

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΔΙΚΤΥΩΝ

ΔΙΑΛΕΞΗ 7

ΔΙΔΑΣΚΩΝ: ΑΝΑΡΓΥΡΟΣ ΣΙΔΕΡΗΣ

`<sideris@epp.teiher.gr>`

`<https://eclass2.teicrete.gr/courses/TP182/>`

ΤΜΗΜΑ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΠΟΛΥΜΕΣΩΝ
ΤΕΙ ΚΡΗΤΗΣ



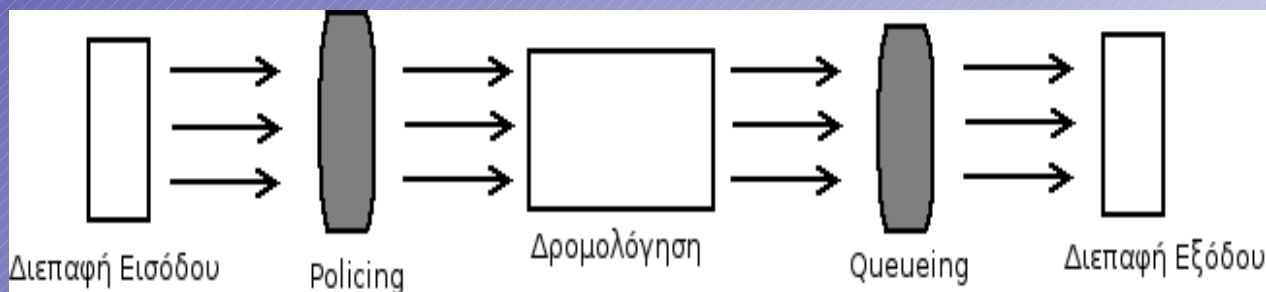
Διαχείριση Δικτυακών πόρων

(Network Resources
Management)



Γενικά (1)

- Τι είναι η διαχείριση δικτυακών πόρων;
 - Η (ανα)κατανομή εύρους ζώνης σε υπηρεσίες.
- Γιατί το χρειαζόμαστε;
 - Για τη παροχή QoS
- Πως;
 - Έλεγχο των ιδιαίτερων χαρακτηριστικών ενός καναλιού μετάδοσης.
 - π.χ. Αλλαγή διαμόρφωσης, κωδικοποίησης.
 - Έλεγχο του ρυθμού μετάδοσης των πακέτων.
 - Τεχνικές επιβολής ρυθμού (policers).
 - Διαχείριση ουρών (queue management)

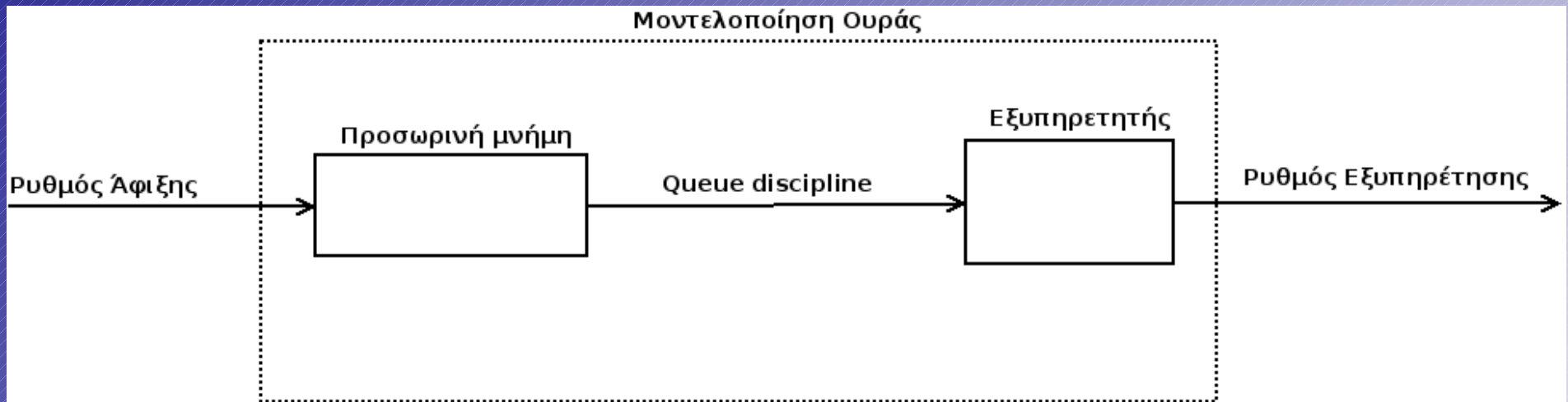


Γενικά (2)

