράδειγμα, το DSCP για συγκεκριμένους πόρους από το X στο Y μπορεί να διαφέρει από αυτό από το Y στο X.

3.3.2.3.2 Αναβολή Υπηρεσίας

Εάν ένα SLA τερματιστεί, η υπηρεσία πρέπει να αναβληθεί. Το μήνυμα Αναβολής Υπηρεσίας (Service Cancellation - SC) περιέχει:

- Το BB_ID
- Το SSU_ID
- Το SLA_ID

Αυτό το αίτημα μπορεί να ξεκινήσει από μια εντολή ενός διαχειριστή δικτύου ή μπορεί να πυροδοτηθεί σε κάποια ημερομηνία λήξης. Ένα SC ελέγχεται για συνέπεια κατά τη λήψη και στέλνεται η Απάντηση Αναβολής (Cancellation Answer - CA). Το CA περιέχει τα ίδια πεδία με το SC. Όταν στέλνεται το CA, ο Bandwidth Broker ανανεώνει την κατάσταση όλων των πόρων που είχαν κατακρατηθεί για τη συγκεκριμένη υπηρεσία. Επειδή από τη φύση της μια υπηρεσία είναι προς μια κατεύθυνση, μια υπηρεσία μπορεί να αναβάλλεται στην κατεύθυνση από τον X στον Y ενώ παραμένει ενεργή από τον Y στον X.

3.3.2.3.3 Επαναδιαπραγμάτευση Υπηρεσίας (προαιρετικό)

Ένα SLA μπορεί να επιτρέπει στον Bandwidth Broker να επανα-διαπραγματεύεται μια υπάρχουσα υπηρεσία. Αυτό επιτυγχάνεται με ένα μήνυμα Επαναδιαπραγμάτευσης Υπηρεσίας (Service Renegotiate - SR). Τα SR μηνύματα μπορούν να δημιουργηθούν είτε χειροκίνητα από τον διαχειριστή ενός δικτύου είτε αυτόματα από τον Bandwidth Broker όταν η κίνηση που λαμβάνει την υπηρεσία υπερβαίνει ένα κατώφλι. Αν χρειάζεται να προστεθούν επιπρόσθετοι πόροι για κάποια υπηρεσία, η κατάσταση των νέων πόρων που διατέθηκαν πρέπει να ενημερωθεί στην βάση δεδομένων του Bandwidth Broker.

Το SR εγκρίνεται από ένα SA μήνυμα. Μετά την έγκριση οι επιπρόσθετοι πόροι έχουν προστεθεί και στα δύο domains και η υπηρεσία είναι έτοιμη για χρήση. Και οι δύο Bandwidth Brokers χρειάζεται να ανανεώσουν τις traffic conditioning και policing παραμέτρους στους routers που βρίσκονται στα domain τους. Το SR αποτελείται από:

- Το BB_ID
- Το SSU_ID (ίδια τιμή όπως και στο προηγούμενο SSU)
- Το SLA_ID
- Το DSCP για τους συγκεκριμένους πόρους στο downstream domain (νέα τιμή)

Όλες οι άλλες παράμετροι αναμένονται να είναι ίδιες με πριν και δεν επαναμεταδίδονται.

3.3.2.4 Κατανομή Πόρων

Αφού ολοκληρωθεί η αρχική διαδικασία εκκίνησης υπηρεσίας, η υπηρεσία είναι διαθέσιμη. Η υπηρεσία δεν έχει ακόμα κατανεμηθεί σε συγκεκριμένες ροές. Ο Bandwidth Broker δεν πρέπει να επιτρέψει στη διαθέσιμη υπηρεσία να χρησιμοποιηθεί έως ότου λάβει RARs για την υπηρεσία μέσα από το domain και SLS μεταξύ των domains.

Τα RAR αιτήματα μέσα σε ένα domain επιτρέπουν σε ένα χρήστη η διαχειριστή δικτύου να ζητήσει πόρους από τον Bandwidth Broker. Τα RARs επιτρέπουν στον Bandwidth Broker μέσα σε ένα domain να καταγράφει την πραγματική ποσότητα μιας διαθέσιμης υπηρεσίας που έχει διανεμηθεί σε κίνηση μέσα στο domain, και βασισμένος σε αυτό να παίρνει αποφάσεις αποδοχής. Ο Bandwidth Broker είναι έτοιμος να επεξεργαστεί RARs από ένα domain όταν εγκριθεί το SSR.

Η κατανομή πόρων περιλαμβάνει μηνύματα αιτήματος και απόκρισης, τα RARs και RAA αντίστοιχα. Τα RARs και RAAs χρησιμοποιούνται και στο intra-domain interface. Το Resource Allocation Request περιλαμβάνει τα πεδία:

- BB_ID
- SSU_ID (το έγκυρο SSU_ID της αρχικοποίησης υπηρεσίας)
- RAR_ID
- DS_egress_router_ID
- DSCP (πιθανόν να είναι περισσότερα από ένα)
- DSCP για τα downstream domains (ένα ανά DSCP)

- Παράμετροι όπως: διεύθυνση αποστολέα, διεύθυνση προορισμού, πρωτόκολλα, αριθμοί των ports, πληροφορία για το προφίλ, παράμετροι διαπραγμάτευσης

Το RAR περιλαμβάνει τα πεδία:

- BB_ID
- SSU_ID
- RAR_ID
- Ένδειξη έγκρισης ή απόρριψης του RAR

3.3.2.5 Διατήρηση της σύνδεσης

Το πρωτόκολλο το οποίο υλοποιεί το inter-domain interface, εκτός από την ανταλλαγή μηνυμάτων που αφορούν κατακράτηση πόρων, θα πρέπει να υποστηρίζει και ένα σύνολο λειτουργιών διατήρησης της σύνδεσης. Αυτές οι λειτουργίες περιλαμβάνουν:

1. Το πρωτόκολλο θα μπορούσε να χρησιμοποιεί ένα ήδη υπάρχον connection oriented transport πρωτόκολλο, όπως το TCP
2. Διαδικασία Keep Alive
3. Ένα μηχανισμό που να αντιμετωπίζει επιτυχώς κάποιο σταμάτημα ή βλάβη κάποιου Bandwidth Broker και τη διακοπή της επικοινωνίας μεταξύ Bandwidth Brokers. Θέματα που θα πρέπει να προσεχτούν είναι :
    - Ένας Bandwidth Broker που ξαναρχίζει να λειτουργεί πρέπει να ξανακατασκευάζει την κατάστασή του.
    - Ένας Bandwidth Broker που πρόσφατα ξανάρχισε να λειτουργεί χρειάζεται να ενημερώσει τους γειτονικούς Bandwidth Brokers για την κατάστασή του.
    - Ένας Bandwidth Broker που πρόσφατα ξανάρχισε να λειτουργεί είναι υπεύθυνος για το συγχρονισμό και τις επαναδιαπραγματεύσεις με τους γειτονικούς Bandwidth Brokers για τις υπηρεσίες και την κατανομή των πόρων.
4. Το interface θα πρέπει να περιλαμβάνει timers οι οποίοι να λήγουν στην περίπτωση χαμένων ή καθυστερημένων απαντήσεων.
5. Εάν ένα αίτημα απορριφτεί μετά από επαναλαμβανόμενες απόπειρες, θα πρέπει να σταλεί ένα μήνυμα στον διαχειριστή του δικτύου το οποίο δημιουργεί τα αιτήματα.
6. Απαιτείται να υπάρχει μια προκαθορισμένη συμπεριφορά που να αφορά τη διαχείριση μη προσδοκώμενων μηνυμάτων (π.χ. απαντήσεις χωρίς να έχει γίνει κάποιο αίτημα) και μη προσδοκώμενες παράμετροι σε μηνύματα

Οι λειτουργίες 1 και 5 είναι προαιρετικές ενώ όλες οι υπόλοιπες είναι υποχρεωτικές

3.3.3 Intra-Domain Interface

Το intra-domain interface είναι το interface που χρησιμοποιείται για την επικοινωνία του Bandwidth Broker με τους routers που βρίσκονται στο τοπικό του domain ώστε να κατανέμονται κατάλληλα οι πόροι αυτού του domain.

3.3.3.1 Μηχανισμοί υλοποίησης του interface

Το intra-domain interface υλοποιείται σαν μια αυτοματοποιημένη διαδικασία. Με αυτήν την αυτοματοποιημένη διαδικασία ο Bandwidth Broker κάνει configure τους routers ώστε να κατανέμονται οι DiffServ πόροι. Κάθε αυτοματοποιημένη διαδικασία θα πρέπει να βασίζεται στους μηχανισμούς του router για configuration, όπως αυτοί παρέχονται από τους κατασκευαστές των routers. Οι κυριότεροι από αυτούς τους μηχανισμούς είναι οι:

1. **COPS**[26]

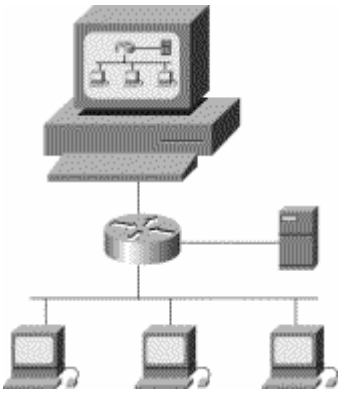
Το πρωτόκολλο COPS (Common Open Policy Service) περιγράφει ένα απλό πρωτόκολλο ερώτησης-απάντησης που μπορεί να χρησιμοποιηθεί για την ανταλλαγή policing πληροφορίας ανάμεσα σε έναν policy server (Policy Decision Point or PDP) και τους clients του (Policy Enforcement Points or PEPs). Έχει σχεδιαστεί να είναι επεκτάσιμο έτσι ώστε και διαφορετικού τύπου policing clients να μπορούν να υποστηριχτούν στο μέλλον. Το μοντέλο δεν κάνει καμία υπόθεση σχετικά με τις μεθόδους του policy server, αλλά στηρίζεται στις αποφάσεις που επιστρέφει ο server στα αιτήματα policing. Κάθε μήνυμα αποτελείται από την COPS header η οποία ακολουθείται από έναν αριθμό από τυποποιημένα αντικείμενα. Το πρωτόκολλο COPS-PR(COPS Usage for Policy Provisioning)[32] αποτελεί επέκταση του πρωτοκόλλου COPS για την υποστήριξη του policing provisioning.

Οι routers οι οποίοι υποστηρίζουν το COPS-PR, γίνονται configure από το πρωτόκολλο COPS. Ο Bandwidth Broker θα περιλαμβάνει έναν COPS server και θα χρησιμοποιεί το πρωτόκολλο ώστε να στέλνει αιτήματα στις DiffServ υπηρεσίες.

2. **SNMP**[27]

Το SNMP(*Simple Network Management Protocol*) πρωτόκολλο είναι ένα πρωτόκολλο του επιπέδου εφαρμογής(application layer) που διευκολύνει την ανταλλαγή πληροφορίας διαχείρισης(management information) ανάμεσα στις δικτυακές συσκευές. Αποτελεί μέρος του σετ πρωτοκόλλων TCP/IP(Transmission Control Protocol/Internet Protocol) και δίνει την δυνατότητα στους διαχειριστές του δικτύου να διαχειρίζονται την απόδοση του δικτύου, να βρίσκουν και να επιλύουν δικτυακά προβλήματα και να σχεδιάζουν την επέκταση του δικτύου.

Υπάρχουν δύο εκδόσεις του SNMP πρωτοκόλλου: η έκδοση 1(SNMPv1) και η έκδοση 2(SNMPv2). Και οι δύο εκδόσεις έχουν έναν αριθμό από κοινά χαρακτηριστικά, αλλά η έκδοση 2 παρέχει επιπλέον χαρακτηριστικά όπως επιπλέον λειτουργίες πρωτοκόλλου. Επίσης η πρωτυποποίηση(standardization) της έκδοσης τρία(SNMPv3) του πρωτοκόλλου είναι σε αναμονή. Στην Εικόνα 17 παρουσιάζεται ένα βασικό δίκτυο που διοικείται (managed) από το πρωτόκολλο SNMP.



Εικόνα 17 Το SNMP διευκολύνει την ανταλλαγή της δικτυακής πληροφορίας ανάμεσα στις συσκευές

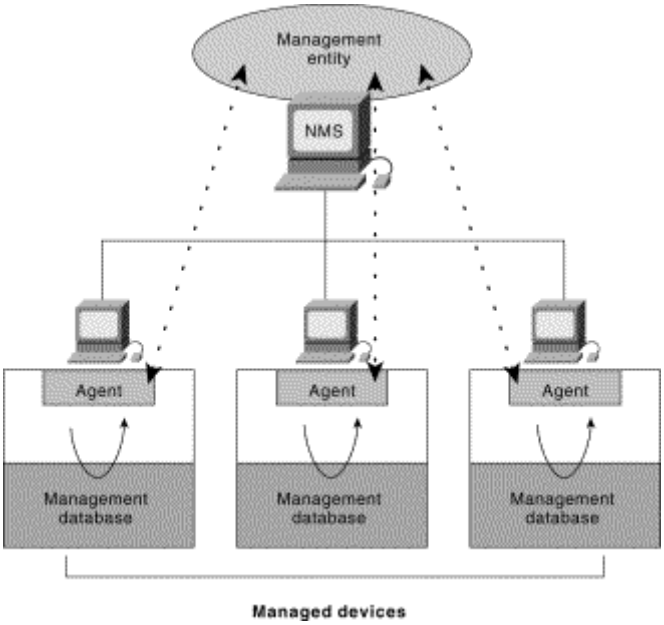
Ένα δίκτυο που διοικείται από το πρωτόκολλο SNMP αποτελείται από τρία στοιχεία κλειδιά: διαχειριζόμενες συσκευές, agents και συστήματα διαχείρισης δικτύου.

Μια διαχειριζόμενη συσκευή είναι ένας κόμβος δικτύου που περιέχει έναν SNMP agent και ανήκει σε ένα διαχειριζόμενο δίκτυο. Οι διαχειριζόμενες συσκευές συλλέγουν και αποθηκεύουν πληροφορία σχετική με τη διαχείριση και κάνουν την πληροφορία αυτή διαθέσιμη στα συστήματα διαχείρισης δικτύου (NMS) με τη χρήση του πρωτοκόλλου SNMP. Οι διαχειριζόμενες συσκευές καμιά φορά ονομάζονται στοιχεία δικτύου και μπορεί να είναι routers και access servers, switches και bridges, hubs, computer hosts, ή εκτυπωτές.

Ένας agent είναι ένα software module διαχείρισης δικτύου που ανήκει σε μία διαχειριζόμενη συσκευή. Ένας agent έχει τοπική γνώση της διαχειριζόμενης πληροφορίας και μεταφράζει την πληροφορία αυτή σε μια μορφή συμβατή με το πρωτόκολλο SNMP.

Ένα σύστημα διαχείρισης δικτύου (NMS) εκτελεί εφαρμογές που παρακολουθούν και ελέγχουν διαχειριζόμενες συσκευές. Τα NMS καταναλώνουν το μεγαλύτερο μέγεθος από επεξεργαστική ισχύς και από πόρους μνήμης που απαιτούνται για τη διαχείριση του δικτύου. Ένα ή περισσότερα NMS πρέπει να υπάρχουν σε οποιοδήποτε διοικούμενο δίκτυο.

Η Εικόνα 18 παρουσιάζει τις σχέσεις των τριών αυτών στοιχείων.



Εικόνα 18 Ένα SNMP διοικούμενο δίκτυο αποτελείται από διαχεριζόμενες συσκευές ,agents και NMS

3. **Telnet**[28]

Το πρωτόκολλο telnet παρέχει στους χρήστες τη δυνατότητα να τρέχουν προγράμματα από απόσταση και διευκολύνει την από απόσταση διαχείριση(administration). Το πρωτόκολλο telnet είναι διαθέσιμο πρακτικά για όλα τα λειτουργικά συστήματα και βοηθάει την ενσωμάτωση σε ετερογενή δικτυακά περιβάλλοντα.

Το πρωτόκολλο telnet είναι περισσότερο κατανοητό στο πλαίσιο ενός χρήστη με ένα απλό τερματικό που χρησιμοποιεί το τοπικό πρόγραμμα telnet (γνωστό και ως πρόγραμμα client) που τρέχει ένα logon session σε ένα υπολογιστή σε απόσταση όπου οι ανάγκες του χρήστη για επικοινωνία διαχειρίζονται από ένα Telnet server πρόγραμμα. Πρέπει να τονιστεί ότι ο Telnet server μπορεί να περάσει τα δεδομένα τα οποία έλαβε από τον client σε πολλούς άλλους τύπους διαδικασίας (process) όπως έναν remote logon server.

Η επικοινωνία εγκαθίσταται με τη χρήση του TCP/IP και στηρίζεται σε ένα Network Virtual Terminal (NVT). Στον client, το πρόγραμμα telnet είναι υπεύθυνο για την μετάφραση εισερχομένων NVT κωδικών σε κώδικες κατανοητούς από την μηχανή απεικόνισης του client όπως και αντίστοιχα είναι υπεύθυνο για την μετάφραση των κωδικών πληκτρολογίου που παράγονται από τον client σε εξερχόμενους NVT κώδικες.

Ένα command line interface ή CLI είναι μια μέθοδος διεπαφής με έναν υπολογιστή διαμέσου ενός τερματικού κειμένου. Οι εντολές εισάγονται σαν γραμμές ενός κειμένου από ένα πληκτρολόγιο και η έξοδος επίσης λαμβάνεται σαν ένα κείμενο.

Οι περισσότεροι routers υποστηρίζουν command line interfaces (CLIs). Ένας Bandwidth Broker μπορεί να επικοινωνεί με έναν router και να τον κάνει configure με το να αυτοματοποιεί ένα telnet session και να εκτελεί τις κατάλληλες configuration εντολές χρησιμοποιώντας το CLI

3.3.3.2 Οι δυνατότητες του intra-domain interface

Το intra-domain interface θα πρέπει να επιτρέπει στον Bandwidth Broker να διαχειρίζεται τις DiffServ υπηρεσίες σε κάθε router που δρομολογεί πακέτα στο domain. Αυτό θα πρέπει υποχρεωτικά να περιλαμβάνει τους ακραίους routers αλλά μπορεί να περιλαμβάνει και άλλους routers που δεν βρίσκονται στα άκρα του domain. Η διαχείριση ενός DiffServ router περιλαμβάνει το configuration και τον έλεγχο των διάφορων Traffic Conditioning τμημάτων μέσα στον router.

3.3.4 Routing Interface

Όταν γίνεται η επεξεργασία ενός αιτήματος για bandwidth, ο Bandwidth Broker χρειάζεται να έχει πρόσβαση στην routing πληροφορία ώστε να ληφθούν οι αποφάσεις του ελέγχου αποδοχής. Αυτή η πληροφορία θα συνδυαστεί με την πληροφορία για το policing. Ο Bandwidth Broker χρειάζεται να γνωρίζει πληροφορία τόσο για την inter-domain όσο και για την intra-domain δρομολόγηση Συγκεκριμένα, ο Bandwidth Broker θα χρειαστεί πληροφορία για:

- τα AS
- τα μονοπάτια που θα πρέπει να προτιμώνται προς τα γειτονικά domains
- πληροφορία για τη διαθεσιμότητα πόρων στα εξερχόμενα links

Μελλοντικά είναι πιθανό να υλοποιηθούν και πρωτόκολλα για QoS routing.

3.3.5 Αποθήκη Δεδομένων, Policy Manager Interface, Network Manager Interface

Η αποθήκη δεδομένων χρησιμοποιείται από όλα τα τμήματα της αρχιτεκτονικής του Bandwidth Broker. Η αποθήκη δεδομένων μπορεί να είναι κοινή για τον Bandwidth Broker και για τον Policy Manager, εάν αυτός υπάρχει. Επίσης μπορεί να είναι κοινή και για τον Bandwidth Broker και για ένα σύστημα διαχείρισης δικτύου (Network Management System - NMS) εάν υπάρχει κάποιο NMS.

