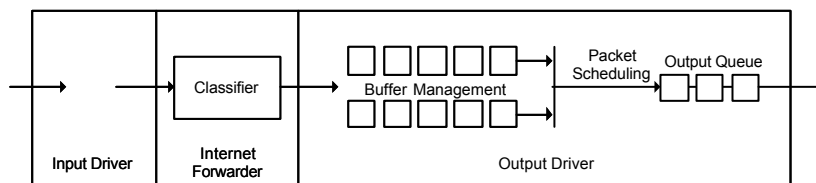


7 ΑΛΓΟΡΙΘΜΟΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ ΕΚΠΟΜΠΗΣ ΠΑΚΕΤΩΝ & ΔΙΑΧΕΙΡΙΣΗΣ ΕΝΤΑΜΙΕΥΤΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

7.1 Εισαγωγή

Σύμφωνα με την ορολογία της τεχνολογίας τηλεπικοινωνιακής κίνησης (teletraffic engineering), όταν μιλάμε για αναμονή (queuing) στους ενταμιευτές εννοούμε την προσωρινή αποθήκευση πακέτων πριν επακολουθήσει η επεξεργασία τους. Τυπικά αναμονή πακέτων μπορεί να έχουμε σε έναν δρομολογητή είτε κατά την είσοδο των πακέτων από τις γραμμές διασύνδεσης του δρομολογητή είτε κατά την εκπομπή των εισερχόμενων πακέτων σε άλλες γραμμές διασύνδεσης του αυτού δρομολογητή. Στην πρώτη περίπτωση μιλάμε για αναμονή στην ουρά εισόδου (input queue) και στη δεύτερη περίπτωση για αναμονή στην ουρά εξόδου του δρομολογητή (output queue). Στην περίπτωση που η επεξεργασία πακέτων σε IP επίπεδο δεν γίνεται συγκεντρωτικά, αλλά τμηματοποιημένα για κάθε γραμμή διασύνδεσης του δρομολογητή, ορίζεται μία ουρά εισόδου και μία ουρά εξόδου, οι οποίες είναι διαφορετικές μεταξύ τους με δικές τους μονάδες επεξεργασίας (CPUs) και ενταμιευτές (buffers). Αντίθετα όταν η επεξεργασία πακέτων σε επίπεδο IP γίνεται συγκεντρωτικά, υπάρχει μια μονάδα επεξεργασίας (CPU) και ένας ενταμιευτής (buffer) για όλα τα εισερχόμενα και τα εξερχόμενα πακέτα των γραμμών διασύνδεσης του δρομολογητή. Στη συνέχεια του κεφαλαίου θα θεωρήσουμε ότι σε IP επίπεδο η επεξεργασία των πακέτων γίνεται τμηματοποιημένα και ότι αναμονή των πακέτων παρατηρείται μόνο στην ουρά εξόδου, δεδομένου ότι κάτι τέτοιο είναι περισσότερο ρεαλιστικό λαμβάνοντας υπόψη τη σύγχρονη σχεδίαση των δρομολογητών.

Η αναμονή είναι η κυρίως συνιστώσα της αρχιτεκτονικής ενός δρομολογητή, όπου ένας αριθμός ασύγχρονων διαδικασιών συνδυάζεται προκειμένου να πραγματοποιηθεί η μεταγωγή των πακέτων διαμέσω των ουρών. Ο θεμελιώδης δρομολογητής αποτελείται από διαδικασίες εισόδου, οι οποίες συγκεντρώνουν τα εισερχόμενα πακέτα και ελέγχουν την ακεραιότητα της βασικής πλαισίωσης των πακέτων, από μία ή περισσότερες διαδικασίες προώθησης, οι οποίες καθορίζουν την προοριζόμενη γραμμή διασύνδεσης των πακέτων και τέλος από τις διαδικασίες εξόδου, οι οποίες πλαισιώνουν και εκπέμπουν τα πακέτα στον επόμενο κόμβο. Κάθε μία από τις παραπάνω διαδικασίες υλοποιείται σε ένα πακέτο κάθε φορά και ασύγχρονα με τις υπόλοιπες.



Σχήμα 7.1
Λειτουργική δομή του δρομολογητή Διαδικτύου

Οι μηχανισμοί του που φαίνονται στο Σχήμα 7.1 καθορίζουν τη διάθεση των πόρων του δικτύου (μήκος ενταμιευτή, εύρος ζώνης διασυνδέσεων) μεταξύ των υποστηριζόμενων κλάσεων εξυπηρέτησης. Συγκεκριμένα, με την ταξινόμηση (classification) των εισερχόμενων πακέτων καθορίζεται η διασύνδεση εξόδου και η κλάση εξυπηρέτησης στην οποία ανήκουν. Η πολιτική διαχείρισης ενταμιευτών (buffer management) είναι αυτή που καθορίζει το μέγεθος της ουράς για κάθε κλάση εξυπηρέτησης καθώς επίσης και το ποιά πακέτα θα απορριφθούν στη περίπτωση συμφόρησης, ενώ ο προγραμματιστής εκπομπής πακέτων (packet scheduler) καθορίζει τη σειρά εξυπηρέτησης των πακέτων.

Γίνεται κατανοητό ότι ο μέγιστος ρυθμός εξυπηρέτησης και η βέλτιστη χρήση του διαθέσιμου εύρους ζώνης επιτυγχάνεται όταν υπάρχει συνεργία των αλγορίθμων διαχείρισης ενταμιευτών και προγραμματισμού εκπομπής πακέτων. Αυτό συμβαίνει γιατί ο αλγόριθμος προγραμματισμού εκπομπής πακέτων από μόνος του δεν ελέγχει το συνολικό μήκος των ουρών του ενταμιευτή είτε το μήκος κάθε μεμονωμένης ουράς. Έτσι ενώ ο ρόλος του προγραμματιστή εκπομπής πακέτων είναι ιδιαίτερα καθοριστικός στη παροχή ποιότητας υπηρεσίας στο δίκτυο, δεν είναι αποτελεσματικός αν δεν υπάρχει χώρος στον ενταμιευτή για την αποθήκευση των νεο-εισερχόμενων πακέτων. Η διαχείριση ενταμιευτών είναι αυτή που μεριμνά για την κατανομή του διαθέσιμου χώρου του ενταμιευτή μεταξύ των κλάσεων εξυπηρέτησης και επηρεάζει αντίστοιχα, μέσω του προγραμματιστή εκπομπής πακέτων, την κατανομή του εύρους ζώνης της διασύνδεσης επικοινωνίας μεταξύ των κλάσεων.

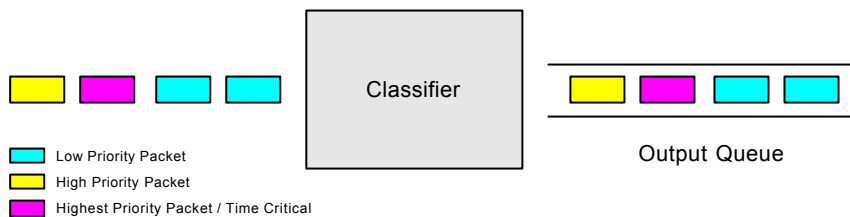
7.2 Αλγόριθμοι Προγραμματισμού Εκπομπής Πακέτων

Στη συνέχεια θα αναλύσουμε τέσσερις κατηγορίες αλγορίθμων προγραμματισμού εκπομπής πακέτων: First-In First-Out (FIFO), Priority Queuing (PQ), Weighted Fair Queuing (WFQ) και Class Based Queuing (CBQ).

7.2.1 Πρώτο-Μέσα Πρώτο-Έξω (First-In First-Out - FIFO)

Είναι ένας από τους απλούστερους αλγόριθμους προγραμματισμού εκπομπής πακέτων και όπως υποδηλώνει η ονομασία του εξυπηρετεί τα πακέτα με τη σειρά με τη οποία έρχονται (Σχήμα 7.2). Πρόκειται δηλαδή για ένα μηχανισμό που παρέχει εξοικονόμηση εργασίας (work conserving), είναι ανεξάρτητος από την αρχιτεκτονική του δικτύου και δεν λειτουργεί βάσει προτεραιότητας. Ένας αλγόριθμος εξοικονόμησης εργασίας στέλνει πακέτο σε κάθε ευκαιρία εκπομπής, εφόσον υπάρχει διαθέσιμο πακέτο, ενώ αντίθετα ένας αλγόριθμος μη εξοικονόμησης εργασίας (non-work

conserving) αποστέλλει μόνο εφόσον ικανοποιούνται κάποιες εσωτερικές συνθήκες, οι οποίες θέτουν τη ροή κίνησης σύμφωνη με προκαθορισμένο προφίλ κίνησης. Αν και ο αλγόριθμος FIFO δεν υποστηρίζει διαφοροποιημένες υπηρεσίες, είναι πολύ απλός στην υλοποίηση. Συγκεκριμένα, η εισαγωγή και η αφαίρεση από την ουρά πακέτων σε αναμονή πραγματοποιείται σε σταθερό χρόνο και δεν απαιτείται η διατήρηση επί μέρους στοιχείων για κάθε ροή (flow) κίνησης. Δικαιολογημένα λοιπόν, η πολιτική FIFO υιοθετείται από τους περισσότερους προμηθευτές των δρομολογητών πακέτων.



