ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ Τ.Ε.

Πτυχιακή εργασία με τίτλο:

Συγκριτική μελέτη ποιότητας υπηρεσιών (QoS) προσομοιωτών (simulators)

Σπουδάστρια: **Κολιοκώτση Βασιλική**

Επιβλέπων Καθηγητής: **Βασιλειάδης Δημήτριος**

Άρτα 2016

Περίληψη

Σκοπός αυτής της πτυχιακής εργασίας είναι να παρουσιάσει και να συγκρίνει την υλοποίηση της Ποιότητας Υπηρεσίας των προσομοιωτών δικτύου ns2 και OMNeT++. Αρχικά, γίνεται μια εισαγωγή στην Ποιότητα Υπηρεσίας και στις μορφές κυκλοφορίας που εγγυάται ένα δίκτυο. Στη συνέχεια, αναλύεται η Θεωρία Ουράς καθώς και οι αλγόριθμοι χρονοδρομολόγησης και διαχείρισης ουράς. Επίσης, αναλύεται ο τρόπος λειτουργίας των δύο αρχιτεκτονικών, ολοκληρωμένων και διαφοροποιημένων υπηρεσιών, της Ποιότητας Υπηρεσίας. Τέλος, εξετάζεται πως υλοποιείται η Ποιότητα Υπηρεσίας σε κάθε προσομοιωτή χωριστά και διατυπώνεται μια σύγκριση μεταξύ τους.

Λέξεις-Κλειδιά: Ποιότητα Υπηρεσίας, κουβάς κουπονιών, ολοκληρωμένες υπηρεσίες, πρωτόκολλο RSVP, διαφοροποιημένες υπηρεσίες, κωδικοσημείο DS, συμπεριφορές ανά κόμβο

Abstract

The purpose of this project is to presents and compares the implementation of Quality of Service (QoS) of ns2 and OMNeT++ network simulators. Initially, there is an introduction to QoS and traffic forms guaranteeing a network. In addition, the Queuing Theory, the scheduling algorithms and the queue management algorithms analyzed. Also, the operating mode of the two architectures, integrated and differentiated services, of the QoS analyzed. Finally, it examines how QoS is implemented in each simulator separately and provides a comparison between them.

Key-Words: Quality of Service, token bucket, integrated services, RSVP protocol, differentiated services, DS codepoint, per hop behaviors

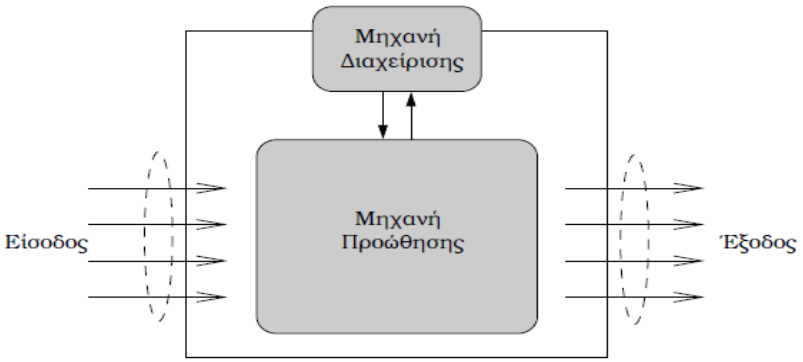
Λίστα σχημάτων

ΠΕΡΙΕΧΟΜΕΝΑ

ΚΕΦΑΛΑΙΟ 1:Ποιότητα Υπηρεσίας (QualityofService)

1.1 Εισαγωγή[1][4][7][8]

Η εξάπλωση του διαδικτύου (Internet) έχει κάνει δυνατή τη διασύνδεση πολλών και διαφορετικών δικτύων με τη χρήση ενός κοινού συνόλου πρωτοκόλλων επικοινωνίας που ονομάζεται InternetProtocolSuite. Όταν τα πακέτα εισέρχονται σε ένα IP δίκτυο διασχίζουν μια σειρά από δρομολογητές, οι οποίοι τα προωθούν στους επόμενους κόμβους (hops), μέχρι να φτάσουν στον προορισμό τους. Οι πολλαπλές διεπαφές εισόδου ενός δρομολογητή δέχονται πακέτα από άλλους δρομολογητές και ο μηχανισμός προώθησης τα περνάει στις κατάλληλες διεπαφές εξόδου με βάση τη διεύθυνση προορισμού (Σχήμα 1). Η συμπεριφορά του μηχανισμού προώθησης ελέγχεται από τον μηχανισμό διαχείρισης.



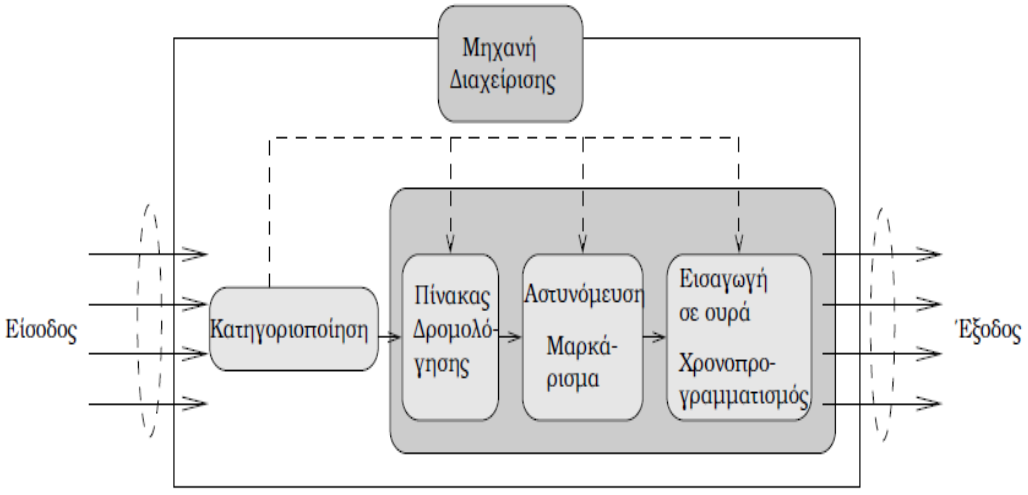
Σχήμα 1: Λειτουργικές οντότητες δρομολογητή

Σε ένα IP δίκτυο και κατά συνέπεια στο διαδίκτυο, η κύρια υπηρεσία που παρέχεται είναι η προώθηση των πακέτων με τη μέθοδο της καλύτερης προσπάθειας (best-effort). Τα πακέτα μεταδίδονται με τη σειρά που εισέρχονται στον δρομολογητή και δεν γίνεται κανένας διαχωρισμός ανάμεσά τους. Η υπηρεσία αυτή δεν δίνει εγγυήσεις για την παράδοση των δεδομένων ή την παροχή προτεραιότητας στην εξυπηρέτηση κάποιων εφαρμογών, με συνέπεια, η εξυπηρέτηση του χρήστη να εξαρτάται από τον τρέχοντα φόρτο εργασίας του δικτύου. Για την επικοινωνία δεδομένων, όπως π.χ. η μεταφορά αρχείων, η αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου, η πλοήγηση στο www κτλ. η παραπάνω υπηρεσία έχει αποδειχθεί πολύ ικανοποιητική.

Σύγχρονες εφαρμογές του διαδικτύου με αδιάλειπτη μετάδοση βίντεο, φωνής και πολυμεσικής πληροφορίας, όπως η IP τηλεφωνία (VoIP) και η τηλεδιάσκεψη (videoconferencing), απαιτούν μια ειδική μεταχείριση από το δίκτυο. Οι εφαρμογές αυτές απαιτούν ένα μηχανισμό διαχείρισης των πόρων του δικτύου και την παροχή κάποιας ποιότητας υπηρεσίας (QualityofService –QoS) για να εξασφαλίσουν μια καλή απόδοση. Λέγοντας παροχή ποιότητας υπηρεσίας αναφερόμαστε στη δυνατότητα που υπάρχει οι ρυθμοί μετάδοσης, οι ρυθμοί λαθών και άλλα χαρακτηριστικά του δικτύου να μετρηθούν, να βελτιωθούν και σε κάποιο βαθμό να εγγυηθούν εκ των προτέρων.

Η ποιότητα υπηρεσίας καθιστά αναγκαίο τον ανασχεδιασμό του μηχανισμού προώθησης ενός δρομολογητή ώστε να αντιμετωπίζει αποτελεσματικότερα το «πότε» στέλνονται τα πακέτα από τις διεπαφές εξόδου στον προορισμό τους. Σε αυτή την περίπτωση, τα πακέτα που εισέρχονται σε έναν δρομολογητή περνάνε από τρεις κύριες επεξεργασίες (Σχήμα 2).

1. Κατηγοριοποίηση (classification) του πακέτου και αναζήτηση (lookup) της διεπαφής εξόδου στον πίνακα δρομολόγησης. Εδώ καθορίζεται η κατηγορία στην οποία ανήκει το πακέτο.
2. Αστυνόμευση (policing) και μαρκάρισμα (marking). Με την αστυνόμευση των πακέτων ελέγχεται αν η ροή των πακέτων υπακούει στα χαρακτηριστικά που είχαν προ-συμφωνηθεί με τον πάροχο του δικτύου (δυναμική ή στατική δρομολόγηση). Στη συνέχεια και ανάλογα με το αποτέλεσμα των σταδίων κατηγοριοποίησης και αστυνόμευσης , κάθε πακέτο μαρκάρεται σύμφωνα με τις εσωτερικές διαδικασίες του δρομολογητή ή του δικτύου το οποίο εποπτεύει.
3. Εισαγωγή σε ουρά εξόδου (queuing) και χρονοπρογραμματισμός (scheduling). Για να μεταδοθούν τα πακέτα στο φυσικό μέσο από την εξερχόμενη διεπαφή εισάγονται στην αντίστοιχη ουρά της κατηγορίας τους και στη συνέχεια ο χρονοπρογραμματιστής καθορίζει από ποια ουρά θα μεταδοθεί το επόμενο πακέτο.



Σχήμα 2: Λειτουργικές οντότητες δρομολογητή που υποστηρίζει ποιότητα υπηρεσίας

1.2 Μετρικές ποιότητας[4][7][9]

Παρακάτω παρουσιάζονται οι μετρικές τις οποίες προσπαθεί να εγγυηθεί η Ποιότητα Υπηρεσίας:

- **Χωρητικότητα (bandwidth):** είναι ο ρυθμός μετάδοσης των δεδομένων από τον έναν χρήστη στον άλλον και μετριέται σε bitspersecond (bps). Όσο αυξάνεται το εύρος ζώνης, τόσο αυξάνονται και ο ρυθμός μετάδοσης των δεδομένων. Η χωρητικότητα χαρακτηρίζεται από τις παρακάτω ποσότητες:
    - Μέγιστο μέγεθος καταιγισμού (maximumburstsize): ο μέγιστος αριθμός πακέτων που μπορούν να βρεθούν στην ουρά του δρομολογητή χωρίς να απορριφθούν.
    - Μέγιστη χωρητικότητα (peakbandwidth): η ανώτατη επιτρεπόμενη τιμή της χωρητικότητας που επιτρέπεται μια ροή να διατηρήσει σταθερή.
    - Ελάχιστη εγγυημένη χωρητικότητα (minimumguaranteedbandwidth)
    - Μέση χωρητικότητα (averagebandwidth): η μέση τιμή της χωρητικότητας που υπολογίζεται διαιρώντας τον αριθμό των bytes που μεταδόθηκαν προς το συγκεκριμένο χρονικό διάστημα.

  Το διαθέσιμο bandwidth είναι αποτέλεσμα της χωρητικότητας του καναλιού μετάδοσης, της τεχνολογίας μετάδοσης που χρησιμοποιείται και του ενδεχόμενου καταμερισμού που μπορεί να πραγματοποιείται στο συνολικό διαθέσιμο bandwidth λόγω της εξυπηρέτησης πολλών χρηστών. Το QoS δεν δημιουργεί bandwidth αλλά διαχειρίζεται το υπάρχον ανάλογα με τις απαιτήσεις της κάθε εφαρμογής.

- **Καθυστέρηση (delay):** είναι ο χρόνος που χρειάζεται ένα πακέτο για να μεταδοθεί, από την έναρξη της μετάδοσης του πρώτου bit ενός πακέτου ως και τη λήψη του τελευταίου bit του πακέτου από τον παραλήπτη. Η καθυστέρηση είναι το άθροισμα των παρακάτω επιμέρους χρόνων:
    - Χρόνος μετάδοσης: ο χρόνος που απαιτείται για την τοποθέτηση κάθε bit ενός πακέτου στο κανάλι μετάδοσης και είναι ανάλογος της ταχύτητας του καναλιού. Είναι ο λόγος του μεγέθους του μεταδιδόμενου πακέτου σε bit προς το διαθέσιμο bandwidth.
    - Χρόνος διάδοσης: ο χρόνος που χρειάζεται το πρώτο bit ενός πακέτου να φτάσει στον προορισμό του και εξαρτάται από την τεχνολογία μετάδοσης και την απόσταση. Είναι ο λόγος της απόστασης του καναλιού μετάδοσης προς την ταχύτητα διάδοσης μέσα στο κανάλι.
    - Χρόνος επεξεργασίας: ο χρόνος που δαπανάται σε ένα δρομολογητή του δικτύου κατά την επεξεργασία του πακέτου. Είναι ανάλογη του πλήθους και της υπολογιστικής πολυπλοκότητας των εργασιών που εκτελούνται και αντιστρόφως ανάλογη της διαθέσιμης υπολογιστικής ισχύος του δρομολογητή.

- Χρόνος αναμονής: ο συνολικός χρόνος αναμονής ενός πακέτου στις ουρές εισόδου και εξόδου ενός δρομολογητή, ώστε να επεξεργαστεί ή να αποσταλεί. Είναι ανάλογος της πληρότητας των ουρών κατά την άφιξη του πακέτου στον δρομολογητή.

- **Διακύμανση καθυστέρησης πακέτων (PacketDelayVariation –PDV) ή jitter:** είναι η διαφορά της καθυστέρησης μεταξύ δυο διαδοχικών πακέτων. Πολλές εφαρμογές (π.χ. πραγματικού χρόνου) χρειάζονται να διατηρείται το jitter σε χαμηλά επίπεδα για να έχουν καλή απόδοση.

- **Απώλεια πακέτων (packetloss):** είναι το ποσοστό των πακέτων που μεταδόθηκαν από την πηγή, αλλά δεν λήφθηκαν ποτέ από τον παραλήπτη τους, ή λήφθηκαν με λάθη που αδυνατεί να διορθώσει ο κώδικας ανίχνευσης και διόρθωσης λαθών του δικτύου. Η απώλεια πακέτων μπορεί να ευθύνεται σε αποτυχία του δικτύου (κόμβων, συνδέσμων), στη συμφόρηση του δικτύου και στην παρουσία υψηλού επιπέδου θορύβου στο κανάλι κατά τη μετάδοση. Η απόδοση των εφαρμογών υποβαθμίζεται λόγω της απώλειας πακέτων. Επιπλέον, η αναμετάδοση των χαμένων πακέτων δυσχεραίνει την λειτουργία κάποιων εφαρμογών (π.χ. τηλεδιάσκεψη).

1.3 Μοντέλα ποιότητας υπηρεσίας[1][4]

Τα δυο μοντέλα παροχής ποιότητας υπηρεσίας που έχουν προταθεί για τα IP δίκτυα είναι οι Ολοκληρωμένες Υπηρεσίες (IntegratedServices –IntServ) και οι Διαφοροποιημένες Υπηρεσίες (DifferentiatedServices –DiffServ). Το μοντέλο IntServ είναι προγενέστερο και βασίζεται στην δέσμευση πόρων σε κάθε ροή που εξυπηρετεί χρησιμοποιώντας το πρωτόκολλο σηματοδοσίας RSVP. Με αυτόν τον τρόπο το δίκτυο εγγυάται στις εφαρμογές τελικού χρήστη τα χαρακτηριστικά ποιότητας που αυτές απαιτούν. Το μοντέλο DiffServ βασίζεται στην ταξινόμηση των ροών σε κατηγορίες με βάση τις απαιτήσεις κάθε εφαρμογής για συγκεκριμένα χαρακτηριστικά ποιότητας. Έτσι, η κυκλοφορία διαφοροποιείται εμπεριέχοντας κάποιες πιο ‘προνομιούχες’ κατηγορίες σε σχέση με άλλες κατηγορίες.

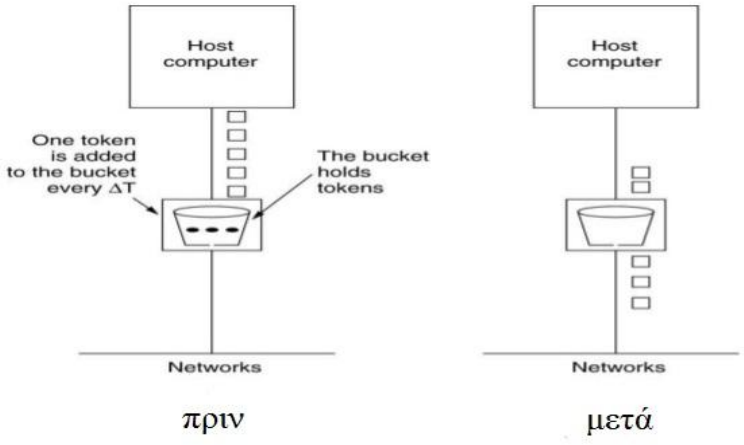
1.4 Μορφοποίηση κυκλοφορίας[3][11]

Τα δίκτυα μπορούν να δώσουν εγγυήσεις με την προϋπόθεση όμως ότι θα γνωρίζουν τι κυκλοφορία εγγυούνται. Η κυκλοφορία στα δίκτυα δεδομένων φτάνει σε μη ομοιόμορφους ρυθμούς καθώς έχει τη μορφή ριπών που κάνει τον ρυθμό κυκλοφορίας να ποικίλει. Οι ριπές κυκλοφορίας είναι δύσκολές στο χειρισμό γιατί μπορεί να γεμίσουν τις μνήμες προσωρινής αποθήκευσης με συνέπεια την απώλεια πακέτων.

Με τον όρο μορφοποίηση κυκλοφορίας (trafficshaping) αναφερόμαστε στη τεχνική που ρυθμίζει τον μέσο ρυθμό και τις ριπές μιας εισερχόμενης ροής δεδομένων στο δίκτυο. Πρόκειται για μια μέθοδο ελέγχου συμφόρησης που διαμορφώνει την ροή δεδομένων προτού γίνει η είσοδος των πακέτων στο δίκτυο. Η μορφοποίηση είναι σημαντική για δεδομένα πραγματικού χρόνου που έχουν αυστηρές απαιτήσεις ως προς την ποιότητα υπηρεσιών.

Η μορφοποίηση κυκλοφορίας επιτυγχάνεται με δυο τύπους αλγορίθμων: τον TokenBucket (κουβάς κουπονιών) και τον LeakyBucket (τρύπιος κουβάς). Παρακάτω θα δούμε τον καθένα αναλυτικά.

- Ο αλγόριθμος **TokenBucket (κουβάς κουπονιών)** δίνει τη δυνατότητα στο ρυθμό εξόδου να ποικίλει ανάλογα με το μέγεθος της ριπής. Στο παρακάτω σχήμα βλέπουμε πως λειτουργεί ο αλγόριθμος αυτός.



Σχήμα 3: Token Bucket

1) Σε κάθε χρόνο Δt προστίθεται ένα κουπόνι (token) μέσα στον κουβά.
2) Ο κουβάς κρατάει όσα κουπόνια χωράει. Αφού γεμίσει ο κουβάς, τα κουπόνια που έρχονται απορρίπτονται.
3) Όταν ένα πακέτο mbytes φτάνει, τότε m κουπόνια μεταφέρονται από τον κουβά και το πακέτο στέλνεται στο δίκτυο.
4) Αν τα διαθέσιμα κουπόνια στον κουβά είναι λιγότερα από τα bytes του πακέτου που φτάνει, τότε το πακέτο ορίζεται ως μη συμμορφωμένο (nonconformant).

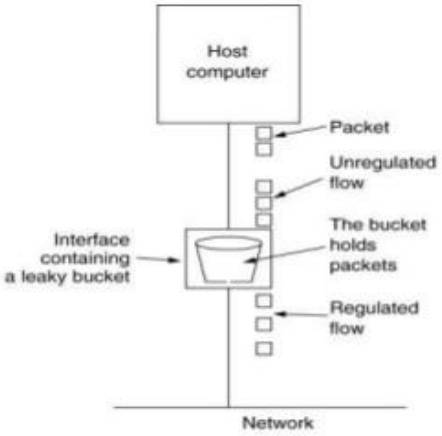
Ένα μη συμμορφωμένο πακέτο ίσως αποθηκευτεί προσωρινά για μετέπειτα μετάδοση όταν θα υπάρχουν επαρκή κουπόνια στον κουβά.

Το μέγεθος τη ριπής (burstlength) S υπολογίζεται από τον παρακάτω τύπο:

$$C+\rho S=MS$$

Όπου C: η μέγιστη χωρητικότητα του κουβά, M: ο μέγιστος ρυθμός εξόδου και ρ: ο ρυθμός άφιξης.

❖ Ο αλγόριθμος **LeakyBucket (τρύπιος κουβάς)** χρησιμοποιείται για να ελεγχθεί ο ρυθμός σε ένα δίκτυο. Ο ρυθμός εισόδου εδώ μπορεί να ποικίλει αλλά ο ρυθμός εξόδου παραμένει σταθερός. Ο αλγόριθμος υλοποιείται ως μια μεμονωμένη ουρά διακομιστή με συνεχή χρόνο εξυπηρέτησης. Ας δούμε τώρα πως λειτουργεί αυτός ο αλγόριθμος (Σχήμα 4).



Σχήμα 4: Leaky Bucket

1) Ο μετρητής αρχικοποιείται στο 'n' σε κάθε tick του ρολογιού.
2) Αν το 'n' είναι μεγαλύτερο από το μέγεθος του πρώτου πακέτου της ουράς τότε το πακέτο στέλνεται στο δίκτυο και μειώνεται ο μετρητής κατά μέγεθος πακέτου. Αυτό επαναλαμβάνεται μέχρι το 'n' του μετρητή να είναι μικρότερο από το μέγεθος του πακέτου.
3) Ο μετρητής επαναφέρεται και πηγαίνουμε στο 1^{ο} βήμα.

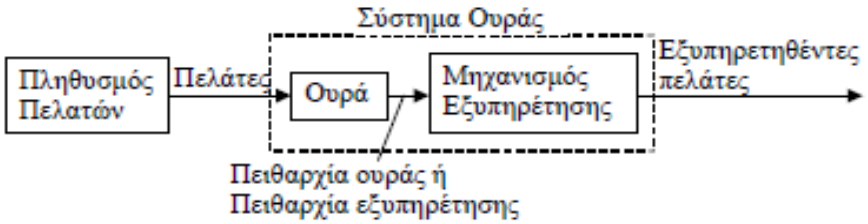
Αν ένα πακέτο φτάσει όταν ο κουβάς γεμίσει τότε το πακέτο είτε θα απορριφθεί είτε θα διατηρηθεί σε μια ουρά μέχρι να αδειάσει αρκετά ο κουβάς.

Συγκρίνοντας τους δυο αλγορίθμους μορφοποίησης βγάζουμε τα εξής συμπεράσματα:

- Ο κουβάς κουπονιών εξαρτάται από τα κουπόνια, ενώ ο τρύπιος κουβάς όχι.
- Στον κουβά κουπονιών τα πακέτα μεταδίδονται μόνο αν υπάρχουν αρκετά κουπόνια. Στον τρύπιο κουβά έχουμε συνεχή μετάδοση πακέτων.
- Αν ο κουβάς είναι γεμάτος, ο κουβάς κουπονιών απορρίπτει τα επιπλέον κουπόνια και όχι τα πακέτα ενώ ο τρύπιος κουβάς απορρίπτει τα επιπλέον πακέτα.
- Ο κουβάς κουπονιών αποθηκεύει κουπόνια για να στείλει μεγάλες ριπές. Ο τρύπιος κουβάς δεν αποθηκεύει κουπόνια.
- Ο κουβάς κουπονιών επιτρέπει στις μεγάλες ριπές να στέλνουν γρηγορότερα ενώ ο τρύπιος κουβάς στέλνει τα πακέτα με σταθερό ρυθμό.