Το Policy Manager interface χρησιμοποιείται για να παρέχει πλήρη έλεγχο αποδοχής. Υλοποιεί τον συντονισμό των συμβολαίων υπηρεσίας και των πόρων του δικτύου. Επίσης, με αυτό το interface παρέχονται οι AAA λειτουργίες.

Το Network Manager interface χρησιμοποιείται για το συντονισμό και την παρακολούθηση του δικτύου.

3.4 ΠΡΩΤΟΚΟΛΛΑ ΧΡΗΣΗΣ ΤΩΝ BANDWIDTH BROKER

Η διαχείριση του δικτύου που στηρίζεται σε policing εξαρτάται από πολλά διαφορετικά πρωτόκολλα.

Ο Bandwidth Broker για να υποστηρίξει τη διαχείριση αυτή μπορεί να χρησιμοποιεί τα εξής πρωτόκολλα ώστε να επικοινωνεί με τον router και να μπορεί να τον κάνει configure: COPS, ένα Command line interface, SNMP ή Diameter[29] .

Το πρωτόκολλο Diameter είναι ένα προτεινόμενο πρωτόκολλο που μπορεί να χρησιμοποιηθεί για αστυνόμευση(policy), AAA (Authentication, Authorization and

Accounting) και έλεγχο πόρων για εφαρμογές όπως πρόσβαση δικτύου και IP mobility. Το πρωτόκολλο έχει κατασκευαστεί για να συνεργάζεται με το πρωτόκολλο RADIUS[33], ένα πρωτόκολλο που έχει ήδη χρησιμοποιηθεί ευρέως.

Στις σύγχρονες υλοποιήσεις του Bandwidth Broker χρησιμοποιούνται ήδη τα πρωτόκολλα COPS, CLI και SNMP ενώ παραδείγματα πρωτοκόλλων τα οποία χρησιμοποιούνται για την επικοινωνία με τους hosts είναι τα: RSVP, COPS, CLI και Diameter. Τέλος ως intra-domain BB πρωτόκολλα μπορούν να χρησιμοποιηθούν τα: Diameter, COPS και άλλα.

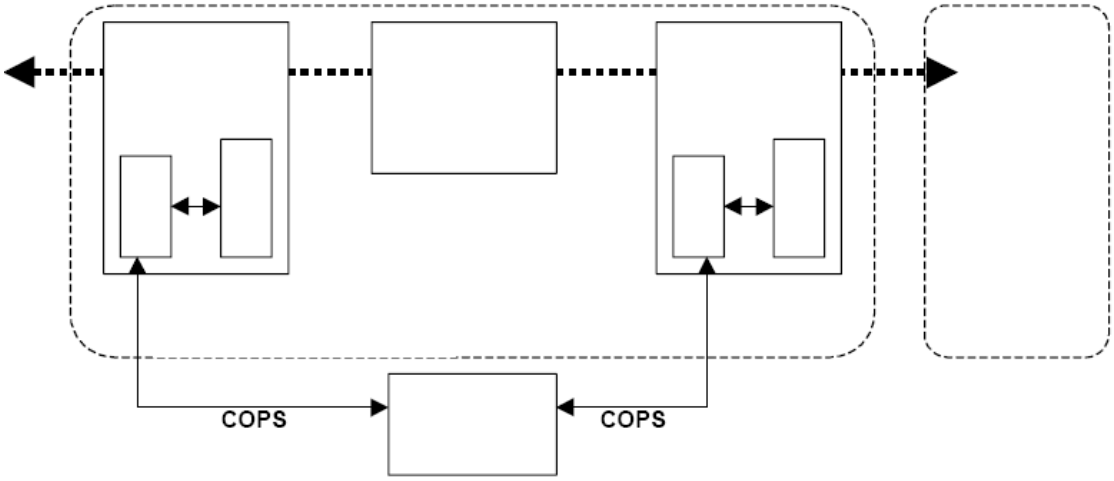
Το interdomain BB πρωτόκολλο ονομάζεται SIBBS(Simple Interdomain Bandwidth Broker Signalling)[30].Το πρωτόκολλο αποτελείται από ένα απλό πρωτόκολλο αίτησης-απάντησης ανάμεσα στους γειτονικούς bandwidth brokers και έχει ως αποστολή να μεταφέρει την απαραίτητη πληροφορία για την αίτηση μια υπηρεσίας γενικά. Το πρωτόκολλο είναι sender-oriented και αναμένεται στο μέλλον να επεκταθεί για να γίνει (προαιρετικά) receiver-oriented.

Το πρωτόκολλο SIBBS αποτελεί πρόταση ως ένα νέο QoS πρωτόκολλο σηματοδότησης που θα χρησιμοποιηθεί σε ερωτήσεις και απαντήσεις απλών συναθροιστικών αιτημάτων υπηρεσίας. Τα γειτονικά domain τα οποία ελέγχονται από τους bandwidth brokers πρέπει να είναι DiffServ (differentiated service). Αυτό που ορίζει το πρωτόκολλο είναι οι διεπαφές σηματοδότησης(signaling interfaces) ανάμεσα στα γειτονικά domains και όχι οι λεπτομέρειες των SLA και τα μέσα διαχείρισης των δικτυακών πόρων του εκάστοτε domain. Άλλωστε θα υπάρξει σημαντική ποικιλία των υλοποιήσεων και των στρατηγικών διαχείρισης πόρων πίσω από την ομοιόμορφη διεπαφή σηματοδότησης. Το πρωτόκολλο αυτό λοιπόν αλληλοσυνδέεται καλά με τις end-to-end δυνατότητες σηματοδότησης των hosts και είναι αρκετά απλό για την διευκόλυνση γρήγορης υλοποίησης ενώ παραμένει αρκετά εύκαμπτο για να υποστηρίξει μελλοντική βελτιστοποίηση απόδοσης και επέκταση πρωτοκόλλου.

3.4.1 Πρωτόκολλο COPS

3.4.1.1 Το βασικό μοντέλο COPS

Το βασικό μοντέλο του COPS φαίνεται στην Εικόνα 19. Η εικόνα επίσης περιγράφει το πλαίσιο εργασίας του policy-based admission control. Οι δικτυακοί κόμβοι μπορεί να είναι routers, switches ή hubs. Οι πόροι δεσμεύονται ή απελευθερώνονται μέσα σε ένα κόμβο από ένα PEP(Policy Enforcement Point). Ένας κόμβος μπορεί να είναι ένας κόμβος ο οποίος δεν υποστηρίζει το πρωτόκολλο COPS. Οι δικτυακοί κόμβοι οργανώνονται μέσα σε διαχειριστικά(administrative) domain. Μέσα στον policy server υπάρχει ένα PDP (Policy Decision Point), που παίρνει την τελική απόφαση σχετικά με τη διαχείριση των πόρων. Επίσης μπορεί να υπάρχει ένα τοπικό PDP μέσα σε ένα δικτυακό κόμβο.



Εικόνα 19 Το μοντέλο COPS

Το πρωτόκολλο COPS απασχολεί ένα client/server μοντέλο όπου το PEP στέλνει αιτήματα, ανανεώνει και διαγράφει στο PDP, και το PDP επιστρέφει αποφάσεις πίσω στον PEP. Εάν ο απομακρυσμένος PDP δεν είναι διαθέσιμος, για παράδειγμα εξαιτίας ενός δικτυακού σφάλματος, το PEP πρέπει να δοκιμάσει να συνδεθεί σε έναν απομακρυσμένο backup PDP ή να επιστρέψει σε έναν τοπικό PDP(LPDP). Παρόλα αυτά, όταν η σύνδεση σε έναν απομακρυσμένο PDP ανακτάται, το PEP πρέπει να ενημερώσει το PDP με τις αποφάσεις, που συνέβησαν τοπικά κατά την διάρκεια της αποσύνδεσης. Η σύνδεση ανάμεσα στον PEP και στον απομακρυσμένο PDP είναι αξιόπιστη, διότι το COPS χρησιμοποιεί το πρωτόκολλο TCP σαν το πρωτόκολλο μεταφοράς. Το PEP αρχίζει την TCP σύνδεση και μετά από αυτό, η επικοινωνία ανάμεσα στον PEP και στον απομακρυσμένο PDP είναι κυρίως ανταλλαγή κατάστασης από αίτημα σε απόφαση και το αντίστροφο.

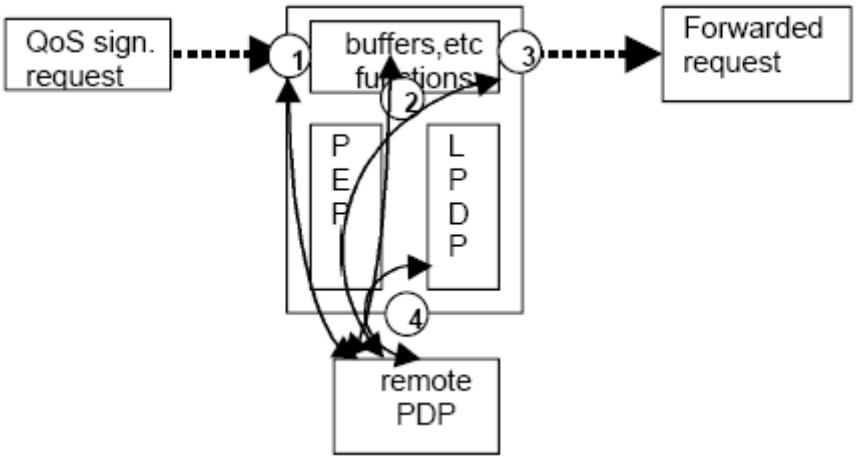
Το πρωτόκολλο COPS είναι πρωτόκολλο κατάστασης και αυτό για δύο κυρίως λόγους:

- Η κατάσταση αίτημα/απόφαση μοιράζεται ανάμεσα στον client και στον server. Αυτό σημαίνει ότι τα αιτήματα από τον client PEP εγκαθίστανται ή ανακαλούνται στην μνήμη από τον απομακρυσμένο PDP μέχρι που να διαγραφούν ρητά από τον PEP. Την ίδια στιγμή, οι αποφάσεις από τον απομακρυσμένο PDP μπορούν να παραχθούν ασύγχρονα οποιαδήποτε στιγμή για μια υπάρχουσα εγκατεστημένη κατάσταση αιτήματος.
- Κατάσταση από διαφορετικά γεγονότα(ζεύγη αιτήματος/απόφασης) μπορεί να αυτοσυσχετίζεται. Αυτό σημαίνει ότι ο server μπορεί να ανταποκριθεί σε νέες ερωτήσεις διαφορετικά, εξαιτίας προηγούμενης εγκατεστημένης κατάστασης αιτήματος/απόφασης η οποία συσχετίζεται με τα νέα αιτήματα.

Το PDP μπορεί επίσης να στείλει αζήτητες αποφάσεις στο PEP, για παράδειγμα το PDP μπορεί να αναγκάσει το PEP να αλλάξει προηγούμενες αποφάσεις. Αντιστοίχως, το PDP μπορεί να στείλει πληροφορία για σκοπούς που έχουν σχέση με λογαριασμούς ή προγράμματα ελέγχου στο PDP.

Το PEP ακολουθεί την απόφαση που παίρνει το PDP. Παρόλο που το PEP μπορεί να παίρνει τοπικές αποφάσεις με το LPDP, η τελική απόφαση παίρνεται από το απομακρυσμένο PDP. Υπάρχουν τρεις διαφορετικοί τύποι

Το PEP αναθέτει τη λήψη της απόφασης στο PDP. Παρόλο που ο PEP μπορεί να παίρνει τοπικές αποφάσεις με το LPDP, η τελική απόφαση παίρνεται από τον απομακρυσμένο PDP. Υπάρχουν τρεις διαφορετικοί τύποι για τα γεγονότα που έχουν ανατεθεί που χρειάζονται την απόφαση από το PDP και ένας τύπος που δεν ανατίθεται Εικόνα 20. Αυτά καθορίζουν το πλαίσιο του κάθε αιτήματος και της κάθε απόφασης. Το εισερχόμενο μήνυμα, για παράδειγμα ένα RSVP μήνυμα, προκαλεί το PEP να ρωτήσει το PDP, πως να διαχειριστεί το εισερχόμενο μήνυμα(1), να το αποδεχθεί ή να το απορρίψει, το επόμενο βήμα που πρέπει να γίνει είναι η δέσμευση τοπικών πόρων(2) με σκοπό την επεξεργασία του μηνύματος και τελικά το PEP ακολουθεί την προωθημένη απόφαση που έχει γίνει από το PDP(3).



Εικόνα 20 Διαδικασία αποδοχής αιτήματος στο COPS

Οι τύποι των γεγονότων είναι οι εξής:

- Η άφιξη ενός εισερχομένου μηνύματος
- η δέσμευση τοπικών πόρων
- η προώθηση ενός εξερχόμενου μηνύματος
- πληροφορία διαμόρφωσης

Ο τέταρτος τύπος γεγονότος, που δεν μετριέται ως ένα γεγονός που ανατίθενται, είναι ένα αίτημα για την πληροφορία διαμόρφωσης από τον απομακρυσμένο PDP. Το PEP μπορεί να κάνει ένα αίτημα, για παράδειγμα όταν ξεκινά. Το PEP μπορεί επίσης να ρωτήσει για την πληροφορία διαμόρφωσης αργότερα, για παράδειγμα , όταν το PEP προσέξει ότι πρέπει να εγκαταστήσει νέα οδηγία με σκοπό να διαχειριστεί ένα νέο module ή interface. Η πληροφορία διαμόρφωσης χρησιμοποιείται για να παίρνονται αποφάσεις τοπικά.

3.4.1.2 Το πρωτόκολλο COPS

Το πρωτόκολλο είναι σχεδιασμένο έτσι ώστε τα μηνύματα να είναι αυτοπροσδιοριζόμενα αντικείμενα αστυνόμευσης(policy). Τα στοιχεία αστυνόμευσης είναι μονάδες πληροφορίας απαραίτητα για την εκτίμηση κανόνων αστυνόμευσης.

Ένα μικρό στοιχείο αστυνόμευσης μπορεί να μεταφέρει τα διαπιστευτήρια ενός πελάτη ή την πληροφορία μιας πιστωτικής κάρτας. Τα στοιχεία αστυνόμευσης από μόνα τους αναμένονται να είναι ανεξάρτητα από το πρωτόκολλο QoS σηματοδότησης που χρησιμοποιείται.

3.4.1.2.1 Η επικεφαλίδα COPS

Το μήνυμα COPS ξεκινά πάντα με μια κοινή επικεφαλίδα Εικόνα 21

0		1	2	3
version	flags	Op Code	Client -type	
Message length				

Εικόνα 21 Η επικεφαλίδα COPS

Τα πεδία είναι τα εξής:

- Το πεδίο version(4 bits) δείχνει την έκδοση του πρωτοκόλλου COPS, η υπάρχουσα έκδοση είναι η 1.
- Flags:Το flag αυτό καθορίζεται όταν το μήνυμα ακολουθείται από ένα άλλο μήνυμα COPS
- Το πεδίο Op Code (8 bits) δείχνει τις λειτουργίες του πρωτοκόλλου που είναι:1 = Request (REQ), 2 = Decision (DEC), 3 = Report State (RPT), 4 = Delete Request State (DRQ), 5 = Synchronize State Req (SSQ), 6 = Client-Open (OPN), 7 = Client-Accept (CAT), 8 = Client-Close (CC), 9 = Keep-Alive (KA), 10= Synchronize Complete (SSC).
- Client type: Ο client-type αναγνωρίζει τον policy client. Η ερμηνεία όλων των encapsulated αντικειμένων σχετίζεται με τον client-type.
- Message length: Το μέγεθος του μηνύματος σε octets, που περιλαμβάνει την στάνταρ COPS επικεφαλίδα και όλα τα ενσωματωμένα αντικείμενα.

3.4.1.2.2 COPS specific object formats

Η επικεφαλίδα COPS καθορίζει τον τύπο του μηνύματος. Μετά την επικεφαλίδα υπάρχει ένας αριθμός από στοιχεία αστυνόμευσης που ακολουθούν, που χρησιμοποιούνται για την μεταφορά την πληροφορίας για το πάρσιμο της απόφασης και για τις ακριβείς αποφάσεις. Τα περιεχόμενα των στοιχείων αστυνόμευσης εξαρτώνται από το QoS πρωτόκολλο σηματοδότησης, αλλά η κύρια δομή του COPS πρωτοκόλλου παραμένει άθικτη. Τα στοιχεία αστυνόμευσης που καθορίζονται στο COPS πρωτόκολλο, διαμορφώνουν την παρακάτω δομή .

0	1	2	3
Length		C-Num	C-Type
(Object contents)			

Εικόνα 22 COPS policy elements

Το μέγεθος(length) είναι μια τιμή 2-octet που περιγράφει των αριθμό των octets(περιλαμβάνοντας και την επικεφαλίδα) που συνθέτει το αντικείμενο. Εάν το μέγεθος σε octets δεν πέφτει σε ένα 32-bit word όριο, πρέπει να προστεθεί ένα κομμάτι στο τέλος του αντικειμένου έτσι ώστε να είναι ευθυγραμμισμένο με το επόμενο 32-bit όριο.

Τυπικά, το C-Num(8 bits) αναγνωρίζει την κλάση της πληροφορίας που περιλαμβάνεται στο αντικείμενο, και το CType(8 bits) αναγνωρίζει τον υποτύπο ή την έκδοση της πληροφορίας που περιλαμβάνεται στο αντικείμενο.

Για παράδειγμα, με In Interface (C-Num = 3) και Out Interface (C-Num = 4) αντικείμενα C-Type = 1 σημαίνει ότι το interface είναι IPv4 και C-Type = 2 σημαίνει ότι το interface είναι IPv6.

Οι πιθανές τιμές των πεδίων C-Num και το τι σημαίνουν παρουσιάζονται στην Εικόνα 23.

C-Num	Name	Explanation
1	Handle	The Handle Object encapsulates a unique value that identifies an installed state. This identification is used by most COPS operations.
2	Context	Specifies the type of event(s) that triggered the query. Required for request messages.
3	In Interface	This object is used to identify the incoming interface on which a particular request applies and the address where the received message originated.
4	Out Interface	This object is used to identify the outgoing interface to which a specific request applies and the address for where the forwarded message is to be sent.
5	Reason Code	This object specifies the reason why the request state was deleted. It appears in the delete request (DRQ) message.
6	Decision	Decision made by the PDP. Appears in replies.
7	LPDP Decision	Decision made by the PEP's local policy decision point (LPDP). May appear in requests.
8	Error	This is used to identify a particular COPS protocol error.
9	Client Specific Info	This contains client-type specific information, e.g. the contents of RSVP Path-message, if the client is RSVP
10	Keep-Alive Timer	Time given in seconds
11	PEP Identification	The PEP Identification Object is used to identify the PEP client to the remote PDP.
12	Report Type	The Type of Report on the request state associated with a handle
13	PDP Redirect Address	A PDP when closing a PEP session for a particular client-type may optionally use this object to redirect the PEP to the specified PDP server address and TCP port number
14	Last PDP Address	When a PEP sends a Client-Open message for a particular client-type the PEP SHOULD specify the last PDP it has successfully opened (meaning it received a Client-Accept) since the PEP last rebooted.
15	Accounting Timer	Optional timer value used to determine the minimum interval between periodic accounting type reports.
16	Message Integrity	The integrity object includes a sequence number and a message digest useful for authenticating and validating the integrity of a COPS message.

Εικόνα 23

Κάθε COPS καθορισμένο αντικείμενο έχει αντίστοιχα περιεχόμενα στο αντικείμενο. Το περιεχόμενο αυτό είναι μια μεταβλητή μεγέθους που εξαρτάται από το C-Num και το C-Type. Τα μηνύματα ανάμεσα στο PEP και στο PDP κατασκευάζονται από τα αντικείμενα αυτά.