Σχήμα 7.2
Προγραμματιστής πακέτων FIFO

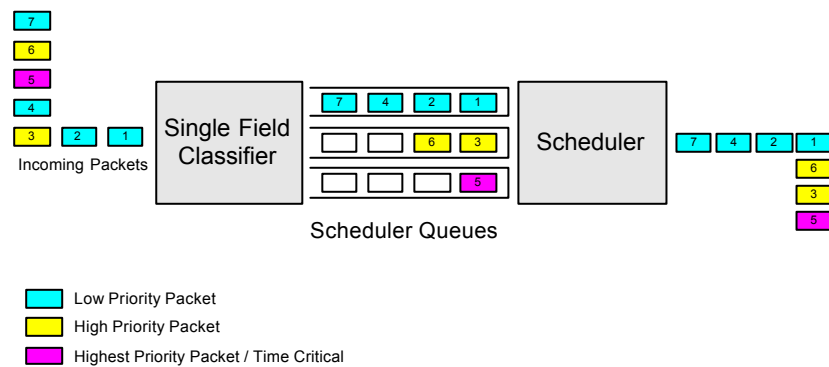
Η πολιτική FIFO, όπως είναι φανερό, δεν παρέχει εγγυήσεις ποιότητας υπηρεσίας αφού όλες οι ροές κίνησης αντιμετωπίζουν την ίδια πιθανότητα καθυστέρησης και απώλειας πακέτων. Ένας τρόπος εξασφάλισης ορίου καθυστέρησης (delay bound) είναι να περιοριστεί το μέγεθος της ουράς αποθήκευσης. Σε μία τέτοια περίπτωση, το πακέτο από τη στιγμή που θα μπει στην ουρά αναμονής σε χρόνο το πολύ ίσο με το χρόνο εξυπηρέτησης μία γεμάτης ουράς πακέτων, θα αποσταλεί στον προορισμό του. Εντούτοις, πακέτα που φθάνουν όταν η ουρά είναι γεμάτη πρέπει να καταστραφούν. Για να διασφαλιστεί ότι η πιθανότητα απώλειας πακέτων θα είναι κάτω από ένα ορισμένο όριο, μόνο ένας περιορισμένος αριθμός ροών κίνησης πρέπει να γίνεται αποδεκτός. Μία απλή τεχνική για τη λήψη της παραπάνω απόφασης είναι αυτή του αποτελεσματικού εύρους ζώνης (effective bandwidth) με την οποία υπολογίζεται το εύρος ζώνης που απαιτείται για κάθε ροή κίνησης.

Οι Clark, Shenker και Zhang έχουν αναπτύξει έναν μηχανισμό που αναφέρεται ως FIFO+, ο οποίος στοχεύει στον περιορισμό των καθυστερήσεων των πακέτων σε κάθε κόμβο. Η χαμηλή απόδοση του αλγορίθμου FIFO οφείλεται στο γεγονός ότι σε κάθε κόμβο το πακέτο μιας ροής κίνησης μπορεί να συναντήσει μία στιγμιαία έξαρση κίνησης άλλων ροών, με αποτέλεσμα μετά τη διέλευση του από μερικούς κόμβους να παρουσιάζει μεγάλη καθυστέρηση. Ο αλγόριθμος FIFO+ στοχεύει στον περιορισμό των καθυστερήσεων στους ενδιάμεσους κόμβους προκειμένου να περιοριστεί κατά το δυνατό η καθυστέρηση χειρίστης περίπτωσης (worst-case delay). Κάθε κόμβος ομαδοποιεί τις ροές κίνησης σε κλάσεις. Κάθε κλάση παρακολουθεί τη μέση τιμή καθυστέρησης για κάθε κόμβο. Για κάθε πακέτο, ο κόμβος υπολογίζει τη διαφορά της καθυστέρησης που

υφίσταται το πακέτο από τη μέση καθυστέρηση της κλάσης στην οποία ανήκει. Στη συνέχεια προσθέτει τη διαφορά σε κάποιο πεδίο (offset) της επικεφαλίδας του πακέτου. Μετά την έλευση του πακέτου από μερικούς κόμβους, το πεδίο αυτό θα καταγράφει τη διαφορά καθυστέρησης του πακέτου από τη μέση τιμή καθυστέρησης της κλάσης του. Κάθε κόμβος θα πρέπει να προγραμματίζει για εκπομπή τα πακέτα στις ουρές του θεωρώντας ότι ο χρόνος άφιξης τους είναι ο πραγματικός χρόνος άφιξης συν το χρόνο του πεδίου offset (ενδεχομένως να είναι αρνητικός). Κατά αυτό το τρόπο, πακέτα που έχουν καθυστέρηση μεγαλύτερη από τη μέση καθυστέρηση της κλάσεως τους για το συγκεκριμένο κόμβο θα μετακινούνται προς τη κορυφή της ουράς, ενώ πακέτα που έχουν καθυστέρηση μικρότερη από τη μέση καθυστέρηση της κλάσεως τους, μετακινούνται προς την βάση της ουράς.

7.2.2 Αναμονή με Προτεραιότητες (Priority Queuing - PQ)

Είναι η πρώτη από τις τροποποιήσεις του αλγορίθμου FIFO που αναπτύχθηκαν. Σε αυτή τη περίπτωση τα πακέτα ταξινομούνται σε επίπεδα προτεραιότητας και ο δρομολογητής διατηρεί μία ξεχωριστή ουρά για κάθε προτεραιότητα. Τα πακέτα της ουράς χαμηλής προτεραιότητας εξυπηρετούνται μόνο εφόσον οι ουρές υψηλότερης προτεραιότητας είναι κενές (Σχήμα 7.3). Κάθε ουρά εξυπηρετείται με μηχανισμό FIFO και ως εκ τούτου η πολιτική PQ είναι περίπου το ίδιο απλή στην υλοποίηση όπως η FIFO με την επιπρόσθετη δυσκολία της διατήρησης περισσότερων ουρών. Η επιλογή του επόμενου πακέτου για εκπομπή είναι διαδικασία σταθερού κόστους και εξαρτάται μόνο από τον αριθμό των επιπέδων προτεραιότητας και όχι από τον αριθμό των πολυπλεγμένων ροών κίνησης. Πρόκειται δηλαδή για ένα μηχανισμό που παρέχει εξοικονόμηση εργασίας (work conserving), είναι ανεξάρτητος από την αρχιτεκτονική του δικτύου και λειτουργεί βάσει προτεραιότητας.



Σχήμα 7.3
Προγραμματιστής πακέτων PQ

Αν και ο αλγόριθμος PQ προσφέρει ένα είδος διαφοροποιημένων υπηρεσιών, δεν εγγυάται ποιότητα υπηρεσίας σε κάθε μία από τις ροές κίνησης, ενώ εξασφαλίζει ότι μία κλάση θα έχει καλύτερη εξυπηρέτηση από μία άλλη. Όπως και στη περίπτωση του FIFO η εξασφάλιση εγγυήσεων όσον αφορά την από άκρη-σε-άκρη καθυστέρηση μπορεί να γίνει με τον περιορισμό του μεγέθους των ουρών. Εντούτοις, θα πρέπει να ληφθεί υπόψη ότι σε αυτή τη περίπτωση θα εξυπηρετηθεί ένα πακέτο χαμηλής προτεραιότητας όταν όλα τα πακέτα των ουρών μεγαλύτερης προτεραιότητας έχουν εξυπηρετηθεί. Συνεπώς, αναμένεται μεγάλη διακύμανση στη καθυστέρηση των πακέτων χαμηλής προτεραιότητας και παρουσιάζεται το ενδεχόμενο κορεσμού των πόρων του δικτύου για όλες τις ουρές προτεραιότητας εκτός από αυτή της υψηλής προτεραιότητας.

Εάν η ταχύτητα του μέσου πρόσβασης είναι ίση με C , το μέγεθος της ουράς υψηλότερης προτεραιότητας S_H , και το μέγιστο μέγεθος πακέτου B_{max} , η μέγιστη καθυστέρηση για ένα πακέτο υψηλής προτεραιότητας θα είναι ίση με $(S_H + B_{max})/C$. Αντίθετα, είναι δύσκολο να εκτιμηθεί η καθυστέρηση των πακέτων χαμηλής προτεραιότητας γιατί εξαρτάται από το φορτίο και τις μεταβολές κίνησης της κλάσης υψηλής προτεραιότητας. Αν το φορτίο υψηλής προτεραιότητας διατηρείται πάντα κάτω από μία συγκεκριμένη τιμή είναι δυνατή η εκτίμηση της μέγιστης καθυστέρησης των πακέτων χαμηλής προτεραιότητας. Δηλαδή αν θεωρήσουμε δύο επίπεδα προτεραιότητας και το φορτίο υψηλής προτεραιότητας να παραμένει λιγότερο από ρ_H για ένα διάστημα T , τότε η μέγιστη καθυστέρηση της κλάσεως χαμηλής προτεραιότητας $D_{L,max}$ θα δίνεται από

$$D_{L,max} < \rho_H * T + (S_H + S_L) / [(1 - \rho_H) * C]$$

όπου S_L είναι το μέγεθος της ουράς χαμηλής προτεραιότητας. Η προσέγγιση αυτή βασίζεται στην υπόθεση ότι κατά την άφιξη ενός πακέτου χαμηλής προτεραιότητας, η ουρά υψηλής προτεραιότητας είναι γεμάτη, και στην ουρά χαμηλής προτεραιότητας ο διαθέσιμος χώρος καλύπτεται πλήρως από το εισερχόμενο πακέτο.

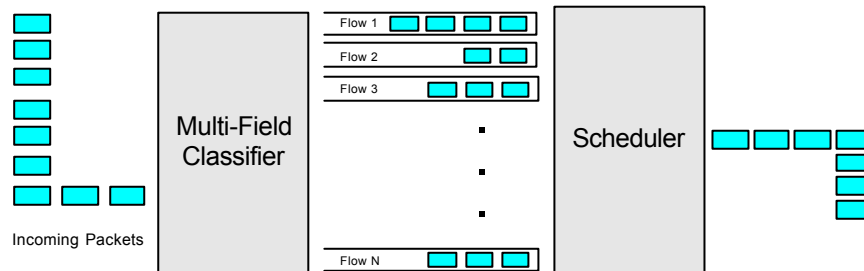
7.2.3 Δίκαιη Αναμονή με Βάρη (Weighted Fair Queuing - WFQ)

Η ιδέα της Δίκαιης Αναμονής (Fair Queuing) αναπτύχθηκε αρχικά από το Nagle και μετά αναθεωρήθηκε από τους Demers, Keshav και Shenker, προκειμένου να ξεπεραστούν τα προβλήματα των αλγορίθμων προγραμματισμού εκπομπής πακέτων FIFO, PQ και να εξασφαλιστούν εγγυήσεις ποιότητας υπηρεσίας για κάθε ροή κίνησης. Όπως θα φανεί στη συνέχεια ο WFQ όπως και ο αλγόριθμος PQ παρέχει εξοικονόμηση εργασίας (work conserving), είναι ανεξάρτητος από την αρχιτεκτονική του δικτύου και λειτουργεί βάσει προτεραιότητας ("βάρος").