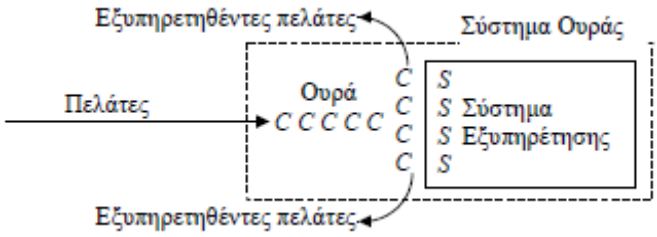
ΚΕΦΑΛΑΙΟ 2: Θεωρία Ουράς (Queuing Theory)

2.1 Συστήματα ουρών[12][13][14]



Σχήμα 5: Η γενική δομή των συστημάτων ουράς.

Η Θεωρία Ουρών ασχολείται με τις διαδικασίες άφιξης και αναχώρησης στοιχείων στις ουρές καθώς και με τα στατιστικά χαρακτηριστικά των ουρών, όπως το πλήθος των στοιχείων κατά την άφιξη και την αναχώρηση, την διάρκεια αναμονής των εισερχόμενων στοιχείων πριν την εξυπηρέτηση από την ουρά, η πιθανότητα η ουρά να γεμίσει κτλ. Στα δίκτυα Η/Υ τα στοιχεία είναι τα δεδομένα, τα οποία μπορεί να είναι και υπό τη μορφή πακέτων. Τα δεδομένα μπορούν να κινηθούν σε νοητούς δρόμους (circuitswitchednetworks), εγκαθιδρύοντας μια επικοινωνία από το ένα σημείο προς το άλλο σημείο (end-to-endconnection) ή με μεταγωγή πακέτων (packetswitching), όπου το μήνυμα διασπάται σε μικρότερα πακέτα όπου το καθένα μπορεί να ακολουθήσει διαφορετική διαδρομή. Σε αυτή την περίπτωση τα πακέτα επανασυναρμολογούνται στον τελικό προορισμό δημιουργώντας το αρχικό μήνυμα.



Σχήμα 6: Μια στοιχειώδης διαδικασία του συστήματος ουράς

Στο Σχήμα 5 βλέπουμε τη γενική δομή ενός συστήματος ουράς που ξεκινάει με την άφιξη των πελατών στο σύστημα. Ο πληθυσμός των πελατών, οι οποίοι είναι πιθανοί πελάτες στο σύστημα, μπορεί να είναι άπειρος (ο ρυθμός αφίξεων δεν επηρεάζεται από τον αριθμό των πελατών στο σύστημα) ή περιορισμένος (ο ρυθμός αφίξεων επηρεάζεται από τον αριθμό των πελατών στο σύστημα). Σ' αυτό το σημείο σημαντικά είναι η κατανομή χρόνου (π.χ. εκθετική, τυχαία) μεταξύ δυο διαδοχικών αφίξεων και η συμπεριφορά των πελατών (π.χ. αποχώρηση λόγω αναμονής σε μεγάλη ουρά). Αν όλοι οι εξυπηρετητές είναι απασχολημένοι, τότε οι πελάτες περιμένουν σε μια ουρά αναμονής μέχρι να επιλεγούν σύμφωνα με κάποια πειθαρχία ουράς (π.χ. εξυπηρέτηση σύμφωνα με τη σειρά άφιξης (FIFO- FirstInFirstOut), κατά τυχαίο τρόπο, σύμφωνα με συγκεκριμένη προτεραιότητα). Στη συνέχεια, (Σχήμα 6) ο πελάτης μπορεί να εξυπηρετηθεί από ένα ή περισσότερα συστήματα εξυπηρέτησης και κάθε σύστημα μπορεί να έχει μία ή περισσότερες θέσεις εξυπηρέτησης. Η κατανομή του χρόνου εξυπηρέτησης για κάθε θέση μπορεί να είναι π.χ. εκθετική. Τέλος, όλοι οι εξυπηρετηθέντες πελάτες εξέρχονται από το σύστημα.

2.1.1 Ο νόμος του Little[12][14]

Σύμφωνα με τον νόμο του Little, ο μέσος όρος των στοιχείων στην ουρά (δηλαδή το μέσο μήκος της) ισούται με τον ρυθμό άφιξης των στοιχείων στην ουρά επί τον μέσο χρόνο αναμονής των στοιχείων στην ουρά:

$$\bar{L} = \lambda \cdot \bar{W}$$

Ο παραπάνω τύπος ισχύει για όλους τους τύπους ουράς και λέγεται, επίσης, 'Μικρός Τύπος'.

2.1.2 Συμβολισμός Kendall[12][14]

Ένας Βρετανός στατιστικός, ο David Kendall ανέπτυξε ένα σύντομο συμβολισμό προκειμένου να δηλώσει τους διαφόρους τύπους ουράς. Αρχικά, ο συμβολισμός περιείχε τρεις παράγοντες

A/B/C

- A:πρότυπο αφίξεων πελατών
- B: πρότυπο εξυπηρέτησης
- C:αριθμός εξυπηρετητών

Όμως, στη συνέχεια επεκτάθηκε για να συμπεριλάβει κι άλλα χαρακτηριστικά των συστημάτων ουράς

A/B/X/Y/Z

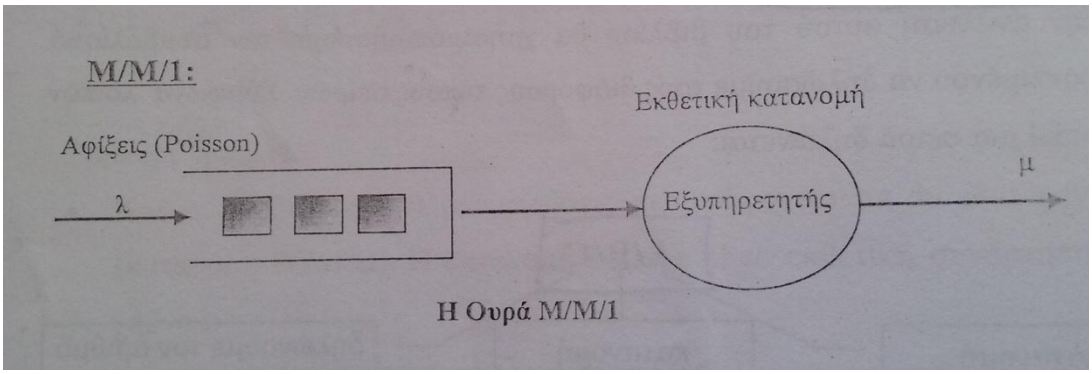
- A:πρότυπο αφίξεων πελατών
- B: πρότυπο εξυπηρέτησης
- X: αριθμός εξυπηρετητών
- Y: χωρητικότητα συστήματος (μέγιστος αριθμός πελατών που μπορούν να φιλοξενηθούν στο σύστημα)
- Z: πληθυσμός πελατών

όπου A,B μπορεί να είναι

- M: εκθετική κατανομή (κατανομή Poisson –Markov)
- G: γενική κατανομή στην οποία η εξυπηρέτηση γίνεται σε αυθαίρετο χρόνο
- D: εκφυλισμένη ή προσδιοριστική κατανομή στην οποία η εξυπηρέτηση γίνεται σε σταθερό (καθορισμένο) χρόνο.

Πολλές φορές που ο πληθυσμός πελατών του συστήματος θεωρείται άπειρος, το σύμβολο Z παραλείπεται. Επίσης, όταν η χωρητικότητα του συστήματος θεωρείται άπειρη το σύμβολο Y παραλείπεται κι αυτό.

2.1.3 Ουρά M/M/1[12]



Σχήμα 7: Ουρά M/M/1

Τα χαρακτηριστικά της ουράς αυτής είναι η κατανομή Poisson που χρησιμοποιείται από τις αφίξεις, η εκθετική κατανομή που χρησιμοποιείται από τις εξυπηρετήσεις και η χρήση ενός εξυπηρετητή με αλγόριθμο δρομολόγησης FIFO. Στην περίπτωση που οι αφίξεις και οι εξυπηρετήσεις δεν έχουν σημαντική χρονική διαφορά, η πιθανότητα μιας κατάστασης μπορεί να ορισθεί σαν την πιθανότητα τα στοιχεία να βρίσκονται σε μια ουρά κάθε χρονική στιγμή. Με τον όρο καταστάσεις μιας ουράς αναφερόμαστε στον αριθμό των στοιχείων που αναμένουν για εξυπηρέτηση. Όταν η

κατάσταση σε μια ουρά είναι σταθερή, τότε η πιθανότητα των στοιχείων που φθάνουν είναι ίση με την πιθανότητα των στοιχείων που φεύγουν.

Έστω ότι **i** και **j** είναι δύο καταστάσεις μιας ουράς. Για να πάμε από την κατάσταση **i** στην κατάσταση **j** η πιθανότητα είναι $\mathbf{p_{ij}}$ ενώ για το αντίστροφο η πιθανότητα είναι $\mathbf{p_{ji}}$. Η πιθανότητα των αφίξεων δίνεται από τον εξής τύπο: $\mathbf{p_{i,i+1} = \lambda \cdot \Delta t}$ ενώ η πιθανότητα των αναχωρήσεων δίνεται από τον τύπο: $\mathbf{p_{i,i-1} = \mu \cdot \Delta t}$, όπου $\boldsymbol{\lambda}$ ο αριθμός αφίξεων στη μονάδα του χρόνου και $\boldsymbol{\mu}$ ο αριθμός εργασιών που εξυπηρετούνται στη μονάδα του χρόνου. Αντικαθιστώντας τους παραπάνω τύπους στην σχέση: $\mathbf{p_i \cdot p_{i,i+1} + p_i \cdot p_{i,i-1} = p_{i+1} \cdot p_{i+1,i} + p_{i-1} \cdot p_{i+1}}$

προκύπτει η εξίσωση: $\mathbf{p_i(\lambda + \mu) = p_{i+1} \cdot \mu + p_{i-1} \cdot \lambda}$

Αν απομονώσουμε τις δυο καταστάσεις **i** και **j** μπορούμε να πούμε ότι η πιθανότητα **pi** ισούται με:

$$\mathbf{\mu \cdot p_i = \lambda \cdot p_{i-1}} \text{ <=> } \mathbf{p_i = \left(\frac{\lambda}{\mu}\right) \cdot p_{i-1}}$$

Όμως μια ουρά έχει ένα σύνολο διακριτών καταστάσεων: (0, 1, 2, 3, .. ,n). Οπότε για κάθε κατάσταση **n** της ουράς ισχύει: $\mathbf{p(n) = \left(\frac{\lambda}{\mu}\right)^n p(0)}$, όπου $\mathbf{\frac{\lambda}{\mu} = \rho}$ (κυκλοφοριακή πυκνότητα)

- για πολύ μεγάλο ή άπειρο αριθμό καταστάσεων: $\mathbf{p(n) = \rho^n p(0) = \rho^n(1-\rho)}$
- για πεπερασμένο αριθμό καταστάσεων: $\mathbf{p(n) = \frac{\rho^n(1-\rho)}{1-\rho^{N+1}}}$, **N**=μέγιστος αριθμός στοιχείων που μπορεί να κρατήσει η ουρά

Πρέπει να αναφέρουμε ότι για να ισχύουν όλα τα παραπάνω, ο ρυθμός αφίξεων στην ουρά πρέπει να είναι μικρότερος από την χωρητικότητα της ουράς ώστε να είναι η ουρά σε σταθερή κατάσταση. Αλλιώς η ουρά θα μεγαλώνει συνεχώς και ποτέ δεν θα φτάσει σε σταθερή κατάσταση.

Η μεταφορική ικανότητα ή ρυθμοαπόδοση (throughput) μιας ουράς M/M/1 είναι: $\mathbf{\gamma = \lambda(1-p_0)}$.

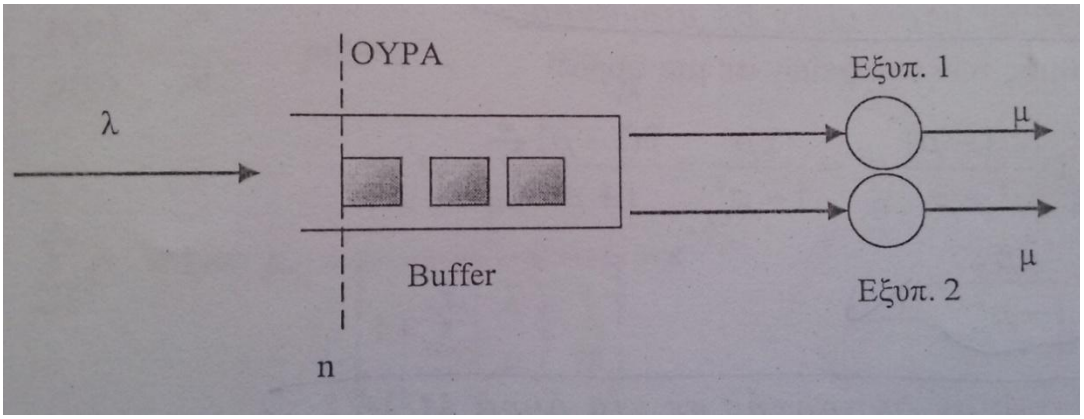
Ο μέσος αριθμός στοιχείων που έχει μια ουρά M/M/1 δίνεται από τον τύπο:

$\mathbf{E(n) = \frac{\rho}{1-\rho}}$.

Ο μέσος συνολικός χρόνος καθυστέρησης μιας M/M/1 ουράς ισούται με:

$\mathbf{E(T) = \frac{E(n)}{\lambda}\frac{1}{\mu(1-\rho)}}$ και αποτελείται από τους χρόνους καθυστέρησης στην ουρά, εξυπηρέτησης και εκπομπής του στοιχείου από την ουρά.

2.1.4 Ουρά M/M/2[12]



Σχήμα 8: Ουρά M/M/2

Στην ουρά M/M/2 υπάρχουν δύο εξυπηρετητές, οι αφίξεις ακολουθούν την κατανομή Poisson και οι εξυπηρετήσεις χρησιμοποιούν την εκθετική κατανομή. Η εξυπηρέτηση είναι ανεξάρτητη από τους εσωτερικούς χρόνους και οι καταστάσεις είναι ανεξάρτητες μεταξύ τους.

Για τις καταστάσεις όπου $\mathbf{i}\geq 2$ έχουμε $\boldsymbol{\mu_i = 2\mu}$ και η πιθανότητα μιας τέτοιας κατάστασης ισούται με:

$\mathbf{2\mu \cdot p_i = \lambda \cdot p_{i-1}}$ <=> $\mathbf{p_i = \frac{\lambda}{2\cdot\mu} \cdot p_{i-1}}$ => $\mathbf{p(i) = \frac{2(1-\rho)}{1+\rho}\rho^i}$, όπου $\boldsymbol{\rho = \frac{\lambda}{2\mu}}$

Ο μέσος αριθμός στοιχείων σε μια ουρά M/M/2 ισούται με: $\mathbf{E(n) = \frac{2\rho}{1-\rho^2}}$.

Η μέση τιμή χρόνου αναμονής σε μια ουρά M/M/2 είναι: $\mathbf{E(T) = \frac{E(n)}{\lambda} = \frac{1}{\mu(1-\rho^2)}}$.

Η μεταφορική ικανότητα μιας ουράς M/M/2 είναι πάντα μικρότερη από 2μ: $\boldsymbol{\gamma=2\mu\rho}$.

2.2 Βασικές κατανομές

2.2.1 Κατανομή Bernoulli (ή δοκιμή Bernoulli)[12]

Η κατανομή Bernoulli χρησιμοποιείται για την περιγραφή ενός τυχαίος πειράματος που έχει ως κατάληξη δυο αντίθετα αποτελέσματα: την επιτυχία (Ε) και την

αποτυχία (Α). Παρατηρούμε ότι τα πιθανά αποτελέσματα είναι αμοιβαίως αποκλειόμενα!

Έστω, μια τυχαία μεταβλητή Χ που μπορεί να πάρει δυο τιμές: τον αριθμό 1 όταν το αποτέλεσμα είναι επιτυχία (Ε) και τον αριθμό 0 όταν το αποτέλεσμα είναι αποτυχία (Α). Αν η πιθανότητα η τυχαία μεταβλητή Χ να πάρει την τιμή 1(Ε) ισούται με p, ενώ η πιθανότητα να πάρει την τομή 0(Α) ισούται με q=1-p, τότε η συνάρτηση πυκνότητας πιθανότητας δίνεται από τον ακόλουθο τύπο:

$$Pr(X=x)=p^{x}\cdot(1-p)^{1-x} \quad \text{, όπου x=0 ή 1}$$

2.2.2 Διωνυμική κατανομή[12]

Εδώ, η τυχαία μεταβλητή Χ παριστάνει τον αριθμό των επιτυχιών σε n ανεξάρτητες δοκιμές Bernoulli, όταν η πιθανότητα επιτυχίας σε κάθε δοκιμή είναι p. Στην περίπτωση αυτή η συνάρτηση πυκνότητας πιθανότητας είναι η εξής:

$$Pr(X=x)=\binom{n}{x}\cdot p^{x}\cdot(1-p)^{n-x} \quad \text{, όπου x=0,1,2,..,n}$$

2.2.3 Κατανομή Poisson[12][13]

Η κατανομή Poisson χρησιμοποιείται όταν θέλουμε να μετρήσουμε τον αριθμό εμφάνισης μιας κατάστασης στη μονάδα του χρόνου. Έτσι, η τυχαία μεταβλητή Χ παριστάνει τον αριθμό εμφάνισης τυχαίων περιστατικών σε καθορισμένο χρονικό διάστημα και η συνάρτηση πυκνότητας πιθανότητα δίνεται από τον παρακάτω τύπο:

$$P(X=x)=\frac{a^{x}\cdot e^{-a}}{x!}\text{, x=0,1,2,….}$$

όπου α =λΤ

λ → ρυθμός αφίξεων

Τ → χρονικό διάστημα μέσα στο οποίο οι αφίξεις λαμβάνουν χώρα.

2.2.4 Εκθετική κατανομή[14]

Μια τυχαία μεταβλητή Χ που ακολουθεί την εκθετική κατανομή, έχει την εξής συνάρτηση πυκνότητας πιθανότητας:

$$f(x)=\lambda e^{-\lambda x}\text{,} \quad x\geq 0$$

Η εκθετική κατανομή είναι άρρηκτα συνδεδεμένη με την κατανομή Poisson. Οι τυχαίες μεταβλητές $T_1,T_2,\ldots,T_n,\ldots$ που εκφράζουν το χρόνο ανάμεσα σε δυο διαδοχικές αφίξεις σε μια Poisson διαδικασία αφίξεων είναι ανεξάρτητες και ισόνομες τυχαίες μεταβλητές που ακολουθούν την εκθετική κατανομή, δηλαδή

$$Pr(T_n \leq x) = 1\text{-}e^{-\lambda x}, \quad x \geq 0, n=1, 2, \ldots$$

Είναι σημαντικό να αναφέρουμε ότι η εκθετική κατανομή έχει την ιδιότητα της αμνησίας, δηλαδή η γνώση των παρελθουσών τιμών δεν παίζει ρόλο στην πρόβλεψη των μελλοντικών τιμών της.

2.3 Χρονοδρομολόγηση

2.3.1 FIFO (First-In-First-Out)[15][16]

Πρόκειται για ένα αλγόριθμο δρομολόγησης που υλοποιείται πολύ εύκολα και απλά. Σύμφωνα με τον αλγόριθμο διατηρείται μια ουρά σε κάθε θύρα εξόδου του δρομολογητή. Τα πακέτα εξυπηρετούνται με βάση τη σειρά άφιξής τους δηλαδή, όταν δρομολογείται ένα πακέτο σε μια θύρα εξόδου, τότε τοποθετείται στο τέλος της ουράς. Όσο η ουρά δεν είναι κενή, η μετάδοση των δεδομένων γίνεται με τέτοιο τρόπο ώστε τα πακέτα που παραμένουν μεγαλύτερο χρονικό διάστημα στην ουρά, να είναι αυτά που επιλέγονται πρώτα.

Παρά την απλότητά του αξίζει να σημειωθεί ότι ο αλγόριθμος αυτός παρουσιάζει μερικά μειονεκτήματα τα οποία αναφέρονται παρακάτω:

- Δεν είναι αποδοτικός σε περιπτώσεις μεγάλης συμφόρησης καθώς ο χρόνος που εξυπηρετούνται τα πακέτα αυξάνει γραμμικά με το πλήθος των πακέτων που βρίσκονται μέσα στην ουρά.
- Τα πακέτα που ανήκουν σε υψηλής προτεραιότητας ουρές είναι πιο ευαίσθητα σε καθυστερήσεις και δεν λαμβάνουν ιδιαίτερη μεταχείριση.
- Σε περίπτωση συμφόρησης του δικτύου, αν μια TCP σύνδεση δεν καταφέρει να υποχωρήσει , τότε οι υπόλοιπες συνδέσεις που βρίσκονται στο ίδιο τμήμα της διαδρομής θα αναγκαστούν να υποχωρήσουν περισσότερο από ότι συνήθως. Επομένως αδυνατεί να διαχειριστεί πολλαπλές ουρές.
- Οι ουρές μεγαλύτερων πακέτων εξυπηρετούνται καλύτερα. Επομένως σε περίπτωση που πακέτα μικρότερου μεγέθους βρίσκονται στην ουρά πίσω από πακέτα μεγαλύτερου μεγέθους, η ουρά FIFO δημιουργεί μεγαλύτερη μέση καθυστέρηση ανά πακέτο από ότι στην περίπτωση που τα μικρότερα πακέτα μεταδίδονταν πρώτα.

Προκειμένου ο αλγόριθμος FIFO να ενεργοποιηθεί και να λειτουργήσει σωστά θα πρέπει να απενεργοποιήσουμε οποιοδήποτε εργαλείο ουράς ή άλλον αλγόριθμο

χρησιμοποιείται τη συγκεκριμένη χρονική στιγμή. Αυτό γίνεται μέσω των παρακάτω εντολών:

router(config-if) #no fair-queue → απενεργοποιείται ο αλγόριθμος WFQ

router(config-if) #hold-queue #_of_packets out → πραγματοποιείται ρύθμιση του μήκους της ουράς από επιτακτικές προεπιλεγμένες ρυθμίσεις. Ο συμβολισμός #_of_packets αντιπροσωπεύει τον αριθμό των πακέτων.

2.3.2 PQ (Priority-Queuing)[17][18]

Ο αλγόριθμος PQ, γνωστός και ως SPQ (StrictPQ), βασίζεται στην προτεραιότητα που έχει κάθε πακέτο που διαχειρίζεται. Τα πακέτα ταξινομούνται σε επίπεδα προτεραιότητας και ο δρομολογητής διατηρεί μια ξεχωριστή ουρά FIFO για κάθε προτεραιότητα. Το πακέτο που έχει την υψηλότερη προτεραιότητα εξυπηρετείται πρώτο σε σχέση με ένα πακέτο χαμηλότερης προτεραιότητας. Τα πακέτα χαμηλής προτεραιότητας εξυπηρετούνται μόνο όταν οι ουρές υψηλής προτεραιότητας είναι άδειες. Στην περίπτωση που δυο πακέτα έχουν την ίδια προτεραιότητα, τότε εξυπηρετούνται βάση της σειράς τους μέσα στην ουρά. Κατά τη λειτουργία του αλγορίθμου αυτού προστίθενται πακέτα με μια σχετική προτεραιότητα. Στην συνέχεια, εξυπηρετείται το πακέτο που έχει την υψηλότερη προτεραιότητα, από την ουρά υψηλής προτεραιότητας. Ακολούθως γίνεται πάλι έλεγχος των πακέτων και ακολουθείται η ίδια διαδικασία.

Όταν στις ουρές υψηλής προτεραιότητας η κίνηση είναι μεγάλη, εξυπηρετούνται συνέχεια πρώτες αυτές, με συνέπεια να μην εξυπηρετούνται οι ουρές χαμηλής προτεραιότητας.

2.3.3 RR (Round-Robin) – WRR (Weighted-Round-Robin)[1][19]

Ο αλγόριθμος ‘Round-Robin’ είναι απλός στην υλοποίησή του. Ξεκινά εξυπηρετώντας ένα πακέτο από κάθε μη κενή ουρά με τη σειρά. Αφού εξυπηρετήσει ένα πακέτο από κάθε σύνδεση, δηλαδή γίνει ένα πέρασμα (RoundTime), τότε επιστρέφει στην σύνδεση που ήταν αρχικά και επαναλαμβάνει τις παραπάνω ενέργειες. Όταν χρησιμοποιούνται πολλαπλές ουρές σε έναν δρομολογητή, δεν έχουμε προτεραιότητες για τη διαχείρισή τους και για αυτό το λόγο ο αλγόριθμος προσφέρει best-effort υπηρεσίες.