3.4.1.3 Ακεραιότητα μηνυμάτων

Τα μηνύματα Client-Open και Client-Accept χρησιμοποιούνται επίσης και στην διαπραγμάτευση της ασφάλειας, εάν είναι ρυθμισμένα έτσι. Εάν το PEP καθορίζεται να χρησιμοποιεί την ασφάλεια του COPS, το πρώτο Client-Open μήνυμα, που το PEP στέλνει στο PDP, πρέπει να έχει το πεδίο Client-Type στο μηδέν. Ως επιπλέον σε αυτό, το αντικείμενο Integrity πρέπει να περιληφθεί. Το αντικείμενο Integrity περιέχει ένα σειριακό αριθμό-πεδίο(32 bits).Το PEP αναθέτει στο πεδίο αυτό τον αρχικό σειριακό αριθμό, που το PEP αναμένει το PDP να αυξήσει, όταν η επικοινωνία συνεχιστεί μετά την αρχική Client-Open/Client-Accept ανταλλαγή. Το σύστημα χρησιμοποιείται για να σταματήσει επιθέσεις απάντησης. Τόσο το PEP όσο και το PDP περιμένουν να λάβουν ένα μήνυμα με συγκεκριμένο σειριακό αριθμό. Εάν ο σειριακός αριθμός του εισερχόμενου μηνύματος δεν είναι ο αναμενόμενος, ο λήπτης στέλνει ένα μήνυμα σφάλματος και κλείνει τη σύνδεση. Επίσης υπάρχουν άλλα δύο πεδία στο αντικείμενο-Integrity: KeyID (32 bits) και Keyed message digest (96 bits). Το πεδίο Key ID χρησιμοποιείται για να αναγνωρίσει το διαμοιραζόμενο κλειδί ανάμεσα στο PEP και στο PDP καθώς και ο κρυπτογραφικός αλγόριθμος που θα χρησιμοποιηθεί. Ο αλγόριθμος που το COPS χρησιμοποιεί για να υπολογίσει το digest είναι ο HMAC (Keyed-Hashing for Message Authentication). Ο ελάχιστος αριθμός που το COPS πρέπει να χρησιμοποιήσει είναι ο HMAC-MD5-96 αλγόριθμος. Ο HMAC υλοποιεί τον MD5 Message Digest αλγόριθμο. Η φυσιολογική 128-bit έξοδος τεμαχίζεται σε 96 bits Keyed Message Digest –field του Integrity αντικειμένου. Το digest υπολογίζεται για κάθε αντικείμενο και πεδίο στο μήνυμα εκτός από το Keyed Message Digest πεδίο. Αυτός είναι ο λόγος γιατί το αντικείμενο integrity πρέπει πάντα να είναι το τελευταίο αντικείμενο στο μήνυμα, εφόσον χρησιμοποιείται. Οι υλοποιήσεις του COPS πρέπει τουλάχιστον να παρέχουν την δυνατότητα να ρυθμίζεις χειροκίνητα τα κλειδιά και τις παραμέτρους τους τοπικά.

3.4.1.4 Πρωτόκολλο COPS για Policy Provisioning(COPS-PR)

Το COPS πρωτόκολλο επεκτάθηκε από το [32] το οποίο περιγράφει τη χρήση του πρωτοκόλλου COPS για την υποστήριξη του policy provisioning. Ο καθορισμός αυτός είναι ανεξάρτητος από τον τύπο της αστυνόμευσης που παρέχεται(QoS, Security, και άλλα). Κυρίως επικεντρώνεται στους μηχανισμούς και στις συμφωνίες που χρησιμοποιούνται για την επικοινωνία των PDPs και των PEPs.

Στο COPS_PR, τα αιτήματα αστυνόμευσης περιγράφουν το PEP και τις παραμέτρους που πρέπει να καθοριστούν. Εάν μια αλλαγή συμβεί σε αυτές τις βασικές παραμέτρους, ένα ενημερωμένο μήνυμα στέλνεται. Για αυτό, τα αιτήματα εκπέμπονται όχι συχνά. Οι αποφάσεις δεν αντικατοπτρίζονται ακριβώς στα αιτήματα, αλλά παίρνονται κυρίως όταν το PDP ανταποκρίνεται σε εξωτερικά γεγονότα ή σε PDP γεγονότα(policy/SLA updates).

3.4.1.5 Υλοποιήσεις COPS

Στη συνέχεια παρουσιάζονται διάφορες υλοποιήσεις του πρωτοκόλλου COPS.

3.4.1.5.1 COPS-PR - Software Development Kit (COPS-PR SDK)

Η εταιρία WebSpectrum[35] έχει αναπτύξει ένα σύνολο από προϊόντα με σκοπό την ενεργοποίηση QoS based υπηρεσιών, στηριζόμενες στο Policy Based Management Framework της IETF,που παρέχει μια βάση για λύση που στηρίζονται σε στάνταρτ.

Έχει αναπτύξει ένα Software Development Kit (SDK) που μπορεί να χρησιμοποιηθεί για την ανάπτυξη Policy Decision Point (PDP) και Policy Enforcement Point (PEP).

Το ποιόν αποτελείται από ένα PIB-to-C++ μεταφραστή και δύο σύνολα από βιβλιοθήκες κλάσεων, μία για την κατασκευή PDP και μία για PEP

3.4.1.5.2 The INTEL COPS Client SDK

Το Intel COPS Client SDK version 3.1[36] δίνει τη δυνατότητα στους developers να χρησιμοποιήσουν το στάνταρ μοντέλο δεδομένων στο COPS με το PIB Control Interface Generator.

Η βασική ιδέα πίσω από το Intel COPS Client SDK είναι να δώσει στους διαχειριστές, σχεδιαστές και κατασκευαστές έναν προτυποποιημένο τρόπο για να καθορίσεις τις δικτυακές συσκευές και να υλοποιήσεις υπηρεσίες σε αυτές. Με το Intel COPS Client SDK, μία δικτυακή συσκευή μπορεί να υποστηρίξει Differentiated Services (DiffServ), VPN, QoS, ασφάλεια, αυθεντικοποίηση, λογαριασμούς, καθώς και άλλες δικτυακές υπηρεσίες απλά τρέχοντας το PIB Control Interface Generator με τις νέες προτυποποιημένες υπηρεσίες που στηρίζονται στα PIB

modules και ενοποιώντας τα με τις αντίστοιχες δικτυακές συσκευές. Η προσέγγιση αυτή διαφέρει πολύ από την κλασική μέθοδο της εκτέλεσης scripts με τη χρήση telnet ή με τη χρήση ενός command-line interface.

3.4.1.5.3 Common Open Policy Server (COPS) Stack

Το [34] κατασκεύασε το Common Open Policy Server (COPS) Stack.

Το Common Open Policy Server (COPS) Stack μπορεί να χρησιμοποιηθεί για την κατασκευή εφαρμογών όπως:

1. Υλοποίηση policy based QoS framework. Για παράδειγμα ένας policy client είναι ένας RSVP router που πρέπει να εκτελέσει policy-based admission control πάνω σε ένα RSVP usage.
2. Υλοποίηση συναρτήσεων του επιπέδου εφαρμογών AAA (Authorization, Authentication and Accouting) σε IP τηλεφωνικό δίκτυο για interdomain κλήσεις. Για παράδειγμα το COPS σε συνεργασία με το OSP (Open Sattelment Protocol) για τη χρήση third party services όπως clearing-houses για την υλοποίηση AAA συναρτήσεων.

3.4.1.6 Σύνοψη

Το πρωτόκολλο COPS ελέγχει κεντρικοποιημένα το QoS της κίνησης των δεδομένων μέσα στο διαχειριζόμενο domain. Η αντικειμενοστραφής σύλληψη του πρωτοκόλλου

COPS διευκολύνει στην πρόσθεση νέων πρωτοκόλλων, client τύπων στο πρωτόκολλο COPS εάν είναι απαραίτητο, επειδή τα περιεχόμενα μεταφέρονται σαν εμφωλευμένα αντικείμενα μέσα στα μηνύματα. Το πρωτόκολλο COPS είναι αξιόπιστο, επειδή χρησιμοποιεί TCP συνδέσεις ανάμεσα στο δικτυακό κόμβο και στο COPS server.

ΚΕΦΑΛΑΙΟ 4: Ο ΕΞΟΜΟΙΩΤΗΣ NS

Ο ΕΞΟΜΟΙΩΤΗΣ NS

4.1 Ο ΕΞΟΜΟΙΩΤΗΣ NS

Η υλοποίηση του Bandwidth Broker έγινε στον εξομοιωτή Network Simulator ns-2[4]. Ο ns-2 είναι ένας εξομοιωτής ανοιχτού κώδικα. Ο ns ξεκίνησε το 1989 ως μια παραλλαγή του Real network simulator και το 1995 η ανάπτυξή του υποστηρίχτηκε από την DARPA μέσα από το έργο VINT. Το VINT είναι ένα έργο στο οποίο συνεργάζονται οι εταιρίες USC/ISI, Xerox PARC, LBNL και το πανεπιστήμιο Berkeley της California. Στις μέρες μας, η υποστήριξη του ns γίνεται από την DARPA μέσα από το SAMAN και από την NSF μέσα από το CONSER. Ο ns διατίθεται στη διεύθυνση [4]όπου μπορεί οποιοσδήποτε να τον κατεβάσει. Ο ns βγαίνει σε διάφορες εκδόσεις για διάφορα λειτουργικά συστήματα όπως τα FreeBSD, Linux, SunOS, Solaris και Windows. Η τελευταία του έκδοση είναι η 2.29.

Ο ns-2 είναι ένα πολύ ισχυρό περιβάλλον εξομοίωσης το οποίο μπορεί να εξομοιώσει πολλά είδη δικτύων, όπως ασύρματα και δορυφορικά δίκτυα. Ένας χρήστης μπορεί να ορίζει τυχαίες τοπολογίες δικτύων που αποτελούνται από κόμβους και συνδέσμους. Στη συνέχεια μπορεί να αναθέτει εφαρμογές σε κάθε κόμβο καθώς και ουρές για τους συνδέσμους. Ένας ερευνητής μπορεί να χρησιμοποιήσει τον ns-2 ώστε να σχεδιάσει νέα πρωτόκολλα, να ελέγξει την συμπεριφορά και την απόδοσή τους και να τα συγκρίνει με τα ήδη υπάρχοντα πρωτόκολλα. Ο ns-2 μπορεί να χρησιμοποιηθεί και για εκπαιδευτικούς σκοπούς διότι μπορεί να οπτικοποιήσει τα δίκτυα και τις συμπεριφορές των πρωτοκόλλων, όπως για παράδειγμα του TCP ή του UDP, σαν animations.

Ο ns-2 έχει γραφτεί σε C++ και χρησιμοποιεί την OTcl σαν γλώσσα εντολών και ρύθμισης των παραμέτρων. Ο ns-2 υποστηρίζει πάρα πολλά δικτυακά πρωτόκολλα, όπως τα TCP και UDP, πολλές πηγές κίνησης όπως FTP, Telnet, Web, CBR και VBR, μηχανισμούς διαχείρισης ουρών όπως τους RED, DropTail και CBQ, αλγορίθμους δρομολόγησης όπως τους Dijkstra και Bellman Ford. Επίσης, ο ns-2 υποστηρίζει σφάλματα της λειτουργίας του δικτύου, όπως ντετερμινιστικές και πιθανοτικές απώλειες πακέτων καθώς και link failures. Ο ns-2 είναι ένας event-driven εξομοιωτής που δέχεται σαν είσοδο tcl scripts και τα εκτελεί. Η έξοδος μπορεί να έχει διάφορες μορφές, ακόμα και γραφική αναπαράσταση του δικτύου. Μια κοινή έξοδος μπορεί να είναι αρχεία τα οποία περιγράφουν με πληρότητα την κίνηση σε ένα σύνδεσμο ή την κατάσταση σε μια ουρά. Με την μετα-επεξεργασία αυτών των αρχείων μπορούν να υπολογιστούν ποσότητες όπως το throughput ή το jitter της κίνησης.

Η έκδοση του ns που χρησιμοποιήθηκε ήταν η ns-allinone-2.26. Αυτή η έκδοση περιείχε τον ns, έκδοση 2.26, τη γλώσσα OTcl καθώς και μερικά ακόμα πακέτα που προσθέτουν λειτουργικότητα. Τα σημαντικότερα από αυτά είναι:

- Το πακέτο nam. Το πακέτο nam δίνει τη δυνατότητα γραφικής αναπαράστασης του δικτύου και των πακέτων που μεταφέρονται μέσα σε αυτό.
- Το πακέτο xgraph. Αυτό το πακέτο χρησιμοποιείται για τη δημιουργία γραφικών παραστάσεων και είναι πολύ χρήσιμο κατά τη μετα-επεξεργασία της εξόδου. Για παράδειγμα μπορεί να αναπαραστήσει το throughput μιας εφαρμογής.

- Τη γλώσσα Perl. Η Perl μπορεί να χρησιμοποιηθεί για την περιγραφή πειραμάτων στον ns-2.
- Το πακέτο tcl-debug που χρησιμοποιείται για debugging.
- Τα πακέτα gt-itm και sgb. Το πακέτο gt-itm χρησιμοποιείται για τη δημιουργία διαφόρων τοπολογιών δικτύων. Το πακέτο sgb χρησιμοποιείται για τη μετατροπή της εξόδου του gt-itm σε μορφή συμβατή με αυτήν των τοπολογιών ns-2.
- Το πακέτο CWeb χρησιμοποιείται για τη δημιουργία ευανάγνωστων C και C++ προγραμμάτων.
- Το πακέτο Zlib χρησιμοποιείται για συμπίεση αρχείων.

Κατά την υλοποίηση του bandwidth broker χρησιμοποιήθηκε ιδιαίτερα το πακέτο diffserv του ns-2. Αυτό το πακέτο παρέχει αρκετούς DiffServ μηχανισμούς. Η DiffServ λειτουργία στον ns-2 μπορεί να υποστηρίξει τέσσερις κλάσεις κίνησης κάθε μία από τις οποίες έχει τρεις προτεραιότητες απόρριψης. Τα πακέτα που ανήκουν σε κάποια κλάση τοποθετούνται στην αντίστοιχη φυσική RED ουρά η οποία περιέχει τρεις ιδεατές ουρές, μία για κάθε προτεραιότητα απόρριψης. Το diffserv πακέτο του ns υλοποιεί τρεις κύριες λειτουργίες:

- Αστυνόμευση
- Λειτουργίες των edge routers, δηλαδή μαρκάρισμα των πακέτων
- Λειτουργίες των core routes, δηλαδή έλεγχο των code points των πακέτων και κατάλληλη προώθησή τους

Το πακέτο diffserv διαθέτει την κλάση dsREDQueue η οποία χρησιμοποιείται για τη δημιουργία και την διαχείριση ουρών που διαθέτουν τον μηχανισμό RED. Οι κλάσεις edgeQueue και coreQueue κληρονομούν την κλάση dsREDQueue και αντιστοιχούν στις ουρές των edge και core routers. Η αστυνόμευση υλοποιείται στην κλάση dsPolicy. Η αστυνόμευση γίνεται με βάση το ζεύγος πηγής-προορισμού. Όλες οι ροές που διαθέτουν κοινό ζεύγος πηγής προορισμού συνενώνονται σε ένα traffic aggregate. Για κάθε διαφορετικό traffic aggregate ορίζεται ένας τύπος αστυνόμευσης, οι παράμετροι του τύπου αστυνόμευσης καθώς και ένα μοναδικό code point. Οι μηχανισμοί αστυνόμευσης που έχουν υλοποιηθεί είναι οι:

- Time Sliding Window with 2 Color Marking
- Time Sliding Window with 3 Color Marking [7]
- Token Bucket
- Single Rate Three Color Marker [5]
- Two Rate Three Color Marker [6]

Όλοι οι παραπάνω μηχανισμοί χρησιμοποιούν έναν αριθμό από προτεραιότητες απόρριψης και μια σειρά παραμέτρων όπως η μέγιστη επιτρεπτή συχνότητα μετάδοσης και το μέγιστο μέγεθος καταιγισμού.

Όσον αφορά τη χρονοδρομολόγηση στις ουρές, ο ns-2 διαθέτει τους μηχανισμούς Priority Queueing, Round Robin, Weighted Round Robin, Weighted Interleaved Round Robin και Deficit Round Robin. Ο προεπιλεγμένος μηχανισμός είναι ο Round Robin.

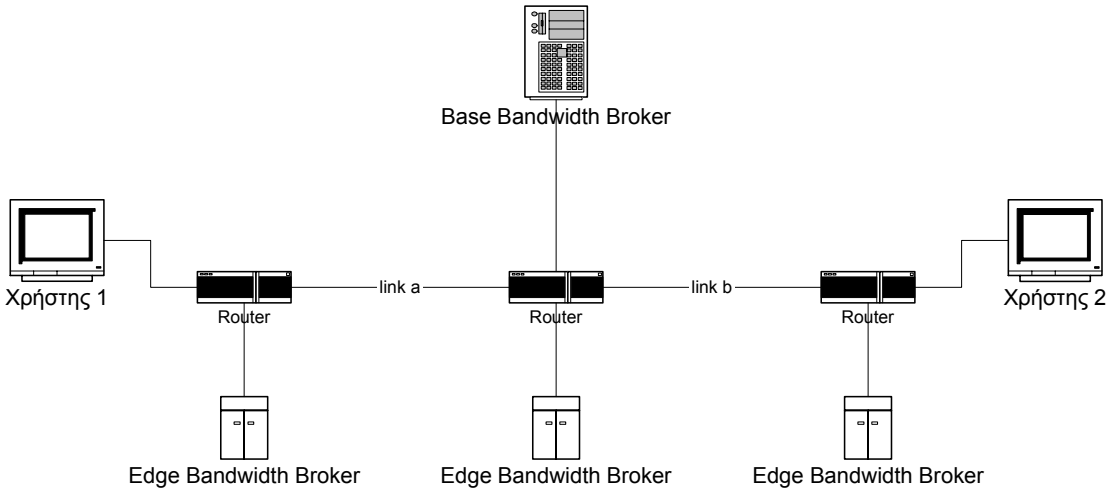
Ο εξομοιωτής ns-2 ανανεώνεται πολύ συχνά και πολλοί άνθρωποι σε όλο τον κόσμο τον υποστηρίζουν είτε με το να διορθώνουν bugs είτε με το να γράφουν νέο κώδικα που προσθέτει λειτουργικότητα στον ns-2. Η επιστημονική κοινότητα τον χρησιμοποιεί ευρύτατα καθώς είναι ένα πολύ ισχυρό και εύκολο στη χρήση περιβάλλον εξομοίωσης.

ΚΕΦΑΛΑΙΟ 5: ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ BANDWIDTH BROKERS

ΣΧΕΔΙΑΣΗ ΚΑΙ ΥΛΟΠΟΙΗΣΗ BANDWIDTH BROKERS

5.1 Η ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΤΩΝ BANDWIDTH BROKER

Οι bandwidth broker που υλοποιήθηκαν στηρίζονται σε agents, τον base bandwidth broker και τον edge bandwidth broker οι οποίοι είχαν υλοποιηθεί στο [8]Σε κάθε κόμβο του δικτύου ανατίθεται ένας edge bandwidth broker. Ο base bandwidth broker ανατίθεται μόνο σε έναν κόμβο. Σε αυτόν τον κόμβο μπορεί να τρέχει και κάποιος edge bandwidth broker(Εικόνα 24) ή μπορεί να είναι ένας επιπλέον κόμβος ο οποίος να συνδέεται με όλους τους άλλους κόμβους του δικτύου (Εικόνα 25).