Σύμφωνα με τον Nagle, οι δρομολογητές θα πρέπει να διατηρούν ουρά για κάθε πηγή κίνησης και να εξυπηρετούν τα πακέτα στις ουρές με κυκλική σειρά (round-robin). Αν n είναι ο αριθμός των ενεργών ουρών (οι άδειες ουρές παραλείπονται) κάθε πηγή κίνησης μπορεί να στέλνει ένα πακέτο κάθε n πακέτα. Η ιδέα αυτή του Nagle θέτει κάποιους

περιορισμούς. Πρώτον, αγνοεί το μέγεθος των πακέτων με αποτέλεσμα ροές πακέτων μεγάλου μεγέθους να απολαμβάνουν περισσότερο εύρος ζώνης από ότι οι ροές πακέτων μικρού μεγέθους. Δεύτερον, παρουσιάζει ευαισθησία στα μοντέλα άφιξης πακέτων. Ένα πακέτο που φτάνει σε μία άδεια ουρά αμέσως αφότου η ουρά αυτή εξετάστηκε, σύμφωνα με τη κυκλική σειρά εξυπηρέτησης, πρέπει να περιμένει να εξεταστούν η άλλες ουρές μέχρι να αποσταλεί .

Για να ξεπεραστούν οι παραπάνω περιορισμοί οι Demers, Keshav και Shenker προχώρησαν σε τροποποίηση της Δίκαιης Αναμονής. Η κεντρική ιδέα της τροποποίησης είναι η κυκλική εξυπηρέτηση bit-by-bit των ουρών. Η κυκλική εξυπηρέτηση bit-by-bit των ουρών μεταφράζεται ως η εξυπηρέτηση των ουρών σε κυκλική σειρά στέλνοντας ένα bit πληροφορίας από κάθε ουρά που έχει πακέτο. Σαφέστατα σε μία τέτοια περίπτωση η πολιτική Δίκαιης Αναμονής είναι πιο "δίκαιη" από ότι αρχικά είχε οριστεί από τον Nagle, αφού λαμβάνεται υπόψη το μέγεθος των πακέτων και επιπλέον ένα πακέτο θα περιμένει το πολύ $n-1$ χρόνους bit πριν αρχίσει να εκπέμπεται. Σύμφωνα με τους Demers, Keshav και Shenker η σειρά εκπομπής των πακέτων καθορίζεται με βάση το χρόνο εκπομπής του τελευταίου τους bit. Όλες οι πηγές κίνησης λαμβάνουν το ίδιο ποσοστό $1/n$ του διαθέσιμου εύρου ζώνης. Σε πολλές περιπτώσεις η ισόποση κατανομή του εύρου ζώνης δεν είναι επιθυμητή. Μία απλή παραλλαγή της Δίκαιης Αναμονής είναι να διαιρεθεί το εύρος ζώνης σε m -bit κύκλους, όπου m είναι μεγαλύτερο από τον αριθμό των ουρών (n), και στη συνέχεια τα επιπρόσθετα bits να διατίθενται στις πηγές που δικαιούνται παραπάνω εύρος ζώνης. Αυτή η παραλλαγή αποκαλείται Weighted Fair Queuing (WFQ) ή Packet Generalized Processor Sharing (PGPS) και δείχνεται στο Σχήμα 7.4.



Σχήμα 7.4
Προγραμματιστής πακέτων WFQ

Αναλυτικότερα, η βασική ιδέα του αλγορίθμου WFQ, όπως παρουσιάστηκε παραπάνω βασίζεται σε αυτή των προγραμματισμών πακέτων Generalized Processor Sharing (GPS) και έχει ως εξής: ένα "βάρος" φ_i συσχετίζεται με κάθε ροή κίνησης i , $i=1,\dots,N$, και το εύρος ζώνης του μέσου πρόσβασης μοιράζεται μεταξύ των ενεργών ροών

ανάλογα με τα "βάρη" τους. Με άλλα λόγια εάν C είναι η ταχύτητα του μέσου πρόσβασης, τότε κάθε ροή i εξασφαλίζει ένα ελάχιστο ρυθμό εξυπηρέτησης ίσο με $(\phi_i / \sum_{j=1}^N \phi_j) * C$, $i=1, \dots, N$. Σε κάποιες χρονικές

στιγμές είναι πολύ πιθανό κάποιες από τις ενεργές ροές να μην έχουν αποθηκευμένα πακέτα που να αναμένουν να αποσταλούν στο μέσο πρόσβασης. Αυτό μεταφράζεται σε διαθέσιμο εύρος ζώνης του μέσου πρόσβασης το οποίο είναι δυνατό να μοιραστεί μεταξύ των υπερφορτωμένων ενεργών ροών σε πλήρη αναλογία προς τα αντίστοιχα "βάρη" τους. Όσο μεγαλύτερο είναι το "βάρος" μίας ροής τόσο μεγαλύτερο είναι και το ελάχιστο δεσμευμένο εύρος ζώνης για τη συγκεκριμένη ροή κίνησης και τόσο μεγαλύτερη είναι και η προτεραιότητα εξυπηρέτησης της. Αν $B(t)$ συμβολίζει το σύνολο των υπερφορτωμένων ροών κίνησης τη χρονική στιγμή $t \geq 0$, τότε ο ελάχιστος ρυθμός εξυπηρέτησης $r_i(t)$ για τη i ροή θα δίνεται από:

$$r_i(t) = \begin{cases} \frac{\phi_i}{\sum_{j \in B(t)} \phi_j} * C, & i \in B(t) \\ 0, & \text{otherwise} \end{cases}$$

Η λειτουργία των προγραμματιστών πακέτων της κατηγορίας των GPS βασίζεται στη συνάρτηση εικονικού χρόνου (virtual time function). Η συνάρτηση εικονικού χρόνου αφορά μόνο τη περίοδο απασχόλησης του εξυπηρετητή (server), η οποία ορίζεται ως το μέγιστο χρονικό διάστημα που ο εξυπηρετητής είναι ενεργός. Για τους αλγόριθμους προγραμματισμού έκπομπής πακέτων Generalized Processor Sharing (GPS) ο πραγματικός χρόνος, $V(t)$, στην αρχή μιας περιόδου απασχόλησης είναι 0 ενώ κατά τη διάρκεια αυτής προσαυξάνεται με ρυθμό :

$$\frac{dV(t)}{dt} = \frac{C}{\sum_{j \in B(t)} \phi_j}$$

Λαμβάνοντας υπόψη τον εικονικό χρόνο, μπορούμε να ορίσουμε για κάθε πακέτο μιας ροής π.χ. το k πακέτο μιας ροής i , τον πραγματικό χρόνο έναρξης (virtual start time) s_i^k , και τον εικονικό χρόνο λήξης (virtual finish time), f_i^k , τα οποία να ανταποκρίνονται στους χρόνους έναρξης και λήξης, αντίστοιχα, σε ένα προγραμματιστή πακέτων GPS. Εάν συμβολίσουμε το χρόνο άφιξης του πακέτου με a_i^k , και το μέγεθος του με l_i^k , τότε προκύπτουν οι παρακάτω σχέσεις:

$$f_i^0 = 0,$$

$$s_i^k = \max\{V(a_i^k), f_i^{k-1}\}, \quad k=1, \dots,$$

$$f_i^k = s_i^k + \frac{l_i^k}{\phi_i}, \quad k=1, \dots,$$

Στη περίπτωση του WFQ (PGPS), το πακέτο που επιλέγεται για εκπομπή στο μέσο πρόσβασης είναι αυτό που έχει το μικρότερο εικονικό χρόνο λήξης (virtual finish time). Αυτό μπορεί να υλοποιηθεί χρησιμοποιώντας μια ουρά προτεραιότητας που θα βασίζεται στους εικονικούς χρόνους λήξης των πακέτων. Η πολυπλοκότητα των λειτουργιών αφαίρεσης και τοποθέτησης πακέτων από την ουρά προτεραιότητας είναι $O(\log(N))$. Αξιοσημείωτη είναι η επιπρόσθετη πολυπλοκότητα στον υπολογισμό του εικονικού χρόνου $V(t)$, αφού το $B(t)$ δεν παραμένει σταθερό κατά τη διάρκεια της περιόδου απασχόλησης του εξυπηρετητή. Στη χειρότερη περίπτωση η πολυπλοκότητα του εικονικού χρόνου μπορεί να είναι ίση με $O(N)$.

Οι αλγόριθμοι WFQ (PGPS) εξασφαλίζουν ανώτατο όριο στην τιμή της καθυστέρησης από άκρη-σε-άκρη για κάθε ροή κίνησης, το οποίο εξαρτάται από το "βάρος" αυτής. Συμβολίζουμε με C^h , $h=1, \dots, H$ την ταχύτητα των μέσων πρόσβασης που διασχίζει μία ροή κίνησης i και R_i τον ελάχιστο ρυθμό διέλευσης που διασφαλίζεται για τη ροή i διαμέσου των μέσων πρόσβασης. Προς χάριν απλοποίησης των υπολογισμών, υποθέτουμε ότι ο φάκελος κίνησης ροής i δίνεται από $A(t)=b_i+r_i*t$, $t \geq 0$. Στην περίπτωση που ικανοποιείται η συνθήκη $R_i \geq r_i$, τότε η καθυστέρηση από άκρη-σε-άκρη για κάθε ροή i θα δίνεται από:

$$\hat{D} = \frac{b_i}{R_i} + \frac{(H-1) * M_i}{R_i} + \sum_{h=1}^H \frac{L_{\max}^h}{C^h},$$

όπου με M_i συμβολίζουμε το μέγιστο μήκος πακέτου για τη ροή i , και με L_{\max}^h το μέγιστο μήκος πακέτου για το μέσο πρόσβασης h . Όπως προκύπτει από την παραπάνω σχέση ο όρος b_i/R_i εμφανίζεται μόνο μία φορά ανεξάρτητα από τον αριθμό των ενδιάμεσων κόμβων. Αυτό συμβαίνει διότι στην περίπτωση κυκλοφοριακής συμφόρησης σε ένα μέσο πρόσβασης, ο προγραμματιστής πακέτων θα επιχειρήσει την εξομάλυνση των πακέτων που αποστέλλονται με ριπές (burst) προκειμένου στη συνέχεια τα πακέτα να μην υφίστανται καθυστέρηση (b_i / R_i) κατά την κίνηση τους προς τον τελικό αποδέκτη. Επίσης όπως προκύπτει από την παραπάνω σχέση το όριο καθυστέρησης ροής είναι ανεξάρτητο από τον αριθμό των ροών που πολυπλέκονται σε κάθε ένα από τα μέσα πρόσβασης από τα οποία διαπερνά. Αυτή η ιδιότητα καθιστά τον μηχανισμό προγραμματισμού εκπομπής πακέτων WFQ (PGPS) ως τον πλέον ιδανικό αλγόριθμο για την παροχή εγγυήσεων ανελαστικού ορίου καθυστέρησης από άκρη-σε-άκρη.