Στην περίπτωση που τα πακέτα έχουν μεταβλητό μήκος και προέρχονται από συνδέσεις με διαφορετικά βάρη εφαρμόζεται ο αλγόριθμος WRR. Σε κάθε πέρασμα εξυπηρετούνται ανά σύνδεση περισσότερα από ένα πακέτα. Βάση των κανονικοποιημένων βαρών γίνεται υπολογισμός των πακέτων ανά πέρασμα. Τα βάρη κανονικοποιούνται αρχικά ως προς το μέσο μέγεθος πακέτου και στη συνέχεια τα βάρη που προκύπτουν κανονικοποιούνται πάλι ώστε να είναι στο

σύνολό τους ακέραιοι αριθμοί. Αξίζει να αναφερθεί ότι για να λειτουργήσει ο αλγόριθμος αυτός χρειάζεται να γνωρίζει κανείς το μέσο μέγεθος των πακέτων για κάθε σύνδεση, το οποίο όμως είναι δύσκολο καθώς δεν μπορεί να υπολογιστεί με ακρίβεια. Ένα δεύτερο μειονέκτημα του αλγορίθμου αυτού είναι ότι δεν λειτουργεί στην περίπτωση που το χρονικό διάστημα εξυπηρέτησης μιας σύνδεσης είναι συγκρίσιμο με το χρόνο που απαιτείται για ένα πέρασμα.

Παρακάτω βλέπουμε κάποιες εντολές με τις οποίες μπορούμε να διαμορφώσουμε έναν αλγόριθμου WRR και να τον κάνουμε να λειτουργήσει. Αρχικά διαμορφώνουμε τις ουρές εξόδου και τι ανήκει σε κάθε ουρά. Στο παράδειγμά μας χρησιμοποιούμε 4 ουρές. Η ουρά 1 περιλαμβάνει πακέτα μαρκαρισμένα με CoS 0 και 1, η ουρά 2 περιλαμβάνει πακέτα μαρκαρισμένα με CoS 2 και 3, η ουρά 3 περιλαμβάνει πακέτα με CoS 4 και 5 και η ουρά 4 περιλαμβάνει πακέτα με CoS 6 και 7.

```
SW3(config-if-range) #wrr-queue cos-map 1 0 1

SW3(config-if-range) #wrr-queue cos-map 2 2 3

SW3(config-if-range) #wrr-queue cos-map 3 4 5

SW3(config-if-range) #wrr-queue cos-map 4 6 7

SW3(config-if-range) #
```

Αφού ορίσαμε τις ουρές, τώρα πρέπει να καθορίσουμε πόσα πακέτα από κάθε ουρά μπορεί να μεταδώσει ο διακόπτης (switch) πριν πάει στην επόμενη ουρά. Ρυθμίζουμε τις ουρές μία- μία, από την ουρά 1 στην ουρά 4 (αριστερά προς δεξιά), έτσι ώστε σε αυτό το παράδειγμα ο διακόπτης να μπορεί να μεταδίδει 10 πακέτα από την ουρά 1, 20 πακέτα από την ουρά 2, 30 πακέτα από την ουρά 3 και 50 πακέτα από την ουρά 4.

```
 SW3(config-if-range) #wrr-queue bandwidth ?
<1-65536> enter bandwidth weight for qid 1

SW3(config-if-range) #wrr-queue bandwidth 10 ?
<1-65536> enter bandwidth weight for qid 2

SW3(config-if-range) #wrr-queue bandwidth 10 20 ?
<1-65536> enter bandwidth weight for qid 3

SW3(config-if-range) #wrr-queue bandwidth 10 20 30 ?
<1-65536> enter bandwidth weight for qid 4

SW3(config-if-range) #wrr-queue bandwidth 10 20 30 50 ?
<cr>

SW3(config-if-range) #wrr-queue bandwidth 10 20 30 50
```

Στη συνέχεια ορίζουμε την ουρά 4 ως την ουρά που έχει προτεραιότητα. Αυτό σημαίνει ότι ο διακόπτης πρώτα ελέγχει την ουρά προτεραιότητας και μεταδίδει όλα τα πακέτα της και μετά ελέγχει τις υπόλοιπες ουρές. Έτσι ο διακόπτης θα ελέγξει την ουρά 3, θα μεταδώσει τα πακέτα τότε ελέγχει την ουρά 4 δεδομένου ότι έχει προτεραιότητα μεταδίδει εκείνα τα πακέτα, μετά ελέγχει την ουρά 2 μεταδίδει τα πακέτα και στη συνέχεια γυρίζει πίσω στην ουρά 4.

```
SW3(config-if-range) #priority-queue out

SW3(config-if-range) #
```

Τέλος, είναι σημαντικό να προσέξουμε την κίνηση και τον αριθμό των πακέτων που θα μεταδίδει η ουρά προτεραιότητας ώστε να αποφεύγεται η άσκοπη καθυστέρηση σε άλλες κυκλοφορίες και οι ουρές με πολύ χαμηλή προτεραιότητα να μπορούν να μεταδίδουν κι αυτές πακέτα.

2.3.4 WFQ (Weighted-Fair-Queuing)[1][20]

Ο αλγόριθμος αυτός αποδίδει καλύτερα ως προς την δίκαιη κατανομή πόρων από τον αλγόριθμο WRR στην περίπτωση που τα πακέτα είναι μεταβλητού μήκους και βάρους. Συγκεκριμένα βρίσκει το χρόνο ολοκλήρωσης κάθε πακέτου και εξυπηρετεί τα πακέτα με βάση τους χρόνους αυτούς. Ο αλγόριθμος παρέχει εγγυήσεις στην απόδοση μιας σύνδεσης σχετικά με το εύρος ζώνης και την καθυστέρηση μετάδοσης. Όμως, απαιτείται να γίνει υλοποίηση της ουράς προτεραιότητας και διατήρηση κατάστασης ανά σύνδεση. Επίσης, σε περίπτωση μικρής καθυστέρησης στη μετάδοση απαιτείται μεγάλο εύρος ζώνης.

Ας δούμε τώρα πως λειτουργεί ο αλγόριθμος WFQ:

- Κατά την άφιξη ενός πακέτου μήκους *pbit* χρησιμοποιούνται οι διευθύνσεις προέλευσης και προορισμού για την ταξινόμησή του. Επίσης, ελέγχεται ο αριθμός ολοκλήρωσης του τελευταίου πακέτου που εξυπηρετήθηκε ή περιμένει να εξυπηρετηθεί. Σε κάθε γύρο εξυπηρετείται 1bit από κάθε *ενεργή σύνδεση* (ενεργή σύνδεση έχουμε αν το τελευταίο πακέτο που έχει εξυπηρετηθεί από αυτή, ή στην ουρά της, έχει αριθμό ολοκλήρωσης μεγαλύτερο από τον τρέχοντα αριθμό γύρων).
- Γίνεται επαναϋπολογισμός του *αριθμού γύρων (Roundnumber - R)*. Με τον όρο αυτό αναφερόμαστε σε μια μεταβλητή πραγματικής τιμής που αυξάνεται με ρυθμό αντιστρόφως ανάλογο του αριθμού των υπαρχόντων ενεργών συνδέσεων.
- Υπολογίζεται ο *αριθμός ολοκλήρωσης (finishnumber)* του πακέτου ο οποίος ισούται με $R+p$. Αν το πακέτο φτάσει σε μια μη κενή ουρά και το προηγούμενο πακέτο έχει αριθμό ολοκλήρωσης f , τότε ο αριθμός ολοκλήρωσης του πακέτου είναι $f+p$.
- Στη συνέχεια εισάγεται το πακέτο στη σειρά προτεραιότητας με βάση τον αριθμό ολοκλήρωσης. Στην περίπτωση που η ουρά είναι γεμάτη απορρίπτεται το πακέτο με τον μεγαλύτερο αριθμό ολοκλήρωσης.

- Αφού εξυπηρετηθεί ένα πακέτο, επιλέγεται το πακέτο με τον μικρότερο αριθμό ολοκλήρωσης.

Αξίζει να τονισθεί ότι ο αλγόριθμος δίνει προτεραιότητα στις διαδραστικές συνομιλίες με χαμηλό εύρος ζώνης και μοιράζει δίκαια το υπόλοιπο εύρος ζώνης μεταξύ των μη διαδραστικών συνομιλιών με υψηλό εύρος ζώνης. Το "κλειδί" του είναι ότι η ροή μεταφοράς ενός αρχείου δεν έχει κανένα είδος προτεραιότητας έναντι των υπολοίπων.

Δεδομένου ότι ο αλγόριθμος WFQ είναι ενεργοποιημένος από προεπιλογή σε όλες τις σειριακές διασυνδέσεις, δεν χρειάζεται να τον ενεργοποιήσουμε εμείς. Αυτό που μπορούμε να κάνουμε είναι να διαμορφώσουμε κάποιες επιλογές του. Παρακάτω θα δούμε τις εντολές με τις οποίες θα το κάνουμε. Ξεκινάμε με την μεταβλητή CongestiveDiscardThreshold που παρουσιάζει τον αριθμό των πακέτων που μπορεί να βρίσκονται σε μια ενιαία ουρά. Ο προκαθορισμένος είναι 64 πακέτα αλλά μπορούμε να τον αλλάξουμε.

```
R1(config)#int  serial0

R1(config-if)#fair-queue  ?

<1-4096>  Congestive Discard Threshold

<cr>
```

Η αλλαγή γίνεται με την παρακάτω εντολή. Θα το αλλάξουμε σε 200 για το παράδειγμά μας:

```
R1(config-if)#fair-queue  200
```

Στη συνέχεια θα αλλάξουμε το DynamicConversationQueues που είναι ο αριθμός των δυναμικών ουρών συνομιλίας και χρησιμοποιούνται για κανονικές και best-effort συνομιλίες, Θα το κάνουμε κι αυτό 200 με την παρακάτω εντολή (όμως μας βγάζει λάθος γιατί η τιμή πρέπει να είναι δύναμη του 2) :

```
R1(config-if)#fair-queue  200  200
Number of dynamic queue must be a power of 2 (16, 32, 64, 128, 256, 512, 1024)
```

Τέλος θα αλλάξουμε το ReservableConversationQueues, δηλαδή τις ουρές που χρησιμοποιούνται για εξειδικευμένες ουρές αναμονής και για χαρακτηριστικά Ποιότητας Υπηρεσιών, όπως το πρωτόκολλο RSVP. Η προεπιλεγμένη τιμή είναι το 0, όμως εμείς θα το κάνουμε 100 με την εντολή που ακολουθεί:

```
R1(config-if)#fair-queue  200  256  100
```

2.3.5 CBWFQ (Class-Based-Weighted-Fair-Queuing)[21]

Ο αλγόριθμος CBWFQ είναι μια επέκταση του αλγορίθμου WeightedFairQueuing για να παρέχει υποστήριξη στις κατηγορίες κυκλοφορίας που ορίζονται από τον

χρήστη. Οι κατηγορίες κυκλοφορίας ορίζονται με βάση κάποια κριτήρια τα οποία περιλαμβάνουν πρωτόκολλα, λίστες ελέγχου πρόσβασης και διεπαφές εισόδου. Τα πακέτα που πληρούν τα κριτήρια μιας κατηγορίας, ανήκουν στην κυκλοφορία της κατηγορίας αυτής. Υπάρχει και μια ουρά για κάθε κατηγορία όπου κατευθύνεται η κυκλοφορία της κατηγορίας. Προκειμένου να γίνει χαρακτηρισμός μια κατηγορίας, θα πρέπει να εκχωρήσουμε το εγγυημένο εύρος ζώνης που διανέμεται στην κατηγορία κατά τη διάρκεια της συμφόρησης, το βάρους και το μέγιστο όριο πακέτων. Επίσης, μπορούμε να καθορίσουμε το όριο ουράς μιας κατηγορίας, που είναι ο μέγιστος αριθμός των πακέτων που μπορούν να συσσωρευτούν στην ουρά. Όταν μια ουρά φτάσει αυτό το όριο, τότε τα πρόσθετα πακέτα που εισέρχονται αναγκάζουν την πτώση της ουράς (droptail) ή την πτώση πακέτων, ανάλογα με την πολιτική της κατηγορίας

Αν μια κατηγορία προεπιλογής έχει ρυθμιστεί με την εντολή διαμόρφωσης κατηγορίας εύρους ζώνης *policy-map*, η αταξινόμητη κυκλοφορία τοποθετείται σε μια ενιαία ουρά και επεξεργάζεται με βάση το διαμορφωμένο εύρος ζώνης. Αν μια κατηγορία προεπιλογής έχει ρυθμιστεί με την εντολή *fair-queue* , η αταξινόμητη κυκλοφορία ταξινομείται με best-effort επεξεργασία. Στην περίπτωση που καμιά κατηγορία προεπιλογής δεν διαμορφώνεται, τότε η κυκλοφορία που δεν ταιριάζει με καμιά από τις διαμορφωμένες κατηγορίες ταξινομείται με best-effort επεξεργασία. Μόλις ένα πακέτο ταξινομηθεί, ισχύουν όλοι οι τυποποιημένοι μηχανισμοί που μπορούν να χρησιμοποιηθούν στις διαφοροποιημένες υπηρεσίες μεταξύ των κατηγοριών.

Η ταξινόμηση της ροής είναι μια τυποποιημένη επεξεργασία WFQ. Τα πακέτα με την ίδια IP διεύθυνση πηγής και προορισμού, το ίδιο TCP ή UDP πρωτόκολλο ανήκουν στην ίδια ροή. Ο αλγόριθμος WFQ διαθέτει ίσο μερίδιο εύρους ζώνης σε κάθε ροή.

Για τον αλγόριθμο CBWFQ, το βάρος μιας κατηγορίας γίνεται το βάρος κάθε πακέτου που πληροί τα κριτήρια αυτής της κατηγορίας. Το βάρος ενός πακέτου που ανήκει σε μια κατηγορία προέρχεται από το εύρος ζώνης που δώσαμε στην κατηγορία αυτή όταν την διαμορφώσαμε κι αυτό σημαίνει ότι το βάρος της κατηγορίας είναι διαμορφώσιμο από τον χρήστη. Αφού οριστεί το βάρος ενός πακέτου, το πακέτο μπαίνει στην κατάλληλη σειρά αναμονής κατηγορίας.

Για να διαμορφώσουμε μια πολιτική κατηγορίας του αλγορίθμου CBWFQ ακολουθούνται οι τρεις παρακάτω διαδικασίες:

- Καθορίζονται οι κατηγορίες κυκλοφορίας προκειμένου να προσδιοριστεί η πολιτική ταξινόμησης (χάρτες κατηγορίας – classmaps). Μέσω αυτού καθορίζονται πόσοι τύποι πακέτων πρέπει να διαφοροποιούνται μεταξύ τους.
- Συνδέονται οι πολιτικές με τις κατηγορίες κυκλοφορίας (πολιτικοί χάρτες – policymaps). Εδώ γίνεται διαμόρφωση των πολιτικών που εφαρμόζονται στα πακέτα μιας κατηγορίας , τα οποία έχουν καθοριστεί μέσω ενός χάρτη κατηγορίας.
- Συνδέονται οι πολιτικές με τις διεπαφές (πολιτικές υπηρεσιών – servicepolicies). Αυτό απαιτεί την σύνδεση ενός υπάρχοντα πολιτικού χάρτη ή μιας πολιτικής υπηρεσιών με μια διεπαφή ώστε να εφαρμοστεί ένα συγκεκριμένο σύνολο πολιτικών για το χάρτη σε εκείνη τη διεπαφή.

Παρακάτω θα δούμε τις εντολές που χρησιμοποιούνται για τη δημιουργία ενός χάρτη κατηγορίας (classmap). Πρώτα καθορίζουμε το όνομα που θα έχει ο χάρτης κατηγορίας και στη συνέχεια διευκρινίζουμε τα κριτήρια (το όνομα του ACL, το όνομα της διεπαφής εισόδου, το πρωτοκόλλου που χρησιμοποιείται, την τιμή του πεδίου EXP) τα οποία πρέπει να πληρούν τα πακέτα που θα ανήκουν στην κατηγορία.

Router(config)# **class-map** *class-map-name*

Router(config-cmap)# **match access-group** {*access-group* | **name** *access-group-name*}

Router(config-cmap)# **match input-interface** *interface-name*

Router(config-cmap)# **match protocol** *protocol*

Router(config-cmap)# **match mpls experimental** *number*

Στη συνέχεια θα δούμε εντολές για τη δημιουργία ή την τροποποίηση ενός πολιτικού χάρτη. Αρχικά, καθορίζουμε το όνομα του πολιτικού χάρτη και το όνομα της κατηγορίας που περιλαμβάνεται στην πολιτική υπηρεσιών. Επίσης, διευκρινίζουμε το ποσό εύρους ζώνης σε kbps και το μέγιστο αριθμό πακέτων της κατηγορίας.

Router(config)# **policy-map** *policy-map*

Router(config-pmap)# **class** *class-name*

Router(config-pmap-c)# **bandwidth** *bandwidth-kbps*

Router(config-pmap-c)# **queue-limit** *number-of-packets*

Η παρακάτω εντολή χρησιμοποιείται για την σύνδεση ενός πολιτικού χάρτη υπηρεσιών με την διεπαφή εξόδου:

Router(config-if)# **service-policy output** *policy-map*

2.3.6 LLQ (Low-Latency-Queuing)[20][22]

Ο αλγόριθμος LLQ είναι μια επέκταση του CBWFQ που προσπαθεί να πετύχει μια εγγύηση για τα δεδομένα που επηρεάζονται αρνητικά από τις καθυστερήσεις δίνοντας τους την πιο πιθανή υψηλή προτεραιότητα μέσα από μια ουρά αναμονής προτεραιότητας. Ένα από τα πλεονεκτήματα του αλγορίθμου είναι ότι επιτρέπει δεδομένα ευαίσθητα στην καθυστέρηση, όπως είναι η φωνή, να στέλνονται πρώτα πριν από τα πακέτα άλλων ουρών. Επίσης, ο αλγόριθμος έχει τη δυνατότητα να συντονίζει και να περιορίζει το βάθος του δακτυλίου μετάδοσης μιας συσκευής, περιορίζοντας τον αριθμό των πακέτων που βρίσκονται σε ένα δακτύλιο μετάδοσης με αποτέλεσμα να μειώνεται η καθυστέρηση. Τέλος, ένα ακόμη πλεονέκτημα του LLQ είναι ο έλεγχος αποδοχής (AdmissionControl), ο οποίος ορίζει ένα μέγιστο

ποσό για το εύρος ζώνης που μπορεί να παρέχεται στα πακέτα δεδομένων μιας κατηγορίας , αποφεύγοντας το φαινόμενο «πείνας» για κυκλοφορίες χωρίς προτεραιότητα.

Παρακάτω θα δούμε κάποιες εντολές για να δώσουμε προτεραιότητα σε μια κατηγορία μέσα σε έναν πολιτικό χάρτη. Αρχικά, διευκρινίζουμε το όνομα του πολιτικού χάρτη που θέλουμε να διαμορφώσουμε και το όνομα της προκαθορισμένης κατηγορίας που περιλαμβάνεται στην πολιτική υπηρεσιών. Επίσης, διατηρούμε μια σειρά προτεραιότητας για την CBWFQ κυκλοφορία.

Router(config)# policy-map *policy-name*

Router(config-pmap)# class-map *class-name*

Router(config-pmap-c)# priority *kbps bytes*

2.4 Διαχείριση ουράς

2.4.1 DropTail[10]

Η DropTail (πτώση ουράς) είναι μια τεχνική που χρησιμοποιείται σε περιόδους συμφόρησης μέσω ελέγχου του μέσου μήκους των ουρών και λειτουργεί ανεξάρτητα σε κάθε ουρά. Έτσι, όταν ο buffer γεμίσει, τα πακέτα που φτάνουν στη ουρά του απορρίπτονται ή αλλιώς πέφτουν, μέχρι να υπάρξει και πάλι διαθέσιμος χώρος στην ουρά. Η απλή της υλοποίηση και η μείωση των απωλειών με buffer μεγάλου μεγέθους είναι τα κυριότερα πλεονεκτήματά της. Όμως, από την άλλη έχει και σημαντικά μειονεκτήματα. Υπάρχει μεγάλη καθυστέρηση αναμονής στην ουρά εξαιτίας των μεγάλων μηκών ουράς. Η αργή αντίληψη της συμφόρησης από τους TCP χρήστες για να μειώσουν τον ρυθμό μετάδοσης τους καθώς ο buffer πρέπει να φτάσει στο 100% της χρησιμοποίησής του για να το αντιληφθούν. Τέλος, έχουμε το φαινόμενο του καθολικού TCP συγχρονισμού, όπου όλες οι TCP πηγές ταυτόχρονα μειώνουν την κίνησή τους λόγω πτώσης πακέτων με συνέπεια να έχουμε αρνητικές επιδράσεις στη χρησιμοποίηση των πόρων του δικτύου.

2.4.2 RED (Random-Early-Detection)[23][24]

Ο σκοπός του μηχανισμού RED είναι να αποτρέψει τον TCP συγχρονισμό από τυχαία απόρριψη πακέτων όταν μια ουρά διεπαφής εξόδου έχει αρχίσει να γεμίζει. Το ποσό των πακέτων που απορρίπτει εξαρτάται κάθε φορά από το τρέχον βάθος που έχει η ουρά. Η λειτουργία του RED είναι να ελέγχει το μέσο μέγεθος ουράς και να υποδεικνύει στο τελικό host πότε πρέπει να επιβραδύνει προσωρινά την

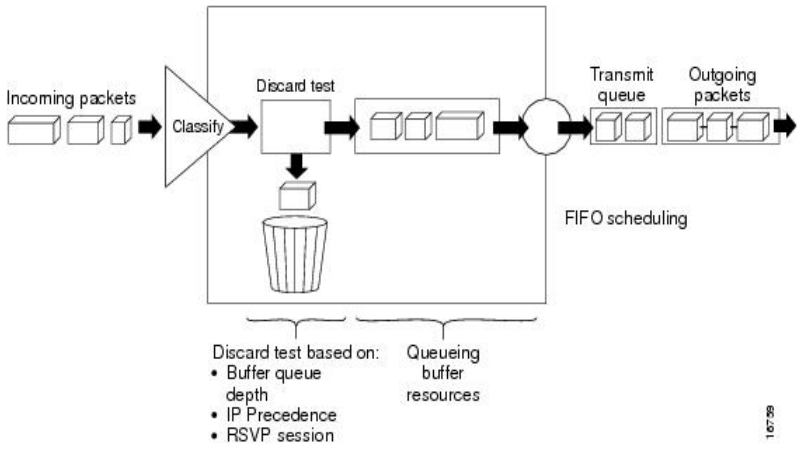
μετάδοση πακέτων. Όταν υπάρχει πτώση πακέτων πριν από περιόδους υψηλής συμφόρησης, ο RED λέει στην πηγή μετάδοσης των πακέτων να μειώσει την ταχύτητα μετάδοσης. Εκμεταλλεύεται τον μηχανισμό συμφόρησης του TCP. Υποθέτοντας ότι η πηγή χρησιμοποιεί πρωτόκολλο TCP, μειώνει την ταχύτητα μετάδοσης μέχρι όλα τα πακέτα να φτάσουν στον προορισμό τους, υποδεικνύοντας ότι δεν υπάρχει κυκλοφοριακή συμφόρηση. Χρησιμοποιούμε τον RED ως τρόπο για να αναγκαστεί το TCP να επιβραδύνει τη μετάδοση πακέτων. Το TCP δεν σταματά μόνο, αλλά ξεκινά ξανά γρήγορα και προσαρμόζει το ποσοστό μετάδοσής του στο ποσοστό που το δίκτυο μπορεί να υποστηρίξει. Ο RED μοιράζει τις απώλειες στο χρόνο και διατηρεί συνήθως χαμηλό βάθος ουράς. Ο μηχανισμός αυτός αρχίζει να ρίχνει πακέτα όταν η συμφόρηση εμφανίζεται με ρυθμό που εμείς θα επιλέξουμε κατά τη διάρκεια της διαμόρφωσης.