Εικόνα 24 Το δίκτυο αποτελείται από τρεις κόμβους, σε κάθε έναν ανατίθεται και ένας Edge Bandwidth Broker και σε οποιοδήποτε από αυτούς τους κόμβους μπορεί να τοποθετηθεί και ο Base Bandwidth Broker

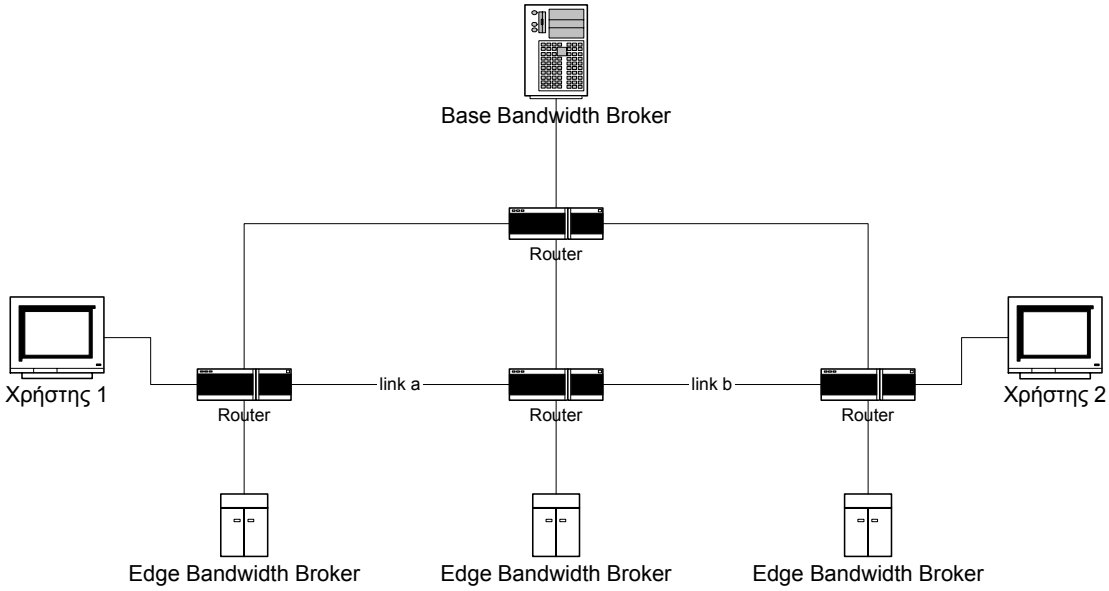
Τέσσερις διαφορετικοί bandwidth brokers υλοποιήθηκαν:

- Serial Distributed Bandwidth Broker model (SDBB model)
- Parallel Distributed Bandwidth Broker model (PDBB model)
- Centralized Bandwidth Broker model (CBB model)
- Centralized Fault-tolerant Bandwidth Broker model (CFBB model)

Ο base bandwidth broker εκτελεί διαφορετική λειτουργία ανάλογα με το μοντέλο ενώ ο edge bandwidth broker είναι κοινός σε όλα τα μοντέλα.

Τα κοινά στοιχεία των base bandwidth broker είναι ότι όλοι ανεξαρτήτως αρχιτεκτονικής συνδέονται με όλους τους edge bandwidth brokers. Η λειτουργία του συστήματος γίνεται με χρήση μηνυμάτων τα οποία πάντα αποστέλλονται είτε από κάποιον edge bandwidth broker προς τον base bandwidth broker, είτε από τον base bandwidth broker προς κάποιον από τους edge bandwidth brokers. Ποτέ δηλαδή δεν

επικοινωνούν δύο edge bandwidth brokers στέλνοντας μηνύματα μεταξύ τους. Στην επικοινωνία πάντα παρεμβάλλεται ο base bandwidth broker.



Εικόνα 25 Σε κάθε κόμβο τρέχει ένας Edge Bandwidth Broker ενώ έχει προστεθεί και ένας επιπλέον κόμβος (ο ανώτερος κόμβος) στον οποίο τρέχει ο Base Bandwidth Broker.

Η λειτουργία του συστήματος ξεκινά όταν κάποιος από τους edge bandwidth brokers κάνει ένα αίτημα ζητώντας εγγυημένο bandwidth x bytes για ένα συγκεκριμένο χρονικό διάστημα από τον κόμβο στον οποίο τρέχει ως κάποιον άλλο κόμβο του δικτύου.

Εάν ο base bandwidth broker εξυπηρετεί εκείνη τη στιγμή κάποιο άλλο αίτημα θα έχουμε διαφορετική αντιμετώπιση ανάλογα με την αρχιτεκτονική.

Στο CBB και στο CFBB μοντέλο στέλνει απάντηση ειδοποίησης μη διαθεσιμότητας στον συγκεκριμένο edge bandwidth broker. Αλλιώς, ξεκινά την εξυπηρέτηση του αιτήματος. Επειδή όμως η απόκριση των base agents ήταν άμεση ανεξαρτήτως του αριθμού των αιτημάτων που έφταναν, προέκυψε η περίπτωση απόρριψης αιτημάτων σε πολύ μικρό ποσοστό.

Αντίθετα στο SDBB και στο PDBB μοντέλο επειδή η απόρριψη ήταν μεγάλη λόγω μη διαθεσιμότητας του Base agent, υλοποιήθηκε ένα buffer στο οποίο αποθηκεύονται με σειριακό τρόπο τα αιτήματα που δεν μπορούσαν να εξυπηρετηθούν. Μόλις το αίτημα έβγαινε από το buffer , ελέγχονταν αν η έναρξη του αιτήματος δεν είχε έρθει ακόμη και αν δεν είχε έρθει όντως, ο base το επεξεργάζονταν . Διαφορετικά απορρίπτονταν.

Όταν ο base bandwidth broker εξυπηρετεί ένα αίτημα, ελέγχει αν υπάρχει διαθέσιμο bandwidth από τον κόμβο στον οποίο τρέχει ο edge bandwidth broker που έκανε το αίτημα μέχρι τον άλλο τελικό κόμβο για το συγκεκριμένο χρονικό διάστημα.

Ανάλογα με το μοντέλο η πληροφορία για το διαθέσιμο bandwidth βρίσκεται είτε στους edge είτε στους ίδιους τους base agents

Στα SDBB ,PDBB και CFBB μοντέλα κάθε edge bandwidth broker διατηρεί πληροφορία για το διαθέσιμο bandwidth μεταξύ του κόμβου στον οποίο τρέχει και

όλων των γειτονικών του κόμβων. Ο base bandwidth broker βρίσκει από τα routing tables το next hop n_1 από τον κόμβο n_0 που έκανε το αίτημα ως τον άλλο τελικό κόμβο n_k.

Στο πρώτο μοντέλο, στέλνει ερώτημα στον edge bandwidth broker που τρέχει στον κόμβο n_0 για το αν υπάρχει διαθέσιμο bandwidth μεταξύ των κόμβων n_0 και n_1. Αν η απάντηση είναι θετική, ο base bandwidth broker υπολογίζει το next hop n_2 από τον n_1 ως τον n_k και στέλνει στον n_1 ερώτημα για το αν υπάρχει διαθέσιμο bandwidth μεταξύ των κόμβων n_1 και n_2. Αν όλες οι απαντήσεις είναι θετικές, αυτή η διαδικασία συνεχίζεται ως ότου φτάσουμε στον κόμβο n_k. Αυτό θα σημαίνει ότι το διαθέσιμο bandwidth από τον n_0 ως τον n_k υπάρχει. Ο base bandwidth broker τότε θα στείλει θετική απάντηση στον edge bandwidth broker που έκανε αρχικά το αίτημα. Η διαδικασία θα ολοκληρωθεί με την αποστολή από τον base bandwidth broker μηνυμάτων προς όλους τους edge bandwidth brokers που βρίσκονται στο μονοπάτι $n_0,n_1,\ldots,n_k$ τα οποία θα τους ειδοποιούν να μειώσουν το διαθέσιμο bandwidth κατά x bytes στα links τα οποία βρίσκονται στο μονοπάτι. Στην περίπτωση που κάποιος από τους edge bandwidth brokers στείλει αρνητική απάντηση διότι δεν υπάρχει διαθέσιμο bandwidth σε κάποιο link, ο base bandwidth broker στέλνει αρνητική απάντηση στον κόμβο από τον οποίο έγινε το αρχικό αίτημα και η διαδικασία τελειώνει εκεί.

Στο PDBB μοντέλο η ερώτηση στους edge agents γίνεται ταυτόχρονα σε όλους. Αν στείλει κάποιος αρνητική απάντηση καθώς ο base agent περιμένει τις απαντήσεις, τότε η επεξεργασία του αιτήματος σταματά, στέλνεται αρνητική απάντηση στον αιτούντα και συνεχίζει την επεξεργασία του επόμενου αιτήματος. Διαφορετικά αν παίρνει θετικές απαντήσεις, περιμένει να του απαντήσουν όλοι οι agents τους οποίους ρώτησε. Αν λάβει από όλους θετική απάντηση ακολουθείται η διαδικασία που έγινε και παραπάνω.

Στο CFBB μοντέλο η πληροφορία είναι αποθηκευμένη και στα δύο σημεία, στους edge agents και στο ίδιο τον base agent. Όταν έρχεται το αίτημα, ο base agent ρωτά την βάση του και αν πάρει θετική απάντηση, αλλάζει τα δεδομένα της, αλλάζει και τα δεδομένα στους edge agents και στέλνει θετική απάντηση στον αιτούντα. Εάν πάρει αρνητική απάντηση, απλά την στέλνει πίσω στον αιτούντα.

Τέλος στο CBB μοντέλο όλη η πληροφορία βρίσκεται στο base agent. Όταν παίρνει ένα αίτημα ρωτά τη βάση του, που έχει όλη την πληροφορία για την κατάσταση του δικτύου που ελέγχει. Αν πάρει θετική απάντηση, αλλάζει τα δεδομένα της βάση του και στέλνει θετική απάντηση στον αιτούντα ενώ διαφορετικά στέλνει αρνητική και προχωρά στην επεξεργασία του επόμενου αιτήματος.

Η κατανομή των πόρων ώστε να παρέχονται εγγυήσεις για το bandwidth γίνεται χρησιμοποιώντας την DiffServ αρχιτεκτονική. Ο base bandwidth broker όταν στέλνει στον edge bandwidth broker μια θετική απάντηση, στο ίδιο μήνυμα στέλνει και το DSCP που πρέπει να φέρουν τα πακέτα της εφαρμογής η οποία θα χρησιμοποιεί το δεσμευμένο bandwidth. Προηγουμένως έχει κάνει configure όλους τους δρομολογητές του δικτύου ώστε να προσφέρουν εγγυήσεις στην εφαρμογή που έκανε το αίτημα για το ζητούμενο bandwidth, με βάση το DSCP.

5.2 ΟΙ AGENTS BBEDGE, BBBASE, CENTRALBBBASE, DISTRIBUTEDBBBASE

Οι agents base bandwidth broker και edge bandwidth broker οι οποίοι αναφέρθηκαν στην περιγραφή της λειτουργίας του συστήματος υλοποιήθηκαν με τα ονόματα BBbase, CentralBBbase, DistributedBBbase και BBedge αντίστοιχα.

Η κλάση BBbaseAgent κληρονομεί τις ιδιότητες της κλάσης Agent. Η κλάση BBbaseAgent έχει τρεις public συναρτήσεις-μέλη και μια protected μεταβλητή-μέλος. Συγκεκριμένα ορίζονται:

- ο constructor της κλάσης
- η συνάρτηση-μέλος command που χρησιμοποιείται για να μπορούν να δοθούν στον agent εντολές μέσω tcl
- η συνάρτηση-μέλος recv χρησιμοποιείται για να καθοριστεί η διαδικασία που θα ακολουθηθεί με την παραλαβή του κάθε πακέτου
- η δυαδική μεταβλητή-μέλος available που χρησιμοποιείται για να δείξει εάν ο base bandwidth broker είναι διαθέσιμος ή εξυπηρετεί ήδη κάποιο αίτημα για bandwidth
- η δυαδική μεταβλητή-μέλος cur_dscp χρησιμοποιείται για να διατηρεί ο base bandwidth broker πληροφορία για το τρέχον διαθέσιμο DSCP που θα ανατεθεί σε κάποια ροή μετά από μια θετική απάντηση από τον έλεγχο αποδοχής.

Όταν δημιουργείται ένας BBbaseAgent χρειάζεται η μεταβλητή-μέλος available να γίνει TRUE αφού ο BBbaseAgent είναι αρχικά διαθέσιμος. Η άλλη μεταβλητή-μέλος cur_dscp χρειάζεται να γίνει 2 αφού αυτό είναι το αρχικό διαθέσιμο DSCP. Αυτές οι αναθέσεις γίνονται στον constructor της κλάσης.

Και η κλάση BBedgeAgent κληρονομεί τις ιδιότητες της κλάσης Agent. Η κλάση BBedgeAgent έχει τέσσερις public συναρτήσεις-μέλη και μια protected μεταβλητή-μέλος. Συγκεκριμένα, ορίζονται:

- ο constructor της κλάσης
- η συνάρτηση-μέλος command χρησιμοποιείται για να μπορούν να δοθούν στον agent εντολές μέσω tcl
- η συνάρτηση-μέλος recv χρησιμοποιείται για να καθοριστεί η διαδικασία που θα ακολουθηθεί με την παραλαβή του κάθε πακέτου
- η συνάρτηση sendto η οποία χρησιμοποιείται για την αποστολή αιτήματος για bandwidth
- η μεταβλητή-μέλος nghbr_list χρησιμοποιείται για να διατηρείται σε κάθε BBedgeAgent πληροφορία για το διαθέσιμο bandwidth μεταξύ του κόμβου στον οποίο τρέχει ο BBedgeAgent και των γειτονικών του κόμβων
- η συνάρτηση inform χρησιμοποιείται για την εισαγωγή της πληροφορίας για την κατάσταση του δικτύου στο χρόνο μηδέν για τα CBB και CFBB μοντέλα.

Η κλάση CentralBBbaseAgent κληρονομεί τις ιδιότητες της κλάσης BBbaseAgent. Η κλάση CentralBBbaseAgent έχει τρεις public συναρτήσεις-μέλη και μια protected μεταβλητή-μέλος. Συγκεκριμένα ορίζονται:

- ο constructor της κλάσης
- η συνάρτηση-μέλος recv χρησιμοποιείται για να καθοριστεί η διαδικασία που θα ακολουθηθεί με την παραλαβή του κάθε πακέτου
- οι υπόλοιπες μεταβλητές κληρονομούνται

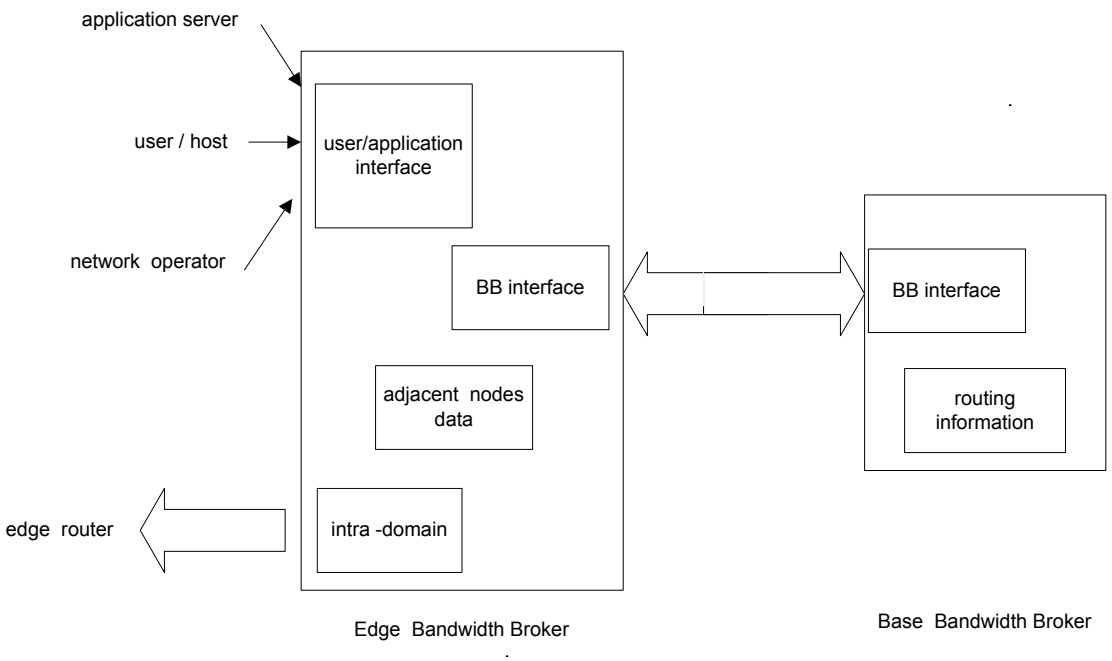
Η κλάση DistributedBBbaseAgent κληρονομεί τις ιδιότητες της κλάσης BBbaseAgent. Η κλάση CentralBBbaseAgent έχει τρεις public συναρτήσεις-μέλη και μια protected μεταβλητή-μέλος. Συγκεκριμένα ορίζονται:

- ο constructor της κλάσης
- η συνάρτηση-μέλος recv χρησιμοποιείται για να καθοριστεί η διαδικασία που θα ακολουθηθεί με την παραλαβή του κάθε πακέτου
- οι υπόλοιπες μεταβλητές κληρονομούνται

Όταν ορίζεται στον ns ένας νέος agent, χρειάζεται να καθοριστεί ο τύπος των πακέτων που θα διαχειρίζεται. Τα πακέτα που διαχειρίζονται οι BBedgeAgent και BBbaseAgent,CentralBBbaseAgent,DistributedBBbaseAgent είναι δύο νέοι τύποι πακέτων, τα πακέτα BBB και BBE, που δημιουργήθηκαν για να εξυπηρετήσουν τη λειτουργικότητα του bandwidth broker.

5.3 Η ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΤΟΥ BANDWIDTH BROKER

Η αρχιτεκτονική των bandwidth broker που υλοποιήθηκαν εμφανίζεται στην Εικόνα 26.



Εικόνα 26 Η αρχιτεκτονική του bandwidth broker που υλοποιήθηκε

Όπως είπαμε, οι bandwidth broker που υλοποιήθηκαν αποτελούνται από τον κεντρικό base bandwidth broker και τους κατανεμημένους edge bandwidth brokers. Όπως φαίνεται στην παραπάνω εικόνα, ο base bandwidth broker αποτελείται από δύο λειτουργικές μονάδες. Αυτές είναι:

- Το BB interface
- Το routing information interface

Ο edge bandwidth broker αποτελείται από τέσσερις λειτουργικές μονάδες. Αυτές είναι:

- Το user/application interface
- Το BB interface
- Το intra-domain interface
- Το adjacent nodes interface

Στη συνέχεια θα αναλύσουμε τη λειτουργία του κάθε interface.

5.3.1 BB Interface

Ολόκληρη η λειτουργία των bandwidth broker βασίζεται στο BB interface. Το BB interface είναι το πρωτόκολλο επικοινωνίας μεταξύ των διαφορετικών base bandwidth broker και κάποιου edge bandwidth broker.

Δημιουργήθηκαν δύο τύποι πακέτων για το BB interface, ο τύπος BBE και ο τύπος BBB. Οι διαφορετικοί base bandwidth broker agents δημιουργούν BBB πακέτα και λαμβάνουν BBE πακέτα. Οι edge bandwidth broker agents δημιουργούν BBE πακέτα και λαμβάνουν BBB πακέτα. Και οι δύο τύποι πακέτων έχουν την ίδια επικεφαλίδα,

δηλαδή οι επικεφαλίδες και των δύο τύπων αποτελούνται από ακριβώς τα ίδια πεδία. Η δημιουργία όμως δύο τύπων πακέτων και όχι ενός επιτρέπει την τοποθέτηση σε κάποιον κόμβο τόσο ενός base bandwidth broker agent όσο και ενός edge bandwidth broker agent. Με αυτόν τον τρόπο δεν απαιτείται ένας επιπλέον κόμβος για την τοποθέτηση του κεντρικού Bandwidth Broker. Το μέγεθος των πακέτων BBE και BBB καθορίστηκε να είναι 64 bytes. Επίσης, στα BBE και BBB πακέτα όλη η πληροφορία που είναι απαραίτητη για τη λειτουργία του BB interface βρίσκεται στην επικεφαλίδα των πακέτων. Δηλαδή τα πακέτα δεν περιλαμβάνουν δεδομένα.