7.2.3.1 Παραλλαγές Δίκαιης Αναμονής

Ένα από τα μεγαλύτερα μειονεκτήματα των αλγορίθμων Δίκαιης Αναμονής είναι η πολυπλοκότητα υπολογισμού της συνάρτησης εικονικού χρόνου (virtual time function). Στην κατεύθυνση αυτή έχουν προταθεί διάφορες παραλλαγές του WFQ και παρουσιάζονται συνοπτικά στη συνέχεια.

Αυτορυθμιζόμενη Δίκαιη Αναμονή (Self Clocked Fair Queuing - SCFQ): προσεγγίζει την τιμή της συνάρτησης εικονικού χρόνου με τον εικονικό χρόνο λήξης του πακέτου που εξυπηρετείται τη δεδομένη χρονική στιγμή. Η προσέγγιση αυτή μειώνει σημαντικά τις απαιτήσεις σε υπολογιστική ισχύ, αλλά συνεπάγεται την αύξηση του ορίου καθυστέρησης από άκρη-σε-άκρη σε σχέση με τον μηχανισμό WFQ (PGPS). Συνοπτικά, στο όριο καθυστέρησης μιας ροής προστίθεται για κάθε μέσο πρόσβασης ο όρος

$$\sum_{i=1}^N M_i / C^h .$$

Σε αντίθεση με τον αλγόριθμο WFQ, το όριο καθυστέρησης

από άκρη-σε-άκρη του SCFQ εξαρτάται από τον αριθμό των ροών που πολυπλέκονται στο μέσο πρόσβασης και αυτό αποτελεί το κύριο μειονέκτημα του.

Δίκαιη Αναμονή με Χρόνους Έναρξης (Start-time Fair Queuing - SFQ): παρουσιάζει ομοιότητες με τον SCFQ. Η βασική διαφορά του από τον SCFQ είναι ότι το πακέτο που επιλέγεται για εκπομπή στο μέσο πρόσβασης είναι αυτό που έχει το μικρότερο εικονικό χρόνο έναρξης και όχι το μικρότερο εικονικό χρόνο λήξης. Η καθυστέρηση από άκρη-σε-άκρη είναι ελαφρώς μικρότερη από ότι για το μηχανισμό SCFQ.

Όπως παρουσιάστηκε παραπάνω ο αλγόριθμος WFQ προσομοιώνει έναν υποθετικό εξυπηρετητή bit-by-bit κυκλικής σειράς με διαφορετικό "βάρος" για κάθε ροή κίνησης, όπου ο αριθμός των bit που εξυπηρετούνται από κάθε ροή σε κάθε γύρο να είναι ανάλογος του "βάρους" αυτής. Ενδέχεται λοιπόν σε κάποια περίπτωση να επιλεγεί πακέτο για εκπομπή στο μέσο πρόσβασης ενώ δεν έχει αρχίσει η εξυπηρέτηση του στον υποθετικό εξυπηρετητή bit-by-bit κυκλικής σειράς γιατί έχει το μικρότερο εικονικό χρόνο λήξης. Ο αλγόριθμος Χείριστη Περίπτωση Δίκαιου WFQ (Worst-case Fair Weighted Fair Queuing - WF²Q) χρησιμοποιεί τον εικονικό χρόνο έναρξης και τον εικονικό χρόνο λήξης για το προγραμματισμό των πακέτων. Πιο συγκεκριμένα, ο προγραμματιστής πακέτων τη χρονική στιγμή t επιλέγει για εκπομπή το πακέτο με το μικρότερο εικονικό χρόνο λήξης με την προϋπόθεση ότι ο εικονικός χρόνος έναρξης είναι μικρότερος από το χρόνο t . Το όριο καθυστέρησης από άκρη-σε-άκρη για τον μηχανισμό WF²Q είναι ίδιο με αυτό του WFQ.

Άλλες παραλλαγές της Δίκαιης Αναμονής είναι ο αλγόριθμος Ελλειμματική Κυκλική Σειρά (Deficit Round Robin – DRR), ο οποίος είναι μια επέκταση του Κυκλική Σειρά με Βάρη (Weighted Round Robin) με μεταβλητό μήκος πακέτων, και ο αλγόριθμος Δίκαιη Αναμονή Βάσει Πλαισίων (Frame-based Fair Queuing – FFQ), ο οποίος παρέχει τις ίδιες

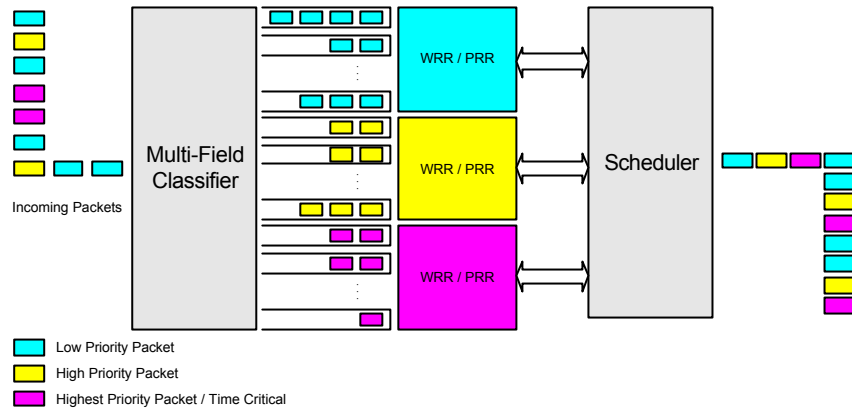
εγγυήσεις καθυστέρησης με τον WFQ αλλά είναι απλούστερος στην υλοποίηση.

Οι παραπάνω παραλλαγές της Δίκαιης Αναμονής λειτουργούν σε επίπεδο ροής και αυτό μειώνει την επεκτασιμότητα αυτών. Εντούτοις, η αρχή λειτουργίας του αλγορίθμου WFQ μπορεί να χρησιμοποιηθεί σε επίπεδο κλάσεων, δηλαδή σε σύνολο ροών με κοινά χαρακτηριστικά ποιότητας υπηρεσίας, προκειμένου να μειωθεί η πολυπλοκότητα των υπολογισμών. Σε μια τέτοια περίπτωση αναφερόμαστε στον αλγόριθμο WFQ Βάσει Κλάσεων εξυπηρέτησης (Class-Based WFQ), ο οποίος έγινε γνωστός από τη Cisco Systems, και σύμφωνα με αυτόν τα πακέτα ανατίθενται σε διαφορετικές ουρές ανάλογα με τη τιμή του πεδίου DS. Συνήθως τα πακέτα με μεγάλες απαιτήσεις σε ποιότητα υπηρεσίας έχουν μεγάλη τιμή DS πεδίου και αντιμετωπίζονται με μεγαλύτερη προτεραιότητα από τα πακέτα με χαμηλή τιμή DS πεδίου. Κάθε κλάση (QoS class) διατηρεί μια διαφορετική ουρά, η οποία έχει το δικό της "βάρος" και περιορισμένη δυνατότητα αποθήκευσης πακέτων (individual queue limit). Επίσης ορίζεται ένας περιορισμένος αριθμός αποθηκευμένων πακέτων για το σύνολο των ουρών (aggregate limit). Σε περίπτωση επικείμενης συμφόρησης (congestion) στο μέσο πρόσβασης, ο αλγόριθμος Class-Based WFQ δεσμεύει για κάθε κλάση εύρος ζώνης ανάλογο του "βάρους" αυτής, διαφορετικά οι κλάσεις μπορούν να χρησιμοποιήσουν ελεύθερα το διαθέσιμο εύρος ζώνης. Κάθε φορά που έρχεται πακέτο υπολογίζεται ο αριθμός των πακέτων σε κάθε ουρά καθώς και ο συνολικός αριθμός πακέτων για όλες τις ουρές. Όταν ο συνολικός αριθμός πακέτων για όλες τις ουρές είναι μικρότερος από το συνολικό όριο (aggregate limit), οι ουρές μπορούν να αποθηκεύουν περισσότερα πακέτα από αυτά που καθορίζει το προκαθορισμένο μέγεθος τους (individual queue limit). Αντίθετα, όταν ξεπεραστεί το συνολικό όριο πακέτων, οι ουρές απορρίπτουν τα νέο-εισερχόμενα πακέτα εφόσον με την είσοδο τους τα αποθηκευμένα πακέτα της ουράς ξεπερνούν το προκαθορισμένο μέγεθος της. Πακέτα που βρίσκονται ήδη στην ουρά δεν απορρίπτονται ακόμα και στη περίπτωση που ο συνολικός αριθμός των αποθηκευμένων πακέτων της ουράς είναι μεγαλύτερος από το προκαθορισμένο όριο της.