- **Policing:**
 - Επιβολή ενός συγκεκριμένου ρυθμού μετάδοσης απορρίπτοντας τα επιπλέον δεδομένα μιας ροής που υπερβαίνει τον εν λόγω ρυθμό.
- **Queue management:**
 - Ο τρόπος αποθήκευσης, απόρριψης και ρυθμού εξαγωγής των πακέτων δεδομένων από την ουρά.

Ουρές (1)

- Τι είναι ουρά;
 - Μια προσωρινή μνήμη (buffer) στην οποία αποθηκεύονται τα πακέτα και ένας εξυπηρετητής (server) που προωθεί τα πακέτα στο δίκτυο.



- Ρυθμός άφιξης: Πόσο γρήγορα εισέρχονται τα πακέτα στην ουρά.
- Ρυθμός εξυπηρέτησης: Πόσο γρήγορα εξέρχονται τα πακέτα από την ουρά.

Ουρές (2)

- **Queue discipline:** Ο τρόπος εξαγωγής των πακέτων από τη προσωρινή μνήμη.
 - π.χ. Εάν έχω τα πακέτα μιας TCP μιας UDP ροής ποιας τα πακέτα να εξάγω πρώτα στον εξυπηρετητή.
- **Δεδομένα:**
 - T ο μέσος χρόνος παραμονής ενός πακέτου μέσα στην ουρά.
 - λ ο ρυθμός άφιξης.
 - N ο μέσος αριθμός των πακέτων στην ουρά.
 - Τότε $N = \lambda * T$ (Little's theorem)
 - Το θεώρημα αυτό μπορεί να εφαρμοστεί και για κάθε ένα υποσύστημα της ουράς ξεχωριστά.



Ουρές (3)

- Υποσύστημα εξυπηρετητή:
 - Το λ εδώ είναι ο ρυθμός εισαγωγής στο server.
 - Ο χρόνος παραμονής είναι ο χρόνος εκπομπής.
 - Άρα μέσος όρος πακέτων μέσα στον εξυπηρετητή είναι ίσος με $N = \lambda * T$.
 - Το N εδώ είναι γνωστό και σαν utilization της ζεύξης και συμβολίζεται με το γράμμα ρ .
- Το ίδιο ισχύει και για τα άλλα δύο υποσυστήματα:
 - Υποσύστημα προσωρινής μνήμης.
 - Υποσύστημα εξερχόμενης ζεύξης
- Τι είναι το N για τα δύο προηγούμενα συστήματα;



Ουρές (4)

- Χρήσιμοι τύποι:

- Δεδομένα:

- N ο μέσος αριθμός πακέτων σε όλο το σύστημα.
 - T ο μέσος χρόνος παραμονής σε όλο το σύστημα.
 - λ ο ρυθμός άφιξης σε όλο το σύστημα.
 - μ ο ρυθμός εξυπηρέτησης.
 - ρ η βελτιστοποίηση της ζεύξης.
 - W ο μέσος χρόνος παραμονής στη ουρά.

- Τότε:

- $N = \rho / (1 - \rho)$
 - $N = \lambda / (\mu - \lambda)$
 - $T = 1 / (\mu - \lambda)$
 - $W = (1 / \mu - \lambda) - (1 / \mu)$
 - $W = T - (1 / \mu)$

Ουρές (5)

- Κατηγορίες ουρών:
 - Δίκαιες/Μη Δίκαιες:
 - οι δίκαιες εξασφαλίζουν την ίδια πιθανότητα εξυπηρέτησης/απόρριψης για τα πακέτα διαφορετικών ροών. Η μη δίκαιες τσου!!!
 - Ενεργητικές/Παθητικές:
 - Ενεργητικές ονομάζονται οι ουρές που απορρίπτουν πακέτα πριν η ουρά γεμίσει πλήρως σε αντιδιαστολή με τις παθητικές οι οποίες απορρίπτουν πακέτα εφόσον έχουν ήδη γεμίσει.
 - Χωρίς/Με κατηγοριοποίηση ροών:
 - Υποστήριξη ή όχι για ταξινόμηση των ροών ανάλογα με τα ιδιαίτερα χαρακτηριστικά τους.