Στον ns για να δημιουργηθεί ένας τύπος πακέτου χρειάζεται να δηλωθεί η επικεφαλίδα του. Η επικεφαλίδα του BBE (ομοίως και του BBB) πακέτου ορίστηκε να έχει τα πεδία :

- type
- answer
- bndw_request
- sender
- endnode
- hop_num
- DSCP
- Start_time
- Stop_time
- status

Ας αναλύσουμε τώρα το κάθε πεδίο της επικεφαλίδας, δηλαδή τι εκφράζει, και από ποιον καθορίζεται.

Το πεδίο type της επικεφαλίδας καθορίζει τον τύπο του πακέτου. Έχουν οριστεί οι ακόλουθοι τύποι:

- Ο τύπος REQUEST δηλώνει πακέτο αιτήματος για bandwidth. Πακέτα τύπου REQUEST στέλνονται τόσο από κάποιον από τους edge bandwidth broker agents όταν στέλνει ένα αρχικό αίτημα για bandwidth όσο και από τον base bandwidth broker agent κατά την εξυπηρέτηση του αιτήματος στον έλεγχο αποδοχής.
- Ο τύπος ANSWER δηλώνει πακέτο απάντησης σε κάποιο αίτημα. Πακέτα τύπου ANSWER στέλνονται τόσο από τον base bandwidth broker agent όσο και από κάποιον από τους edge bandwidth broker agents.
- Ο τύπος FIX_BWD δηλώνει πακέτο εντολής μείωσης του διαθέσιμου bandwidth σε κάποιο link. Πακέτα τύπου FIX_BWD στέλνονται μόνο από τον base bandwidth broker agent.
- Ο τύπος UNAVAILABLE δηλώνει μη διαθεσιμότητα του bandwidth broker. Πακέτα τύπου UNAVAILABLE στέλνονται μόνο από τον base bandwidth broker agent.
- Ο τύπος DISTR_QUESTION δηλώνει το τύπο πακέτου που στέλνει ο DistributedBBbase στο BBedge για το αν υπάρχει το διαθέσιμο bandwidth

- Ο τύπος DISTR_ANSWER δηλώνει το τύπο πακέτου με το οποίο απαντά ο BBedge στον DistributedBBbase για το αν όντως υπάρχει το διαθέσιμο bandwidth
- Ο τύπος CENTR_QUESTION δηλώνει το τύπο πακέτου που στέλνει ο CentralBBbase στο BBedge για το αν υπάρχει το διαθέσιμο bandwidth
- Ο τύπος CENTR_ANSWER δηλώνει το τύπο πακέτου με το οποίο απαντά ο BBedge στον CentralBBbase για το αν όντως υπάρχει το διαθέσιμο bandwidth

Το πεδίο answer της επικεφαλίδας καθορίζει την απάντηση σε κάποιο αίτημα για bandwidth. Το συγκεκριμένο πεδίο χρησιμοποιείται μόνο στα πακέτα τύπου ANSWER. Η απάντηση μπορεί να είναι είτε θετική είτε αρνητική. Για τους υπόλοιπους τύπους πακέτων η τιμή στο συγκεκριμένο πεδίο δεν έχει καμία σημασία.

Το πεδίο bndw_request της επικεφαλίδας καθορίζει το ζητούμενο bandwidth όταν το πακέτο είναι τύπου REQUEST ή ANSWER. Στην περίπτωση πακέτου τύπου FIX_BWD το συγκεκριμένο πεδίο καθορίζει την μείωση του διαθέσιμου bandwidth σε κάποιο link. Για τους υπόλοιπους τύπους πακέτων η τιμή στο συγκεκριμένο πεδίο δεν έχει καμία σημασία.

Το πεδίο sender της επικεφαλίδας καθορίζει την διεύθυνση του κόμβου ο οποίος αρχικά δημιούργησε το αίτημα. Αρχικά συμπληρώνεται από τον edge bandwidth broker που στέλνει το αρχικό αίτημα και στη συνέχεια αντιγράφεται σε όλες τις επικεφαλίδες των μηνυμάτων που στέλνονται κατά τον έλεγχο αποδοχής.

Σε πακέτα τύπου REQUEST το πεδίο endnode της επικεφαλίδας καθορίζει τη διεύθυνση του τελικού κόμβου. Σε πακέτα τύπου FIX_BWD το συγκεκριμένο πεδίο χρησιμοποιείται για να καθορίσει ένα link. Δηλαδή σε πακέτα τύπου FIX_BWD το πεδίο endnode πρέπει να είναι ένας κόμβος γειτονικός με τον κόμβο στον οποίο στέλνεται το πακέτο ενώ σε πακέτα τύπου REQUEST μπορεί να είναι οποιοσδήποτε κόμβος του δικτύου.

Το πεδίο hop_num καθορίζει τον αριθμό των hops που το αίτημα έχει επισκεφτεί εκτός του base bandwidth broker agent. Κάθε edge bandwidth broker που δέχεται ένα αίτημα κατά τον έλεγχο αποδοχής, αυξάνει την τιμή στο συγκεκριμένο πεδίο κατά μία μονάδα.

Όταν ένα αίτημα για bandwidth γίνει δεκτό, ανατίθεται στην εφαρμογή που έκανε το αίτημα ένα μοναδικό DSCP. Για αυτό απαιτείται το πεδίο DSCP στην επικεφαλίδα των πακέτων BBE και BBB. Το πεδίο DSCP το συμπληρώνει μόνο ο base bandwidth broker.

Τα πεδία start_time και stop_time περιλαμβάνουν το χρόνο έναρξης και λήξης του αιτήματος για bandwidth.

Το πεδίο status εκφράζει την κατάσταση στην οποία βρίσκεται ένας κόμβος.

5.3.2 Routing Information

Ο base bandwidth broker, όταν δέχεται ένα αίτημα για bandwidth θα πρέπει ή να το προωθήσει σε όλους τους edge bandwidth brokers που βρίσκονται στο μονοπάτι πηγής-προορισμού ή να ρωτήσει τη βάση δεδομένων του για την απαιτούμενη πληροφορία. Και στις δύο περιπτώσεις απαιτείται λοιπόν να γνωρίζουν οι base agents όλους τους κόμβους που βρίσκονται στο μονοπάτι. Συγκεκριμένα, αρχικά υπολογίζεται ο επόμενος κόμβος n_1 από τον κόμβο που προήλθε το μήνυμα, στη

συνέχεια ο επόμενος κόμβος n_2 από τον n_1 και ούτω καθεξής μέχρι να φτάσουμε στον προορισμό.

Αυτή η διαδικασία εύρεσης του επόμενου κόμβου γίνεται με το να προσπελαύνει ο base bandwidth broker τους πίνακες δρομολόγησης. Συγκεκριμένα χρησιμοποιούνται διαδοχικά οι εντολές ***compute-routes***, ***get-routelogic*** και ***lookup.***Στη συνέχεια παρουσιάζουμε συνοπτικά αυτές τις εντολές.

compute-routes

Η παραπάνω εντολή υπολογίζει από την τοπολογία του δικτύου την πληροφορία για το next hop για κάθε κόμβο του δικτύου. Στη συνέχεια δημιουργεί τα routing tables.

get-routelogic

Με αυτήν την εντολή μπορούμε να χειριστούμε τα routing tables και να λάβουμε πληροφορία από αυτά. Χρειάζεται λοιπόν να έχουν υπολογιστεί τα routing tables, άρα να έχει εκτελεστεί η εντολή compute-routes.

lookup <srcid> <destid>

Αφού έχει εκτελεστεί η εντολή get-routelogic, η εντολή lookup επιστρέφει το id του επόμενου κόμβου από το <srcid> ως το <destid>.

Και οι τρεις παραπάνω εντολές είναι της γλώσσας tcl. Όταν χρειάζεται ένα πρόγραμμα που είναι γραμμένο σε C++ να χρησιμοποιήσει tcl εντολές μπορεί να το κάνει με χρήση των *εντολών* ***Tcl::instance()***,***tcl.evalc tcl.evalf*** και ***tcl.result()***.

5.3.3 Adjacent Nodes Interface

Το adjacent nodes interface χρησιμοποιείται για να διατηρεί ο κάθε edge bandwidth broker πληροφορία για το διαθέσιμο bandwidth στα links που συνδέουν τον κόμβο πάνω στον οποίο τρέχει με τους γειτονικούς του κόμβους.

Ο κάθε edge bandwidth broker χρειάζεται να γνωρίζει ποιοι είναι οι γειτονικοί του edge bandwidth brokers. Επίσης, για κάθε γειτονικό edge bandwidth broker χρειάζεται να γνωρίζει πόσο είναι το διαθέσιμο bandwidth στο αντίστοιχο link. Για τη διαχείριση του bandwidth σε κάθε link δημιουργήθηκε η δομή *neighbor*. Η δομή neighbor αποτελείται από ένα node-id και από το διαθέσιμο bandwidth. Ακόμα, δημιουργήθηκε η μεταβλητή-μέλος *nghbr_list* σε κάθε edge bandwidth broker. Η *nghbr_list* είναι μια λίστα από δομές neighbor. Για κάθε γειτονικό κόμβο δημιουργείται μια δομή neighbor και καθορίζεται το node-id του γειτονικού κόμβου καθώς και το διαθέσιμο bandwidth στο link προς αυτόν τον γειτονικό κόμβο. Στη συνέχεια γίνεται η καταχώρηση της δομής στη λίστα nghbr_list. Με αυτόν τον τρόπο κάθε edge bandwidth broker διατηρεί πληροφορία για το bandwidth σε κάθε link μεταξύ του κόμβου στον οποίο έχει τοποθετηθεί και των γειτονικών κόμβων.

Για να μπορεί ο χρήστης να έχει πρόσβαση στο adjacent nodes interface δημιουργήθηκε η tcl εντολή ***set_bndw***. Η εντολή συντάσσεται ως:

BBedgeAgent set_bndw *node_id bandwidth start_time stop_time*

όπου *BBedgeAgent* κάποιος edge bandwidth broker που τρέχει σε κάποιον κόμβο n1, *node_id* το node-id κάποιου κόμβου n2 γειτονικού του n1 , *bandwidth* το bandwidth που θα διαχειρίζεται ο bandwidth broker στο link n1-n2 και start_time- stop_time το χρονικό διάστημα για το οποίο θέλουμε το bandwidth.

Στα SDBB,PDBB μοντέλα όταν προωθείται ένα αίτημα από τον base bandwidth broker σε κάποιον edge bandwidth broker που βρίσκεται στο μονοπάτι, ουσιαστικά ο edge bandwidth broker ερωτάται για το εάν υπάρχει διαθέσιμο bandwidth στο link που οδηγεί στον επόμενο κόμβο. Ο έλεγχος γίνεται με το να αναζητείται στη λίστα *nghbr_list* η εγγραφή *neighbor* της οποίας το node-id ισούται με το node-id του επόμενου κόμβου. Στη συνέχεια γίνεται έλεγχος εάν το bandwidth στη συγκεκριμένη εγγραφή είναι μεγαλύτερο από το αιτούμενο bandwidth για το συγκεκριμένο χρονικό διάστημα. Εάν τελικά το αίτημα γίνει δεκτό, ο base bandwidth broker ενημερώνει για το γεγονός όλους τους edge bandwidth brokers που βρίσκονται στο μονοπάτι. Αφού ειδοποιηθεί, ο κάθε edge bandwidth broker αναζητεί και πάλι στη λίστα *nghbr_list* την εγγραφή *neighbor* της οποίας το node-id ισούται με το node-id του επόμενου κόμβου και μειώνει το bandwidth στη συγκεκριμένη εγγραφή τόσο όσο είναι το bandwidth του αιτήματος για το συγκεκριμένο χρονικό διάστημα.

Στα CBB/CFBB μοντέλα όλες οι λίστες *nghbr_list* όλων των κόμβων του δικτύου βρίσκονται στους base agents. Η αναζήτηση γίνεται με ακριβώς των ίδιο τρόπο μόνο που δεν γίνεται μέσω δικτύου αλλά τοπικά. Εάν τελικά το αίτημα γίνει δεκτό, ο base bandwidth broker ενημερώνει για το γεγονός την τοπική του βάση δεδομένων. Ειδικά στο CFBB μοντέλο, έχουμε και ενημέρωση της βάσης δεδομένων που βρίσκεται στους Edges.

Υπάρχουν πολλά πλεονεκτήματα με το να διαθέτει κάθε edge bandwidth broker τη δική του λίστα *nghbr_list*. Το σημαντικότερο είναι ότι για κάθε link σε ένα μονοπάτι, το διαθέσιμο bandwidth μπορεί να έχει διαφορετική τιμή. Επίσης, στην περίπτωση που ένα duplex link αποτελείται από δύο simplex links, δηλαδή στην περίπτωση που το link αποτελείται από δύο links μιας κατεύθυνσης, μπορεί να οριστεί διαφορετικό διαθέσιμο bandwidth για τη μια κατεύθυνση και διαφορετικό για την άλλη. Αυτό θα σημαίνει ότι για δύο γειτονικούς edge bandwidth brokers, το bandwidth στην εγγραφή neighbor του πρώτου για τον δεύτερο δεν είναι απαραίτητα ίσο με το bandwidth στην εγγραφή neighbor του δεύτερου για τον πρώτο διότι το διαθέσιμο bandwidth σε κάθε ένα από τα δύο simplex links μπορεί να διαφέρει.

5.3.4 User/Application Interface

Το ανώτερο επίπεδο στην αρχιτεκτονική του bandwidth broker που υλοποιήθηκε είναι το user/application interface. Το user/application interface είναι το interface που χρησιμοποιείται για την επικοινωνία του χρήστη με το σύστημα του bandwidth broker. Όταν ένας χρήστης επιθυμεί να χρησιμοποιήσει ένα μέρος των δικτυακών πόρων με εγγυήσεις για το bandwidth, μπορεί να χρησιμοποιήσει το user/application interface για να κάνει ένα αίτημα στον bandwidth broker. Επίσης, όταν ο bandwidth broker ολοκληρώσει τη διαδικασία του ελέγχου αποδοχής του αιτήματος (και στην περίπτωση θετικής απάντησης ολοκληρώσει και την κατανομή των πόρων) χρησιμοποιεί το user/application interface ώστε να δηλώσει την απάντηση στον χρήστη.

Το user/application interface υλοποιήθηκε σε Tcl. Τα αιτήματα για bandwidth στέλνονται πάντα από κάποιον Edge bandwidth broker agent. Έτσι δημιουργήθηκε η Tcl εντολή ***sendto***. Η εντολή συντάσσεται ως:

BBedgeAgent *sendto* node_id bandwidth *start_time stop_time*

όπου *BBedgeAgent* κάποιος edge bandwidth broker που τρέχει σε κάποιον κόμβο n1, *node_id* το node-id κάποιου κόμβου n2 , *bandwidth* το αιτούμενο bandwidth που ζητά ο BBedgeAgent από τον κόμβο n1 ως τον κόμβο n2 και start_time-stop_time το χρονικό διάστημα για το οποίο επιθυμούμε το bandwidth.

Όταν ο edge bandwidth broker ο οποίος έστειλε ένα αίτημα λάβει απάντηση, τη διαβιβάζει στην εφαρμογή που ζήτησε το εγγυημένο bandwidth. Αυτό γίνεται με την Tcl συνάρτηση **recv** του edge bandwidth broker που επιστρέφει την απάντηση στην εφαρμογή. Σε περίπτωση θετικής απάντησης επιστρέφεται και το DSCP που έχει ανατεθεί στη συγκεκριμένη κίνηση του χρήστη. Από εκεί και πέρα είναι θέμα της εφαρμογής το πώς θα μεταχειριστεί το εγγυημένο bandwidth που της παραχωρήθηκε. Παρόλα αυτά, για να έχει τις εγγυήσεις ποιότητας που ζήτησε, η εφαρμογή χρειάζεται να μαρκάρει όλα τα πακέτα της με το DSCP που της επιστράφηκε.

5.3.5 Intra Domain Interface

Το intra-domain interface χρησιμοποιείται από τους edge bandwidth brokers για την κατανομή των πόρων του δικτύου. Όταν ολοκληρωθεί ο έλεγχος αποδοχής ενός αιτήματος για bandwidth, στην περίπτωση που το αίτημα γίνει δεκτό απαιτείται να γίνει η κατάλληλη κατανομή των πόρων του δικτύου. Η κατανομή των πόρων έχει τρεις κύριους στόχους:

- Να παρέχεται στην εφαρμογή που έστειλε το αίτημα το εγγυημένο bandwidth που ζήτησε και έγινε δεκτό από τον bandwidth broker.
- Στην περίπτωση που η συγκεκριμένη εφαρμογή αποπειραθεί να στείλει στο δίκτυο κίνηση που να απαιτεί ελαφρώς περισσότερο bandwidth από αυτό που της έχει ανατεθεί, αυτό να γίνεται ομαλά χωρίς απώλειες πακέτων.
- Στην περίπτωση που η συγκεκριμένη εφαρμογή αποπειραθεί να στείλει στο δίκτυο κίνηση που να απαιτεί αρκετά περισσότερο bandwidth από αυτό που της έχει ανατεθεί, τα πακέτα της πλεονάζουσας κίνησης να απορρίπτονται από τους routers του δικτύου.

Η λειτουργία της κατανομής πόρων αρχίζει μόλις γίνει δεκτό ένα αίτημα για bandwidth. Εκείνη τη στιγμή ο edge bandwidth broker χρησιμοποιεί το intra-domain interface για να κάνει configure τους routers ώστε να ικανοποιούν κατάλληλα το αίτημα. Όλη η διαδικασία κατανομής των πόρων γίνεται μέσα στην Tcl συνάρτηση recv που αναφέρθηκε και στο user/application interface.

Για τη χρονοδρομολόγηση σε όλες τις ουρές έχουν επιλεγεί οι ουρές προτεραιοτήτων. Έτσι, ορίζονται δύο φυσικές ουρές εκ των οποίων η μία θα εξυπηρετεί την κίνηση των εφαρμογών που έχουν πάρει την έγκριση από τον bandwidth broker και έχουν εξασφαλισμένο bandwidth και η άλλη ουρά θα εξυπηρετεί όλη την υπόλοιπη κίνηση. Η πρώτη φυσική ουρά αποτελείται από δύο ιδεατές ουρές (virtual queues). Στον ns, όταν χρησιμοποιούνται ουρές προτεραιοτήτων για τη χρονοδρομολόγηση στις ουρές, μπορεί να ρυθμιστεί συγκεκριμένα το μέγιστο bandwidth που θα εξυπηρετείται από κάθε φυσική ουρά (π.χ. ότι η φυσική ουρά 0 θα εξυπηρετεί το πολύ 1Mbps). Αντίστοιχα, για την Weighted Round Robin χρονοδρομολόγηση ή την Modified Deficit Round Robin το μέγιστο bandwidth που θα εξυπηρετεί η κάθε ουρά δεν ρυθμίζεται συγκεκριμένα αλλά σχετικά με τις υπόλοιπες ουρές. Σε κάθε ουρά ανατίθεται ένα βάρος και όσο μεγαλύτερο βάρος έχει μια ουρά σε σχέση με μια άλλη τόσο περισσότερο εξυπηρετείται σε κάθε κύκλο.

Στην υλοποίηση θεωρούμε ότι η κίνηση στην οποία παρέχεται εξασφαλισμένο bandwidth εξυπηρετείται από τη φυσική ουρά 0 και η υπόλοιπη κίνηση από τη φυσική ουρά 1. Εδώ απαιτείται να γίνει ορθά η κατανομή των βαρών ή το μέγιστο bandwidth που θα εξυπηρετεί η κάθε ουρά (ανάλογα με το μηχανισμό χρονοδρομολόγησης) ώστε το bandwidth που θα εξυπηρετεί ή ουρά 0 να είναι ίσο με το bandwidth που διαχειρίζεται ο bandwidth broker στο συγκεκριμένο link, όπως καθορίζεται από τη συνάρτηση *set_bndw*.