7.2.4 Αναμονή Βάσει Κλάσεως Εξυπηρέτησης (Class Based Queuing - CBQ)

Ο αλγόριθμος CBQ δεν παρέχει εξοικονόμηση εργασίας (non-work conserving), εξαρτάται από την αρχιτεκτονική του δικτύου και λειτουργεί βάσει προτεραιότητας (Σχήμα 7.5). Ο CBQ αποτελεί εξέλιξη των προηγούμενων αλγορίθμων (π.χ. PQ, WFQ) και στοχεύει στη διαχείριση υπηρεσιών με διαφορετικές απαιτήσεις σε ποιότητα υπηρεσίας. Πιο συγκεκριμένα, η απόδοση των διαθέσιμων πόρων του δικτύου υλοποιείται, σύμφωνα με το CBQ, δίνοντας διαφορετικές προτεραιότητες στις κλάσεις και ταυτόχρονα εξασφαλίζοντας ότι καμία κλάση δεν θα μονοπωλήσει τους διαθέσιμους πόρους. Επιπρόσθετα, ο αλγόριθμος CBQ σχεδιάστηκε

για να υποστηρίξει την καταμέριση του μέσου πρόσβασης (link sharing) μεταξύ κλάσεων δομημένων σε ιεραρχική δομή.



Σχήμα 7.5
Προγραμματιστής πακέτων CBQ

Για κάθε κλάση διατηρείται ξεχωριστή ουρά και ορίζονται οι παρακάτω παράμετροι: επιτρεπόμενος ρυθμός εύρους ζώνης (percent), προτεραιότητα εξυπηρέτησης (priority), μέσο μέγεθος πακέτου (avg_size) και δυνατότητα δανεισμού (borrow) με επιτρεπόμενες τιμές 0 ή 1 στη περίπτωση που η κλάση έχει ή δεν έχει, αντίστοιχα, την δυνατότητα δανεισμού αδαπάνητου εύρους ζώνης. Μία κλάση υπηρεσίας χαρακτηρίζεται ως υπό το όριο (underlimit) όταν δεν έχει χρησιμοποιήσει το αποδιδόμενο σε αυτή, σύμφωνα με τη παράμετρο percent, εύρος ζώνης και αντίστοιχα υπέρ το όριο (overlimit) όταν επιχειρεί να χρησιμοποιήσει παραπάνω εύρος ζώνης από το επιτρεπόμενο. Εκτός των παραπάνω παραμέτρων, για κάθε κλάση υπηρεσίας καθορίζεται ένας μέγιστος αριθμός πακέτων (maxburst) που μπορούν να σταλούν κατά ριπές σε περίπτωση που η κλάση μέχρι τότε ήταν υπό το όριο και ένας αριθμός πακέτων (minburst) που μία υπέρ το όριο κλάση πρέπει να περιμένει μέχρι να αποκτήσει ξανά το δικαίωμα εκπομπής στο μέσο πρόσβασης. Οι δύο τελευταίες μεταβλητές χρησιμοποιούνται, αντίστοιχα, για τον υπολογισμό του επιτρεπόμενου βαθμού "εκρηκτικότητας" (burstiness) μιας κλάσεως και του χρονικού διαστήματος που πρέπει να περιμένει μία υπέρ το όριο κλάση μέχρι να στείλει άλλο πακέτο.

Ο προγραμματισμός των πακέτων, σύμφωνα με τον CBQ, καθορίζεται από τη προτεραιότητα και την κατάσταση (underlimit, overlimit) των κλάσεων. Ο CBQ χρησιμοποιεί δύο τύπους προγραμματισμού εκπομπής πακέτων τον γενικό προγραμματιστή (general scheduler) και τον προγραμματιστή καταμερισμού εύρους ζώνης (link-sharing scheduler). Ο γενικός προγραμματιστής λειτουργεί βάσει προτεραιότητας και καθορίζει αποκλειστικά τη δρομολόγηση των πακέτων όταν δεν παρατηρείται συμφόρηση κίνησης. Αντίθετα, ο προγραμματιστής καταμερισμού εύρους

ζώνης ελέγχει την εκπομπή των πακέτων διαφορετικών κλάσεων σε κατάσταση συμφόρησης.

Η λειτουργία του CBQ περιγράφεται συνοπτικά στη συνέχεια. Με την άφιξη ενός πακέτου κλάσεως υπό το όριο, το πακέτο εξυπηρετείται από τον εξυπηρετητή και δρομολογείται στον προορισμό του μέσω εξωτερικής διασύνδεσης του δρομολογητή. Οι καθυστερήσεις οι οποίες υφίσταται σχετίζονται με την μέθοδο προώθησης του δρομολογητή και όχι με τον προγραμματιστή πακέτων. Διαφορετικά, στη περίπτωση πακέτου κλάσεως υπέρ το όριο το πακέτο τοποθετείται στην ουρά, εφόσον υπάρχει διαθέσιμο χώρος αποθήκευσης, και στη συνέχεια είτε περιορίζεται σύμφωνα με τη μεταβλητή *minburst* ή διαφορετικά δανείζεται από το διαθέσιμο εύρος ζώνης άλλων κλάσεων (*borrow=1*). Επιπλέον, μία ουρά υπό το όριο μπορεί να ξεπεράσει το επιτρεπόμενο εύρος ζώνης κατά *maxburst*, σε περιπτώσεις έκρηξης πακέτων (*bursts*).

Εάν η παράμετρος *borrow* είναι ίση με 1 για όλες τις κλάσεις, ο προγραμματιστής πακέτων CBQ έχει τα ίδια αποτελέσματα με τον PQ. Το ίδιο ισχύει και στην περίπτωση διασυνδέσεων μεγάλης ταχύτητας. Επιπλέον, ο αλγόριθμος CBQ είναι δυνατόν να λειτουργήσει με εξοικονόμηση εργασίας (*work conserving*). Ο τρόπος αυτός λειτουργίας ονομάζεται "αποτελεσματικός τρόπος" (*efficient mode*) και επιτρέπει, στην περίπτωση που όλες οι κλάσεις της ιεραρχικής δομής είναι υπέρ το όριο, την αποστολή πακέτου στο μέσο διασύνδεσης από υπό το όριο κλάση. Στην πράξη, η κλάση αυτή θα είναι η πρώτη *overlimit* της ιεραρχικής δομής που εξετάζεται.

Στο CBQ, οι κλάσεις της ίδιας προτεραιότητας εξυπηρετούνται σε κυκλική σειρά (*round-robin*). Η πολιτική εξυπηρέτησης των ουρών ίδιας προτεραιότητας μπορεί αν είναι είτε *packet-by-packet round-robin* (PRR) ή *weighted round-robin* (WRR). Για κάθε κλάση που προγραμματίζεται για εκπομπή σύμφωνα με τη κυκλική σειρά, το μέγεθος της πληροφορίας που αποστέλλει εξαρτάται από το μηχανισμό εξυπηρέτησης (PRR ή WRR) και από το χαρακτηρισμό της ως *overlimit* ή *underlimit*. Στη περίπτωση υπέρ το όριο κλάσεως, η εκπομπή της στο μέσο πρόσβασης περιορίζεται από το προσυμφωνημένο ποσοστό εύρος ζώνης και τις παραμέτρους ορισμού της. Με τη πολιτική εξυπηρέτησης PRR, οι κλάσεις της ίδιας προτεραιότητας εξυπηρετούνται σε κυκλική σειρά και κάθε κλάση που δεν είναι *overlimit* στέλνει ακριβώς ένα πακέτο. Ο αλγόριθμος εξυπηρέτησης WRR διαφέρει στο ότι ορίζονται βάρη ανάλογα με το προσυμφωνημένο εύρος ζώνης κάθε κλάσεως. Το βάρος καθορίζει τον αριθμό των bytes που στέλνει κάθε κλάση σε κάθε κύκλο εξυπηρέτησης. Εάν το μέγεθος ενός πακέτου που αποστέλλεται είναι μεγαλύτερο από το βάρος της κλάσεως του και η κλάση είναι *underlimit*, το πακέτο στέλνεται επιτρέποντας κατά κάποιο τρόπο στην κλάση να δανείζεται εύρος ζώνης από αυτό που της αντιστοιχεί στους επόμενους κύκλους εξυπηρέτησης.

7.3 Αλγόριθμοι Διαχείρισης Ενταμιευτών

Οι αλγόριθμοι διαχείρισης ενταμιευτών χωρίζονται σε δύο κατηγορίες ανάλογα με τον σκοπό τον οποίο υπηρετούν. Στους αλγορίθμους που αφορούν την αποφυγή της συμφόρησης (congestion avoidance) και σε αυτούς που αφορούν την αντιμετώπιση της συμφόρησης (congestion control). Επίσης, οι ροές κίνησης στο Διαδίκτυο μπορούν να χωριστούν στις TCP-συμβατές ροές δηλ. ροές που προσαρμόζουν το ρυθμό έκπομπής πακέτων σε περίπτωση συμφόρησης και στις μη TCP-συμβατές ροές που δεν ανήκουν στην παραπάνω κατηγορία. Ως γνωστό, η μεγάλη διάδοση και επιτυχία του Διαδικτύου οφείλεται στο μηχανισμό του TCP. Επειδή το TCP μειώνει το ρυθμό έκπομπής πακέτων σε κατάσταση συμφόρησης, ένας μεγάλος αριθμός από TCP συνδέσεις μπορεί να μοιραστεί το εύρος ζώνης μίας συμφορημένης διασύνδεσης. Αντίθετα, για τις μη TCP-συμβατές ροές, κυρίως εφαρμογές UDP, δεν υπάρχουν μηχανισμοί αποφυγής συμφόρησης ή όσοι υπάρχουν είναι ανεπαρκείς.

Πιο συγκεκριμένα, οι αλγόριθμοι αποφυγής συμφόρησης (congestion avoidance) αφορούν τις TCP-συμβατές ροές και υλοποιούνται με την απόρριψη πακέτων πριν επέλθει η πλήρωση των ουρών, έτσι ώστε οι πηγές πακέτων να ανταποκριθούν στη συμφόρηση χωρίς να έχει προηγηθεί η υπερχειλίση των αντίστοιχων ουρών. Η προσέγγιση αυτή καλείται ενεργή διαχείριση ουρών (active queue management). Με την υιοθέτηση της ανωτέρω προσέγγισης, οι δρομολογητές θα έχουν την δυνατότητα να ελέγχουν πότε και πόσα πακέτα θα χαθούν στη συνέχεια.