Η πιθανότητα πτώσης πακέτων βασίζεται στο ελάχιστο κατώτατο όριο (minimumthreshold), στο μέγιστο κατώτατο όριο (maximumthreshold) και στον παρονομαστή πιθανότητας σημαδιών(MarkProbabilityDenominator –MPD). Όταν το μέσο βάθος ουράς είναι πάνω από το ελάχιστο κατώτατο όριο, ο RED αρχίζει να ρίχνει πακέτα. Το ποσοστό πτώσης πακέτων αυξάνεται γραμμικά καθώς το μέσο μέγεθος ουράς αυξάνεται μέχρι να φτάσει το μέγιστο κατώτατο όριο. Ο MPD είναι το μέρος των πακέτων που πέφτουν όταν το μέσο βάθος ουράς είναι στο μέγιστο κατώτατο όριο. Για παράδειγμα, αν ο MPD είναι 50 τότε 1 για κάθε 50 πακέτα «πέφτει» όταν το μέσο βάθος ουράς είναι στο μέγιστο κατώτατο όριο. Όταν το μέσο μέγεθος ουράς είναι πάνω από το μέγιστο κατώτατο όριο, όλα τα πακέτα «πέφτουν». Η τιμή ελάχιστου κατώτατου ορίου πρέπει να είναι αρκετά υψηλή για να μεγιστοποιηθεί η χρησιμοποίηση της σύνδεσης. Αν το ελάχιστο κατώτατο όριο είναι πολύ χαμηλό, τα πακέτα μπορούν να «πέσουν» χωρίς λόγο και η σύνδεση μετάδοσης δεν θα χρησιμοποιηθεί πλήρως. Η διαφορά μεταξύ των δυο κατώτατων ορίων πρέπει να είναι αρκετά μεγάλη για να αποφύγουμε την ξαφνική πτώση πολλών πακέτων με συνέπεια τον καθολικό συγχρονισμό των TCP hosts και μείωση των ποσοστών μετάδοσής τους.

2.4.3 WRED (Weighted-Random-Early-Detection)[21][23][24]

Ο μηχανισμός WRED συνδυάζει τις δυνατότητες του μηχανισμού RED με την IP προτεραιότητα παρέχοντας μια προνομιακή μεταχείριση της κυκλοφορίας των πακέτων υψηλότερης προτεραιότητας. Διαφέρει από τους άλλους μηχανισμούς αποφυγής συμφόρησης επειδή προσπαθεί να προβλέψει και να αποφύγει την συμφόρηση αντί να την ελέγξει τη στιγμή που συμβαίνει. Χρησιμοποιούμε τον WRED γιατί προσφέρει έγκαιρη ανίχνευση της πιθανής συμφόρησης, επιτρέπει την κυκλοφορία πολλών κατηγοριών και προστατεύει από τον καθολικό συγχρονισμό. Ο WRED χρησιμοποιείται συνήθως στους κεντρικούς δρομολογητές του δικτύου και όχι στις άκρες του. Οι δρομολογητές των άκρων ορίζουν τις IP προτεραιότητες στα πακέτα καθώς εισάγονται στο δίκτυο, τις οποίες χρησιμοποιεί ο WRED για να καθορίσει την μεταχείριση των διαφορετικών τύπων κυκλοφορίας. Ο WRED παρέχει ξεχωριστά κατώτατα όρια και βάρη για τις διαφορετικές IP προτεραιότητες, που μας επιτρέπουν να παρέχουμε διαφορετική ποιότητα υπηρεσιών όσον αφορά

την πτώση πακέτων για διαφορετικούς τύπους κυκλοφορίας. Σε περιόδους συμφόρησης, η «κανονική» κυκλοφορία ίσως πέφτει περισσότερο συχνά από την «προνομιούχο» κυκλοφορία. Επίσης, ο WRED γνωρίζει το RSVP και αυτό μπορεί να του παρέχει την QoS ολοκληρωμένη υπηρεσία ελεγχόμενου φορτίου.



Σχήμα 9: Μηχανισμός WRED

Ο WRED είναι χρήσιμος μόνο όταν ο όγκος της κυκλοφορίας είναι TCP/IP κυκλοφορία. Όταν υπάρχει ξαφνική πτώση πακέτων πριν τις περιόδους υψηλής συμφόρησης, ο WRED λέει στη πηγή πακέτων να μειώσει το ποσοστό μετάδοσής της. Αν η πηγή χρησιμοποιεί το πρωτόκολλο TCP, τότε αυτό θα μειώσει το ποσοστό μετάδοσής της μέχρι όλα τα πακέτα να φτάσουν στον προορισμός τους με αποτέλεσμα να τελειώσει η συμφόρηση. Αν η πηγή χρησιμοποιεί άλλο πρωτόκολλο, τότε μπορεί να μην αποκριθεί ή να μην ξαναστείλει τα πακέτα που έπεσαν με το ίδιο ποσοστό μετάδοσης. Στατιστικά, οι πηγές που παράγουν την περισσότερη κυκλοφορία είναι πιθανότερο να επιβραδυνθούν από τις πηγές που παράγουν μικρή κυκλοφορία Γενικά, ο WRED ρίχνει επιλεκτικά τα πακέτα βασιζόμενος στην IP προτεραιότητά (IPPrecedence) τους. Μεταχειρίζεται την μη IP κυκλοφορία ως προτεραιότητα 0, η οποία είναι η χαμηλότερη προτεραιότητα. Πακέτα με χαμηλότερη IP προτεραιότητα είναι περισσότερο πιθανό να πέσουν από τα πακέτα με υψηλότερη IP προτεραιότητα. Όσο υψηλότερη είναι η προτεραιότητα ενός πακέτου τόσο υψηλότερη είναι η πιθανότητα να παραδοθεί. Όταν η διεπαφή εξόδου δείχνει σημάδια συμφόρησης, ο WRED ρίχνει μερικά πακέτα χωρίς να έχει γεμίσει η ουρά. Με αυτόν τον τρόπο μειώνει τις πιθανότητες της πτώσης της ουράς, αποφεύγει την ξαφνική πτώση πακέτων, ελαχιστοποιεί τις πιθανότητες του καθολικού συγχρονισμού και επιτρέπει στη γραμμή μετάδοσης να χρησιμοποιείται πλήρως πάντα. Καθολικό συγχρονισμό έχουμε όταν πολλά TCP hosts μαζί μειώνουν τα ποσοστά μετάδοσής τους σε απάντηση για την πτώση πακέτων και τότε αυξάνουν τα ποσοστά μετάδοσής τους αμέσως ξανά όταν η συμφόρηση μειωθεί.

Για να ενεργοποιηθεί ο μηχανισμός αυτός και για να καθοριστεί η σήμανση που χρησιμοποιεί (IPPrecedence), χρησιμοποιούμε την εντολή:

```
Router(config-pmap-c)#random-detect  prec-based
```

Αφού διαμορφώσουμε τον WRED, μπορούμε αν θέλουμε να αλλάξουμε τις προεπιλεγμένες τιμές για το ελάχιστο κατώτατο όριο, το μέγιστο κατώτατο όριο και τον MPD για όσες IP προτεραιότητες θέλουμε.

Router(config-pmap-c)#random-detect precedence *precedence-value minimum-threshold maximum-threshold mark-probability-denominator*

Στη συνέχεια θα διαμορφώσουμε μια πολιτική κατηγορίας χρησιμοποιώντας την WRED πτώση πακέτων (WREDpacketdrop) αντί για την πτώση ουράς (droptail). Αρχικά, καθορίζουμε το όνομα του πολιτικού χάρτη που δημιουργείται ή τροποποιείται και το όνομα της κατηγορίας που δημιουργείται και περιλαμβάνεται στην πολιτική υπηρεσιών. Έπειτα, διευκρινίζουμε το ποσό εύρους ζώνης σε kbps που διαθέτεται στην κατηγορία. Στη συνέχεια, διαμορφώνουμε την πολιτική κατηγορίας ώστε να χρησιμοποιεί τον WRED για την απόρριψη των πακέτων. Τέλος, προσδιορίζουμε τον εκθετικό παράγοντα βάρους που χρησιμοποιείται για να υπολογισθεί το μέσο μήκος της ουράς.

Router(config)# **policy-map** *policy-map*

Router(config-pmap)# **class** *class-name*

Router(config-pmap-c)# **bandwidth** *bandwidth-kbps*

Router(config-pmap-c)# **random-detect**

Router(config-pmap-c)# **random-detect exponential-weighting-constant** *exponent*

ΚΕΦΑΛΑΙΟ 3:Αρχιτεκτονικές Ποιότητας Υπηρεσίας

3.1 Ολοκληρωμένες υπηρεσίες

3.1.1 Εισαγωγή[3][15][25]

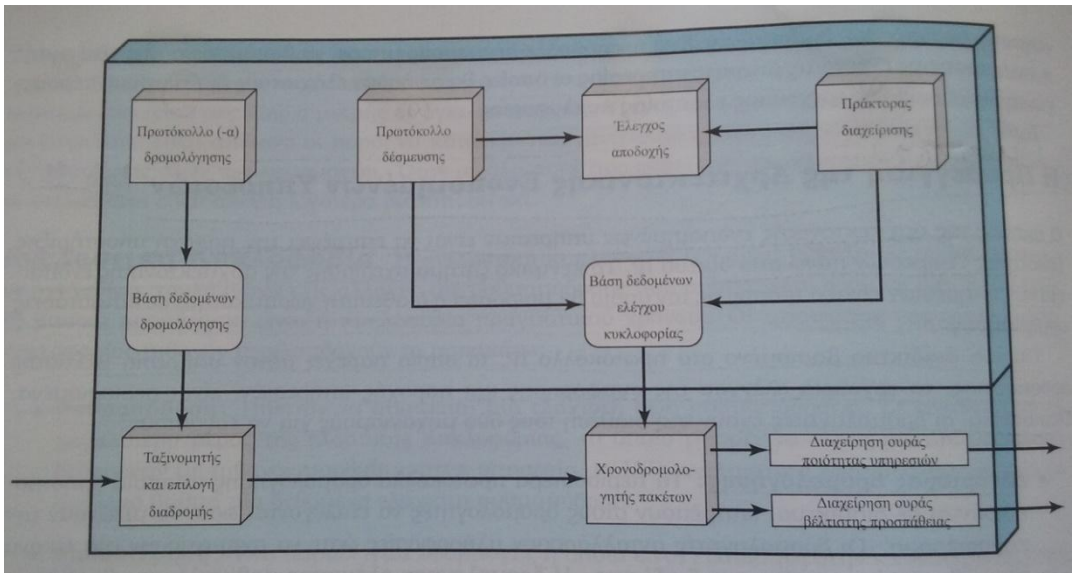
Η IETF (InternetEngineeringTaskForce) επινόησε το μοντέλο **Ολοκληρωμένων ή Ενοποιημένων Υπηρεσιών (IntegratedServices – IntServ)** για τη συνεχή ροή πολυμέσων τόσο σε εφαρμογές αποκλειστικής διανομής όσο και σε εφαρμογές πολυδιανομής. Ο σκοπός του μοντέλου αυτού είναι να παρέχει Ποιότητα Υπηρεσίας για μεταφορές πάνω από IP δίκτυα. Το κύριο ζήτημα σχεδίασης της αρχιτεκτονικής ολοκληρωμένων υπηρεσιών είναι ο τρόπος με τον οποίο θα μοιραστεί η διαθέσιμη χωρητικότητα σε περιόδους συμφόρησης του δικτύου. Ένα IP διαδίκτυο παρέχει μόνο την υπηρεσία βέλτιστης προσπάθειας. Τα εργαλεία ελέγχου της συμφόρησης και παροχής υπηρεσιών είναι περιορισμένα. Ο αλγόριθμος δρομολόγησης και η απόρριψη πακέτων είναι οι δυο μοναδικοί μηχανισμοί με τους οποίους μπορούν να εργαστούν οι δρομολογητές. Αν και οι μηχανισμοί αυτοί λειτουργούν αρκετά ικανοποιητικά, δεν είναι επαρκείς για την ποικιλία κυκλοφορίας των διαδικτύων.

Οι ολοκληρωμένες υπηρεσίες χρησιμοποιούν κάποιες λειτουργίες για να διαχειριστούν τη συμφόρηση και να παρέχουν ποιότητα υπηρεσιών. Οι λειτουργίες αυτές είναι ο έλεγχος αποδοχής, ο αλγόριθμος δρομολόγησης, η πειθαρχία ουράς και η πολιτική απόρριψης. Ο έλεγχος ροής απαιτεί να γίνεται μια κράτηση, μέσω του πρωτοκόλλου RSVP, για κάθε νέα ροή που βασίζεται στην ποιότητα υπηρεσιών. Αν οι διαθέσιμοι πόροι, που προσδιορίζουν οι δρομολογητές, δεν μπορούν να εγγυηθούν την απαιτούμενη Ποιότητα Υπηρεσίας τότε η ροή δεν γίνεται δεκτή. Ο αλγόριθμος δρομολόγησης είναι πιθανό να βασίζεται σε διάφορες παραμέτρους της Ποιότητας Υπηρεσιών και όχι μόνο στην ελάχιστη καθυστέρηση. Η πειθαρχία ουράς χρησιμοποιείται από τους δρομολογητές και είναι ένα σημαντικό συστατικό της υλοποίησης της αρχιτεκτονικής που βασίζεται σε αλγορίθμους χρονοδρομολόγησης (όπως FIFO, Δίκαια διαχείριση) λαμβάνοντας υπόψη τις διαφορετικές απαιτήσεις των ροών. Η πολιτική απόρριψης καθορίζει ποιά πακέτα θα απορριφθούν όταν γεμίσει μια ενδιάμεση μνήμη και συνεχίζουν να φτάνουν νέα πακέτα.

Σύμφωνα με τις ολοκληρωμένες υπηρεσίες, κάθε δρομολογητής που τις χρησιμοποιεί και κάθε εφαρμογή που απαιτεί εγγυήσεις οφείλουν να κάνουν ατομική κράτηση. Ο λόγος για τον οποίο γίνεται η κράτηση περιγράφεται στα

FlowSpecs (χαρακτηριστικά ροής), ενώ η κράτηση γίνεται μέσω του πρωτοκόλλου RSVP. Υπάρχουν δυο είδη FlowSpecs: το TSpec και το RSpec. Το TSpec (TrafficSpecification) μας δείχνει τα χαρακτηριστικά κυκλοφορίας της ροής δεδομένων που στέλνει ο αποστολέας. Τα TSpecs περιλαμβάνουν παραμέτρους του αλγόριθμου token bucket (κουβάς κουπονιών). Το RSpec προσδιορίζει τις απαιτήσεις που υπάρχουν για τη ροή δεδομένων. Για παράδειγμα, μπορεί να είναι μια φυσιολογική "best-effort" ροή για την οποία δεν χρειάζεται κράτηση. Η ρύθμιση "controlled-load" δείχνει ένα «ελαφρά» φορτωμένο δίκτυο με σταθερή καθυστέρηση και απόρριψη πακέτων. Η ρύθμιση "guaranteed" υπόσχεται ένα άνω όριο στην καθυστέρηση και μηδενικές απώλειες πακέτων με την προϋπόθεση η κυκλοφορία να είναι εντός προδιαγραφών.

3.1.2 Τρόπος λειτουργίας[15]



Σχήμα 10: Υλοποίηση ολοκληρωμένων υπηρεσιών

Στο παραπάνω σχήμα βλέπουμε την γενική απεικόνιση της αρχιτεκτονικής ολοκληρωμένων υπηρεσιών στο εσωτερικό ενός δρομολογητή. Κάτω από την οριζόντια γραμμή είναι οι λειτουργίες προώθησης του δρομολογητή. Πάνω από τη γραμμή βρίσκονται οι λειτουργίες υπόβαθρου που δημιουργούν δομές δεδομένων υποστηρίζοντας τη βασική λειτουργία του δρομολογητή, την προώθηση πακέτων.

Οι βασικές λειτουργίες υπόβαθρου είναι οι εξής:

- **Πρωτόκολλο δέσμευσης**: Χρησιμοποιείται ανάμεσα στους δρομολογητές και μεταξύ δρομολογητών και τερματικών. Το πρωτόκολλο δέσμευσης που χρησιμοποιείται εδώ είναι το RSVP, το οποίο δεσμεύει πόρους για μια νέα ροή σε ένα συγκεκριμένο επίπεδο Ποιότητας Υπηρεσίας. Διατηρεί πληροφορίες κατάστασης για μια συγκεκριμένη ροή σε όλους τους

δρομολογητές και τερματικά κατά μήκος της διαδρομής της. Ενημερώνει τη βάση δεδομένων ελέγχου κυκλοφορίας που χρησιμοποιείται από τον χρονοδρομολογητή πακέτων για να προσδιοριστεί η υπηρεσία κάθε ροής.

- **Έλεγχος αποδοχής:** Η εκτέλεσή της προκαλείται από το πρωτόκολλο δέσμευσης όταν ζητείται μια νέα ροή. Καθορίζει αν υπάρχουν επαρκείς πόροι για τη συγκεκριμένη ροή με βάση τη ζητούμενη Ποιότητα Υπηρεσίας, τις υποχρεώσεις προς άλλες δεσμεύσεις και το τρέχον φορτίο του δικτύου.

- **Πράκτορας διαχείρισης**: Τροποποιεί τη βάση δεδομένων του ελέγχου κυκλοφορίας και κατευθύνει τη μονάδα ελέγχου αποδοχής για να καθορίσει τις πολιτικές ελέγχου αποδοχής.

- **Πρωτόκολλο δρομολόγησης**: Διατηρεί μια βάση δεδομένων δρομολόγησης που δίνει τον επόμενο κόμβο για κάθε διεύθυνση προορισμού και κάθε ροή.

Αυτές οι λειτουργίες υπόβαθρου υποστηρίζουν τις δυο παρακάτω βασικές περιοχές της λειτουργίας προώθησης ενός δρομολογητή:

- **Ταξινομητής και επιλογή διαδρομής**: Τα εισερχόμενα πακέτα ταξινομούνται σε κατηγορίες για λόγους προώθησης και ελέγχου κυκλοφορίας. Μια κατηγορία μπορεί να αντιστοιχεί σε μια ροή ή σε μια ομάδα ροών με τις ίδιες απαιτήσεις σε Ποιότητα Υπηρεσίας. Η επιλογή της κατηγορίας βασίζεται στα πεδία της IP επικεφαλίδας. Βάσει της κατηγορίας του πακέτου και της IP διεύθυνσης προορισμού του, αυτή η λειτουργία προσδιορίζει τη διεύθυνση του επόμενου ενδιάμεσου κόμβου.

- **Χρονοδρομολογητής πακέτων:** Διαχειρίζεται μία ή περισσότερες ουρές για κάθε θύρα εξόδου. Καθορίζει τη σειρά μετάδοσης των πακέτων μιας ουράς και, αν είναι αναγκαίο, ποια πακέτα θα απορριφθούν. Αυτά γίνονται με βάση την κατηγορία του πακέτου, τα περιεχόμενα της βάσης δεδομένων ελέγχου κυκλοφορίας και την δραστηριότητα της θύρας εξόδου. Επίσης, ένα μέρος της λειτουργίας του χρονοδρομολογητή πακέτων είναι η πολιτική που καθορίζει αν η κυκλοφορία πακέτων σε μια ροή ξεπερνά τη ζητούμενη χωρητικότητα και, αν ναι, αποφασίζει πως θα χειριστούν τα επιπλέον πακέτα.

3.1.3 Υπηρεσίες αρχιτεκτονικής[1][15]

Στις ολοκληρωμένες υπηρεσίες εκτός της βέλτιστης προσπάθειας (best-effort) υπάρχουν ακόμη δυο κατηγορίες υπηρεσιών: η εγγυημένη υπηρεσία και η υπηρεσία ελεγχόμενου φορτίου. Για καθεμιά από αυτές τις δυο κατηγορίες μπορούν να οριστούν "άπειρες" υπηρεσίες , για παράδειγμα στην Guaranteed: 5Mbps και end-to-enddelay 2ms, 5Mbps και end-to-enddelay 10ms, κοκ. . Αυτό είναι και ένα πλεονέκτημα των ολοκληρωμένων υπηρεσιών.

- Η **εγγυημένη υπηρεσία (guaranteed)** δίνει αυστηρές εγγυήσεις για την καθυστέρηση και το εύρος ζώνης που θα παρέχονται σε μια ροή πληροφορίας. Παρέχει εξασφαλισμένη χωρητικότητα ή ρυθμό μετάδοσης δεδομένων, ένα καθορισμένο άνω όριο της καθυστέρησης ουράς (buffering) και μηδενικές απώλειες ουράς. Η καθυστέρηση ουράς εξαρτάται από την εγγυημένη υπηρεσία που επιλέγουμε, ενώ η καθυστέρηση διάδοσης εξαρτάται από τη διαδρομή που επιλέγουμε. Η συνολική καθυστέρηση που θα έχει ένα πακέτο είναι το άθροισμά τους. Πρέπει να αναφέρουμε ότι η υπηρεσία δεν παρέχει καμιά εγγύηση για την διακύμανση της καθυστέρησης (jitter) και πακέτα είναι πιθανό να απορριφθούν λόγω βλαβών στο δίκτυο ή αλλαγών στις διαδρομές δρομολόγησης. Η υπηρεσία απευθύνεται για εφαρμογές με αυστηρές προθεσμίες πραγματικού χρόνου (π.χ. μια εφαρμογή ήχου καθιστά άχρηστα όσα πακέτα φτάσουν μετά από κάποιο χρονικό όριο) και για εκείνες που χρειάζονται ένα άνω όριο σχετικά με την καθυστέρηση.
  Με αυτή την υπηρεσία, ο χρήστης περιγράφει στο δίκτυο το προφίλ κυκλοφορίας του. Έπειτα, κάθε δρομολογητής υπολογίζει τις παραμέτρους που δείχνουν την συμπεριφορά που θα έχει αυτή η κυκλοφορία στο δίκτυο. Αφού αθροιστούν οι διάφορες παράμετροι που επιστρέφουν οι δρομολογητές μιας διαδρομής, μπορεί να υπολογιστεί η μέγιστη δυνατή καθυστέρηση ενός πακέτου που θα κάνει αυτή τη διαδρομή. Σε όλο το μήκος της διαδρομής που θα χρησιμοποιηθεί, η υπηρεσία δεσμεύει πόρους για κάθε ροή. Αν μια ροή τηρεί το συμβόλαιό της, τότε το δίκτυο της παρέχει την εγγυημένη υπηρεσία. Ο χρήστης για να χρησιμοποιήσει την εγγυημένη υπηρεσία πρέπει να περιγράψει στο δίκτυο την επιθυμητή κυκλοφορία (TSpec) και την ποιότητα υπηρεσίας (RSpec). Οι δρομολογητές που παρέχουν εγγυημένη υπηρεσία, με τη σειρά τους, στέλνουν πληροφορίες στο χρήστη σχετικά με την απόκλιση από την επιθυμητή κυκλοφορία. Για να αποφεύγεται η απόκλιση, εφαρμόζονται τεχνικές αστυνόμευσης. Η κανονική αστυνόμευση (policing) ελέγχει αν οι ροές τηρούν το συμβόλαιο και η μορφοποίηση (shaping) αναδιαμορφώνει τα κατεστραμμένα χαρακτηριστικά μιας ροής ώστε να τηρεί το συμβόλαιό της.