Το DSCP της κίνησης που δεν έχει εξασφαλίσει bandwidth από τον bandwidth broker είναι το 0. Το DSCP των πακέτων BBB και BBE είναι το 1. Τα DSCPs των ροών που έχουν εξασφαλίσει bandwidth από τον bandwidth broker είναι τα 2-255. Η επιλογή της ουράς στην οποία τοποθετείται κάθε πακέτο γίνεται με βάση το DSCP της επικεφαλίδας του. Συγκεκριμένα, τα πακέτα με DSCP 0 τοποθετείται στην φυσική ουρά 1. Τα πακέτα με DSCP 1 τοποθετούνται στην ιδεατή ουρά 0 της φυσικής ουράς 0. Τα πακέτα με DSCP 2-255 τοποθετούνται στην ιδεατή ουρά 1 της φυσικής ουράς 0. Ο καθορισμός των ουρών στις οποίες τοποθετούνται τα πακέτα με DSCP 0 και 1 γίνεται από την αρχή της λειτουργίας του συστήματος. Αρχικά δεν έχει ανατεθεί κανένα από τα DSCPs 2-255 σε κάποια εφαρμογή και δεν προβλέπεται η τοποθέτηση πακέτων που φέρουν κάποιο από αυτά τα DSCPs σε κάποια ουρά.

Όταν ολοκληρωθεί ένας έλεγχος αποδοχής κάποιου αιτήματος με επιτυχία, δηλαδή το αίτημα γίνει δεκτό, αρχικά καθορίζεται το DSCP που θα χρησιμοποιεί η εφαρμογή που έκανε το αίτημα. Ο καθορισμός αυτού του DSCP γίνεται από τον base bandwidth broker. Ο base bandwidth broker διαθέτει την μεταβλητή-μέλος *cur_dscp* η τιμή της οποίας ανά πάσα στιγμή εκφράζει το επόμενο διαθέσιμο DSCP που θα ανατεθεί σε εφαρμογές των οποίων τα αιτήματα έγιναν δεκτά. Το αρχικό διαθέσιμο DSCP είναι 2 και μετά από κάθε ανάθεση αυξάνεται κατά 1. Συνεπώς μπορούν να υπάρχουν 254 διαφορετικές συνενώσεις ροών και η διαφοροποίηση τους θα γίνεται με βάση το DSCP.

Όταν υπολογιστεί το DSCP που θα χρησιμοποιεί μια εφαρμογή, επιστρέφεται με το μήνυμα θετικής απάντησης στον edge bandwidth broker ο οποίος έστειλε το αίτημα και από αυτόν μέσω της Tcl συνάρτησης *recv* στην εφαρμογή που θα το χρησιμοποιεί. Από εκεί και μετά είναι ευθύνη της εφαρμογής να μαρκάρει τα πακέτα της με το συγκεκριμένο DSCP.

Στη συνάρτηση recv γίνονται configure και όλοι οι δρομολογητές του δικτύου ώστε να εξυπηρετούν κατάλληλα τα πακέτα που φέρουν το συγκεκριμένο DSCP. Σε κάθε ουρά του δικτύου κορμού καθορίζεται ότι τα πακέτα με το συγκεκριμένο DSCP θα τοποθετούνται στην ιδεατή ουρά 1 της φυσικής ουράς 0. Επίσης, καθορίζεται η αστυνόμευση που θα γίνεται στα πακέτα με το συγκεκριμένο DSCP. Ως μηχανισμός αστυνόμευσης έχει οριστεί ο Token Bucket. Η αστυνόμευση γίνεται σε όλους τους κόμβους του δικτύου. Ο μηχανισμός Token Bucket, έχει δύο σημαντικές παραμέτρους, το μέσο επιτρεπτό bandwidth σε bps και το μέγεθος του κάδου σε bytes. Το μέσο επιτρεπτό bandwidth συνήθως είναι λίγο μεγαλύτερο από το bandwidth που είχε ζητηθεί από τον bandwidth broker. Αυτό συμβαίνει για να μην χάνονται πακέτα μιας ροής όταν ο ρυθμός τους υπερβαίνει κατά ένα μικρό ποσοστό το προκαθορισμένο προφίλ.

Όλες οι ροές που έχουν διασφαλίσει εγγυημένο bandwidth καθώς και τα πακέτα του intra-domain πρωτοκόλλου τοποθετούνται πάντα στη φυσική ουρά 0. Τα πακέτα του intra-domain πρωτοκόλλου τοποθετούνται στην ιδεατή ουρά 0 της φυσικής ουράς 0 ενώ τα άλλα πακέτα στην ιδεατή ουρά 1 της φυσικής ουράς 0. Ο λόγος που τα πακέτα

του intra-domain πρωτοκόλλου τοποθετούνται σε διαφορετική ιδεατή ουρά από τα υπόλοιπα πακέτα είναι ότι απαιτείται ιδιαίτερη προσοχή ώστε να μην χάνονται τέτοια πακέτα, καθώς απώλεια BBB ή BBE πακέτων μπορεί να οδηγήσει σε μη ορθή λειτουργία του πρωτοκόλλου. Ο ns δίνει τη δυνατότητα ανάθεσης σε κάθε ιδεατή ουρά της μέγιστης πιθανότητας απόρριψης πακέτων. Αν λοιπόν ανατεθεί σχεδόν μηδενική πιθανότητα απόρριψης πακέτων στην ιδεατή ουρά 1 της φυσικής ουράς 0, διασφαλίζεται ότι πολύ σπάνια θα χαθούν πακέτα του intra-domain πρωτοκόλλου, ακόμα και σε περιπτώσεις συμφόρησης του δικτύου.

5.4 ΖΗΤΗΜΑΤΑ ΥΛΟΠΟΙΗΣΗΣ

Στην παράγραφο αυτή θα γίνει μια περιγραφή των Bandwidth Broker που αναπτύχθηκαν, υλοποιήθηκαν και δοκιμάστηκαν στον ns-2. Όπως έχει ήδη αναφερθεί τα μοντέλα αυτά είναι:

- Serial Distributed Bandwidth Broker model (SDBB model)
- Parallel Distributed Bandwidth Broker model (PDBB model)
- Centralized Bandwidth Broker model (CBB model)
- Centralized Fault-tolerant Bandwidth Broker model (CFBB model)

Σε κάθε υλοποίηση, δύο είδη agents υπάρχουν, ο Edge Bandwidth Broker(BBedgeAgent) και ο Base Bandwidth Broker(BBbaseAgent). Ένας agent στον ns-2 είναι ένα σημείο όπου τα πακέτα καταναλώνονται και κατασκευάζονται χρησιμοποιώντας ένα συγκεκριμένο πρωτόκολλο. Ένας BBedge agent είναι ένα απλό module τοποθετημένο σε κάθε κόμβο(router) του δικτύου που αναπαριστά ένα client (ένας χρήστης ή μια εφαρμογή) και έχει ως λειτουργία το να στέλνει αιτήματα για κίνηση με συγκεκριμένο profile για μια συγκεκριμένη χρονική περίοδο. Ένα τέτοιο αίτημα λαμβάνεται από τον BBbase agent, που αναπαριστά έναν server, και προστίθεται σαν ένα αίτημα με συγκεκριμένες παραμέτρους(sender,end node, bandwidth, time limit και status, που στο στάδιο αυτό μαρκάρεται ως pending) στην τοπική βάση δεδομένων. Όταν τελειώνει η επεξεργασία του αιτήματος, το status του αιτήματος αλλάζει ανάλογα με το αποτέλεσμα της επεξεργασίας, είτε “satisfied” είτε “rejected”. Η απάντηση μετά στέλνεται πίσω στον BBedge agent.

Ένα άλλο κοινό στοιχείο των BBbase agents για όλα τα μοντέλα είναι το buffer για τα αιτήματα. Οι BBbase agents επεξεργάζονται ένα αίτημα τη φορά ώστε εάν δύο αιτήματα φτάσουν κοντινά το ένα με το άλλο, το πρώτο πρόκειται να ικανοποιηθεί και το άλλο πρόκειται να προστεθεί σε ένα buffer που υπάρχει στους BBbase agents. Εάν ένα αίτημα φτάσει και το buffer είναι άδειο, το αίτημα θα ικανοποιηθεί αμέσως. Διαφορετικά, εάν το buffer εμπεριέχει προηγούμενα αιτήματα αλλά δεν είναι ακόμα γεμάτο, το αίτημα θα προστεθεί στο τέλος του buffer με σκοπό να επεξεργαστεί στο μέλλον. Τελικά, εάν το buffer είναι γεμάτο, το αίτημα απορρίπτεται. Το μήκος του buffer μπορούμε να το διαμορφώσουμε οι ίδιοι και είναι ένα σημείο έρευνας για την αποτελεσματική λειτουργία του bandwidth broker.

Παρόλο που η δομή των BBedge agents είναι ίδια και για τα τέσσερα μοντέλα που έχουν υλοποιηθεί, κάθε ένα λειτουργεί με τον τρόπο του, όταν διάφορα πακέτα λαμβάνονται κατά τη διάρκεια της εξομοίωσης. Οι διαφορές ανάμεσα στις τέσσερις αρχιτεκτονικές έχουν κυρίως να κάνουν κυρίως με τη δομή και τη συμπεριφορά των υλοποιημένων πρωτοκόλλων. Τα διάφορα μοντέλα χρησιμοποιούν μεθόδους για να

επεξεργαστούν τα αιτήματα(admission control) και να αποθηκέυσουν την απαραίτητη πληροφορία. Οι μέθοδοι επεξεργασίας αιτημάτων περιγράφονται λεπτομερώς στη συνέχεια ενώ το υπόλοιπο κομμάτι αφορά την αποθήκευση της πληροφορίας.

Τα SDBB και PDBB μοντέλα αποθηκεύουν την πληροφορία σχετικά με την κατάσταση της κάθε σύνδεσης (δεσμευμένο bandwidth για τα υπάρχοντα αιτήματα, ελεύθερο bandwidth) στους αντίστοιχους BBedge agents. Αντίθετα, το CBB μοντέλο αποθηκεύει την πληροφορία στον BBbase agent και το CFBB μοντέλο αποθηκεύει την πληροφορία τόσο στους BBedge agents όσο και στον BBbase agent. Το format της πληροφορίας αυτής είναι το παρακάτω: σκεφτείτε για παράδειγμα τον κόμβο 1 που είναι συνδεδεμένος με τον κόμβο 2 και τον κόμβο 2 που είναι συνδεδεμένος με τον κόμβο 1. Η σχετική πληροφορία για τον κόμβο ένα είναι ότι "Το διαθέσιμο bandwidth από τον κόμβο 1 στον κόμβο 2 είναι b για την χρονική περίοδο t2-t1 " όπου t2,t1 είναι σημεία στο χρόνο και b το bandwidth που προκύπτει. Η σχετική πληροφορία για τον κόμβο 2 είναι ότι "το διαθέσιμο bandwidth από τον κόμβο 2 στον κόμβο 1 είναι b' για την χρονική περίοδο t2'-t1' ", όπου t2',t1' είναι σημεία στο χρόνο και b' το bandwidth. Στο δικό μας μοντέλο εξομοίωσης κάνουμε την υπόθεση ότι b'=b, t2'=t2 και t1=t1' ανάμεσα στους δύο κόμβους, με άλλα λόγια ότι όλα τα αιτήματα και οι δεσμεύσεις είναι bidirectional. Για τα CBB και CFBB μοντέλα, η τοπική βάση δεδομένων του BBbase agent, έχει για κάθε κόμβο που ελέγχει, ολόκληρη την πληροφορία του διαθέσιμου bandwidth για κάθε σύνδεση που έχει με ένα γειτονικό κόμβο για όλες τις χρονικές στιγμές.

5.4.1 Περιγραφή των υλοποιημένων μοντέλων

Σε κάθε ένα από τα υλοποιημένα μοντέλα, χρησιμοποιούμε διαφορετικό μοντέλο επικοινωνίας με σκοπό την ολοκλήρωση της επεξεργασίας των αιτημάτων. Σκοπός μας είναι να εξομοιώσουμε, όσο ακριβές γίνεται, ένα δίκτυο που ελέγχεται από ένα Bandwidth Broker. Η επικοινωνία ανάμεσα στο BBbase και στον BBedge agent επιτυγχάνεται με τη χρήση μηνυμάτων που στέλνονται συνέχεια είτε από ένα BBedge σε ένα BBbase είτε από έναν BBbase σε έναν BBedge agent. Δύο BBedge agents δεν επικοινωνούν ποτέ με το να στέλνουν μηνύματα ο ένας στον άλλο, αφού ο BBbase agent πάντα μεσολαβεί στην επικοινωνία.

Για το SDBB μοντέλο η επεξεργασία παρουσιάζεται στη συνέχεια. Στο μοντέλο αυτό, όλη η πληροφορία σχετικά με την κατάσταση των γραμμών όσον αφορά το διαθέσιμο bandwidth αποθηκεύεται τοπικά στους BBedge agents, όπως έχει ήδη αναφερθεί. Στην αρχή, ο BBedge στέλνει ένα αίτημα για bandwidth στον BBbase agent. Όταν ο BBbase agent λαμβάνει το αίτημα, αποθηκεύει την διεύθυνση του αποστολέα. Μετά χρησιμοποιώντας μια tcl εντολή του ns-2 περιβάλλοντος εξομοίωσης("lookup"), βρίσκει τον γειτονικό κόμβο του αποστολέα αιτήματος. Συγκεκριμένα, με την εισαγωγή του κόμβου έναρξης και λήξης της επιθυμητής διαδρομής, ο ns-2 χρησιμοποιεί το OSFP πρωτόκολλο για να βρει το πιο σύντομο μονοπάτι ανάμεσα στους δύο αυτούς κόμβους και μετά επιστρέφει στον χρήστη τον πρώτο κόμβο της διαδρομής αυτής που βρίσκεται δίπλα στον αποστολέα αιτήματος. Έτσι, αρχικά, ο BBbase στέλνει ένα πακέτο στον BBedge agent που ξεκίνησε το αίτημα για bandwidth, ρωτώντας τον για το κατά πόσο υπάρχει διαθέσιμο bandwidth από τον εαυτό του μέχρι τον επόμενο γειτονικό κόμβο. Εάν ο BBbase λάβει μια θετική απάντηση από τον πρώτο BBedge agent της διαδρομής, τότε ο BBbase ρωτά τον γείτονα του πρώτου κόμβου για bandwidth και πάει λέγοντας, μέχρι ο BBbase λάβει μια αρνητική ενέργεια, στην οποία περίπτωση σταματά την επεξεργασία του

αιτήματος, στέλνει αρνητική απάντηση στον αποστολέα του αιτήματος και συνεχίζει με το επόμενο αίτημα, ή μέχρι ο BBbase λάβει θετικές απαντήσεις από όλους τους κόμβους που βρίσκονται στη διαδρομή του αιτήματος, περίπτωση στην οποία ο BBbase στέλνει θετική απάντηση στον αποστολέα του αιτήματος και από ένα πακέτο σε κάθε κόμβο που βρίσκεται πάνω στη διαδρομή του αιτήματος, με σκοπό να ανανεώσει την πληροφορία τους σχετικά με τη διαθεσιμότητα του bandwidth.Μετά από αυτό, ο BBbase συνεχίζει στην επεξεργασία του επόμενου αιτήματος(εάν υπάρχει). Μια θετική απάντηση σημαίνει ότι ο BBbase agent επιτρέπει στον αποστολέα του αιτήματος να χρησιμοποιήσει το bandwidth που ζητήθηκε δίνοντας εγγυήσεις σχετικά με την δέσμευση των απαιτούμενων πόρων και μια αρνητική απάντηση σημαίνει ότι το αίτημα απορρίφθηκε λόγω έλλειψης bandwidth.

Το PDBB μοντέλο, παρόμοια με το SDBB μοντέλο, αποθηκεύει και αυτό την πληροφορία στους BBedge agents αλλά έχει διαφορετικό τρόπο απόκτησης της πληροφορίας αυτής στην πορεία επεξεργασίας ενός αιτήματος. Όταν το αίτημα φτάνει, ο BBbase agent στέλνει ένα πακέτο σε κάθε ένα από τους κόμβους που βρίσκονται στη διαδρομή του αιτήματος ρωτώντας τους για bandwidth σε μια συγκεκριμένη χρονική περίοδο. Εάν λάβει έστω και μια αρνητική απάντηση, δεν περιμένει για οποιαδήποτε άλλη απάντηση από τους BBedge agents, αλλά σταματά την επεξεργασία του αιτήματος, στέλνει μια αρνητική απάντηση στον αποστολέα του αιτήματος και συνεχίζει στο επόμενο αίτημα. Εάν ο BBbase agent συνεχίζει να παίρνει θετικές απαντήσεις από τους κόμβους, περιμένει να απαντήσουν όλοι οι κόμβοι στη ερώτησή του. Εάν όλοι οι κόμβοι έχουν απαντήσει θετικά, τότε το υπόλοιπο μέρος της επεξεργασίας του αιτήματος είναι πανομοιότυπο με το SDBB μοντέλο. Η κύρια διαφορά από τον SDBB είναι επομένως ότι τώρα ο BBbase agent δεν ρωτάει σειριακά τους κόμβους, αλλά παράλληλα (ένα είδος πλημμύρας των μηνυμάτων ερώτησης) και περιμένει μέχρι όλοι οι BBedge agents απαντήσουν. Το PDBB μοντέλο στοχεύει στη μείωση των χρόνων απόκρισης των αιτημάτων σε σχέση με το SDBB μοντέλο με την παραλληλοποίηση των ερωτήσεων πάνω στο δίκτυο.

Το CBB μοντέλο είναι ένα κεντρικοποιημένο μοντέλο σε σχέση με την κατανεμημένη φύση των προηγούμενων μοντέλων. Ο BBbase agent αποθηκεύει όλη την πληροφορία σχετικά με την κατάσταση και την διαθεσιμότητα των γραμμών σε μια τοπική βάση δεδομένων, την οποία λαμβάνει υπόψη με σκοπό να επεξεργαστεί ένα αίτημα. Έτσι ο χρόνος απόκρισης μειώνεται με την μείωση της διαδικτυακής επικοινωνίας, αλλά το CBB μοντέλο μείωσε και την ανεκτικότητα εξαιτίας της εξάρτησής του από έναν μόνον κόμβο.

Το CFBB μοντέλο συνδυάζει τα πλεονεκτήματα του CBB μοντέλου και προσθέτει πλεονασμό πληροφορίας με σκοπό την αύξηση της ανεκτικότητας σε αποτυχίες κόμβων. Τα δεδομένα που αφορούν την κατάσταση των γραμμών και την διαθεσιμότητα αποθηκεύονται και στους BBedge και στους BBbase agents. Ένα εισερχόμενο αίτημα επεξεργάζεται όπως ακριβώς και στο CBB μοντέλο, αλλά με το που έχουμε μια θετική απάντηση, ο BBbase agent όχι μόνο ενημερώνει την τοπική βάση δεδομένων του, αλλά και επίσης στέλνει ένα πακέτο σε κάθε κόμβο της διαδρομής του αιτήματος με σκοπό την ενημέρωση της σχετικής τους πληροφορίας. Η πληροφορία αυτή δεν μπορεί να αποκτηθεί κατά την διάρκεια της επεξεργασίας του αιτήματος, αλλά μπορεί να χρησιμοποιηθεί στην περίπτωση μιας αποτυχίας στον κόμβο που φιλοξενεί τον BBbase agent. Ένας άλλος κόμβος μπορεί μετά να τρέξει ένα νέο BBbase agent ο οποίος γίνεται ενήμερος για την κατάσταση του δικτύου από όλους τους BBedge agents(ο καθένας στέλνει ένα απλό μήνυμα στο νέο BBbase agent με την σχετική πληροφορία) σχετικά με την διαθεσιμότητα του bandwidth και τις

υπάρχουσες δεσμεύσεις. Το μειονέκτημα για το CFBB μοντέλο είναι η παραγωγή κάποιου επιπλέον φόρτου στο δίκτυο σε σχέση με το CBB μοντέλο.

ΚΕΦΑΛΑΙΟ 6: ΠΕΙΡΑΜΑΤΑ

ΠΕΙΡΑΜΑΤΑ

6.1 ΕΙΣΑΓΩΓΗ

Αφού παρουσιάστηκε η υλοποίηση των bandwidth broker, σε αυτό το κεφάλαιο θα παρουσιάσουμε διάφορα πειράματα που εκτελέστηκαν χρησιμοποιώντας τις υλοποιήσεις αυτές καθώς και μια συγκριτική μελέτη μεταξύ τους.