Οι αλγόριθμοι αντιμετώπισης συμφόρησης (congestion control) λαμβάνουν χώρα όταν επέλθει η συμφόρηση (συμπλήρωση ενταμιευτή με πακέτα) και αποσκοπούν στην εξασφάλιση ενός όσο το δυνατό μικρότερου ρυθμού απόρριψης πακέτων. Οι αλγόριθμοι αυτοί είναι ιδιαίτερα σημαντικοί για τις μη TCP-συμβατές ροές για τις οποίες, όπως έχουμε ήδη αναφέρει, δεν υπάρχει δυνατότητα προσαρμογής σε περίπτωση συμφόρησης. Στη σπουδαιότητα των αλγορίθμων αντιμετώπισης συμφόρησης συντείνει η μεγάλη έξαρση των εφαρμογών πολυμέσων με υψηλές απαιτήσεις σε εύρος ζώνης και μικρές ανοχές στις χρονικές καθυστερήσεις.

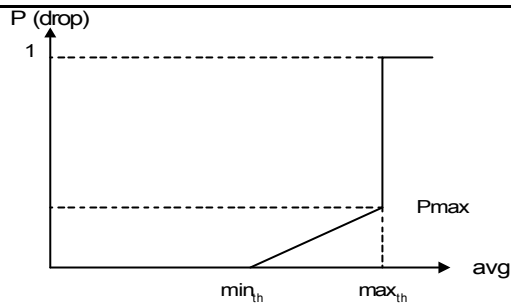
7.3.1 Αλγόριθμοι Διαχείρισης Ενταμιευτών για Αποφυγή Συμφόρησης

Ο πρωταρχικός σκοπός των αλγορίθμων αποφυγής συμφόρησης σε ένα δρομολογητή είναι η ανίχνευση επερχόμενης συμφόρησης. Ένας αλγόριθμος αποφυγής συμφόρησης πρέπει να διατηρεί το δίκτυο σε κατάσταση χαμηλών καθυστερήσεων και υψηλών ρυθμών εξυπηρέτησης. Συνεπώς, το μέσο μήκος των ουρών θα πρέπει να διατηρείται μικρό, αλλά θα πρέπει να επιτρέπονται οι διακυμάνσεις στο πραγματικό μήκος των ουρών λόγω της εκρηκτικότητας της κίνησης και των παροδικών συμφορήσεων.

Αρκετοί αλγόριθμοι έχουν προταθεί κατά καιρούς για την αποφυγή συμφόρησης στο Διαδίκτυο όπως ο DECbit, ο αλγόριθμος Έγκαιρης Τυχαίας Απόρριψης (Early Random Drop – ERD) και ο αλγόριθμος Τυχαίας Έγκαιρης Ανίχνευσης (Random Early Detection – RED), ο οποίος θεωρείται ως ο επικρατέστερος. Ο DECbit χρησιμοποιεί ένα bit ως ένδειξη συμφόρησης (congestion indication bit) στην επικεφαλίδα των πακέτων για να παρέχει πληροφορίες σχετικά με τη συμφόρηση στο δίκτυο. Για τους δρομολογητές που χρησιμοποιούν το αλγόριθμο DECbit, το επίπεδο μεταφοράς (transport layer) θα πρέπει να υπολογίζει το ποσοστό των πακέτων που έχουν το bit συμφόρησης ίσο με 1. Ο εν λόγω αλγόριθμος δεν ευνοεί τις ροές που παρουσιάζουν εκρηκτικότητα. Με τον ERD εάν το μήκος της ουράς ξεπερνάει ένα προκαθορισμένο επίπεδο (drop level), τότε ο δρομολογητής απορρίπτει κάθε εισερχόμενο πακέτο με σταθερή πιθανότητα (drop probability). Τα αποτελέσματα που δίνονται σε αποδεικνύουν ότι ο αλγόριθμος ERD δεν επιτυγχάνει τον έλεγχο των ροών με κακή συμπεριφορά. Τέλος, οι δρομολογητές που υιοθετούν τον RED (ακολουθεί αναλυτική παρουσίαση αυτού), διατηρούν το μέσο μήκος της ουράς μικρό και ταυτόχρονα έχουν τη δυνατότητα να εξυπηρετήσουν περιστασιακές αποστολές πακέτων με ριπές. Όταν το μέσο μήκος ουράς έχει ξεπεράσει προκαθορισμένο όριο, οι RED δρομολογητές απορρίπτουν τυχαία πακέτα έτσι ώστε οι ροές TCP να μειώνουν το ρυθμό εκπομπής σε διαφορετικές χρονικές στιγμές. Με αυτό το τρόπο αποφεύγεται ο συγχρονισμένος περιορισμός ρυθμών εκπομπής των ροών και ο ρυθμός εξυπηρέτησης διατηρείται υψηλός. Αντίθετα με τον DECbit, ο RED δεν ζημιώνει τις ροές που παρουσιάζουν εκρηκτικότητα και δεν έχει απαιτήσεις από το επίπεδο μεταφοράς. Επιπλέον, ενώ στον ERD υπάρχει αδυναμία ελέγχου των misbehaving ροών, στον RED η πιθανότητα μείωσης του ρυθμού εκπομπής πακέτων μίας σύνδεσης καθορίζεται από το διατιθέμενο για τη σύνδεση ποσοστό εύρου ζώνης.

7.3.1.1 Τυχαία Έγκαιρη Ανίχνευση (Random Early Detection - RED)

Ένας δρομολογητής RED λειτουργεί όπως περιγράφεται παρακάτω. Υπολογίζει το μέσο μήκος ουράς και όταν αυτό ξεπεράσει ένα προκαθορισμένο όριο, απορρίπτει κάθε εισερχόμενο πακέτο με πιθανότητα που μεταβάλλεται συναρτήσει του μέσου μήκους ουράς. Το μέσο μήκος ουράς υπολογίζεται χρησιμοποιώντας ένα κατωδιαβατό φίλτρο με εκθετική μεταβολή (Exponentially Weighted Moving Average-EWMA) έτσι ώστε να επιτρέπει πρόσκαιρες ριπές πακέτων στον δρομολογητή. Στη περίπτωση συνεχούς συμφόρησης στο δρομολογητή, το μέσο μήκος ουράς θα είναι μεγάλο καθώς και η πιθανότητα απόρριψης. Η μεγάλη πιθανότητα απόρριψης θα επιτρέψει την ανίχνευση καθώς και τον έλεγχο της συμφόρησης.



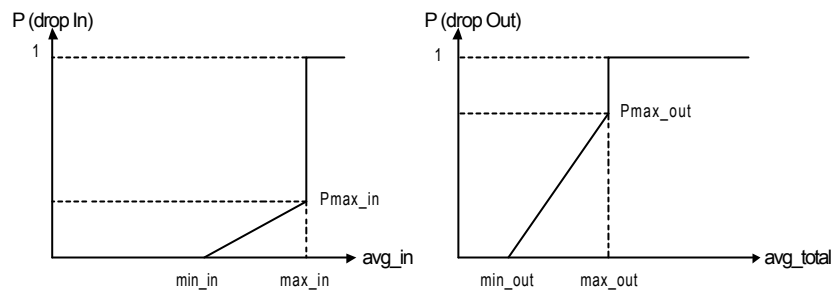
Σχήμα 7.6
Αλγόριθμος RED

Οι παράμετροι του αλγορίθμου RED είναι οι εξής: \min_{th} , \max_{th} και P_{max} . Στο Σχήμα 7.6 ο άξονας x είναι το μέσο μήκος ουράς το οποίο υπολογίζεται με την άφιξη κάθε πακέτου όπως αναφέρθηκε παραπάνω από το στιγμιαίο μήκος ουράς χρησιμοποιώντας το φίλτρο EWMA. Ο άξονας y συμβολίζει τη πιθανότητα απόρριψης εισερχόμενου πακέτου. Στον RED διακρίνουμε τρεις καταστάσεις ανάλογα με τη τιμή του μέσου μήκους ουράς δηλ. τα διαστήματα $[0, \min_{th})$, $[\min_{th}, \max_{th})$, και $[\max_{th}, \infty)$ τις οποίες ονομάζουμε αντίστοιχα ομαλή λειτουργία, αποφυγή συμφόρησης και έλεγχος συμφόρησης, αντίστοιχα. Κατά τη διάρκεια της ομαλής λειτουργίας, όταν το μέσο μήκος ουράς είναι χαμηλότερο από \min_{th} , ο δρομολογητής δεν απορρίπτει κανένα πακέτο. Όταν το μέσο μήκος ουράς είναι μεταξύ των δύο ορίων, ο δρομολογητής λειτουργεί σε κατάσταση αποφυγής συμφόρησης και με κάθε απόρριψη πακέτου ειδοποιείται το επίπεδο μεταφοράς να μειώσει το ρυθμό έκπομπής πακέτων. Για αυτό το λόγο η πιθανότητα απόρριψης είναι συνήθως μικρή με μέγιστη τιμή τη P_{max} . Όταν το μέσο μήκος ουράς είναι μεγαλύτερο από \max_{th} , ο δρομολογητής απορρίπτει κάθε εισερχόμενο πακέτο προκειμένου να διατηρήσει μικρό το μήκος της ουράς.

Μία ευρέως χρησιμοποιούμενη παραλλαγή του RED είναι ο αλγόριθμος RIO. Ο τελευταίος, όπως και ο RED, εφαρμόζεται ανά ουρά και επιπλέον επιτρέπει τη διάκριση μεταξύ πακέτων διαφορετικών προτεραιοτήτων. Στη συνέχεια θα παρουσιάσουμε τον μηχανισμό RIO.

7.3.1.2 RED In Out (RIO)

Η ονομασία RIO σημαίνει RED με In και Out bit. Ο RIO λειτουργεί όπως ο RED με δύο σει παραμέτρων, ένα για τα in πακέτα και ένα για τα out πακέτα. Με κάθε άφιξη πακέτου ο δρομολογητής ελέγχει κατά πόσο τα πακέτα είναι μαρκαρισμένα ως in ή out. Εάν είναι in πακέτο, ο δρομολογητής υπολογίζει το μέσο μήκος in πακέτων στην ουρά (avg_{in}). Εάν είναι out πακέτο, ο δρομολογητής υπολογίζει το μέσο μήκος πακέτων (in και out) στην ουρά (avg_{total}). Η πιθανότητα απόρριψης in πακέτου είναι ανάλογη του avg_{in} , ενώ η πιθανότητα απόρριψης out πακέτου εξαρτάται από avg_{total} .