- Η **υπηρεσία ελεγχόμενου φορτίου (controlled-load**) το μόνο που εγγυάται είναι ότι οι ροές κυκλοφορίας θα έχουν μεταχείριση σαν να ανήκουν στην best-effort υπηρεσία, χωρίς όμως να υπάρχει συμφόρηση στο δίκτυο. Εξασφαλίζει ότι τα περισσότερα πακέτα δεν θα συναντήσουν καθυστερήσεις μεγαλύτερες από την ελάχιστη καθυστέρηση διέλευσης, η οποία αποτελείται από την καθυστέρηση διάδοσης και την καθυστέρηση λόγω επεξεργασίας από τον δρομολογητή. Επιπλέον, εξασφαλίζει ότι ένα πολύ μεγάλο ποσοστό των πακέτων θα παραδοθεί στον προορισμό του και σχεδόν όλα τα υπόλοιπα πακέτα δεν θα καταφέρνουν να φτάσουν λόγω λαθών στο φυσικό επίπεδο. Η υπηρεσία απευθύνεται σε προσαρμοστικές εφαρμογές πραγματικού χρόνου (π.χ. μια εφαρμογή φωνής που ρυθμίζει τις σιωπηρές περιόδους) και για εφαρμογές που λειτουργούν καλά σε δίκτυο χωρίς συμφόρηση αλλά υπολειτουργούν σε συνθήκες συμφόρησης.
  Για να εγγυηθεί το δίκτυο αυτή την υπηρεσία, ο χρήστης πρέπει να δηλώσει με τη χρήση του TSpec ορισμένα χαρακτηριστικά της κυκλοφορίας που αποστέλλει στο δίκτυο. Αυτό έχει ως αποτέλεσμα την δέσμευση πόρων από το δίκτυο και έτσι το δίκτυο παρέχει στον χρήστη την απαιτούμενη ποιότητα

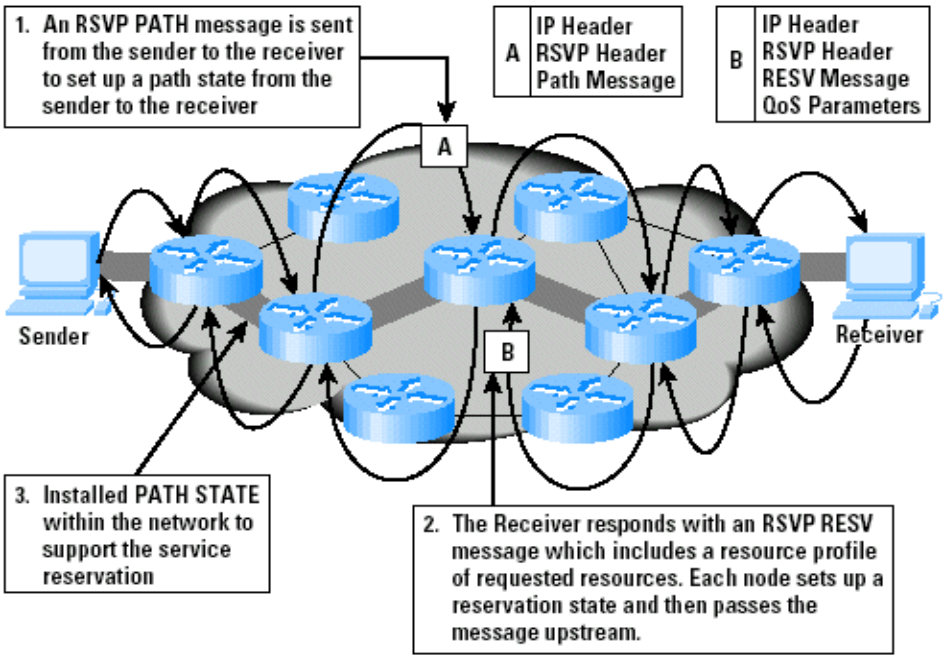
υπηρεσίας για την κυκλοφορία που ζήτησε. Όσες ροές δεν παραβιάζουν το συμβόλαιο κυκλοφορίας με το δίκτυο, λαμβάνουν την ποιότητα υπηρεσίας που τους εγγυήθηκε η υπηρεσία ελεγχόμενου φορτίου. Σε περίπτωση που μια ροή παραβιάζει το συμβόλαιο κυκλοφορίας, το δίκτυο συνεχίζει την παροχή ποιότητας υπηρεσιών στις υπόλοιπες ροές αλλά τα πακέτα αυτής της ροής μεταφέρονται μέσω της κατηγορίας best-effort.

3.1.4 Το πρωτόκολλο RSVP[3][8][15][26][27]

Το **Πρωτόκολλο Δέσμευσης Πόρων (ResourcereSerVationProtocol - RSVP)** παρέχει υποστήριξη στη λειτουργία της αρχιτεκτονικής ολοκληρωμένων υπηρεσιών καθώς χρησιμοποιείται για την δέσμευση των πόρων του δικτύου. Η παράδοση δεδομένων με την επιθυμητή Ποιότητα Υπηρεσίας σε ένα δίκτυο είναι δύσκολη υπόθεση όσο αυξάνεται ο αριθμός των χρηστών, ο ρυθμός μετάδοσης δεδομένων των εφαρμογών και η χρήση πολυεκπομπής. Το RSVP επιτρέπει στις εφαρμογές να δεσμεύουν τους πόρους για μια δεδομένη Ποιότητα Υπηρεσίας, βελτιστοποιώντας τη χρήση του εύρους ζώνης και παράλληλα αποτρέποντας τη συμφόρηση του δικτύου.

Για εφαρμογές μονοεκπομπής, το πρωτόκολλο επιτρέπει τη δέσμευση πόρων για να λάβουν την επιθυμητή ποιότητα υπηρεσίας. Οι δρομολογητές της επιδιωκόμενης διαδρομής, μπορούν να δεσμεύσουν από πριν τους πόρους (χώρος ουράς και χωρητικότητα εξόδου) για να εξασφαλίσουν την επιθυμητή ποιότητα υπηρεσίας. Αν ένας δρομολογητής δεν μπορεί να ανταποκριθεί λόγω προηγούμενων υπερβολικών δεσμεύσεων, τότε η εφαρμογή ενημερώνεται. Στη συνέχεια, η εφαρμογή επιλέγει αν θέλει μια νέα δέσμευση πόρων, όμως μειωμένης ποιότητας υπηρεσιών, ή να δοκιμάσει πάλι αργότερα. Σε εφαρμογές πολυεκπομπής, ένα μεγάλο μέρος του πιθανού φορτίου που παράγουν μπορεί να αποτραπεί εύκολα καθώς κάποια μέλη της ομάδας πολυεκπομπής ίσως να μην χρειάζονται την παράδοση δεδομένων από μια συγκεκριμένη πηγή για μια χρονική στιγμή ή να μπορούν να διαχειριστούν μόνο ένα μέρος της μετάδοσης πηγής. Έτσι, το RSVP, αν είναι εφικτό, δίνει τη δυνατότητα στους δρομολογητές να αποφασίσουν από πριν αν μπορούν να ανταποκριθούν στις απαιτήσεις της εκάστοτε εφαρμογής πολυεκπομπής.

Πριν γίνει η δέσμευση πόρων από μια εφαρμογή, η εφαρμογή αυτή θα πρέπει πρώτα να κάνει μια αίτηση για την επιθυμητή ποιότητα υπηρεσίας της ροής δεδομένων της. Σε αυτό το σημείο να αναφέρουμε ότι το RSVP δεν είναι πρωτόκολλο δρομολόγησης αλλά σηματοδοσίας και λειτουργεί ως εξής (Σχήμα 10):



Σχήμα 11: Λειτουργία RSVP

A. Ο αποστολέας (Sender) στέλνει μηνύματα PATH, δηλαδή αιτήσεις καταγραφής διαδρομής και χαρακτηριστικών ροής δεδομένων, στον παραλήπτη (Receiver) . Τα μηνύματα PATH εγκαθιστούν «κατάσταση διαδρομής» (pathstate) σε κάθε κόμβο κατά μήκος της διαδρομής. Η κατάσταση διαδρομής κρατάει την IP διεύθυνση του κόμβου του προηγούμενου βήματος, η οποία χρησιμοποιείται για την δρομολόγηση των Resv μηνυμάτων στην αντίθετη κατεύθυνση.

B. Ο δέκτης στέλνει μηνύματα Resv, δηλαδή αιτήσεις δέσμευσης πόρων, προς τον αποστολέα. Αυτά τα μηνύματα ακολουθούν την αντίθετη διαδρομή από τα πακέτα δεδομένων και τα μηνύματα PATH, χρησιμοποιώντας την κατάσταση διαδρομής που είχαν εγκαταστήσει τα μηνύματα PATH. Δημιουργούν και διατηρούν «κατάσταση δέσμευσης» (reservationstate) σε κάθε κόμβο κατά μήκος του μονοπατιού. Όταν παραδοθούν τα μηνύματα Resv στον αποστολέα τότε τίθενται οι κατάλληλες παράμετροι ελέγχου κυκλοφορίας για το πρώτο βήμα.

- Έστω και αν ένας δρομολογητής δεν μπορεί να εξασφαλίσει τους ζητούμενους πόρους στέλνει μήνυμα λάθους στον παραλήπτη, Αλλιώς το μήνυμα προωθείται στον επόμενο δρομολογητή.
- Όταν ο πιο κοντινός στον αποστολέα δρομολογητής μπορεί να εξασφαλίσει τους πόρους στέλνει ένα μήνυμα επιβεβαίωσης στον αποστολέα. Στη συνέχεια αρχίζει η μετάδοση.

Μειονεκτήματα

Το RSVP παρέχει QoS δεσμεύοντας πόρους, προς μια κατεύθυνση ενός υπάρχοντος μονοπατιού. Δηλαδή, διανύοντας ένα μονοπάτι hop-by-hop ζητάει τους απαιτούμενους πόρους. Εάν δεν υπάρχουν διαθέσιμοι πόροι τότε δεν είναι εφικτή η μετάδοση των πακέτων δεδομένων. Τρία θέματα δρομολόγησης αντιμετωπίζονται επομένως σε ένα πρωτόκολλο σαν αυτό: να βρεθεί μια διαδρομή που να υποστηρίζει δέσμευση πόρων, να βρεθεί μια διαδρομή που να διαθέτει επαρκή και μη δεσμευμένη χωρητικότητα για την εισερχόμενη ροή και να υπάρχει τρόπος να αντιμετωπιστεί η απώλεια ενός δρομολογητή και η αντικατάστασή του.

3.1.5 Μειονεκτήματα[1][3][25][28]

Παρά το γεγονός ότι οι ολοκληρωμένες υπηρεσίες παρέχουν αρκετά πλεονεκτήματα κατά τη χρήση τους, όπως είναι η εγγυημένη παροχή ποιότητας υπηρεσιών σε κάθε ροή δεδομένων, παρουσιάζουν και μειονεκτήματα μερικά από τα οποία αναφέρονται παρακάτω:

- Το μεγαλύτερο μειονέκτημα είναι το γεγονός ότι η σηματοδότηση /τροφοδότηση συμβαίνει χωριστά από την διαδικασία δρομολόγησης. Για παράδειγμα, μπορεί να υπάρχει μια διαδρομή στο δίκτυο που διαθέτει τους απαιτούμενους πόρους ακόμα κι όταν το πρωτόκολλο RSVP αποτυγχάνει να βρει τους πόρους.

- Προκειμένου να λειτουργούν αποδοτικά οι ολοκληρωμένες υπηρεσίες, όλοι οι δρομολογητές κατά μήκος της διαδρομής της κυκλοφορίας στο δίκτυο θα πρέπει να τις υποστηρίζουν. Κάθε δρομολογητής θα πρέπει να αποθηκεύει πληροφορίες για πολλές καταστάσεις. Με αποτέλεσμα, οι ολοκληρωμένες υπηρεσίες να λειτουργούν σε μικρή κλίμακα. Αλλά σε ένα αναβαθμισμένο δίκτυο στο μέγεθος του Διαδικτύου, είναι δύσκολο να παρακολουθούνται όλες οι κρατήσεις. Επομένως, είναι εμφανής η δυσκολία επεκτασιμότητας.

- Οι ολοκληρωμένες υπηρεσίες για να εξασφαλίσουν τη δέσμευση πόρων χρησιμοποιούν ένα πρωτόκολλο σηματοδοσίας, το RSVP, το οποίο όμως είναι πολύπλοκο.

- Για την εγκαθίδρυση κάθε ροής απαιτείται εκ των προτέρων διευθέτηση και αυτό δεν είναι καλό όταν υπάρχουν χιλιάδες ή εκατομμύρια ροές.

- Οι δρομολογητές διατηρούν μια εσωτερική κατάσταση ανά ροή και στην περίπτωση κατάρρευσής τους θα δημιουργούνταν προβλήματα.

- Απαιτείται αστυνόμευση για κάθε ξεχωριστή ροή που έχει και διαφορετικές απαιτήσεις ανάλογα με το συμβόλαιο ροής ανάμεσα στο χρήστη και το δίκτυο.

3.2 Διαφοροποιημένες υπηρεσίες

3.2.1 Εισαγωγή[1][3][15]

Λόγω των μειονεκτημάτων των Ολοκληρωμένων Υπηρεσιών, η IETF επινόησε ένα απλούστερο μοντέλο που μπορεί να υλοποιηθεί τοπικά σε κάθε δρομολογητή χωρίς εκ των προτέρων διευθέτηση και χωρίς να εμπλέκεται όλη η διαδρομή. Το μοντέλο των **Διαφοροποιημένων Υπηρεσιών (DifferentiatedServices – DiffServ)** βασίζεται σε κατηγορίες, με τα πακέτα που ανήκουν σε διαφορετικές κατηγορίες να έχουν διαφορετική μεταχείριση, και όχι σε ροές όπως οι ολοκληρωμένες υπηρεσίες. Η κυκλοφορία μέσα σε μια κατηγορία πιθανόν να διαμορφώνεται με κάποια συγκεκριμένη μορφή, όπως με έναν τρύπιο κουβά (leakybucket). Οι διαφοροποιημένες υπηρεσίες κάνουν ενοποίηση πολλών διαφορετικών ροών καθώς ολόκληρη η κυκλοφορία που χαρακτηρίζεται από την ίδια τιμή του πεδίου DS, αντιμετωπίζεται με τον ίδιο τρόπο. Επιπλέον, αυτές οι υπηρεσίες δεν χρησιμοποιούν πρωτόκολλο για τη δέσμευση πόρων. Στην περίπτωση που οι διαφοροποιημένες και οι ολοκληρωμένες υπηρεσίες συνυπάρχουν στο δίκτυο, θα πρέπει να "προστατεύσουμε" τους πόρους που χρησιμοποιούν οι διαφοροποιημένες υπηρεσίες. Αυτό το κάνουμε επειδή οι ολοκληρωμένες υπηρεσίες, λόγω του πρωτοκόλλου RSVP που χρησιμοποιούν, μπορεί να δεσμεύσουν όλους τους πόρους του δικτύου με αποτέλεσμα να μην υπάρχουν πόροι για τις διαφοροποιημένες υπηρεσίες.

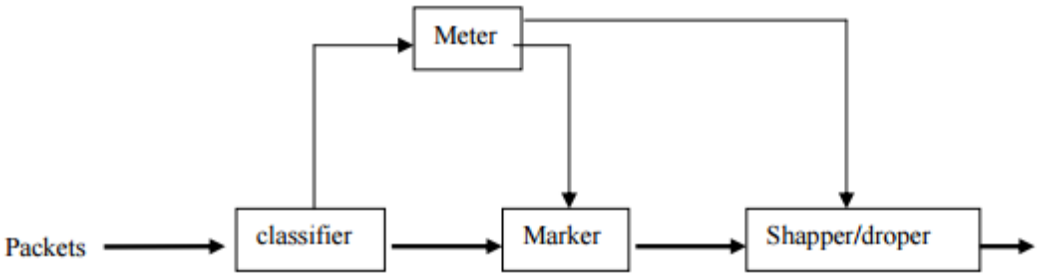
Οι διαφοροποιημένες υπηρεσίες παρέχονται από ένα σύνολο δρομολογητών οι οποίοι σχηματίζουν μια περιοχή DS. Η περιοχή DS είναι ένα συνεχές τμήμα του Διαδικτύου, όπου παρέχεται ένα σταθερό σύνολο πολιτικών των διαφοροποιημένων υπηρεσιών. Οι υπηρεσίες που παρέχονται μέσα σε μια περιοχή DS ορίζονται από κάποια Συμφωνία σε Επίπεδο Υπηρεσίας (ServiceLevelAgreement – SLA) μεταξύ ενός πελάτη και ενός παρόχου, η οποία καθορίζει την υπηρεσία προώθησης που πρέπει να λαμβάνει ο πελάτης για διάφορες κατηγορίες πακέτων. Πελάτης μπορεί να είναι ένας οργανισμός ή μια άλλη περιοχή DS. Τα εισερχόμενα πακέτα του πελάτη στην περιοχή DS επισημαίνονται ώστε να δείχνουν σε ποια κατηγορία ανήκουν. Η επισήμανση γίνεται μέσω του πεδίου DS των πρωτοκόλλων IPv4 και IPv6. Ο πάροχος πρέπει να εξασφαλίζει ότι ο πελάτης λαμβάνει τη συμφωνηθείσα Ποιότητα Υπηρεσίας για κάθε κατηγορία πακέτου. Αυτό το κατορθώνει διαμορφώνοντας τις κατάλληλες πολιτικές προώθησης σε κάθε δρομολογητή, με βάση την τιμή του πεδίου DS, και μετρώντας διαρκώς την απόδοση που παρέχεται για κάθε κατηγορία. Όταν ο πελάτης παραδίδει πακέτα με προορισμό εντός της περιοχής DS, αναμένεται να παρέχεται η συμφωνηθείσα υπηρεσία. Στην περίπτωση όμως που ο προορισμός βρίσκεται εκτός της περιοχής DS, η περιοχή DS

θα προσπαθήσει να προωθήσει τα πακέτα μέσω άλλων περιοχών, ζητώντας την καταλληλότερη υπηρεσία για να ικανοποιήσει εκείνη που έχει ζητηθεί.

3.2.2 Τρόπος λειτουργίας[1][3][15]

Εντός μιας περιοχής DS, η μεταβίβαση από έναν δρομολογητή σε έναν άλλον γίνεται μέσω μιας διαδρομής που δεν συμπεριλαμβάνει κανένα δρομολογητή εκτός της περιοχής. Σε μια τέτοια περιοχή, παρέχεται μια ομοιόμορφη, σταθερή υπηρεσία λόγω της ομοιόμορφης ερμηνείας των κωδικοσημείων DS. Οι δρομολογητές αυτής της περιοχής μπορεί να είναι εσωτερικοί κόμβοι ή κόμβοι ορίου. Οι εσωτερικοί κόμβοι υλοποιούν την πειθαρχία ουράς, η οποία παρέχει προνομιακή μεταχείριση βάσει της τιμής του κωδικοσημείου, τους κανόνες απόρριψης πακέτων, οι οποίοι υπαγορεύουν ποια πακέτα θα απορριφθούν πρώτα σε περίπτωση κορεσμού της ενδιάμεσης μνήμης, και την PHB (PerHopBehavior). Η PHB είναι το μοναδικό τμήμα των διαφοροποιημένων υπηρεσιών που υλοποιείται στους εσωτερικούς κόμβους καθώς αυτοί οι κόμβοι τυπικά υλοποιούν απλούς μηχανισμούς διαχείρισης πακέτων. Οι κόμβοι ορίου υλοποιούν μηχανισμούς PHB και πιο εξελιγμένους μηχανισμούς συνθηκών κυκλοφορίας για να παρέχεται η επιθυμητή υπηρεσία.

Στη συνέχεια θα δούμε τα πέντε στοιχεία από τα οποία αποτελούνται οι μηχανισμοί συνθηκών κυκλοφορίας (Σχήμα 12):



Σχήμα 12: Συνθήκες κυκλοφορίας DS

- **Ταξινομητής (classifier)**: Αποτελεί την βάση για την παροχή διαφοροποιημένων υπηρεσιών και διαχωρίζει τα πακέτα που παραδίδονται σε διαφορετικές κατηγορίες. Πιθανόν να διαχωρίζει την κυκλοφορία μόνο βάσει του κωδικοσημείου DS (συμπεριφορά ταξινομητή συνάθροισης) ή βάσει πολλαπλών τιμών πεδίων της επικεφαλίδας του πακέτου ή ακόμη και του ωφέλιμου φορτίου του πακέτου (ταξινομητής πολλαπλών πεδίων).

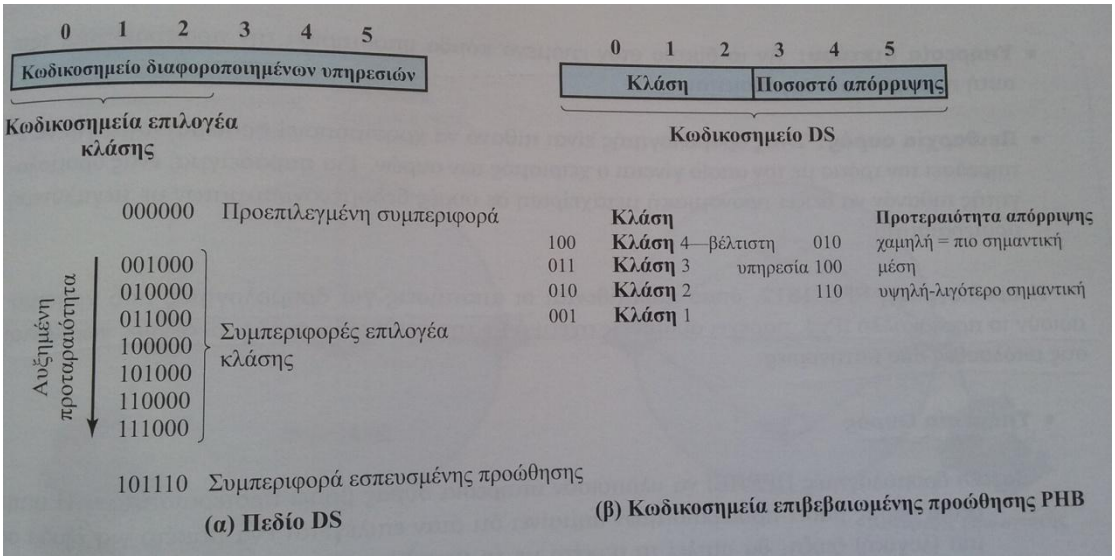
- **Καταμετρητής (meter)**: Πραγματοποιεί μετρήσεις του όγκου των πακέτων σε ένα συγκεκριμένο χρονικό διάστημα για να καθορίσει το βαθμό συμμόρφωσης της ροής με το προφίλ κυκλοφορίας, ώστε να αποφασίσει αν μια συγκεκριμένη κατηγορία είναι εντός του εγγυημένου επιπέδου υπηρεσίας για αυτήν την κατηγορία ή αν το υπερβαίνει.

- **Μηχανισμός σήμανσης (marker)**: Στα πακέτα που υπερβαίνουν το προφίλ κυκλοφορίας, είναι αναγκαίο να γίνει εκ νέου σήμανση με διαφορετική τιμή κωδικοσημείου. Ίσως τα πακέτα σημανθούν για να διακινηθούν με υπηρεσίες χαμηλότερης ποιότητας επιτρέποντάς τους έτσι την πρόσβαση στην περιοχή διαφοροποιημένων υπηρεσιών. Επίσης, η εκ νέου σήμανση ίσως είναι αναγκαία να γίνεται στο όριο μεταξύ δύο περιοχών διαφοροποιημένων υπηρεσιών.

- **Μηχανισμός διαμόρφωσης (shapper)**: Καθυστερεί τα πακέτα όσο χρειάζεται, για να μην υπερβεί η ροή πακέτων μιας κατηγορίας το ρυθμό κυκλοφορίας του προφίλ αυτής της κατηγορίας. Επίσης, είναι πιθανό να απορροφά μια ριπή πακέτων, τοποθετώντας τα σε μια ενδιάμεση μνήμη και ρυθμίζοντάς τα για μεγαλύτερη χρονική περίοδο.

- **Μηχανισμός απόρριψης (dropper)**: Στην περίπτωση που ο ρυθμός κυκλοφορίας μιας κατηγορίας υπερβαίνει εκείνον του προφίλ της κατηγορίας, ο μηχανισμός απορρίπτει πακέτα. Επίσης, ίσως να απορρίπτει πακέτα αν κορεστεί η ενδιάμεση μνήμη που χρησιμοποιείται για την περαιτέρω ρύθμιση.

3.2.3 Κωδικοσημεία DS[1][15]



Σχήμα 13: Κωδικοσημεία DS

Η **τιμή του πεδίου DS** ή, αλλιώς, **κωδικοσημείο DS (DScodepoint)** είναι η ετικέτα που χρησιμοποιείται για την ταξινόμηση των πακέτων ώστε να λαμβάνουν διαφοροποιημένες υπηρεσίες. Η ετικέτα είναι το πεδίο DS μεγέθους 6 bits που βρίσκεται στην επικεφαλίδα του IPv4 ή του IPv6. Με ένα κωδικοσημείο DS μεγέθους 6 bits μπορούν να οριστούν $2^6 = 64$ κατηγορίες κυκλοφορίας. Τα κωδικοσημεία μπορούν να κατανεμηθούν σε τρεις περιοχές ως εξής:

1. Κωδικοσημεία της μορφής 'xxxxx0', όπου x=0 ή 1: δεσμεύονται για καθορισμό ως πρότυπα.
2. Κωδικοσημεία της μορφής 'xxxx11': δεσμεύονται για πειραματική ή τοπική χρήση.
3. Κωδικοσημεία της μορφής 'xxxx01': δεσμεύονται για πειραματική ή τοπική χρήση, αλλά ίσως κατανέμονται και για μελλοντικά πρότυπα.

Το κωδικοσημείο '000000' αποτελεί την προεπιλεγμένη κατηγορία πακέτων, η οποία είναι η συμπεριφορά προώθησης βέλτιστης προσπάθειας. Όταν η χωρητικότητα της ζεύξης είναι διαθέσιμη, τότε αυτά τα πακέτα προωθούνται με τη σειρά που λαμβάνονται. Αν είναι διαθέσιμα για μετάδοση πακέτα άλλων κατηγοριών DS με υψηλότερη προτεραιότητα, τότε θα προτιμηθούν αυτά έναντι των πακέτων βέλτιστης προσπάθειας.