Τα πειράματα αυτά έγιναν με σκοπό να επιβεβαιώσουμε την υλοποίηση μας όσο αφορά την ορθότητα της και να αναλύσουμε τα χαρακτηριστικά της απόδοσης των διαφορετικών υλοποιήσεων.

Γίνεται παρουσίαση των πειραμάτων με σκοπό την έρευνα σε ένα αριθμό από θέματα που σχετίζονται με την λειτουργία των Bandwidth Broker modules, και συγκεκριμένα κατά πόσο το ποσοστό αποδοχής αιτημάτων (acceptance rate) ,ο φόρτος δικτύου (network overhead) και ο χρόνος απόκρισης (response time) για κάθε έναν από τους Bandwidth Brokers επηρεάζονται από την τοπολογία δικτύου, τα διαθέσιμα buffers για τα εισερχόμενα αιτήματα και η διαφορά χρόνου από τη στιγμή που ένα αίτημα υποβλήθηκε μέχρι τη στιγμή που πού έγινε η δέσμευση bandwidth.

6.2 ΔΟΜΗ ΤΩΝ ΠΕΙΡΑΜΑΤΩΝ

Όλα τα πειράματα στον ns γράφονται σε Tcl και απαιτείται να έχουν μια συγκεκριμένη δομή ώστε να μπορέσει ο εξομοιωτής να τα εκτελέσει.

Αρχικά χρειάζεται να οριστεί η τοπολογία του δικτύου. Ορίζονται αρχικά οι κόμβοι από τους οποίους αποτελείται το δίκτυο και στη συνέχεια τα links που τους συνδέουν. Τα links μπορούν να είναι simplex links ή duplex links, ανάλογα αν επιθυμούμε η κατεύθυνση της κίνησης μέσα από αυτά να είναι μονόδρομη ή αμφίδρομη αντίστοιχα. Οι σημαντικότερες παράμετροι κατά τη δημιουργία του κάθε link είναι το μέγιστο bandwidth σε bits per second, η καθυστέρηση μετάδοσης σε second ή milliseconds και το είδος της ουράς, για παράδειγμα droptail, dsRed/edge κλπ. Μετά την τοπολογία του δικτύου γίνονται configure οι ουρές των links ώστε να μπορέσουν να εξυπηρετήσουν την DiffServ αρχιτεκτονική, αν βέβαια αυτό κρίνεται απαραίτητο. Σε κάθε ουρά, ορίζεται ο μηχανισμός χρονοδρομολόγησης, ο αριθμός των φυσικών και των ιδεατών ουρών, οι αντιστοιχίσεις του κάθε DSCP με την αντίστοιχη φυσική και ιδεατή ουρά, για κάθε DSCP ο μηχανισμός αστυνόμευσης με τις παραμέτρους του (όπως το μέγιστο επιτρεπτό bandwidth και το μέγιστο μέγεθος καταιγισμού), και για κάθε ιδεατή ουρά η πιθανότητα απόρριψης πακέτων που βρίσκονται σε αυτήν.

Στο τελευταίο μέρος του πειράματος δηλώνονται όλες οι λειτουργίες που θα συμβούν και η χρονική στιγμή την οποία θα συμβεί η καθεμία.

Σε γενικές γραμμές η παραπάνω μεθοδολογία ακολουθείται κατά τη δημιουργία ενός Tcl πειράματος στον ns. Όταν χρησιμοποιείται η υλοποίηση του bandwidth broker απαιτείται να γίνουν ορισμένες διευκρινήσεις επιπλέον. Συγκεκριμένα, σε κάθε ουρά ο αριθμός των ιδεατών και των φυσικών ουρών χρειάζεται να οριστεί ως 2. Η φυσική ουρά 1 θα χρησιμοποιείται για την εξυπηρέτηση της κίνησης που δεν έχει διασφαλίσει εγγυημένο bandwidth παίρνοντας την έγκριση από τον bandwidth

broker. Τα πακέτα αυτής της κίνησης χρειάζεται να μαρκάρονται ώστε να έχουν DSCP 0. Χρειάζεται λοιπόν να γίνει αντιστοίχηση του DSCP 0 με τη φυσική ουρά 1. Όμοια, η φυσική ουρά 0 θα χρησιμοποιείται από την κίνηση που έχει διασφαλίσει εγγυημένο bandwidth. Χρειάζεται δηλαδή να γίνει αντιστοίχηση των πακέτων που χρησιμοποιούνται στο BB interface πρωτόκολλο , δηλαδή του DSCP 1, με τη φυσική ουρά 0 και την ιδεατή ουρά 0.

Σε κάθε κόμβο του δικτύου χρειάζεται να ανατεθεί ένας BBedge agent και σε έναν μοναδικό κόμβο πρέπει να ανατεθεί ένας BBbase agent. Όλοι οι BBedge agents χρειάζεται να συνδεθούν με τον BBbase agent. Επίσης, απαιτείται για κάθε link να χρησιμοποιηθεί η εντολή set_bndw ώστε να δημιουργηθούν σε όλους τους BBedge agents οι λίστες nghbr_list και, κατά συνέπεια, να καθοριστεί το bandwidth που θα διαχειρίζεται ο bandwidth broker σε όλο το εύρος του δικτύου.

Στο Tcl αρχείο χρειάζεται να δημιουργηθεί και η συνάρτηση recv του BBedge. Η συνάρτηση recv δέχεται έξι ορίσματα:

- τη διεύθυνση του κόμβου προορισμού
- την απάντηση στο αίτημα (θετική ή αρνητική)
- το DSCP που θα πρέπει να χρησιμοποιεί η εφαρμογή (σε περίπτωση θετικής απάντησης)
- το bandwidth που ζητήθηκε
- ο χρόνος έναρξης του αιτήματος
- ο χρόνος λήξης του αιτήματος

Στη συνάρτηση recv θα πρέπει:

- Να ελέγχεται αν η απάντηση είναι θετική η αρνητική.
- Να καθορίζεται η λειτουργία του συστήματος σε περίπτωση θετικής απάντησης. Αυτή η λειτουργία θα πρέπει κυρίως να περιλαμβάνει το μηχανισμό κατανομής πόρων ώστε να μπορεί να εξυπηρετηθεί η κίνηση της εφαρμογής που έστειλε το αίτημα. Μετά την κατανομή των πόρων μπορεί να ξεκινά η δημιουργία των πακέτων της νέας κίνησης, τα οποία απαιτείται να μαρκάρονται με το συγκεκριμένο DSCP.
- Να καθορίζεται η λειτουργία του συστήματος σε περίπτωση αρνητικής απάντησης. Για παράδειγμα, σε περίπτωση αρνητικής απάντησης μπορεί να στέλνει ο BBedge ένα νέο αίτημα στον bandwidth broker.

Στο τελευταίο τμήμα του tcl script μπορούν οι BBedge agents να στέλνουν αιτήματα στον BBbase agent χρησιμοποιώντας την εντολή sendto.

Η απόδοση των Bandwidth Broker models συγκρίθηκε σε ένα αριθμό από διαφορετικές δικτυακές τοπολογίες χρησιμοποιώντας διαφορετικά κριτήρια και μετρικές. Μία σημαντική μετρική είναι το ποσοστό των θετικών απαντήσεων (τα αιτήματα που έγιναν αποδεκτά) σε σχέση με τον συνολικό αριθμό από τα αιτήματα που έχουν υποβληθεί για κάθε πείραμα. Επίσης μελετήσαμε τον χρόνο απόκρισης(το χρόνο από την στιγμή που ένα αίτημα υποβάλλεται μέχρι τη στιγμή που ο BBbase agent απαντά είτε με το να δεχθεί το αίτημα είτε να το απορρίψει) καθώς και το

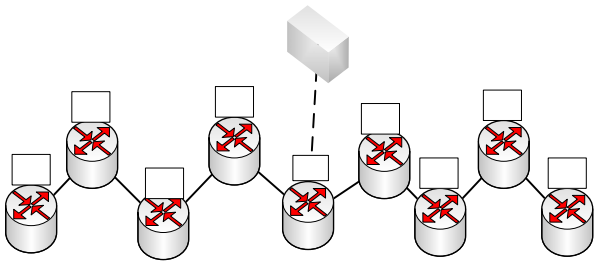
φόρτο δικτύου εξαιτίας το μηνυμάτων ελέγχου που ανταλλάσσονται μεταξύ BBbase και BBedge agents.

Τα αιτήματα στα πειράματα μας παρήχθησαν τυχαία από ένα ns-2 tcl script που χρησιμοποιούσε την μεταβλητή variable για να καθορίσει τον αριθμό των αιτημάτων που θα παραχθούν. Οι παράγοντες για κάθε αίτημα παρήχθησαν τυχαία μέσα στα κατάλληλα όρια (λαμβάνοντας υπόψη τη συνολική διάρκεια της κάθε εξομοίωσης, το συνολικό διαθέσιμο bandwidth, το μέγιστο και τον ελάχιστο αριθμό αιτημάτων δέσμευσης bandwidth) για κάθε περίπτωση που θέλαμε να εξομοιώσουμε. Συγκεκριμένα, παρήγαμε τυχαία αιτήματα με τυχαίο χρόνο αποστολής αιτήματος, τυχαίο χρόνο έναρξης αιτήματος, τυχαίο χρόνο λήξης αιτήματος, τυχαίο bandwidth και τυχαίο neighbor id. Τα αιτήματα είχαν συγκεκριμένα χρονικά όρια όσο αφορά τη διάρκειά τους (χρόνος λήξης μείον χρόνος έναρξης) , πολύ μικρότερο όριο από την συνολική διάρκεια του κάθε πειράματος, με σκοπό να αντισταθμίσουμε οποιεσδήποτε επιρροές έναρξης.

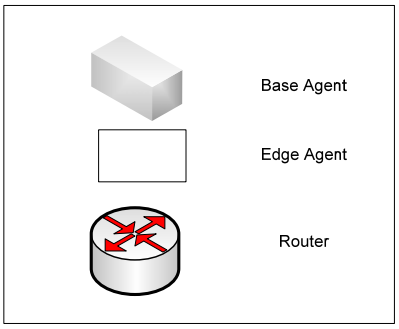
Η τυχαιότητα αποκτήθηκε χρησιμοποιώντας την ns-2 RNG κλάση. Η κλάση αυτή περιέχει μια υλοποίηση του συνδυασμένου πολλαπλού αναδρομικού γεννήτορα[31]. Ο MRG32k3a γεννήτορας παρέχει $1.8x10^{19}$ ανεξάρτητες συνεχείς ροές από τυχαίους αριθμούς, κάθε μια από τις οποίες αποτελείται από $2.3x10^{15}$ συνεχείς υποροές. Κάθε υποροή έχει μια περίοδο (για παράδειγμα τον αριθμό των τυχαίων αριθμών πριν την υπέρθεση) των $7.6x10^{22}$. Η περίοδος του συνολικού γεννήτορα είναι $3.1x10^{57}$,πολύ παραπάνω από αρκετή για να παράγουμε τυχαιότητα για τους σκοπούς μας. Ο γεννήτορας τυχαιότητας παρήγαγε ανεξάρτητα παραγόμενους αριθμούς που αναθέτονταν σε κάθε μια από τις ιδιότητες του νέου αιτήματος. Εάν ο τυχαίος συνδυασμός των ιδιοτήτων ήταν έγκυρος (για παράδειγμα ο χρόνος λήξης του αιτήματος δεν ήταν πριν από τον χρόνο έναρξης) το αίτημα παράγονταν από τον κόμβο και στέλνονταν στο Bandwidth Broker για επεξεργασία.

6.3 ΜΕΛΕΤΩΝΤΑΣ ΤΙΣ ΕΠΙΡΡΟΕΣ ΤΟΥ ΜΕΓΕΘΟΥΣ ΤΟΥ BUFFER

Στο πρώτο μας πείραμα, μελετήσαμε πως οι απαιτήσεις για buffer του BBbase agent επηρεάζουν την ολική απόδοση και πιο συγκεκριμένα πως οι απαιτήσεις αυτές σχετίζονται με το ποσοστό των αιτημάτων που έγιναν αποδεκτά. Εξετάσαμε μόνο τα SDBB και PDBB μοντέλα αφού είναι τα μόνα που πρέπει να χρησιμοποιήσουν το δίκτυο για να ολοκληρώσουν την επεξεργασία ενός αιτήματος, ενώ τα κεντρικοποιημένα μοντέλα (CBB, CFBB) εκτελούν την επεξεργασία ενός αιτήματος εσωτερικά με αποτέλεσμα πιο γρήγορα και για αυτό έχουν πολύ μικρότερη ανάγκη για buffer. Η τοπολογία δικτύου που χρησιμοποιήθηκε για αυτό το πείραμα ήταν μία τοπολογία σε σειρά, με τον BBbase Agent τοποθετημένο στο κέντρο της τοπολογίας όπως φαίνεται στην Εικόνα 27.

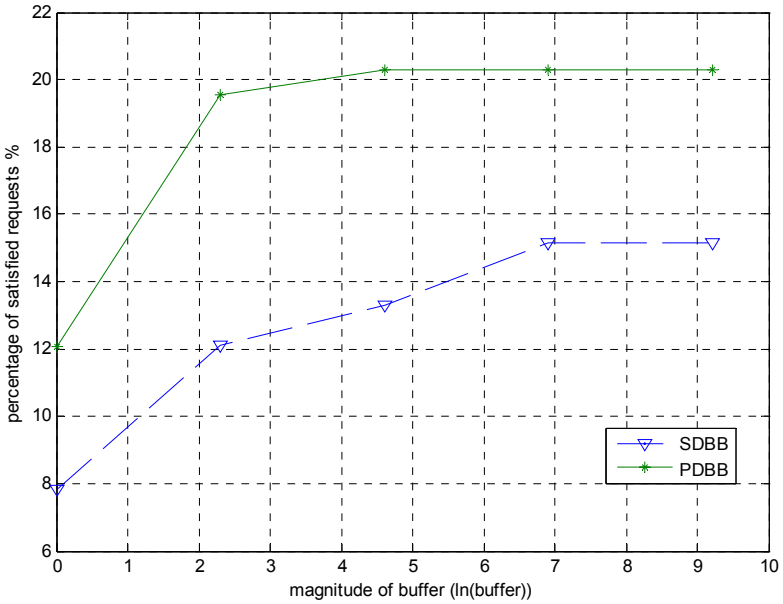


Εικόνα 27 Σειριακή τοπολογία



Εικόνα 28

Το buffer των αιτημάτων για τα SDBB και τα PDBB μοντέλα αποθηκεύει αιτήματα που δεν μπορούν να επεξεργαστούν αμέσως. Εάν ο χρόνος έναρξης του αιτήματος έχει περάσει και το αίτημα δεν έχει επεξεργαστεί, τότε το αίτημα απορρίπτεται.



Εικόνα 29 Ποσοστό αποδοχής vs μέγεθος του buffer(1ms καθυστέρηση γραμμής)

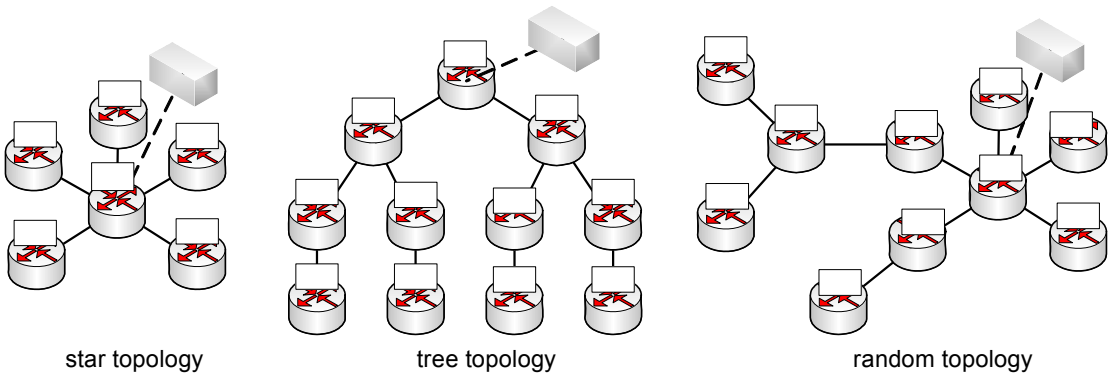
Ο οριζόντιος άξονας της Εικόνα 29 απεικονίζει το μέγεθος του buffer σε λογαριθμική (ln) κλίμακα και ο κάθετος άξονας μετράει το ποσοστό των αιτημάτων που έχουν γίνει αποδεκτά. Όταν δεν υπάρχει buffer (μηδενικό μέγεθος), υπάρχει μεγάλη αστοχία

απόδοσης, ειδικά για το SDBB μοντέλο, αλλά χρησιμοποιώντας ένα buffer μεγέθους δέκα ή μεγαλύτερο, οι αρνητικές συνέπειες μειώνονται σημαντικά. Για το PDBB μοντέλο, αυτό σημαίνει ότι με την τοποθέτηση ενός πολύ μικρού buffer στους agents, κανένα αίτημα δεν χάνεται λόγω μη διαθεσιμότητας. Αυτό το συμπέρασμα μπορεί επίσης να ενισχυθεί από το γεγονός ότι σε παρόμοια πειράματα, τα CBB/CFBB μοντέλα επίσης παρουσιάζουν παρόμοια επίδοση (20% αιτήματα που έχουν ικανοποιηθεί). Για το SDBB μοντέλο, παρόλο που το buffer βελτιώνει την κατάσταση, η απόδοση ποτέ δεν φτάνει το ποσοστό απόδοσης των υπόλοιπων μοντέλων, εξαιτίας της γραμμικής και για αυτό αργής φύσης του SDBB μοντέλου. Ακόμα και αν το buffer πρακτικά γίνει άπειρο (για παράδειγμα, κανένα αίτημα δεν απορρίπτεται εξαιτίας της έλλειψης buffer space), η αργή λειτουργία του SDBB αναγκάζει πολλά αιτήματα να απορρίπτονται απλώς επειδή ο χρόνος έναρξης του αιτήματος έχει λήξει λόγω του χρόνου που πήρε στα αιτήματα για να εξεταστούν.

Έχει παρατηρηθεί ότι οι γραμμές στο δίκτυο εξομοιώθηκαν με latency 1ms, που είναι ένας τυπικός αριθμός για μη τοπικά δίκτυα. Μεγαλύτερες τιμές καθυστέρησης θα είχαν φυσικά καταστρεπτικές συνέπειες στην απόδοση του SDBB.

6.4 ΑΠΟΤΙΜΗΣΗ ΤΩΝ ΑΡΧΙΤΕΚΤΟΝΙΚΩΝ ΜΕ ΤΗ ΧΡΗΣΗ ΔΙΑΦΟΡΕΤΙΚΩΝ ΤΟΠΟΛΟΓΙΩΝ

Για το κύριο σύνολο των πειραμάτων μας χρησιμοποιήσαμε έναν αριθμό από διαφορετικές τοπολογίες σε συνδυασμό με τη βασική τοπολογία της Εικόνα 27. Οι υπόλοιπες τοπολογίες φαίνονται στην Εικόνα 30, και σχεδιάστηκαν έτσι ώστε όσα περισσότερα χαρακτηριστικά τοπολογίας δικτύου γίνεται, να συμπεριληφθούν και να ληφθούν υπόψη. Για παράδειγμα, ενώ η τοπολογία αστέρα ελαχιστοποιεί την απόσταση μεταξύ των BBbase και των BBedge agents, η τοπολογία δέντρου και η σειριακή τοπολογία την αυξάνουν και μεγεθύνουν την συνέπειά της στην απόδοση του Bandwidth Broker. Η τοπολογία δέντρου ξεχωρίζει κυρίως από την σειριακή (που είναι και αυτή τυπικά ένα δέντρο) στο ότι η σειριακή έχει τα πιο πιθανά bottlenecks από οποιαδήποτε άλλη τοπολογία, αφού τα περισσότερα αιτήματα πιθανότατα πρόκειται να ζητήσουν μέρος από τους πόρους των μεσαίων γραμμών. Τέλος, η τυχαία τοπολογία συνδυάζει χαρακτηριστικά και από τις τοπολογίες αστέρα, δέντρου και τη σειριακή τοπολογία με σκοπό να έχει ένα πιο ισορροπημένο σύνολο από αποτελέσματα. Για όλα τα πειράματα, εκτός και αν τονιστεί διαφορετικά, το latency των γραμμών ορίστηκε ίσο με 1ms.