Σχήμα 7.7
Αλγόριθμος RIO

Όπως απεικονίζεται στο Σχήμα 7.7, οι τρεις παράμετροι min_in , max_in , και $P_{\text{max_in}}$ ορίζουν τις καταστάσεις ομαλής λειτουργίας $[0, \text{min_in})$, αποφυγής συμφόρησης $[\text{min_in}, \text{max_in})$ και ελέγχου συμφόρησης $[\text{max_in}, \infty)$ για τα in πακέτα. Ομοίως ορίζουμε τις παραμέτρους min_out , max_out , και $P_{\text{max_out}}$ για τις αντίστοιχες καταστάσεις λειτουργίας των out πακέτων.

Η διάκριση μεταξύ in και out πακέτων στο μηχανισμό RIO πραγματοποιείται επιλέγοντας κατάλληλα τις παραμέτρους (min_in , max_in , $P_{\text{max_in}}$) και (min_out , max_out , $P_{\text{max_out}}$). Στο Σχήμα 7.7 παρατηρούμε ότι ένας RIO δρομολογητής είναι πιο “επιθετικός” στα out πακέτα. Αυτό συμβαίνει γιατί πρώτον ο δρομολογητής απορρίπτει τα out πακέτα νωρίτερα από ότι τα in πακέτα, επιλέγοντας min_out μικρότερο από min_in . Δεύτερον, στην κατάσταση αποφυγής συμφόρησης ο δρομολογητής απορρίπτει τα out πακέτα με μεγαλύτερη πιθανότητα επιλέγοντας $P_{\text{max_out}}$ μεγαλύτερο από $P_{\text{max_in}}$. Τρίτον, η κατάσταση ελέγχου συμφόρησης επέρχεται νωρίτερα για τα out πακέτα επιλέγοντας max_out πολύ μικρότερο από max_in . Ουσιαστικά, ο RIO απορρίπτει out πακέτα όταν ανιχνεύει επερχόμενη συμφόρηση και στη συνέχεια απορρίπτει όλα τα out πακέτα εάν η συμφόρηση παρατείνεται. Σαν τελευταία λύση όταν ο δρομολογητής έχει πλημμυρίσει από in πακέτα, απορρίπτει in πακέτα με την ελπίδα να καταφέρει να αντιμετωπίσει τη συμφόρηση. Είναι φανερό ότι κάτι τέτοιο δεν μπορεί να συμβεί σε ένα καλά σχεδιασμένο δίκτυο.

Η επιλογή του avg_total δηλ. του μέσου μήκους ουράς για in και out πακέτα, να καθορίζει τη πιθανότητα απόρριψης ενός out πακέτου είναι λογική. Σε αντίθεση με τα in πακέτα, τα out πακέτα εκφράζουν μη ελεγχόμενη κίνηση και δεν είναι γνωστός ο αριθμός αυτών που πρέπει να γίνεται αποδεκτός. Στη περίπτωση που το μέσο μήκος ουράς των out πακέτων καθόριζε την απόρριψη των out πακέτων, τότε δεν θα λαμβανόταν υπόψη η αύξηση του μήκους της ουράς λόγω άφιξης in πακέτων. Επίσης θα μπορούσε να χρησιμοποιηθεί το avg_in για να υπολογιστεί ο διαθέσιμος χώρος για τα out πακέτα. Σε μία τέτοια περίπτωση θα είχαμε απόρριψη λιγότερων out πακέτων όταν το avg_in είναι μικρό, και απόρριψη περισσότερων out πακέτων όταν το avg_in είναι μεγάλο. Κάτι τέτοιο θα αποδώσει μόνο εφόσον ο αριθμός

των in πακέτων στην ουρά είναι μεγάλος, έτσι ώστε οι δρομολογητές να μπορούν να ελέγξουν τον αριθμό των out πακέτων και το συνολικό μήκος ουράς. Τέλος, χρησιμοποιώντας το avg_total για τον υπολογισμό της πιθανότητας απόρριψης ενός out πακέτου, οι RIO δρομολογητές εξασφαλίζουν μικρού μήκους ουρές και υψηλό ρυθμό εξυπηρέτησης ανεξάρτητα από την αριθμητική αναλογία μεταξύ των in και out πακέτων.

7.3.1.3 Αδυναμίες αλγορίθμου RED

Οι May, Bolot, Diot και Lyles προσπάθησαν να απαντήσουν στα εξής ερωτήματα: Πρώτον εάν είναι σκόπιμο να απορρίπτονται πακέτα όταν δεν υπάρχει εμφανής λόγος και δεύτερον γιατί να χρησιμοποιείται ένας σχετικά πολύπλοκος αλγόριθμος, όπως είναι ο RED, για τον έλεγχο του μεγέθους των ουρών. Στα πειράματα που πραγματοποιήθηκαν χρησιμοποιήθηκε η υλοποίηση CISCO IOS 12.0 του μηχανισμού RED και η γεννήτρια φορτίου Chariot 2.2 για την παραγωγή της κίνησης του Διαδικτύου. Η παραγόμενη κίνηση ήταν κατά 80% TCP συνδέσεις και κατά 20% UDP συνδέσεις. Η σύγκριση της απόδοσης της προσέγγισης RED έγινε σε σχέση με τον απλό αλγόριθμο διαχείρισης ουρών ενταμιευτή, Complete Partitioning (CP). Ο εν λόγω αλγόριθμος εντάσσεται στους μηχανισμούς αντιμετώπισης συμφόρησης και απορρίπτει κάθε εισερχόμενο πακέτο όταν το μέγεθος της ουράς είναι μεγαλύτερο από ένα προκαθορισμένο όριο.

Τα αποτελέσματα της προσομοίωσης έδειξαν ότι η απόδοση του αλγορίθμου RED δεν διαφέρει πολύ από αυτή του CP. Στη περίπτωση UDP κίνησης φαίνεται ότι η τεχνική CP είναι πιο "επιθετική" στην απόρριψη πακέτων από ότι ο RED. Επίσης από τα αποτελέσματα προκύπτει ότι οι RED παράμετροι επηρεάζουν σε μικρό βαθμό την απόδοση του όταν ο χώρος του ενταμιευτή είναι μικρός. Αντίθετα όταν ο χώρος του ενταμιευτή είναι μεγάλος ο RED, με σωστή επιλογή των παραμέτρων του, βελτιώνει σημαντικά την απόδοση του συστήματος. Το πρόβλημα που προκύπτει στη συνέχεια είναι η παραμετροποίηση του μηχανισμού RED όπου διαπιστώνεται ότι δεν είναι μία απλή υπόθεση. Λόγω της δυναμικής φύσης των RED παραμέτρων του, ένας στατικός RED δεν μπορεί να προσφέρει καλύτερη απόδοση από ότι ο CP στη γενική περίπτωση. Τέλος, οι συγγραφείς προτείνουν περισσότερη ερευνητική εργασία για την πλήρη κατανόηση του μηχανισμού RED, τον καθορισμό των παραμέτρων του και αντιπροτείνουν πιο εξεζητημένους αλγορίθμους σαν αυτούς που παρουσιάζονται στην επόμενη ενότητα.

7.3.2 Αλγόριθμοι Διαχείρισης Ενταμιευτών για Αντιμετώπιση Συμφόρησης

Πολλοί αλγόριθμοι αντιμετώπισης συμφόρησης έχουν προταθεί στη υπάρχουσα βιβλιογραφία, οι οποίοι βελτιώνουν την απόδοση των μεταγωγέων και των δρομολογητών πακέτων. Οι τεχνικές αντιμετώπισης συμφόρησης μπορούν να διακριθούν σε αυτές που επιτρέπουν και σε

αυτές που δεν επιτρέπουν την αφαίρεση πακέτου από μία ουρά και την αντικατάσταση του από άλλο πακέτο (push-out). Οι τεχνικές push-out αποδέχονται πακέτα όλων των κλάσεων μέχρι να συμπληρωθεί ο συνολικός χώρος του ενταμιευτή. Όταν συμβεί αυτό ένα νεοεισερχόμενο πακέτο μπορεί να αντικαταστήσει ένα ήδη αποθηκευμένο πακέτο. Αντίθετα, για τις τεχνικές non push-out από τη στιγμή που ένα πακέτο εισήλθε σε ουρά του ενταμιευτή, τότε θα δρομολογηθεί σε διασύνδεση εξόδου του δρομολογητή.

Οι τεχνικές non push-out σύμφωνα με τη βιβλιογραφία επιτυγχάνουν τη βέλτιστη αξιοποίηση του χώρου του ενταμιευτή. Το μόνο μειονέκτημα που παρουσιάζουν είναι η πολυπλοκότητα υλοποίησής τους. Μέχρι τώρα οι τεχνικές push-out έχουν παρουσιαστεί σε μεταγωγείς πακέτων και κυρίως σε μεταγωγείς ATM (μεταγωγείς σταθερού μήκους πακέτου). Εντούτοις σε αντίθεση με τους μεταγωγείς ATM, οι δρομολογητές Διαδικτύου αποθηκεύουν σχετικά μικρότερους περιγραφητές πακέτων και επιπλέον αποθηκεύουν και στη συνέχεια προωθούν πακέτα (store and forward mode) με αποτέλεσμα οι επιπρόσθετες διαδικασίες εισαγωγής και αφαίρεσης πακέτων που απαιτούνται για τις τεχνικές push-out να μην δημιουργούν υπερβολικές απαιτήσεις σε μνήμη και πρόσθετες καθυστερήσεις.