Τα κωδικοσημεία της μορφής 'xxx000' δεσμεύονται για να παρέχουν συμβατότητα με την υπηρεσία προτεραιότητας του IPv4. Το πεδίο Τύπου Υπηρεσίας (TypeofService – ToS) του IPv4 αποτελείται από δυο υποπεδία: ένα προτεραιότητας των 3 bits και ένα ToS των 4 bits. Το υποπεδίο προτεραιότητας παρέχει καθοδήγηση για την κατανομή πόρων του δρομολογητή και το υποπεδίο ToS παρέχει καθοδήγηση στην πηγή ή στον δρομολογητή ώστε να επιλέξουν τον επόμενο κόμβο. Το πεδίο ToS χρησιμοποιείται μόνο για να προσδιοριστεί η ποιότητα υπηρεσίας. Η δρομολόγηση των πακέτων γίνεται με τον κλασσικό τρόπο. Τα πακέτα διαφορετικών ροών που έχουν την ίδια τιμή στο πεδίο ToS τοποθετούνται στην ίδια ουρά αλλά δρομολογούνται από άλλα πεδία του IP πακέτου.

3.2.4 Συμπεριφορές ανά κόμβο(PHBs)[3][15][28]

Οι **Συμπεριφορές Ανά Κόμβο (PerHopBehavior – PHB)** αντιστοιχούν στη μεταχείριση που θα έχει το πακέτο σε κάθε ενδιάμεσο δρομολογητή του δικτύου διαφοροποιημένων υπηρεσιών, και όχι σε μια εγγύηση κατά μήκος του δικτύου. Στα πακέτα κάποιων συμπεριφορών ανά κόμβο παρέχονται καλύτερες υπηρεσίες (όπως άμεση εξυπηρέτηση) από ότι σε άλλα. Υπάρχουν τέσσερις διαθέσιμες συμπεριφορές για την κατασκευή ενός δικτύου διαφοροποιημένων υπηρεσιών και την επίτευξη Ποιότητας υπηρεσίας από άκρο σε άκρο:

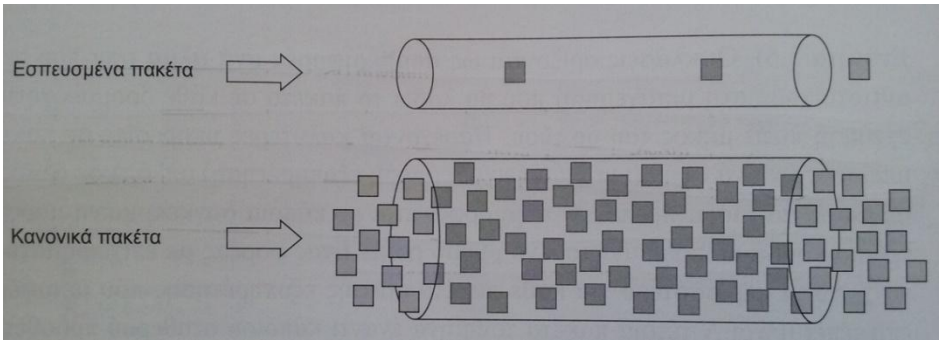
- **Προεπιλεγμένη συμπεριφορά (DefaultPHB):** Πρόκειται για την υπηρεσία βέλτιστης προσπάθειας. Τα πακέτα με κωδικοσημείο DS '000000' (Σχήμα 13(α)) εξυπηρετούνται με τον παραδοσιακό τρόπο από έναν συμβατό DS κόμβο. Όταν μια κυκλοφορία, κατά συνέπεια και τα πακέτα της, δεν πληρούν τις απαιτήσεις από τις υπόλοιπες καθορισμένες συμπεριφορές, τότε μεταχειρίζονται από την προεπιλεγμένη συμπεριφορά.

- **Συμπεριφορές επιλογέα κατηγορίας (Class-SelectorPHBs):** Προκειμένου να διατηρηθεί η συμβατότητα με το IP σύστημα προτεραιότητας ορίζονται κάποια κωδικοσημεία DS. Τα πρώτα τρία bits αυτών των κωδικοσημείων είναι τα IP bits προτεραιότητας. Τα κωδικοσημεία DS (Σχήμα 13(α)) των πακέτων είναι τα εξής:

  1) '001000'
  2) '010000'
  3) '011000'
  4) '100000'
  5) '101000'
  6) '110000'
  7) '111000'

  Να αναφέρουμε ότι το κωδικοσημείο '000000' ανήκει κι αυτό στην παραπάνω ομάδα κωδικοσημείων. Κάθε IP τιμή προτεραιότητας μπορεί να αντιστοιχεί σε μια κατηγορία διαφοροποιημένων υπηρεσιών. Με αυτόν τον τρόπο εξασφαλίζεται ότι οι συμβατοί DS κόμβοι μπορούν να συνυπάρχουν με τους κόμβους IP προτεραιότητας. Η προτεραιότητα των πακέτων με τα παραπάνω κωδικοσημεία αυξάνεται όσο πηγαίνουμε προς το 7. Έτσι ένα πακέτο με κωδικοσημείο '110000' προωθείται πρώτο σε σχέση με ένα πακέτο με κωδικοσημείο '010000'.

- **Συμπεριφορά εσπευσμένης προώθησης (ExpeditedForwardingPHB – EFPHB):** Αυτή η συμπεριφορά είναι παρόμοια της εγγυημένης υπηρεσίας των ολοκληρωμένων υπηρεσιών και τα πακέτα που υπακούουν σε αυτή έχουν κωδικοσημείο DS '101110' (Σχήμα 13(α)). Η υπηρεσία εσπευσμένης προώθησης έχει ως χαρακτηριστικά το χαμηλό κόστος, την χαμηλή καθυστέρηση και διακύμανση καθυστέρησης από άκρο σε άκρο, τις μικρές απώλειες πακέτων και το εξασφαλισμένο εύρος ζώνης. Κάποιες εφαρμογές, όπως το VoIP, τα video και τα διαδικτυακά εμπορικά προγράμματα απαιτούν τέτοιου είδους χαρακτηριστικά.

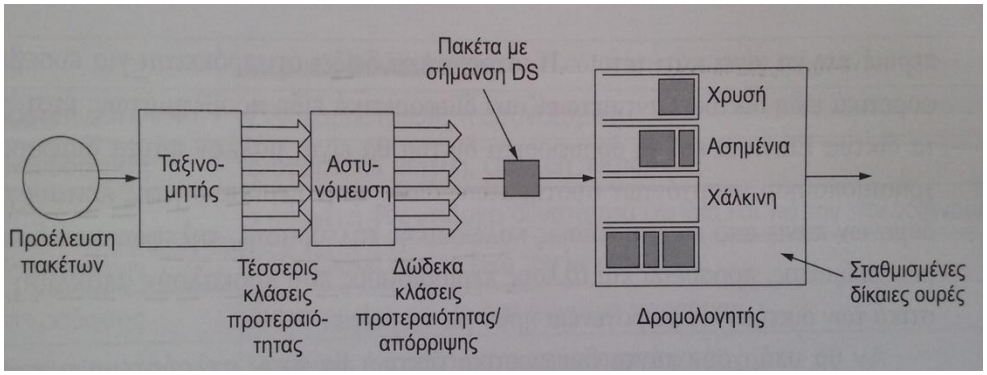


Σχήμα 14: Τα εσπευσμένα πακέτα 'βλέπουν' ένα μη φορτωμένο δίκτυο.

Τα πακέτα εδώ κατηγοριοποιούνται σε εσπευσμένα ή απλά και τους τοποθετείται η αντίστοιχη σήμανση. Η σήμανση πραγματοποιείται στον υπολογιστή υπηρεσίας αποστολής ή στον δρομολογητή εισόδου. Στόχος της συμπεριφοράς είναι τα εσπευσμένα πακέτα να συναντούν συνήθως μικρού μήκους ή κενές ουρές με αποτέλεσμα να ελαχιστοποιείται η καθυστέρηση και η διακύμανσή της. Επίσης, όταν οι ουρές θα παραμένουν μικρές σε σχέση με την ενδιάμεση μνήμη, η απώλεια πακέτων θα ελαχιστοποιείται.

Η εσπευσμένη προώθηση διαμορφώνει τους κόμβους με τέτοιο τρόπο ανεξάρτητα από την ένταση της υπόλοιπης κυκλοφορίας. Οι κόμβοι ορίου ελέγχουν τη συνολική κυκλοφορία περιορίζοντας τον ρυθμό και την ανομοιομορφία της. Οι εσωτερικοί κόμβοι διαχειρίζονται την εισερχόμενη κυκλοφορία ώστε να μην υπάρχουν επιπτώσεις από τον σχηματισμό ουρών. Με αυτόν τον τρόπο ο μέγιστος ρυθμός αφίξεων της συνολικής κυκλοφορίας είναι μικρότερος από τον ελάχιστο ρυθμό αναχωρήσεών της.
Τα εσπευσμένα πακέτα θα πρέπει να μπορούν να διασχίζουν το δίκτυο σαν να μην υπάρχουν άλλα πακέτα (Σχήμα 14). Οι δρομολογητές ίσως έχουν δυο ουρές εξόδου για κάθε εξερχόμενη γραμμή, μία για τα εσπευσμένα πακέτα και μία για τα απλά. Όταν φτάνει ένα πακέτο, με βάση την κατηγορία του, μπαίνει στην κατάλληλη ουρά. Μέσω ενός χρονοπρογραμματιστή προτεραιότητας τα εσπευσμένα πακέτα προωθούνται πρώτα.
Αυτή η συμπεριφορά όμως έχει ένα μειονέκτημα, λόγω των προτεραιοτήτων πρέπει να διασπαστούν οι ροές πακέτων οποιασδήποτε άλλης κυκλοφορίας PHB, με συνέπεια την ανάγκη για χρήση μιας άλλης πιο εξελιγμένης πολιτικής διαχείρισης ουρών.

- **Συμπεριφορά επιβεβαιωμένης προώθησης (AssuredForwardingPHB – AFPHB):** Η συμπεριφορά είναι παρόμοια της υπηρεσίας ελεγχόμενου φορτίου των ολοκληρωμένων υπηρεσιών. Παρέχει μια υπηρεσία που είναι ανώτερη από την βέλτιστη προσπάθεια και δεν απαιτεί τη δέσμευση πόρων εντός του διαδικτύου και διάκριση ανάμεσα σε ροές διαφορετικών χρηστών. Το πλεονέκτημά της είναι η απλότητα καθώς απαιτείται λίγη εργασία από τους εσωτερικούς κόμβους. Οι κόμβοι ορίου κάνουν την σήμανση της κυκλοφορίας, με βάση τα προφίλ κυκλοφορίας, παρέχοντας διαφορετικά επίπεδα υπηρεσιών σε διαφορετικές κατηγορίες. Οι προτεινόμενες τιμές κωδικοσημείων για την υπηρεσία επιβεβαιωμένης προώθησης απεικονίζονται στο Σχήμα 13(β).
Η επιβεβαιωμένη προώθηση ορίζει τέσσερις κατηγορίες προτεραιότητας, επιτρέποντας τον ορισμό τεσσάρων προφίλ κυκλοφορίας και με κάθε κατηγορία να έχει τους δικούς της πόρους. Οι τρεις ανώτερες κατηγορίες μπορεί να αποκαλούνται χρυσή, ασημένια και χάλκινη. Ο χρήστης μπορεί να επιλέξει μία ή περισσότερες κατηγορίες. Εντός κάθε κατηγορίας, τα πακέτα αποκτούν σήμανση από τον πελάτη ή τον πάροχο υπηρεσίας, με μία από τις τρεις πιθανότητες απόρριψης των πακέτων που αντιμετωπίζουν συμφόρηση: χαμηλή, μεσαία και υψηλή. Συνδυάζοντας τις τέσσερις κατηγορίες προτεραιότητας με τις τρεις πιθανότητες απόρριψης ορίζονται 12 κατηγορίες υπηρεσιών (AFxyPHBs).



Σχήμα 15: Μια πιθανή υλοποίηση της επιβεβαιωμένης προώθησης.

Στο παραπάνω σχήμα (Σχήμα 15) βλέπουμε έναν τρόπο επεξεργασίας των πακέτων. Αρχικά, τα πακέτα κατηγοριοποιούνται σε μια από τις τέσσερις κατηγορίες προτεραιότητας, είτε από τον δρομολογητή υπηρεσίας αποστολής είτε από τον δρομολογητή εισόδου. Με βάση τη συμφωνηθείσα παρεχόμενη υπηρεσία, ο φορέας μπορεί να περιορίσει τον ρυθμό των υψηλότερης προτεραιότητας πακέτων. Στη συνέχεια, γίνεται μεταβίβαση των πακέτων κάθε κατηγορίας προτεραιότητας σε ένα σύστημα αστυνόμευσης κυκλοφορίας (π.χ. σε ένα κουβά κουπονιών), ώστε να προσδιοριστεί η κατηγορία απόρριψης για κάθε πακέτο. Με αυτόν τον τρόπο περνάει όλη η κυκλοφορία, και προσδιορίζονται τα πακέτα που εμπίπτουν σε μικρές ριπές με χαμηλή απόρριψη, τα πακέτα που υπερβαίνουν τις μικρές ριπές με μεσαία απόρριψη και τα πακέτα που υπερβαίνουν τις μεγάλες ριπές με υψηλή απόρριψη. Έπειτα, κωδικοποιείται σε κάθε πακέτο ο συνδυασμός προτεραιότητας με απόρριψη (Σχήμα 16).

DROP Precedence	Class #1	Class #2	Class #3	Class #4
Low Drop Precedence	(AF11) 001010	(AF21) 010010	(AF31) 011010	(AF41) 100010
Medium Drop Precedence	(AF12) 001100	(AF22) 010100	(AF32) 011100	(AF42) 100100
High Drop Precedence	(AF13) 001110	(AF23) 010110	(AF33) 011110	(AF43) 100110

Σχήμα 16: Κωδικοσημεία DS επιβεβαιωμένης προώθησης

Τέλος, στους δρομολογητές του δικτύου μέσω ενός χρονοπρογραμματιστή πακέτων, τα πακέτα διακρίνονται στις διαφορετικές κατηγορίες προτεραιότητας. Αυτό συνήθως γίνεται με την χρήση τεσσάρων σταθμισμένων δίκαιων ουρών με υψηλότερα βάρη στις υψηλότερες κατηγορίες, ώστε οι υψηλότερες κατηγορίες να λαμβάνουν το μεγαλύτερο ποσοστό εύρους ζώνης και οι χαμηλότερες να μην μένουν τελείως χωρίς εύρος ζώνης. Σε μια κατηγόρια προτεραιότητας, τα πακέτα με υψηλότερη πιθανότητα απόρριψης θα προτιμούνται για απόρριψη από κάποιον αλγόριθμο όπως ο RED.

3.2.5 Μειονεκτήματα[28]

Οι διαφοροποιημένες υπηρεσίες έχουν προσφέρει αρκετά πλεονεκτήματα αλλά έχουν και μειονεκτήματα. Μερικές από τις προκλήσεις που έχουν να αντιμετωπίσουν στο μέλλον και ευκαιρίες για βελτίωση είναι οι εξής:

- **Ποιότητα υπηρεσίας και δρομολόγηση:** Ένα από τα μεγαλύτερα μειονεκτήματα των διαφοροποιημένων υπηρεσιών είναι το γεγονός ότι η διαδικασία δρομολόγησης γίνεται χωριστά από τη διαδικασία σηματοδότησης/τροφοδότησης. Ίσως να υπάρχει μια διαδρομή στο δίκτυο που διαθέτει τους απαιτούμενους πόρους ακόμα κι όταν οι διαφοροποιημένες υπηρεσίες αδυνατούν. Αυτό συμβαίνει όταν το TE (TrafficEngineering) και το MPLS τίθενται σε λειτουργία. Προκειμένου να επιτευχθεί ποιότητα υπηρεσιών με την καλύτερη δυνατή αξιοποίηση δικτύου, απαιτείται συνδυασμός της παραδοσιακής Ποιότητας Υπηρεσίας και μεθόδων δρομολόγησης.

- **Απώλεια διακριτότητας:** Συνήθως οι επιβεβαιώσεις Ποιότητας Υπηρεσίας γίνονται στο επίπεδο κατηγορίας. Όμως, σε μερικές περιπτώσεις μπορεί να χρειαστεί να γίνουν σε κατώτερο επίπεδο ώστε να εξασφαλιστεί η απαραίτητη Ποιότητα Υπηρεσίας. Για παράδειγμα, ενώ η κυκλοφορία HTTP είναι στην χρυσή κατηγορία, με 100Mbps εύρος ζώνης, δεν υπάρχει μηχανισμός που να εγγυάται ότι αυτό το εύρος ζώνης δεν χρησιμοποιείται από μια μόνο ενιαία ροή.

- **Τροφοδοσία:** Οι διαφοροποιημένες υπηρεσίες χρειάζονται τροφοδοσία. Για τη σωστή ρύθμιση των διαφόρων κατηγοριών σε όλο το δίκτυο απαιτείται γνώση των εφαρμογών και των στατιστικών κυκλοφορίας για τα σύνολα κυκλοφορίας στο δίκτυο. Προκειμένου να ανακαλυφθούν τα στατιστικά και οι εφαρμογές απαιτείται πολύς χρόνος. Αυτές οι δραστηριότητες διευκολύνονται αν χρησιμοποιηθούν εργαλεία όπως εφαρμογή ανακάλυψης NBAR, αναλυτές πρωτοκόλλου και ανιχνευτές απομακρυσμένης παρακολούθησης (RMON).

- **Χρέωση και παρακολούθηση:** Η χρέωση και η παρακολούθηση εξακολουθούν να είναι δύσκολα στην πραγματοποίησή τους, παρά το γεγονός ότι οι μετρητές (bytes/sec, πακέτα/sec, κ.ά.) είναι διαθέσιμοι μέσω της βάσης διαχείρισης πληροφοριών (ManagementInformationBase – MIB).

3.3 Σύγκριση ολοκληρωμένων με διαφοροποιημένων υπηρεσιών[1][29]

- Για κάθε κατηγορία (guaranteed και controlled-load) των ολοκληρωμένων υπηρεσιών μπορούν να οριστούν ‘άπειρες’ υπηρεσίες, σε αντίθεση με τις διαφοροποιημένες υπηρεσίες που προσφέρουν ένα πεπερασμένο σύνολο υπηρεσιών.

- Οι διαφοροποιημένες υπηρεσίες δεν διαθέτουν πρωτόκολλο για την δέσμευση πόρων, σε αντίθεση με τις ολοκληρωμένες υπηρεσίες που χρησιμοποιούν το πρωτόκολλο RSVP.

- Η απουσία πρωτοκόλλου για τη δέσμευση πόρων στις διαφοροποιημένες υπηρεσίες, έχει ως συνέπεια την μη παροχή αυστηρών εγγυήσεων όσον αφορά την Ποιότητα Υπηρεσιών.

- Οι ολοκληρωμένες υπηρεσίες λόγω της χρήσης του RSVP είναι περισσότερο πολύπλοκες στην υλοποίηση από ότι οι διαφοροποιημένες υπηρεσίες.

- Στις ολοκληρωμένες υπηρεσίες κάθε ροή έχει τα δικά της χαρακτηριστικά, ενώ στις διαφοροποιημένες υπηρεσίες ο αριθμός ταξινομήσεων των ροών είναι περιορισμένος λόγω των συγκεκριμένων PHBs.

- Οι ολοκληρωμένες υπηρεσίες μπορούν να συνδυαστούν με ένα πρωτόκολλο σηματοδοσίας όπως το RSVP, ενώ οι διαφοροποιημένες υπηρεσίες έχουν μικρότερη ακρίβεια και απαιτείται από κοινού διαμόρφωση όλων των δρομολογητών που παίρνουν μέρος.

- Οι ολοκληρωμένες υπηρεσίες παρέχουν καλύτερο έλεγχο της ποιότητας πραγματικού χρόνου της παράδοσης της κυκλοφορίας σε σχέση με τις διαφοροποιημένες.

- Στις διαφοροποιημένες υπηρεσίες η πρόβλεψη συμπεριφοράς του δικτύου από άκρη σε άκρη είναι δύσκολη.

Κλείνοντας αυτό το κεφάλαιο αξίζει να σημειωθεί το γεγονός ότι οι διαφοροποιημένες υπηρεσίες μπορούν να υποστηρίξουν καλύτερα τις διαφορετικές ροές δεδομένων, οι οποίες με τη σειρά τους έχουν διαφορετικές QoS απαιτήσεις. Για αυτό το λόγο, οι ολοκληρωμένες υπηρεσίες δεν χρησιμοποιούνται ευρέως.

ΚΕΦΑΛΑΙΟ 4: Ποιότητα Υπηρεσιών Προσομοιωτών

4.1 Υλοποίηση DiffServ στον προσομοιωτή ns2[5]

Η αρχιτεκτονική DiffServ παρέχει ποιότητα υπηρεσιών διαχωρίζοντας την κυκλοφορία σε κατηγορίες, μαρκάροντας κάθε πακέτο με ένα κωδικοσημείο που υποδηλώνει την κατηγορία του και δρομολογώντας τα πακέτα ανάλογα. Υποστηρίζει τέσσερις κατηγορίες κυκλοφορίας και η καθεμιά έχει τρεις προτεραιότητες απόρριψης επιτρέποντας διαφορετική επεξεργασία της κυκλοφορίας εντός μιας κατηγορίας. Τα πακέτα μιας κατηγορίας κυκλοφορίας προσθέτονται σε μια αντίστοιχη φυσική RED ουρά, που περιλαμβάνει τρεις εικονικές ουρές (μία για κάθε προτεραιότητα απόρριψης).

Οι διάφοροι RED παράμετροι μπορούν να διαμορφωθούν για τις εικονικές ουρές, λόγω ότι τα πακέτα από μια εικονική ουρά απορρίπτονται συχνότερα από ότι τα πακέτα άλλων. Ένα πακέτο με μια χαμηλότερη προτεραιότητα απόρριψης επεξεργάζεται καλύτερα σε περιόδους συμφόρησης επειδή του ανατίθεται ένα κωδικοσημείο που αντιστοιχεί σε μια εικονική ουρά με σχετικά επιεικούς RED παραμέτρους.

Το μοντέλο DiffServ, εδώ, έχει τρία κύρια στοιχεία:

- **Policy:** Η πολιτική καθορίζεται από τον διαχειριστή του δικτύου σχετικά με το επίπεδο υπηρεσίας που μια κατηγορία κυκλοφορίας πρέπει να λάβει στο δίκτυο.
- **EdgeRouter:** Οι δρομολογητές ορίου μαρκάρουν τα πακέτα με ένα κωδικοσημείο σύμφωνα με την πολιτική που καθορίζεται.
- **CoreRouter:** Οι εσωτερικοί δρομολογητές εξετάζουν τα κωδικοσημεία των μαρκαρισμένων πακέτων και τα προωθούν ανάλογα.

Οι διαδικασίες και οι συναρτήσεις: dsred.{h,cc}, dsredq.{h,cc}, dsCore.{h,cc}, dsEdge.{h,cc}, dsPolicy.{h,cc}, που περιγράφονται παρακάτω βρίσκονται στον ιστότοπο: http://www.ns2.sourcearchive.com/documentation/2.35~RC4-1/files.html .

4.1.1 Η ουρά RED

Μια DiffServ ουρά της κλάσης dsREDQueue προέρχεται από την βασική κλάση Queue που παρέχει τη βασική λειτουργία του DiffServ δρομολογητή (δες dsred.{h.cc}) . Η dsREDQueue έχει τις εξής ικανότητες:

- Υλοποιεί πολλαπλές φυσικές ουρές RED κατά μήκος μιας σύνδεσης.
- Υλοποιεί πολλαπλές εικονικές ουρές εντός μιας φυσικής ουράς με ξεχωριστά σύνολα παραμέτρων για κάθε εικονική ουρά.

- Καθορίζει σε ποια φυσική και εικονική ουρά, προσθέτεται ένα πακέτο σύμφωνα με το κωδικοσημείο του.
- Καθορίζει από ποιά φυσική και εικονική ουρά, αφαιρείται ένα πακέτο σύμφωνα με το σχεδιάγραμμα δρομολόγησης που επιλέχθηκε.

Η κλάση dsREDQueue αποτελείται από τέσσερις φυσικές ουρές RED και καθεμιά περιέχει τρεις εικονικές ουρές. Ο αριθμός των φυσικών και εικονικών ουρών ορίζονται στα numPrec και numQueues_ . Κάθε συνδυασμός αριθμού φυσικής και εικονικής ουράς σχετίζεται με ένα κωδικοσημείο (ή μια προτεραιότητα απόρριψης) που προσδιορίζει ένα ορισμένο επίπεδο υπηρεσίας.