Εικόνα 30 Επιπλέον τοπολογίες που χρησιμοποιήθηκαν για τα πειράματα

6.4.1 Overhead δικτύου

Ο Πίνακας 1 συνοψίζει το overhead δικτύου που προκαλείται από το καθένα Bandwidth Broker μοντέλο για κάθε μία τοπολογία, υπολογισμένο από το μέσο αριθμό πακέτων που ανταλλάσσονται για κάθε αίτημα.

Εξαιτίας της κατανεμημένης φύσης τους, υπάρχει περισσότερο overhead δικτύου για τους SDBB, PDBB σε σύγκριση με τα CBB και CFBB μοντέλα, ειδικά για την σειριακή τοπολογία και για την τοπολογία δέντρου. Η διαφορά αυτή μειώνεται σημαντικά για την τοπολογία αστέρα, επειδή η τοπολογία αστέρα ταιριάζει περισσότερο στους υλοποιημένους αλγορίθμους των SDBB και PDBB μοντέλων. Σε όλες τις περιπτώσεις, το CFBB μοντέλο παρουσιάζει την μέση εναλλακτική οδό για τα SDBB/PDBB και CBB μοντέλα.

Επίσης, αυτά τα πειράματα δείχνουν ότι το ολικό overhead δικτύου επηρεάζεται από την τοπολογία και από την θέση του BBbase agent συνδυαζόμενο από την κατανομή των QoS αιτημάτων κατά μήκος των δικτυακών κόμβων. Είναι ένα ήδη γνωστό πρόβλημα που αντιμετωπίζεται με τη μελέτη της τοπολογίας, την κατανομή της χρήσης QoS και γι' αυτό συστήνεται η εφαρμογή μιας περιοδικής βέλτιστης επιλογής του κόμβου που θα φιλοξενήσει των BBbase agent.

Model \ Topology	Serial	Star	Tree	Random
SDBB	7.65	4.76	7.39	5.99
PDBB	9.68	4.92	9.92	6.61
CBB	2.01	1.70	1.94	1.90
CFBB	3.66	2.44	3.63	3.01

Πίνακας 1 Overhead δικτύου(μέσος όρος αριθμών πακέτου ανά αίτημα)

6.4.2 Ρυθμός αποδοχής αιτημάτων

Ο Πίνακας 2 δείχνει το ποσοστό αποδοχής ανά μοντέλο για κάθε τοπολογία. Όλα τα μοντέλα επωφελούνται από την τοπολογία αστέρα και παράγουν καλύτερα αποτελέσματα. Ο λόγος είναι ότι οι γραμμές είναι κατά μέσο όρο λιγότερο φορτωμένες στην τοπολογία αστέρα από ότι στις υπόλοιπες τοπολογίες, όπου γραμμές με bottleneck εμφανίζονται.

Model \ Topology	Serial	Star	Tree	Random
SDBB	0.1514	0.2216	0.1699	0.2012
PDBB	0.2029	0.2227	0.2171	0.2130
CBB	0.2035	0.2259	0.2088	0.2106
CFBB	0.2060	0.2323	0.2083	0.2118

Πίνακας 2 Ρυθμός αποδοχής (ρυθμός αποδοχής των αιτημάτων που υποβλήθηκαν)

Γενικά οι διαφορές μεταξύ των μοντέλων είναι μάλλον μικρές, εκτός από το SDBB μοντέλο το οποίο, ειδικά για τοπολογίες με μεγαλύτερες κατά μέσο όρο αποστάσεις παρουσιάζει σημαντικά χειρότερη συμπεριφορά από τα υπόλοιπα μοντέλα. Οι μεγάλες του καθυστερήσεις που παρουσιάζει στην απάντηση των αιτημάτων προκαλούν σε πολλά από αυτά, να έχουν λήξει την χρονική στιγμή που οδηγούνται προς επεξεργασία. Αυτό φαίνεται και από τον μέσο όρο των χρόνων απόκρισης που παρέχονται στον Πίνακας 4. Κατά την διάρκεια των πειραμάτων μας, αποφασίσαμε να ερευνήσουμε το κατά πόσο το να εξαναγκάσουμε τους χρήστες να υποβάλλουν αιτήματα με ένα ελάχιστο όριο στο advance time(μια ελάχιστη χρονική περίοδο από τη στιγμή που ένα αίτημα υποβλήθηκε στη στιγμή στην οποία οι ζητούμενοι πόροι πρέπει να δεσμευτούν) θα βοηθούσε το SDBB να ξεπεράσει το πρόβλημα του περιορισμένου ρυθμού αποδοχής αιτημάτων. Παρόλα αυτά, εξαιτίας των μεγάλων χρόνων απόκρισης σε σύγκριση με τα άλλα μοντέλα, ο ρυθμός άφιξης των αιτημάτων στο SDBB buffer είναι μεγαλύτερος από τον ρυθμό επεξεργασίας (με δεδομένο ένα σταθερό ρυθμό εισερχομένων αιτημάτων, όπως στα πειράματά μας). Γι' αυτό, το SDBB μοντέλο πάντα σταδιακά θα αποτυγχάνει να επεξεργαστεί κάποια αιτήματα έγκαιρα (πριν τη χρονική στιγμή που η κράτησή τους θα ξεκινούσε).

Εξαιτίας της έλλειψης πιθανών bottleneck γραμμών, η τοπολογία αστέρα επιτρέπει το μεγαλύτερο ποσοστό αιτημάτων που θα γίνουν αποδεκτά. Από τη στιγμή που γενικά η τοπολογία πρόκειται να είναι σταθερή και η μόνη ρεαλιστική επιλογή πρόκειται να γίνει ανάμεσα στα Bandwidth Broker μοντέλα, ο Πίνακας 2προτείνει ότι για μια τυχαία τοπολογία το SDBB μοντέλο είναι το μόνο που φαίνεται μη αποδοτικό, ενώ τα υπόλοιπα παρουσιάζουν παρόμοια συμπεριφορά, που υπονοεί ότι αυτή είναι μάλλον η καλύτερη συμπεριφορά που μπορεί κανείς να περιμένει, στερούμενος τη χρήση τη χρήση πιο πολύπλοκων admission control αλγορίθμων [11] ή πιο πολύπλοκης τοποθέτησης του BBbase agent[10].

Με σκοπό την περαιτέρω μελέτη της επίδρασης του latency, επαναλάβαμε τα πειράματα για τα SDBB/PDBB μοντέλα με latency 10ms. Τέτοιο latency θα μπορούσε να εξομοιώσει έναν Bandwidth Broker που λειτουργεί σε ένα domain που καλύπτει απομακρυσμένες περιοχές, όπως για παράδειγμα ένα εθνικό δίκτυο. Όπως μπορούμε να δούμε στον Πίνακας 3, αυτή η αύξηση του latency έχει επιβλαβείς συνέπειες στην απόδοση των κατανεμημένων μοντέλων. Αυτό μας οδηγεί στο συμπέρασμα ότι για Bandwidth Brokers που λειτουργούν σε domain ευρείας περιοχής, η κεντρικοποιημένη προσέγγιση έχει το πλεονέκτημα απέναντι στην κατανεμημένη, τουλάχιστον όσο αφορά τις διαδικασίες που αφορούν το admission control που χρησιμοποιήθηκαν από τα μοντέλα που περιγράφηκαν.

Model \ Topology	Serial	Star	Tree	Random
SDBB	0.0201	0.1263	0.0265	0.0366
PDBB	0.0515	0.2099	0.0714	0.0861

Πίνακας 3 Ρυθμός αποδοχής για latency 10ms

6.4.3 Η απόκριση χρόνου των μοντέλων

Ο Πίνακας 4 παρουσιάζει τους κατά μέσο όρο χρόνους απόκρισης για όλα τα μοντέλα σε εξεταζόμενες τοπολογίες. Οι χρόνοι απόκρισης για το SDBB μοντέλο είναι τάξεις

μεγέθους μεγαλύτεροι από τους υπόλοιπους, που εξηγεί την κατώτερη απόδοση στην μέτρηση που έγινε για το ρυθμό αποδοχής και εκφράζει την ακαταλληλότητα που έχει ως μοντέλο για όλες τις περιπτώσεις με εξαίρεση τις πιο βολικές(για παράδειγμα, τοπολογία αστέρα).

Model \ Topology	Serial	Star	Tree	Random
SDBB	1919	4.077	1325	314.3
PDBB	22.19	2.520	9.382	6.041
CBB	5.502	1.597	4.097	2.879
CFBB	5.502	1.597	4.097	2.879

Πίνακας 4 Χρόνος απόκρισης(ο μέσος χρόνος που πέρασε μέχρι τη στιγμή που η απάντηση επιστρέφει στο αποστολέα αιτήματος)(msec)

Με σκοπό την καλύτερη κατανόηση της ακριβής επίδρασης των αριθμών στον Πίνακας 4, υπολογίσαμε επίσης την τυπική απόκλιση(standard deviation) των χρόνων απόκρισης, το οποίο παρουσιάζεται στον Πίνακας 5. Τα αποτελέσματα αυτά δείχνουν ότι οι χρόνοι απόκρισης του SDBB όχι μόνο είναι πολύ μεγαλύτεροι από τους υπόλοιπους των άλλων μοντέλων κατά μέσο όρο, αλλά επίσης έχουν πολύ μεγαλύτερη διακύμανση. Σε όλες τις περιπτώσεις, τα CBB και CFBB μοντέλα είναι πανομοιότυπα όπως αναμένεται, από τη στιγμή που οι διαδικασίες επεξεργασίας αιτημάτων είναι ακριβώς ίδιες.

Model \ Topology	Serial	Star	Tree	Random
SDBB	757	2.180	591.7	224.1
PDBB	19.83	1.485	6.404	4.726
CBB	2.576	0.8128	2.176	1.770
CFBB	2.576	0.8128	2.176	1.770

Πίνακας 5 τυπική απόκλιση του χρόνου απόκρισης(10^{-3})

Η κοντινή γειτνίαση των κόμβων στον BBbase agent στην τοπολογία αστέρα ευνοεί το PDBB κατανεμημένο μοντέλο, γεγονός που μειώνει τη διαφορά με τα CBB/CFBB κεντρικοποιημένα μοντέλα. Αλλά και για τις υπόλοιπες τοπολογίες ο χρόνος απόκρισης παραμένει το πολύ 2-4 φορές το χρόνο των CBB\CFBB μοντέλων, που μπορεί να είναι μια λογική εξισορρόπηση για τα κατανεμημένα πλεονεκτήματα του PDBB μοντέλου.

ΚΕΦΑΛΑΙΟ 7: ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΓΑΣΙΑ

Η κύρια δουλειά μας επικεντρώθηκε στην υλοποίηση και στην σύγκριση των τεσσάρων διαφορετικών Bandwidth Broker μοντέλων χρησιμοποιώντας τον ns-2. Οι Bandwidth Brokers παρέχουν μια QoS υπηρεσία στην κίνηση που υποβλήθηκε χρησιμοποιώντας την ευρέως γνωστή DiffServ λειτουργικότητα. Τα τεστ που περιγράφηκαν παραπάνω, παρουσιάζουν τις διαφορές ανάμεσα στους Bandwidth Broker agents. Τα αποτελέσματα μας διευκρίνισαν τις εξισορροπήσεις που έγιναν ανάμεσα στην κεντρικοποιημένη και την κατανεμημένη προσέγγιση, στην ανεκτική σε σφάλματα και στην απλή κεντρικοποιημένη έκδοση, και ανάμεσα σε μία σειριακή και σε μία παράλληλα κατανεμημένη μέθοδο επεξεργασίας αιτημάτων. Επιπρόσθετα, είδαμε ότι με τη χρήση ενός μικρού buffer στον BBbase agent, μπορούμε αισθητά να αυξήσουμε το ποσοστό των θετικών αιτημάτων.

Στην μελλοντική μας εργασία σκοπεύουμε να επεκτείνουμε την εμβέλεια και την ποικιλία των υλοποιήσεών μας, καθώς επίσης και των πειραματικών αποτελεσμάτων. Συγκεκριμένα, σκοπεύουμε να εμπλουτίσουμε το CFBB μοντέλο με προσαρμοστικές ικανότητες που θα του επιτρέψουν να είναι ικανό να μετακινεί τη θέση του Base Bandwidth Broker agent. Η λειτουργικότητα αυτή μπορεί να είναι χρήσιμη και για να ξεπεραστεί μια αποτυχία ενός κόμβου που φιλοξενεί τον Base Bandwidth Broker agent, και επίσης να είναι χρήσιμη στην τοποθέτηση του BBbase agent σύμφωνα με την καταλληλότερη (ή προσεγγιστικά καταλληλότερη) τοποθέτηση που μπορεί να υπολογιστεί όπως παρουσιάζεται στο [12]. Επίσης, σκοπεύουμε να αναλύσουμε πιο εξεζητημένους admission control αλγορίθμους που λαμβάνουν υπόψη την χρήση του δικτύου και το ρυθμό αποδοχής που έχει επιτευχθεί [13]. Τέλος, σκοπεύουμε να μελετήσουμε και να υλοποιήσουμε λειτουργικότητα με την οποία θα εξομοιώνουμε την inter-domain λειτουργία, που προτείνει επιπλέον προκλήσεις, όπως το peering μοντέλο, λειτουργίες εύρεσης διαδρομών(path finding procedures) και η αποτελεσματικότητά τους, και τα SLAs ανάμεσα στα domains και στις δυναμικές διαπραγματεύσεις τους.

ΠΑΡΑΡΤΗΜΑ Ι: ΑΝΑΦΟΡΕΣ

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] S. Vegesna, 'IP Quality of Service: the complete resource for understanding and deploying IP quality of service for Cisco networks', Cisco Press, 2001

[2] R. Braden, D. Clark, "Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994

[3] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, "An Architecture for Differentiated Services", RFC 2475, December 1998

[4] http://www.isi.edu/nsnam/ns/

[5] J. Heinanen, R. Guerin, "A single rate three color marker", RFC 2697, September 1999

[6] J. Heinanen, R. Guerin, "A Two Rate Three Color Marker", RFC 2698, September 1999

[7] W. Fang, N. Seddigh, "A Time Sliding Window Three Color Marker (TSWTCM)," RFC 2859, June 2000

[8] Ν. Στάθης, " Ποιότητα Υπηρεσίας σε IP Δίκτυα " Διπλωματική εργασία, Μάρτιος 2004

[9] Pierre L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. Operations Research, 47(1):159–164, 1999.

[10] C. Bouras, D. Primpas, "An admission control and deployment optimization algorithm for an implemented distributed Bandwidth Broker in a simulation environment", 4th International Conference on Networking – ICN 2005, Reunion Island, France, 17 - 21 April 2005, pp. 766 - 773

[11] C. Bouras, K. Stamos, "An Adaptive Admission Control Algorithm for Bandwidth Brokers", 3rd IEEE International Symposium on Network Computing and Applications (NCA04), Cambridge, MA, USA, 30 August - 1 September 2004, pp. 243 - 250

[12] C. Bouras, D. Primpas, "An admission control and deployment optimization algorithm for an implemented distributed Bandwidth Broker in a simulation environment", 4th International Conference on Networking – ICN 2005, Reunion Island, France, 17 - 21 April 2005, pp. 766 - 773

[13] C. Bouras, K. Stamos, "An Adaptive Admission Control Algorithm for Bandwidth Brokers", 3rd IEEE International Symposium on Network Computing and Applications (NCA04), Cambridge, MA, USA, 30 August - 1 September 2004, pp. 243 – 250

[14] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin "Resource ReSerVation Protocol (RSVP)", RFC 2205, September 1997

[15] K.Ramakrishman and S.Floyd, "A proposal to Add Explicit Congestion Notification (ECN) to IP", RFC 2481, January 1999

[16] Δίκτυα Υπολογιστών, Τρίτη Έκδοση, Andrew Tanenbaum

[17] http://www.cisco.com

[18] V. Jacobson, K. Nicholsm K.Poduri, "An Expedited Forwarding PHB", RFC 2598, June 1999

[19] S.Shenker, R.Guerin, "Specification of the Guaranteed Quality of Service", RFC 2212, September 1997

[20] http://www.geant.net

[21] Chahed T. "IP QoS Parameters", private communication to TF-NGN November 2000

[22] M. Campanella, Implementation Architecture specification for the Premium IP service", Deliverable D2.1-Addendum 1, SEQUIN Project (IST-1999-20841)

[23] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group", RFC 2597, 1999

[24] RFC 2905 "AAA Authorization Application Examples", J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, D. Spence, August 2000

[25] Service Level Agreement Trading for the Differentiated Services Architecture George Fankhauser, David Schweikert, Bernhard Plattner

[26] RFC 2748 "The COPS (Common Open Policy Service) Protocol" D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry, January 2000

[27] RFC 1157 "A Simple Network Management Protocol (SNMP)" J. Case, M. Fedor, M. Schoffstall, J.Davin, May 1990

[28] RFC 854 "Telnet Protocol Specification" J. Postel, J. Reynolds, May 1983

[29] RFC 3588 "Diameter Base Protocol" P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, September 2003

[30] http://qbone.internet2.edu/bb/

[31] Pierre L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. Operations Research, 47(1):159–164, 1999.

[32] RFC 3084 "COPS Usage for Policy Provisioning" K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith, March 2001.

[33] RFC 2138 "Remote Authentication Dial In User Service (RADIUS)" C. Rigney, A. Rubens, W. Simpson, S. Willens, April 1997.

[34] http://www.vovida.org/

[35] www.wsspl.com

[36] www.intel.com/technology/magazine/communications/nc05021.pdf

WEB SITES

[1] ru6.cti.gr

[2] http://www.isi.edu/nsnam/ns/

[3] http://www.cisco.com

[4] http://www.geant.net

[5] http://qbone.internet2.edu/bb/

ΠΑΡΑΡΤΗΜΑ ΙΙ: ΕΓΚΑΤΑΣΤΑΣΗ ΚΩΔΙΚΑ

ΕΓΚΑΤΑΣΤΑΣΗ ΚΩΔΙΚΑ

Ακολουθεί η διαδικασία εγκατάστασης του κώδικα ο οποίος βρίσκεται στο cd που συνοδεύει τη διπλωματική εργασία.

Τα αρχεία που μας ενδιαφέρουν για την εγκατάσταση βρίσκονται όλα στο αρχείο zip με το όνομα : "QoS_ns_codes"

Κάνουμε unzip το αρχείο και ακολουθούμε την παρακάτω διαδικασία:

- Προσθέτουμε/αντικαθιστούμε τα αρχεία στα αντίστοιχα paths του φακέλου ns-2.26
- Διαγράφουμε όλα τα αρχεία με κατάληξη *.o των αρχείων που αντικαταστήσαμε για παράδειγμα: rm packet.o
- Προσθέτουμε τα ονόματα των αρχείων που προσθέσαμε και όχι αντικαταστήσαμε μέσα στο αρχείο MAKEFILE
  για παράδειγμα κοίτα το αρχείο MAKEFILE που περιλαμβάνεται στον .zip φάκελο,
  στη γραμμή OBJ_CC=\
  προσθέτουμε το εξής :
  apps/Request.o apps/List_Requests.o apps/BBbase.o apps/BBedge.o\
  apps/BBagent.o apps/BBagent_Participants.o apps/DistributedBBbase.o apps/CentralBBbase.o\
  apps/vbr.o apps/H323.o apps/SupFRP.o common/myClassify.o \
- Στη συνέχεια θα κάνουμε make depend
- Και τέλος make