Το επόμενο θέμα που προκύπτει είναι αν το πακέτο που αντικαθίσταται από το νεο-εισερχόμενο πακέτο πρέπει να είναι στην αρχή, στο τέλος ή σε οποιοδήποτε άλλη θέση της ουράς στην οποία ανήκει. Επιλέγεται η τεχνική απόρριψης από την αρχή της ουράς, γιατί αυτή βελτιώνει την απόδοση του TCP. Συγκεκριμένα, αυξάνει το ρυθμό εξυπηρέτησης μειώνοντας τη διάρκεια συμφόρησης, αφού για την απόρριψη πακέτου που υποδηλώνει συμφόρηση ενημερώνεται νωρίτερα ο αποστολέας. Επιπλέον, αυξάνεται η αμεροληψία της υλοποίησης, αφού σε διαφορετική περίπτωση η αντικατάσταση οποιουδήποτε πακέτου στην ουρά, όταν ο χώρος του ενταμιευτή έχει συμπληρωθεί, θα οδηγούσε σε απώλειες πακέτων που θα καθορίζονταν από τη κατανομή του χώρου του ενταμιευτή μεταξύ των ενεργών συνδέσεων.

Στη συνέχεια της ενότητας αυτής παρουσιάζονται αντιπροσωπευτικοί μηχανισμοί των κατηγοριών push-out και non push-out.

7.3.2.1 Τεχνικές non push-out

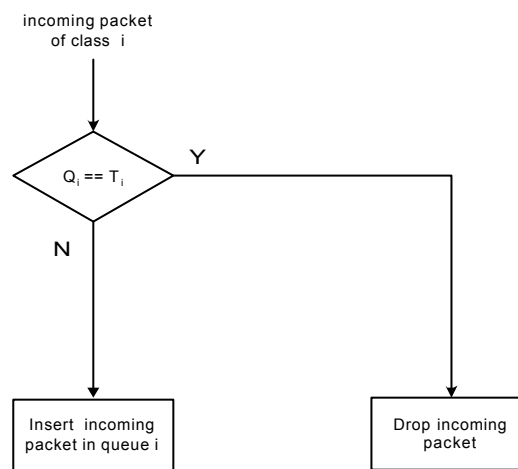
Η πιο συνηθισμένη τεχνική για τη διαχείριση των ουρών ενταμιευτή (μία ουρά για κάθε κλάση εξυπηρέτησης) ενός δρομολογητή είναι ο καθορισμός ενός μέγιστου μήκους (σε πακέτα) για κάθε ουρά, αποδοχή πακέτων για κάθε ουρά όσο επιτρέπει το μέγιστο μήκος αυτής και στη συνέχεια απόρριψη των επακόλουθων πακέτων μέχρι να μειωθεί το μέγεθος της ουράς με τη εκπομπή κάποιου πακέτου από την ουρά. Όπως έχει προαναφερθεί η τεχνική αυτή είναι γνωστή ως Complete Partitioning (CP). Το σημαντικό μειονέκτημα της τεχνικής CP όπως και κάθε τεχνικής non push-out είναι ότι πακέτα ενδέχεται να απορριφθούν ενώ υπάρχει διαθέσιμος χώρος στον ενταμιευτή. Από την άλλη, με τη χρήση της τεχνικής CP εξασφαλίζεται η απομόνωση (isolation) των κλάσεων

εξυπηρέτησης με αποτέλεσμα να ελέγχονται πλήρως οι ροές με κακή συμπεριφορά (misbehaving flows).

Στο Σχήμα 7.8 δίνεται το διάγραμμα ροής της τεχνικής CP, όπου με N συμβολίζουμε το συνολικό αριθμό των υποστηριζόμενων διαφοροποιημένων υπηρεσιών, με T_i το ανώτερο επιτρεπόμενο όριο

πακέτων (queue threshold) για την ουρά i , $i=1...N$ και με $B = \sum_{i=1}^N T_i$ το

συνολικό μήκος του ενταμιευτή. Ο αριθμός των πακέτων κλάσεως-ουράς i (queue occupancy), $i=1...N$ ορίζεται ως Q_i .



Σχήμα 7.8
Διάγραμμα ροής CP

7.3.2.2 Τεχνικές push-out

Οι τεχνικές της κατηγορίας αυτής καθυστερούν την απόρριψη πακέτων μέχρι να γίνει αναπόφευκτο δηλ. να συμπληρωθεί ο χώρος του ενταμιευτή και για αυτό το λόγο παρουσιάζουν συνήθως μικρότερο ρυθμό απόρριψης πακέτων από ότι οι τεχνικές non push-out.

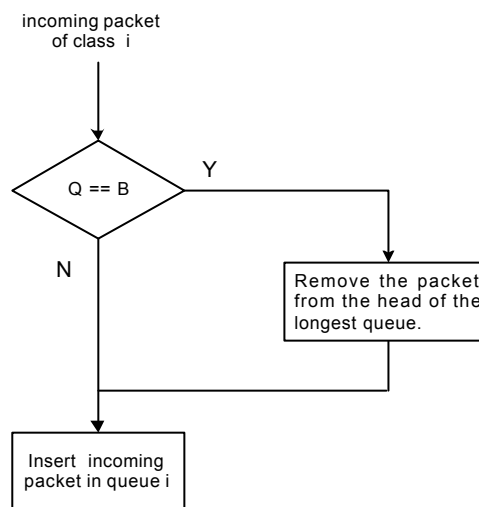
Αποβολή Πακέτου (Push-Out – PO)

Στο μηχανισμό PO, ο χώρος του ενταμιευτή, B μοιράζεται μεταξύ των διαφορετικών κλάσεων εξυπηρέτησης. Εάν ένα νεο-εισερχόμενο πακέτο βρίσκει το χώρο του ενταμιευτή συμπληρωμένο, τότε αντικαθιστά το πακέτο που βρίσκεται στην αρχή της ουράς με το μεγαλύτερο μήκος. Στο Σχήμα 7.9

Σχήμα 7.9 δίνεται το διάγραμμα ροής της τεχνικής PO, όπου με Q συμβολίζουμε το συνολικό αριθμό πακέτων στο χώρο του ενταμιευτή,

$$Q = \sum_{j=1}^N Q_j .$$

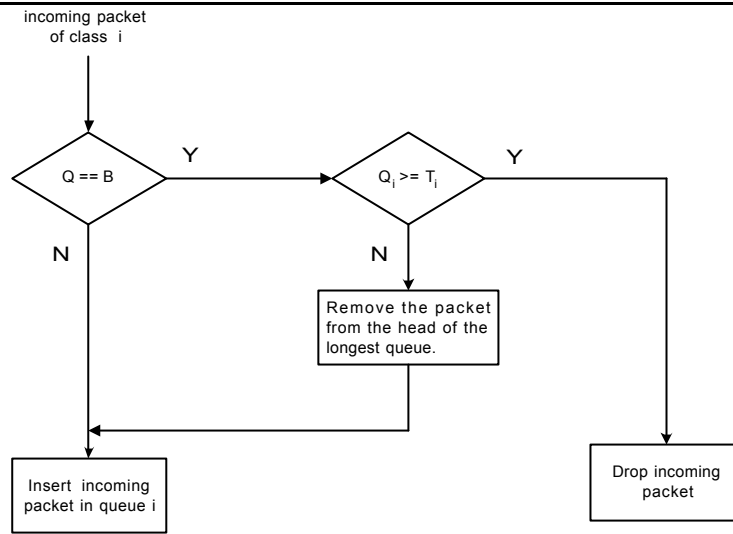
Η προσέγγιση PO παρουσιάζει πολλά πλεονεκτήματα. Πρώτον, είναι “δίκαιη” (fair) αφού επιτρέπει στις ουρές μικρότερου μήκους να μεγαλώνουν εις βάρος μεγαλύτερων ουρών. Δεύτερον, είναι αποτελεσματική αφού διατηρεί υψηλό ρυθμό εξυπηρέτησης. Αυτό συμβαίνει διότι με τη τεχνική PO καμία ουρά εξόδου δεν στερείται χώρο αποθήκευσης και επιπλέον σε καμία περίπτωση χώρος του ενταμιευτή δεν μένει ανεκμετάλλευτος, ενώ κάποιες κλάσεις επιζητούν περισσότερο. Τέλος, η προσέγγιση PO χαρακτηρίζεται ως προσαρμόσιμη (adaptive), αφού όταν υπάρχουν πολλές ενεργές ουρές το μήκος των ουρών αυτών διατηρείται μικρό, ενώ στην περίπτωση μίας ενεργής ροής, το μήκος αυτής μπορεί να γίνει μεγάλο.



Σχήμα 7.9
Διάγραμμα ροής PO

Αποβολή Πακέτου με Κατώφλι (Push-Out with Threshold – POT)

Κάθε κλάση εξυπηρέτησης i δεσμεύει έναν ορισμένο χώρο του ενταμιευτή ίσο με το ανώτερο επιτρεπόμενο όριο πακέτων της κλάσεως του T_i , $i=1\dots N$. Όσο υπάρχει ελεύθερος χώρος στον ενταμιευτή, αυτός μοιράζεται μεταξύ των N κλάσεων εξυπηρέτησης. Όταν ένα νεο-εισερχόμενο πακέτο κλάσεως i , $i=1\dots N$ βρίσκει ενταμιευτή συμπληρωμένο και ο αριθμός των αποθηκευμένων πακέτων της κλάσεως του είναι μικρότερος από T_i , το πακέτο εισέρχεται στην ουρά της κλάσεως του και απορρίπτεται το πακέτο στην αρχή της ουράς μεγαλύτερου μήκους. Διαφορετικά, όταν ο αριθμός των αποθηκευμένων πακέτων της i κλάσεως είναι μεγαλύτερος ή ίσος με T_i , το πακέτο απορρίπτεται. Στο επόμενο σχήμα δίνεται το διάγραμμα ροής του μηχανισμού POT.



Σχήμα 7.10
Διάγραμμα ροής POT

Η ουρά μεγαλύτερου μήκους σε αυτή τη περίπτωση θα μπορούσε να είναι η ουρά με τη μεγαλύτερη διαφορά μεταξύ του χρησιμοποιούμενου χώρου ενταμιευτή και του αντίστοιχου επιτρεπόμενου ορίου πακέτων για την εν λόγω κλάση. Τα πλεονεκτήματα της τεχνικής PO ισχύουν και για τον μηχανισμό POT. Επιπρόσθετα, εξ'ορισμού προκύπτει ότι ο POT επιδιώκει τον έλεγχο και τον περιορισμό των ροών ή κλάσεων με κακή συμπεριφορά.