Η φυσική ουρά ορίζεται στην κλάση redQueue που επιτρέπει την διάκριση της κυκλοφορίας ορίζοντας εικονικές ουρές με ανεξάρτητη διαμόρφωση και παραμέτρους κατάστασης (δες dsredq.{h,cc}). Για παράδειγμα, το μήκος κάθε εικονικής ουράς υπολογίζεται μόνο στα πακέτα που αντιστοιχούν σε εκείνη την ουρά. Έτσι, οι αποφάσεις απόρριψης πακέτων μπορούν να εφαρμοστούν βασιζόμενοι στην κατάσταση και στους παραμέτρους διαμόρφωσης εκείνων των εικονικών ουρών. Η κλάση redQueue δεν είναι ισοδύναμη με την κλάση REDQueue. Αντιθέτως, είναι μια τροποποιημένη μορφή της υλοποίησης της RED με την έννοια των εικονικών ουρών και χρησιμοποιείται μόνο από την κλάση redQueue για να πραγματοποιεί φυσικές ουρές.

Η κλάση dsREDQueue περιέχει μια δομή δεδομένων γνωστή ως Πίνακας Συμπεριφοράς ανά Κόμβο (PHB Table). Οι δρομολογητές ορίου μαρκάρουν τα πακέτα με κωδικοσημεία και οι εσωτερικοί δρομολογητές απλά ανταποκρίνονται στα υπάρχοντα κωδικοσημεία. Και οι δυο χρησιμοποιούν πίνακα PHB για να αντιστοιχίσουν ένα κωδικοσημείο με μια συγκεκριμένη φυσική και εικονική ουρά. Ο πίνακας PHB ορίζεται ως μια παράταξη με τρία πεδία:

```
struct phbParam {
    int  codePt_; // αντίστοιχο κωδικοσημείο
    int  queue_;  // φυσική ουρά
    int  prec_;   // εικονική ουρά (προτεραιότητα απόρριψης)
};
```

4.1.2 Δρομολογητές ορίου και εσωτερικοί δρομολογητές

Αυτοί οι δρομολογητές στο μοντέλο DiffServ ορίζονται αντίστοιχα στις κλάσεις edgeQueue και coreQueue που προέρχονται από την κλάση dsREDQueue (δες dsEdge, dsCore.{h,cc}). Το μαρκάρισμα των πακέτων υλοποιείται στην κλάση edgeQueue. Ένα πακέτο μαρκάρεται με ένα κωδικοσημείο σύμφωνα με την πολιτική που καθορίζεται πριν μπει στην αντίστοιχη φυσική και εικονική ουρά. Η κλάση edgeQueue έχει μια αναφορά στην κλάση PolicyClassifier η οποία περιέχει πολιτικές για το μαρκάρισμα πακέτων. Το DiffServ προσπαθεί να περιορίσει την πολυπλοκότητα μόνο στους δρομολογητές ορίου.

4.1.3 Πολιτική

Η κλάση Policy και οι υπο-κλάσεις (δες dsPolisy.{h,cc}) ορίζουν τις πολιτικές που χρησιμοποιούνται από του δρομολογητές ορίου για να μαρκάρουν τα εισερχόμενα πακέτα. Μια πολιτική εγκαθίσταται ανάμεσα σε μια πηγή και στον κόμβο προορισμού. Όλες οι ροές που ταιριάζουν με αυτό το ζευγάρι προέλευσης-προορισμού αντιμετωπίζονται ως μια ενιαία συνολική κυκλοφορία. Η πολιτική για κάθε διαφορετικό σύνολο κυκλοφορίας έχει ένα σχετιζόμενο τύπο αστυνόμευσης, έναν τύπο μετρητή και ένα αρχικό κωδικοσημείο. Ο τύπος μετρητή καθορίζει τη μέθοδο για την μέτρηση των μεταβλητών κατάστασης που απαιτούνται από την αστυνόμευση. Για παράδειγμα, ο TSW Tagger είναι ένας μετρητής που μετρά το μέσο ποσοστό κυκλοφορίας χρησιμοποιώντας ένα καθορισμένο παράθυρο χρόνου.

Όταν ένα πακέτο φτάνει στον δρομολογητή ορίου, εξετάζεται για να προσδιοριστεί σε ποιο σύνολο ανήκει. Ο μετρητής καθορίζεται από την αντίστοιχη πολιτική που επικαλείται για να ενημερώσει όλες τις μεταβλητές κατάστασης. Η αστυνόμευση επικαλείται για να προσδιορίσει τον τρόπο που θα μαρκαριστεί το πακέτο ανάλογα με τις μεταβλητές κατάστασης του συνόλου: το καθορισμένο αρχικό κωδικοσημείο ή ένα υποβαθμισμένο κωδικοσημείο. Τότε το πακέτο προσθέτεται στην ανάλογη ουρά.

Αυτή τη στιγμή, έξι διαφορετικά μοντέλα πολιτικής ορίζονται:

1) Παράθυρο ολίσθησης στο χρόνο με 2 χρώματα σήμανσης (TSW2CMPolicer): χρησιμοποιεί ένα CIR και δύο προτεραιότητες απόρριψης. Η χαμηλότερη προτεραιότητα χρησιμοποιείται πιθανόν όταν το CIR υπερβαίνεται.
2) Παράθυρο ολίσθησης στο χρόνο με 3 χρώματα σήμανσης (TSW3CMPolicer): χρησιμοποιεί ένα CIR, ένα PIR και τρεις προτεραιότητες απόρριψης. Η μεσαία προτεραιότητα απόρριψης χρησιμοποιείται πιθανόν όταν το CIR υπερβαίνεται και η πιο χαμηλή προτεραιότητα απόρριψης χρησιμοποιείται πιθανόν όταν το PIR υπερβαίνεται.
3) Κουβάς κουπονιών (tokenBucketPolicer): χρησιμοποιεί ένα CIR, ένα CBS και δύο προτεραιότητες απόρριψης. Ένα πακέτο που φτάνει μαρκάρεται με την χαμηλότερη προτεραιότητα αν και μόνο αν είναι μεγαλύτερο από τον κουβά κουπονιών.
4) Μηχανισμός σήμανσης ενιαίου ρυθμού τριών χρωμάτων (srTCMPolicer): χρησιμοποιεί ένα CIR, CBS και ένα EBS για να επιλέξει από τις τρεις προτεραιότητες απόρριψης.
5) Μηχανισμός σήμανσης δύο ρυθμών τριών χρωμάτων(trTCMPolicer): χρησιμοποιεί ένα CIR, CBS, PIR και ένα PBS για να επιλέξει από τις τρεις πιθανότητες απόρριψης.
6) NullPolicer: δεν υποβιβάζει καθόλου πακέτα.

Αυτά τα έξι μοντέλα πολιτικών ορίζονται ως μια υπο-κλάση της dsPolicy. Ο συγκεκριμένος μετρητής και η συγκεκριμένη αστυνόμευση υλοποιούνται στις

συναρτήσεις applyMeter και applyPolicer, που ορίζονται ως εικονικές συναρτήσεις στην κλάση dsPolicy. Η καθορισμένη πολιτική του χρήστη μπορεί να προστεθεί με τον ίδιο τρόπο.

Όλες οι πολιτικές αποθηκεύονται στον πίνακα πολιτικής της κλάσης PolicyClassifier. Αυτός ο πίνακας είναι μια παράταξη που περιλαμβάνει πεδία για την πηγή και τους κόμβους προορισμού, έναν τύπο αστυνόμευσης, έναν τύπο μετρητή, ένα αρχικό κωδικοσημείο και ποικίλες πληροφορίες κατάστασης όπως φαίνεται παρακάτω:

- CIR (CommittedInformationRate): δεσμευμένος ρυθμός πληροφοριών (σε bps)
- PIR (PeakInformationRate): μέγιστος ρυθμός πληροφοριών (σε bps)
- CBS (CommittedBurstSize): δεσμευμένο μέγεθος καταιγισμού(σε bytes)
- EBS (ExcessBurstSize): οριακό μέγεθος καταιγισμού(σε bytes)
- PBS (PeakBurstSize): μέγιστο μέγεθος καταιγισμού(σε bytes)
- Cbucket: τρέχον μέγεθος δεσμευμένου κουβά
- Ebucket: τρέχον μέγεθος υπέρβασης κουβά
- Pbucket: τρέχον μέγεθος κορυφής κουβά
- Arrival time of last packet: χρόνος άφιξης του τελευταίου πακέτου
- Average sending rate: μέσος ρυθμός αποστολής
- TSW window length: μήκος παραθύρου TSW

4.2Υλοποίηση DiffServ στο πλαίσιο προσομοίωσης OMNeT++[6]

Η υλοποίηση στο OMNeT++ βασίζεται στα RFC 2474, RFC 2475, RFC 2597, RFC 2697, RFC 2698, RFC 3246 και RFC 3290.

4.2.1 Η αρχιτεκτονική των καρτών διεπαφής δικτύου

Οι μονάδες καρτών διεπαφής δικτύου (NetworkInterfaceCards – NICs), όπως η PPPInterface και η EthernetInterface, ίσως περιέχουν conditioners κυκλοφορίας στην είσοδο και στη έξοδο της διαδρομής των δεδομένων τους. Επίσης, ίσως περιέχουν ένα εξωτερικό στοιχείο ουράς. Αν η παράμετρος queueType έχει οριστεί, θα πρέπει να περιέχει ένα τύπο μονάδας εφαρμόζοντας τη διεπαφή μονάδας IOutputQueue. Αν δεν καθορίζεται, τότε τα PPP και EtherMAC χρησιμοποιούν μια εσωτερική ουρά απόρριψης πακέτων για να αποθηκεύουν τα πακέτα μέχρι η γραμμή να είναι κατειλημμένη.

4.2.1.1 Conditioners κυκλοφορίας

Οι conditioners κυκλοφορίας έχουν μια είσοδο και μια θύρα εξόδου όπως ορίζεται στη διεπαφή ITrafficConditiner. Μπορούν να μετατρέψουν την εισερχόμενη κυκλοφορία απορρίπτοντας ή καθυστερώντας τα πακέτα. Επίσης, μπορούν να θέσουν το κωδικοσημείο του πακέτου, ή να τα μαρκάρουν με άλλον τρόπο, για έναν διαφοροποιημένο χειρισμός στις ουρές. Οι conditioners κυκλοφορίας εκτελούν τις εξής ενέργειες:

- Ταξινομούν τα εισερχόμενα πακέτα σε κατηγορίες.
- Μετρούν την κυκλοφορία της κάθε κατηγορίας.
- Μαρκάρουν/ απορρίπτουν πακέτα ανάλογα με το αποτέλεσμα της μέτρησης.
- Διαμορφώνουν την κυκλοφορία καθυστερώντας τα πακέτα για να προσαρμοστεί στο επιθυμητό προφίλ κυκλοφορίας.

Ο INET παρέχει τις μονάδες: ταξινομητής (classifier) , μετρητής (meter) και μηχανισμός σήμανσης (marker), οι οποίες συνθέτονται για την κατασκευή ενός conditioner κυκλοφορίας ως μια σύνθετη μονάδα.

4.2.1.2 Ουρές εξόδου

Το στοιχείο ουράς έχει μια είσοδο και μια θύρα εξόδου. Αυτά τα στοιχεία πρέπει να υλοποιούν μια παθητική συμπεριφορά ουράς: να παραδίδουν μόνο ένα πακέτο, όταν τους ζητά ρητά η μονάδα που συνδέεται με την έξοδό τους. Από την άποψη της C++ αυτό σημαίνει, ότι η απλή μονάδα κατέχει την θύρα εξόδου, ή όποια συνδέεται με την θύρα εξόδου της σύνθετης μονάδας, πρέπει να υλοποιήσει την διεπαφή IPassiveQueue. Η επόμενη μονάδα ζητά ένα πακέτο καλώντας την μέθοδο requestPacket() αυτής της διεπαφής.

4.2.2 Απλές μονάδες

Σε αυτή την ενότητα περιγράφονται τα αρχικά στοιχεία από τα οποία μπορούν να κατασκευαστούν οι conditioners κυκλοφορίας και οι ουρές εξόδου. Αυτά τα στοιχεία είναι: η ουρά (queue), ο μηχανισμός απόρριψης (dropper), ο χρονοπρογραμματιστής (scheduler), ο ταξινομητής (classifier), ο μετρητής (meter) και ο μηχανισμός σήμανσης (marker).

4.2.2.1 Ουρές

Όταν τα πακέτα φτάνουν με μεγαλύτερο ρυθμό από ότι η διασύνδεση μπορεί να μεταβιβάσει τότε μπαίνουν σε μια ουρά αναμονής. Η ουρά αποθηκεύει πακέτα μέχρι να μπορούν να μεταβιβαστούν. Έχει μια είσοδο και μια θύρα εξόδου. Οι

ουρές ίσως έχουν ένα ή περισσότερα κατώτατα όρια που συνδέονται με αυτά. Τα ληφθέντα πακέτα μπαίνουν στην ουρά και αποθηκεύονται μέχρι η μονάδα που συνδέεται στην έξοδο της ουράς να ζητήσει ένα πακέτο καλώντας τη μέθοδο requestPacket(). Θα πρέπει να είναι ικανές να ενημερώσουν την συνδεδεμένη μονάδα στην έξοδό τους σχετικά με την άφιξη των νέων πακέτων.

FIFO

Η μονάδα FIFOQueue υλοποιεί μια παθητική ουρά FIFO με απεριόριστο χώρο αποθήκευσης. Μπορεί να συνδυαστεί με αλγοριθμικούς μηχανισμούς απόρριψης και χρονοπρογραμματιστές για να σχηματίσουν μια σύνθετη μονάδα IOutputQueue. Η κατηγορία C++ υλοποιεί τις διεπαφές IQueueAccess και IPassiveQueue.

DropTail

Η χωρητικότητα της μονάδας DropTailQueue μπορεί να καθοριστεί από την παράμετρο frameCapacity. Όταν ο αριθμός των αποθηκευμένων πακέτων φτάσει την χωρητικότητα της ουράς, τα επόμενα πακέτα απορρίπτονται. Αυτή η μονάδα έχει ενσωματωμένη την στρατηγική ρίψης οπότε δεν μπορεί να συνδυαστεί με τους αλγοριθμικούς μηχανισμούς απόρριψης όπως η FIFOQueue. Όμως η έξοδό της μπορεί να συνδεθεί με χρονοπρογραμματιστές. Αυτή η μονάδα υλοποιεί την διεπαφή IOutputQueue, έτσι μπορεί να χρησιμοποιηθεί ως το στοιχείο ουράς της κάρτας διεπαφής per se.

4.2.2.2 Μηχανισμοί απόρριψης

Οι αλγοριθμικοί μηχανισμοί απόρριψης επιλεκτικά απορρίπτουν τα ληφθέντα πακέτα βασιζόμενοι σε κάποια κατάσταση. Η κατάσταση μπορεί να είναι ντετερμινιστική (π.χ. περιορισμένο μήκος ουράς) ή πιθανολογική (π.χ. ουρές RED). Ένα είδος μηχανισμού απόρριψης είναι ο απόλυτος μηχανισμός απόρριψης ο οποίος απορρίπτει κάθε πακέτο που λαμβάνει. Χρησιμοποιείται για να αποτρέψει την περίσσεια κυκλοφορία, δηλαδή τα πακέτα που ο ρυθμός άφιξής τους υπερβαίνει τον επιτρεπόμενο. Η μονάδα Sink μπορεί να χρησιμοποιηθεί ως ένας απόλυτος μηχανισμός απόρριψης. Οι αλγοριθμικοί μηχανισμοί απόρριψης είναι οι ThresholdDropper και REDDropper. Αυτές οι μονάδες έχουν πολλαπλές εισόδους και πολλαπλές εξόδους και τα πακέτα που φτάνουν στην θύρα εισόδου[i] διαβιβάζονται στην θύρα εξόδου[i] (εκτός κι αν πέσουν). Οι ουρές συνδέονται στις θύρες εξόδου και θεωρούνται σαν σύνολο, δηλαδή η παράμετρος του μήκους ουράς του αλγόριθμου απόρριψης αποτελείται από το άθροισμα των επιμέρους μηκών ουράς. Έτσι, μιμούμαστε κοινόχρηστους buffers των ουρών. Να σημειώσουμε ότι μπορεί να συνδεθεί κάθε έξοδος στην ίδια μονάδα ουράς.

Threshold Dropper

Η μονάδα ThresholdDropper απορρίπτει επιλεκτικά πακέτα, βάσει του διαθέσιμου χώρου του buffer των ουρών που συνδέονται στην έξοδό του. Ο χώρος του buffer μπορεί να καθοριστεί ως ο αριθμός των πακέτων ή ως το μέγεθος σε bytes. Η μονάδα αθροίζει τα μήκη των εξόδων του buffer και αν ένα πακέτο, που προστίθεται στην ουρά, υπερβαίνει τις διαμορφωμένες ικανότητες τότε θα απορρίπτεται. Με την σύνδεση ενός ThresholdDropper στην είσοδο μιας FIFO ουράς, συνθέτεται μια drop tail ουρά. Ο κοινόχρηστος χώρος του buffer μπορεί να μοντελοποιηθεί με την σύνδεση περισσότερων ουρών FIFO στην έξοδο.

RED Dropper

Η μονάδα REDDropper υλοποιεί RED. Έχει *n* θύρες εισόδου και *n* θύρες εξόδου που καθορίζονται από την παράμετρο numGates. Τα πακέτα που φτάνουν στην *i* είσοδο προωθούνται στην *i* έξοδο, ή απορρίπτονται. Οι θύρες εξόδου πρέπει να συνδέονται σε απλές μονάδες υλοποιώντας την C++ διεπαφή IQueueAccess (π.χ. στη μονάδα FIFOQueue). Αυτή η μονάδα αθροίζει τον χρησιμοποιημένο χώρο του buffer των ουρών που συνδέονται στις θύρες εξόδου. Αν είναι κάτω από ένα ελάχιστο όριο, το πακέτο δεν θα απορριφθεί. Αν είναι πάνω από ένα μέγιστο όριο, το πακέτο θα απορριφθεί. Αν είναι μεταξύ του ελάχιστου και μέγιστου ορίου, το πακέτο θα απορριφθεί κατά μια δεδομένη πιθανότητα. Αυτή η δεδομένη πιθανότητα καθορίζεται από μια γραμμική συνάρτηση η οποία είναι 0 στο minth και maxp στο maxth. Οι παράμετροι minth, maxth και maxp μπορούν να καθοριστούν χωριστά για κάθε θύρα εισόδου. Έτσι, αυτή η μονάδα μπορεί να χρησιμοποιηθεί για την υλοποίηση διαφορετικών προτεραιοτήτων απόρριψης πακέτων.

4.2.2.3 Χρονοπρογραμματιστές

Οι χρονοπρογραμματιστές αποφασίζουν ποια ουρά μπορεί να στείλει ένα πακέτο, όταν το περιβάλλον είναι έτοιμο να μεταδώσει ένα. Έχουν πολλές θύρες εισόδου και μια θύρα εξόδου. Οι μονάδες που συνδέονται στις θύρες εισόδου ενός χρονοπρογραμματιστή πρέπει να υλοποιούν την C++ διεπαφή IPassiveQueue. Οι χρονοπρογραμματιστές υλοποιούν κι αυτοί IPassiveQueue, έτσι μπορούν να κλιμακωθούν με άλλους χρονοπρογραμματιστές και να χρησιμοποιηθούν ως μονάδα εξόδου του IOutputQueues. Υπάρχουν αρκετές πιθανές πειθαρχίες χρονοπρογραμματισμού (πρώτο έρχεται/πρώτο εξυπηρετείται (first come/first served), κατά προτεραιότητα (priority), δίκαιη με βάρη (weighted fair), round-robin με βάρη (weighted round-robin), βασισμένη σε προθεσμία (deadline-based), βασισμένη σε ποσοστό (rate-based)). Ο INET παρέχει τους χρονοπρογραμματιστές priority και weighted round-robin.

Priority Scheduler

Η μονάδα PriorityScheduler υλοποιεί έναν αυστηρό χρονοπρογραμματιστή κατά προτεραιότητα. Τα πακέτα που φτάνουν στην είσοδο in[0] έχουν την υψηλότερη προτεραιότητα, ακολουθούν τα πακέτα που φτάνουν στην είσοδο in[1], κ.ο.κ. .

Αν είναι διαθέσιμα περισσότερα πακέτα όταν ζητείται ένα, τότε επιλέγεται αυτό με την υψηλότερη προτεραιότητα. Τα πακέτα με χαμηλότερη προτεραιότητα μεταδίδονται μόνο όταν δεν υπάρχουν πακέτα στις εισόδους με μεγαλύτερες προτεραιότητες. Αυτή η μονάδα πρέπει να χρησιμοποιείται με προσοχή, καθώς αν υπάρχει μεγάλος όγκος από πακέτα υψηλής προτεραιότητας θα μεταδίδονται μόνο αυτά με συνέπεια να «λιμοκτονήσουν» τα πακέτα με χαμηλότερη προτεραιότητα. Για αυτό το λόγο είναι αναγκαίο να περιοριστεί το ποσοστό των πακέτων υψηλότερης προτεραιότητας σε ένα κλάσμα του ποσοστού δεδομένων της εξόδου. Ο μπορεί να χρησιμοποιηθεί για την υλοποίηση της εσπευσμένης προώθησης (EF PHB).

Weighted Round-Robin Scheduler

Η μονάδα WRRScheduler υλοποιεί έναν weighted round-robin χρονοπρογραμματιστή. Ο χρονοπρογραμματιστής επισκέπτεται με τη σειρά τις θύρες εισόδου και επιλέγει τον αριθμό των πακέτων με βάση το βάρος της κάθε εισόδου. Για παράδειγμα, έχουμε τρεις θύρες εισόδου in[0], in[1], in[2]. Στην in[0] φτάνουν Α πακέτα, στην in[1] φτάνουν Β πακέτα και στην in[2] φτάνουν C πακέτα. Τα βάρη των εισόδων είναι αντίστοιχα 3, 2, 1. Οπότε, τα πακέτα θα μεταδίδονται με την ακόλουθη σειρά: A A A B B C A A A B B C… . Αν δεν υπάρχουν πακέτα στην τρέχουσα θύρα όταν ζητείται ένα πακέτο, τότε η επόμενη που επιλέγεται έχει αρκετά κουπόνια. Αν το μέγεθος των πακέτων είναι ίσο, τότε ο WRRScheduler μοιράζει το διαθέσιμο εύρος ζώνης σύμφωνα με τα βάρη. Κάθε ροή λαμβάνει ένα ελάχιστο εγγυημένο εύρος ζώνης, το οποίο είναι εξασφαλισμένο ακόμη κι αν οι άλλες ροές υπερβαίνουν το μερίδιό τους (απομόνωση ροής). Επίσης, χρησιμοποιείται αποτελεσματικά το κανάλι, καθώς αν κάποια κυκλοφορία είναι μικρότερη από το λαμβανόμενο εύρος ζώνης, τότε αυτό που περισσεύει κατανέμεται στις υπόλοιπες ροές. Ο WRRScheduler μπορεί να χρησιμοποιηθεί για την υλοποίηση των AFxy PHBs.

4.2.2.4 Ταξινομητές

Οι μονάδες ταξινομητών έχουν μια θύρα εισόδου και πολλές θύρες εξόδου. Εξετάζουν τα πακέτα που έχουν ληφθεί και τα προωθούν στην κατάλληλη θύρα εξόδου με βάση το περιεχόμενο ενός τμήματος της επικεφαλίδας του πακέτου. Παρέχονται δύο ταξινομητές: ο MultiFieldClassifier που ταξινομεί τα πακέτα στους δρομολογητές ορίου της DiffServ περιοχής, και ο BehaviorAggregateClassifier που ταξινομεί τα πακέτα στους εσωτερικούς δρομολογητές.

Multi-field Classifier

Η μονάδα MultiFieldClassifier χρησιμοποιείται για την αναγνώριση μικροροών στην εισερχόμενη κυκλοφορία. Η ροή αναγνωρίζεται από την πηγή και τις

διευθύνσεις προορισμού, την ταυτότητα του πρωτοκόλλου και τις θύρες προορισμού του IP πακέτου. Ο ταξινομητής μπορεί να διαμορφωθεί με τον καθορισμό μιας λίστας φίλτρων. Κάθε φίλτρο μπορεί να καθορίσει μια μάσκα διεύθυνσης πηγής/προορισμού, ένα πρωτόκολλο, το εύρος θύρας πηγής/προορισμού και τα bits των πεδίων TypeOf Service/TrafficClass για να συνδυαστούν. Επίσης, καθορίζουν τον δείκτη της θύρας εξόδου συνδυάζοντας τα πακέτα που θα πρέπει να προωθηθούν. Το πρώτο φίλτρο που ταιριάζει προσδιορίζει την θύρα εξόδου. Στην περίπτωση που δεν υπάρχουν φίλτρα τα οποία να ταιριάζουν τότε επιλέγεται η defaultOut, η προκαθορισμένη θύρα. Η διαμόρφωση αυτής της μονάδας δίνεται σε ένα αρχείο XML, το οποίο πρέπει να περιέχει μια λίστα των στοιχείων <filter>. Μέσα σε αυτή τη λίστα υπάρχει ένα υποχρεωτικό χαρακτηριστικό, @gate, που είναι ο δείκτης της θύρας εξόδου των πακέτων που ταιριάζουν με το φίλτρο. Υπάρχουν κι άλλα χαρακτηριστικά που δεν είναι υποχρεωτικά και καθορίζουν την συνθήκη του ταιριάσματος:

- @srcAddress, @srcPrefixLength: για το ταίριασμα της IP διεύθυνσης της πηγής
- @destAddress, @destPrefixLength: για το ταίριασμα της IP διεύθυνσης του προορισμού
- @protocol: ταιριάζει το πεδίο πρωτοκόλλου του IP πακέτου. Η τιμή του πεδίου μπορεί να είναι ένα όνομα (π.χ. “udp”, “tcp”) ή ο αριθμητικός κωδικός του πρωτοκόλλου
- @tos, @tosMask: ταιριάζει τα bits του πεδίου TypeOfService/TrafficClass του IP πακέτου
- @srcPort: ταιριάζει την θύρα πηγής του TCP ή UDP πακέτου
- @srcPortMin, @srcPortMax: ταιριάζει μια σειρά από θύρες πηγής
- @destPort: ταιριάζει την θύρα προορισμού του TCP ή UDP πακέτου
- @destPortMin, @destPortMax: ταιριάζει μια σειρά από θύρες προορισμού

Behavior Aggregate Classifier

Η μονάδα BehaviorAggregateClassifier μπορεί να χρησιμοποιηθεί για να διαβάσει το πεδίο DSCP (DS κωδικοσημείο) από το IP διάγραμμα και να κατευθύνει τα πακέτα στη σωστή θύρα εξόδου. Η τιμή DSCP είναι τα χαμηλότερα 6 bits του πεδίου TypeOfService/TrafficClass. Οι εσωτερικοί δρομολογητές συνήθως χρησιμοποιούν αυτόν τον ταξινομητή για να καθοδηγήσει το πακέτο στην κατάλληλη ουρά. Οι τιμές DSCP απαριθμούνται στην παράμετρο dscp. Η πρώτη τιμή είναι για την θύρα εξόδου out[0], η δεύτερη για την out[1], κ.ο.κ. . Αν ένα λαμβανόμενο πακέτο έχει μια τιμή DSCP που δεν απαριθμείται, τότε θα προωθηθεί στην προκαθορισμένη θύρα defaultOut.

4.2.2.5 Μετρητές

Οι μετρητές ταξινομούν τα πακέτα με βάση τα προσωρινά χαρακτηριστικά της άφιξής τους. Ο ρυθμός άφιξης πακέτων συγκρίνεται με το επιτρεπόμενο προφίλ κυκλοφορίας, και αποφασίζεται αν τα πακέτα είναι πράσινα (in-profile) ή κόκκινα

(out-profile). Κάποιοι μετρητές εφαρμόζουν περισσότερα από δύο επίπεδα συμμόρφωσης, για παράδειγμα σε μετρητές τριών χρωμάτων τα μερικώς συμμορφούμενα πακέτα ταξινομούνται ως κίτρινα. Το επιτρεπόμενο προφίλ κυκλοφορίας συνήθως καθορίζεται από έναν κουβά κουπονιών. Σε αυτό το μοντέλο, ένας κουβάς γεμίζει με κουπόνια με έναν καθορισμένο ρυθμό, μέχρι να φτάσει την μέγιστη χωρητικότητα. Όταν φτάνει ένα πακέτο, ο κουβάς ελέγχεται. Αν περιέχει τουλάχιστον τόσα κουπόνια όσα και το μήκος του πακέτου, τότε τα κουπόνια μεταφέρονται και το πακέτο μαρκάρεται ως συμμορφωμένο με το προφίλ κυκλοφορίας. Στην περίπτωση που ο κουβάς περιέχει λιγότερα κουπόνια από αυτά που χρειάζονται, μένει αμετάβλητος, αλλά το πακέτο μαρκάρεται ως μη-συμμορφωμένο. Οι μετρητές έχουν δύο λειτουργίες: color-blind (αχρωματοψία) και color-aware (επίγνωση χρώματος). Στην πρώτη περίπτωση, το χρώμα που έχει δοθεί από έναν προηγούμενο μετρητή δεν επηρεάζει την ταξινόμηση των πακέτων στους επόμενους μετρητές. Στην δεύτερη περίπτωση, το χρώμα του πακέτου δεν μπορεί να αλλάξει σε ένα λιγότερο συμμορφωμένο χρώμα: αν ένα πακέτο ταξινομήθηκε ως μη-συμμορφωμένο από έναν μετρητή, τότε θα αντιμετωπιστεί ως μη-συμμορφωμένο και στους επόμενους μετρητές. Αξίζει να σημειώσουμε ότι ένας μετρητής λαμβάνει υπόψη μόνο το μήκος του IP πακέτου, χωρίς τις L2 επικεφαλίδες. Αν λάβει πακέτο που δεν είναι IP διάγραμμα ή δεν εμπεριέχει IP διάγραμμα, τότε παρουσιάζεται σφάλμα.

Token Bucket Meter

Η μονάδα TokenBucketMeter υλοποιεί έναν απλό μετρητή κουβά κουπονιών. Η μονάδα έχει δύο εξόδους, μία για τα πράσινα πακέτα και μία για τα κόκκινα. Όταν φτάνει ένα πακέτο, τα κουπόνια που αποκτήθηκαν προστίθενται στον κουβά και αφαιρούνται τόσα κουπόνια όσα και το μήκος του πακέτου. Τα πακέτα ταξινομούνται σύμφωνα με δύο παραμέτρους, Committed Information Rate (*cir*) και Committed Burst Size (*cbs*), για να είναι είτε πράσινα ή κόκκινα. Η πράσινη κυκλοφορία εγγυάται ότι θα είναι κάτω από *cir*(t1-t0)+8*cbs* σε κάθε διάστημα [*t0*,*t1*].

Single Rate Three Color Meter

Η μονάδα SingleRateThreeColorMeter υλοποιεί ένα μετρητή ενιαίου ρυθμού τριών χρωμάτων. Αυτή η μονάδα έχει τρεις θύρες εξόδου για πράσινα, κίτρινα και κόκκινα πακέτα. Τα πακέτα ταξινομούνται σύμφωνα με τρεις παραμέτρους, Committed Information Rate (*cir*), Committed Burst Size (*cbs*) και Excess Burst Size (*ebs*), για να είναι είτε πράσινα, κίτρινα ή κόκκινα. Η πράσινη κυκλοφορία εγγυάται ότι θα είναι κάτω από *cir*(t1-t0)+8*cbs* , ενώ η πράσινη και η κίτρινη κυκλοφορία μαζί θα είναι κάτω από *cir*(t1-t0)+8*(cbs+ebs)* σε κάθε διάστημα [*t0*,*t1*].

Two Rate Three Color Meter

Η μονάδα TwoRateThreeColorMeter υλοποιεί ένα μετρητή δύο ρυθμών τριών χρωμάτων. Η μονάδα έχει τρεις θύρες εξόδου για πράσινα, κίτρινα και κόκκινα πακέτα. Τα πακέτα ταξινομούνται βάσει δύο ρυθμών, Peak Information Rate (*pir*) και Committed Information Rate (*cir*), και των συνδεδεμένων μεγεθών τους *pbs* και *cbs*, γιθα να είναι είτε πράσινα, κίτρινα ή κόκκινα. Η πράσινη κυκλοφορία

είναι κάτω από *pir*(t1-t0)+8*pbs* και *cir*(t1-t0)+8*cbs*, η κίτρινη κυκλοφορία είναι κάτω από *pir*(t1-t0)+8*pbs* σε κάθε διάστημα [*t0,t1*].

4.2.2.6 Μηχανισμοί σήμανσης

Οι μηχανισμοί σήμανσης DSCP (Differentiated Services Code Point) θέτουν το κωδικοσημείο στα διερχόμενα πακέτα. Το κωδικοσημείο προσδιορίζει την περαιτέρω επεξεργασία του πακέτου στον δρομολογητή ή στο εσωτερικό της περιοχής DiffServ. Η μονάδα DSCPMarker θέτει το DSCP πεδίο (λιγότερα από 6 bit των TypeOfService/TrafficClass) των IP διαγραμμάτων στην τιμή που καθορίζεται από την παράμετρο dscps. Η παράμετρος dscps είναι ένας χώρος με διαχωρισμένη λίστα κωδικοσημείων. Μπορούμε να καθορίσουμε μια διαφορετική τιμή για κάθε θύρα εισόδου· τα ληφθέντα πακέτα στη $i^{η}$ θύρα εισόδου μαρκάρονται με την $i^{η}$ τιμή. Αν οι τιμές είναι λιγότερες από τις θύρες τότε η τελευταία χρησιμοποιείται για επιπλέον θύρες. Οι τιμές DSCP απαριθμούνται στον φάκελο DSCP.msg. Στην παράμετρο dscps μπορούμε να χρησιμοποιήσουμε ονόματα και ακέραιες τιμές μαζί.

4.2.3 Σύνθετες μονάδες

4.2.3.1 AFxyQueue

Η μονάδα AFxyQueue είναι ένα παράδειγμα ουράς που υλοποιεί μια κατηγορία της συμπεριφοράς επιβεβαιωμένης προώθησης (AF PHB). Τα πακέτα με την ίδια κατηγορία AFx, αλλά με διαφορετικές προτεραιότητες απόρριψης φτάνουν στις θύρες afx1In, afx2In και afx3In. Τα ληφθέντα πακέτα αποθηκεύονται στην ίδια ουρά. Πριν το πακέτο προστεθεί στην ουρά, ένας αλγόριθμος απόρριψης RED ίσως αποφασίζει επιλεκτικά την απόρριψή τους, βασιζόμενος στο μέσο μήκος ουράς και στις παραμέτρους RED της προτεραιότητας απόρριψης του πακέτου. Οι παράμετροι afxMinth, afxMaxth και afxMaxp πρέπει να έχουν τιμές που να διασφαλίζουν ότι τα πακέτα με χαμηλότερη προτεραιότητα απόρριψης απορρίπτονται με χαμηλότερη ή ίση πιθανότητα από τα πακέτα με υψηλότερες προτεραιότητες απόρριψης.

4.2.3.2 DiffservQueue

Η μονάδα DiffservQueue είναι ένα παράδειγμα ουράς που μπορεί να χρησιμοποιηθεί σε διεπαφές εσωτερικών DS κόμβων και DS κόμβων ορίου για να

υποστηρίξει τις συμπεριφορές εσπευσμένης προώθησης (EF PHB) και επιβεβαιωμένης προώθησης (AFxy PHB). Τα εισερχόμενα πακέτα πρώτα ταξινομούνται σύμφωνα με το DSCP πεδίο τους. Τα DSCP αντιμετωπίζονται ως BE (best-effort), εκτός από τα AFxy και EF. Τα EF πακέτα αποθηκεύονται σε μια ειδική ουρά και εξυπηρετούνται πρώτα όταν ζητείται ένα πακέτο. Επειδή μπορούν να κατέχουν πρώτα τις άλλες ουρές, το ποσοστό των EF πακέτων θα πρέπει να περιορίζεται σε ένα κλάσμα του εύρους ζώνης της σύνδεσης. Αυτό γίνεται με τη μέτρηση της EF κυκλοφορίας με ένα μετρητή κουβά κουπονιών και με την απόρριψη πακέτων που δεν συμμορφώνονται με το προφίλ κυκλοφορίας. Υπάρχουν και άλλες ουρές για AFx κατηγορίες και BE. Οι ουρές AFx χρησιμοποιούν RED για να υλοποιήσουν 3 διαφορετικές προτεραιότητες απόρριψης μέσα στην κατηγορία. Τα BE πακέτα αποθηκεύονται σε μια ουρά drop tail. Τα πακέτα από τις ουρές AFx και BE προγραμματίζονται από ένα WRR χρονοπρογραμματιστή, ο οποίος διασφαλίζει ότι το υπόλοιπο εύρος ζώνης κατανέμεται μεταξύ των κατηγοριών σύμφωνα με τα καθορισμένα βάρη.

4.3 Σύγκριση

- Στο πλαίσιο προσομοίωσης OMNeT++ η υλοποίηση DiffServ επιτυγχάνεται μέσα από ένα σύνολο μονάδων που χωρίζονται σε απλές και σύνθετες. Οι απλές μονάδες χρησιμοποιούνται για την κατασκευή των conditioners κυκλοφορίας και των ουρών εξόδου, ενώ οι σύνθετες αποτελούν παραδείγματα ουρών που μπορούν να χρησιμοποιηθούν από δρομολογητές ορίου και εσωτερικούς δρομολογητές. Στον προσομοιωτή ns2 τα κύρια στοιχεία του μοντέλου DiffServ είναι οι δρομολογητές ορίου, οι εσωτερικοί δρομολογητές και η πολιτική που χρησιμοποιούν οι δρομολογητές ορίου. Από τα παραπάνω συμπεραίνουμε ότι το OMNeT++ μας δίνει περισσότερα εργαλεία ώστε να διαμορφώσουμε το επιθυμητό DiffServ μοντέλο.

- Στον ns2 η πολιτική για κάθε διαφορετικό σύνολο κυκλοφορίας έχει και έναν μετρητή ο οποίος ‘συμμορφώνει’ την κυκλοφορία. Αυτή τη στιγμή υπάρχουν 6 διαφορετικά μοντέλα πολιτικής: παράθυρο ολίσθησης στο χρόνο με 2 χρώματα σήμανσης, παράθυρο ολίσθησης στο χρόνο με 3 χρώματα σήμανσης, κουβάς κουπονιών, μηχανισμός σήμανσης ενιαίου ρυθμού τριών χρωμάτων, μηχανισμός σήμανσης δύο ρυθμών τριών χρωμάτων και NullPolicer. Στο OMNeT++ ο μετρητής ενός conditioner κυκλοφορίας μπορεί να υλοποιεί έναν απλό κουβά κουπονιών, έναν μετρητή ενιαίου ρυθμού τριών χρωμάτων ή έναν μετρητή δύο ρυθμών τριών χρωμάτων.

- Στον ns2 μια DiffServ ουρά αποτελείται από τέσσερις φυσικές RED ουρές και καθεμιά περιέχει τρεις εικονικές ουρές. Κάθε συνδυασμός φυσικής και εικονικής ουράς σχετίζεται με ένα κωδικοσημείο ή μια προτεραιότητα απόρριψης. Όμως, στο OMNeT++ υπάρχουν οι σύνθετες μονάδες AFxyQueue και DiffservQueue που υλοποιούν αντίστοιχα μια κατηγορία της συμπεριφοράς επιβεβαιωμένης προώθησης και εσπευσμένη ή επιβεβαιωμένη προώθηση.

Βιβλιογραφία

[1]Βαρτζιώτης Φ. (2011), *Προχωρημένα Θέματα Προγραμματισμού Δικτύων* [πανεπιστημιακές σημειώσεις], Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου, Τμήμα Τεχνολογίας Πληροφορικής και τηλεπικοινωνιών, Άρτα

[2]*QoS Τι είναι και πως δουλεύει*…(2005). Ανακτήθηκε στης 11/6/2015 από τον ιστότοπο http://www.adslgr.com/forum/threads/33661-QoS-%CE%A4%CE%B9-%CE%B5%CE%AF%CE%BD%CE%B1%CE%B9-%CE%BA%CE%B1%CE%B9-%CF%80%CF%8E%CF%82-%CE%B4%CE%BF%CF%85%CE%BB%CE%B5%CF%8D%CE%B5%CE%B9

[3] Tanenbaum W. (2011), *Δίκτυα Υπολογιστών* (5η έκδοση), Κλειδάριθμος

[4]Νιανιάκας Χ.(2013), *Μελέτη και Υλοποίηση Scheduling Αλγορίθμων στον ns-3* [διπλωματική εργασία], Πανεπιστήμιο Πατρών, Τμήμα Μηχανικών Ηλεκτρονικών Υπολογιστών & Πληροφορικής, Πάτρα. Ανακτήθηκε στης 12/6/2015 από τον ιστότοπο http://ru6.cti.gr/ru6/system/files/bouras_site/ergasies/diplwmatikes/131_nianiakas_thesis_0.pdf?language=el

[5] *'The ns Manual'* (2011) . Ανακτήθηκε στης 23/4/2016 από τον ιστότοπο http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf

[6] *'INET Framework for OMNet++'* (2016). Ανακτήθηκε στης 25/4/2016 από τον ιστότοπο https://omnetpp.org/doc/inet/api-current/inet-manual-draft.pdf

[7] Μητρόπουλος Χ. (2004), *QoS σε IP δίκτυα*, Πανεπιστήμιο Μακεδονίας, ΠΜΣ Πληροφοριακά Συστήματα , Τεχνολογίες Τηλεπικοινωνιών και Δικτύων. Ανακτήθηκε στης 25/6/2015 από τον ιστότοπο http://conta.uom.gr/conta/ekpaideysh/metaptyxiaka/technologies_diktywn/ergasies/2004/Mitropoulos-QoS.pdf

[8] Πασκάλης Σ. (2004), *Εισαγωγή Αποδοτικών Μηχανισμών Υποστήριξης Ποιότητας Υπηρεσίας σε IP Δίκτυα Κινητών Εφαρμογών* [διδακτορική διατριβή], Εθνικό και Καποδιστριακό Πανεπιστήμιο Αθηνών, Σχολή Θετικών Επιστημών, Τμήμα Πληροφορικής και Τηλεπικοινωνιών, Αθήνα. Ανακτήθηκε στης 25/6/2015 από τον ιστότοπο http://cgi.di.uoa.gr/~paskalis/papers/thesis.pdf

[9] Γκιαούρης Χ. & Θεόδωρης Ι. (2006), *Υπηρεσίες Τιμολόγησης και Χρέωσης σε Ευρυζωνικά Δίκτυα* [πτυχιακή εργασία], Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου, Σχολή Διοίκησης και Οικονομίας, Τμήμα Τηλεπληροφορικής και διοίκησης, Άρτα. Ανακτήθηκε στης 25/6/2015 από τον ιστότοπο http://apothetirio.teiep.gr/xmlui/bitstream/handle/123456789/78/tlp_000345.pdf

[10] Ανδριοπούλου Φ. (2009), *Μελέτη Αρχιτεκτονικής Υπηρεσιών-QoS πάνω σε Τηλεπικοινωνιακά Δίκτυα Νέας Γενιάς (NGN) (με χρήση εξομοιωτή OPNET)* [διπλωματική εργασία], Πανεπιστήμιο Πατρών, Τμήμα Ηλεκτρολόγων Μηχανικών και Τεχνολογίας Υπολογιστών της Πολυτεχνικής Σχολής, Πάτρα. Ανακτήθηκε στης 5/4/2016 από τον ιστότοπο http://nemertes.lis.upatras.gr/jspui/bitstream/10889/3860/3/Nimertis_Andriopoulou(ele).pdf

[11] *Leaky Bucket & Token Bucket - Traffic shaping* (2014). Ανακτήθηκε στης 9/4/2016 από τον ιστότοπο www.slideshare.net/vimal25792/leaky-bucket-tocken-buckettraffic-shaping

[12] Στεργίου Ελ. (2012), *Ανάλυση και Προσομοίωση Δικτύων* [πανεπιστημιακές σημειώσεις], Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου, Τμήμα Τεχνολογίας Πληροφορικής και τηλεπικοινωνιών, Άρτα

[13] *Θεωρία Ουράς*. Ανακτήθηκε στης 5/10/2015 από τον ιστότοπο http://www.mie.uth.gr/ekp_yliko/QueueingTheory8.1-8.5.pdf

[14] Μπασδάρας Π. (2011), *Δίκτυα ουρών αναμονής και Προσομοίωση: εφαρμογή σε συστήματα εξυπηρέτησης* [διπλωματική εργασία], Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης, Διατμηματικό ΠΜΣ «Πληροφορική Και Διοίκηση» Τμημάτων Πληροφορικής Και Οικονομικών Επιστήμων, Θεσσαλονίκη. Ανακτήθηκε στης 5/10/2015 από τον ιστότοπο http://invenio.lib.auth.gr/record/126641/files/GRI-2011-6764.pdf

[15] Stallings W. (2012), *Επικοινωνίες Υπολογιστών και Δεδομένων* (8η έκδοση), Εκδόσεις Τζιόλα

[16] *Configuring FIFO queuing on Cisco routers* (2009). Ανακτήθηκε στης 4/2/2016 από τον ιστότοπο http://nil.uniza.sk/practical-cisco/qos/configuring-fifo-queuing-cisco-routers

[17] Ανακτήθηκε στης 8/10/2015 από τον ιστότοπο http://www.icbnet.ntua.gr/website/Mathimata/InternetTech/slides/7.ppt

[18] *Priority queue*. Ανακτήθηκε στης 29/1/2016 από τον ιστότοπο https://en.wikipedia.org/wiki/Priority_queue

[19] *A quick look at WRR* (2012). Ανακτήθηκε στης 4/2/2016 από τον ιστότοπο http://ccie-or-null.net/tag/weighted-round-robin

[20] *Queuing: FIFO, WFQ, CBWFQ, LLQ + Global Synchronisation* (2013). Ανακτήθηκε στης 1/3/2016 από τον ιστότοπο https://ccieme.wordpress.com/2013/05/26/queuing-fifo-wfq-cbwfq-llq-global-synchronisation/

[21] *Class-Based Weighted Fair Queueing*. Ανακτήθηκε στης 30/3/2016 από τον ιστότοπο www.cisco.com/en/US/docs/ios/12_0t/12_0t5/feature/guide/cbwfq.html

[22] *Low Latency Queueing*. Ανακτήθηκε στης 31/3/2016 από τον ιστότοπο www.cisco.com/c/en/us/td/docs/ios/12_0s/feature/guide/fsllq26.html#wp1019660

[23] Wallace K. (2004), *Cisco IP Telephone Flash Cards: Weighted Random Early Detection (WRED)*. Ανακτήθηκε στης 2/4/2016 από τον ιστότοπο www.ciscopress.com/articles/article.asp?p=352991&seqNum=8

[24] *Cisco IOS Quality of Service Solutions Configuration Guide, Release 12.2* (2014). Ανακτήθηκε στης 2/4/2016 από τον ιστότοπο www.cisco.com/c/en/us/td/docs/ios/12_2/qos/configuration/guide/fqos_c/qcfconav.html#wpxref11086

[25] *Integrated services*. Ανακτήθηκε στης 9/4/2016 από τον ιστότοπο https://en.wikipedia.org/wiki/Integrated_services

[26] *6 Τι είναι η Ποιότητα Υπηρεσίας (Quality of Service-QoS).* Ανακτήθηκε στης 10/4/2016 από τον ιστότοπο http://conta.uom.gr/conta/ekpaideysh/metaptyxiaka/technologies_diktywn/teaching_m/management/qos.htm

[27] Κονταράκη Μ. (2005), *Παροχή Ποιότητας Υπηρεσίας στο UMTS με τη χρήση Ενοποιημένων Υπηρεσιών (Quality of Service in UMTS – Integrated Services)* [διπλωματική εργασία], Οικονομικό Πανεπιστήμιο Αθηνών, ΠΜΣ στην Επιστήμη των Υπολογιστών, Αθήνα. Ανακτήθηκε στης 10/4/2016 από τον ιστότοπο http://www.mm.aueb.gr/master_theses/xylomenos/2005_kontaraki.pdf

[28] *DiffServ – The Scalable End-to-End QoS Model.* Ανακτήθηκε στης 16/4/2016 από τον ιστότοπο http://www.cisco.com/en/US/technologies/tk543/tk766/technologies_white_paper09186a00800a3e2f.html

[29] Τερζόγλου Απ. (2008), *Πρωτόκολλα Επικοινωνίας Δικτύων* [πανεπιστημιακές σημειώσεις] , Τεχνολογικό Εκπαιδευτικό Ίδρυμα Ηπείρου, Τμήμα Τεχνολογίας Πληροφορικής και Τηλεπικοινωνιών, Άρτα