Ασκήσεις

- Σε ένα δρομολογητή (αναγάγουμε σε σύστημα ουράς) καταφθάνουν πακέτα με ρυθμό 25/sec. Ο χρόνος εξυπηρέτησης είναι 30 ms.
- Υπολογίστε:
 - Το μέσο χρόνο που μένουν μέσα στο δρομολογητή.
 - $\mu = 1/0,03 = 34$ pps, $T = 1/(34 - 25) = 0.1$ sec
 - Το μέσο χρόνο παραμονής μέσα στη ουρά.
 - $W = T - (1/\mu) = 0,1 - (1/34) = 0.07$ sec.
 - Το μέσο αριθμό πακέτων μέσα στο δρομολογητή.
 - $N = \lambda * T = 25 * 0,1 = 2,5$ packets.
 - Τη βελτιστοποίηση του εξυπηρετητή των πακέτων.
 - $\rho = \lambda / \mu = 25 / 34 = 0,74$

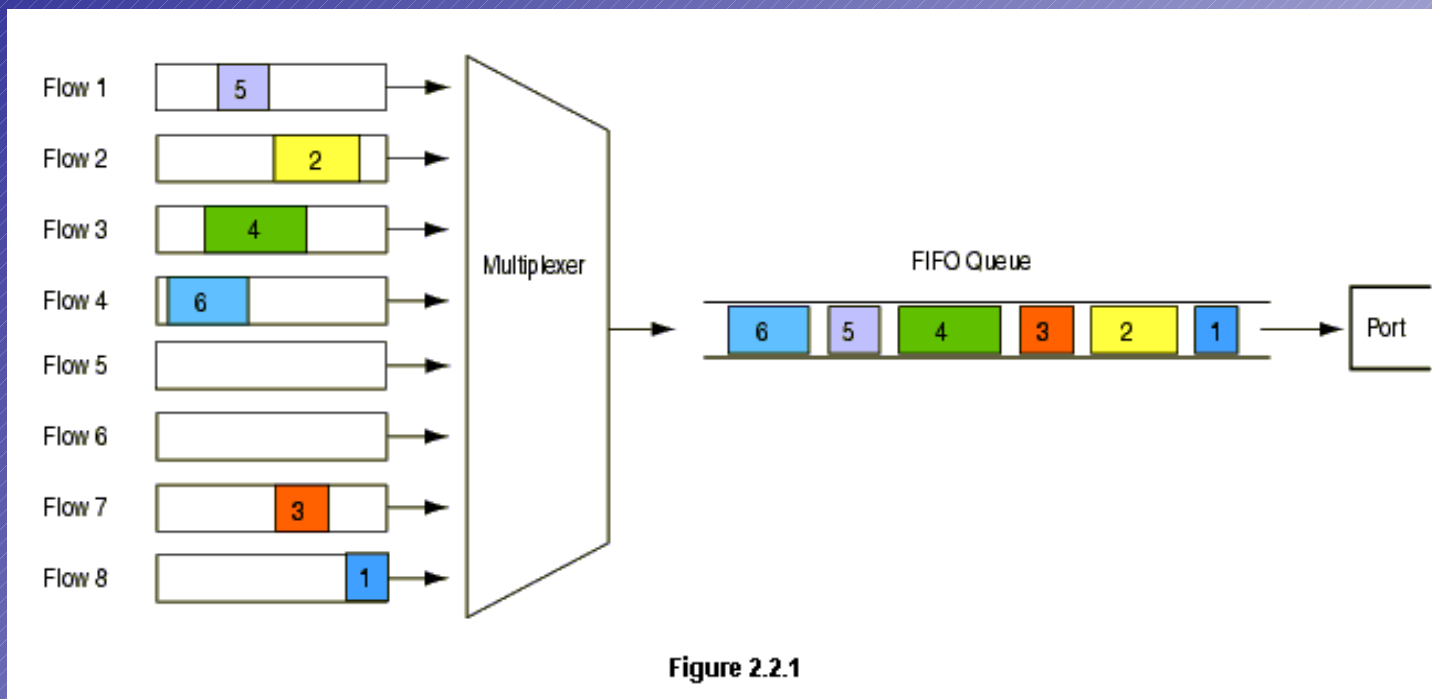


Queue disciplines



FIFO (1)

- Τα πακέτα εξέρχονται με τη σειρά που εισόδου τους στην ουρά.



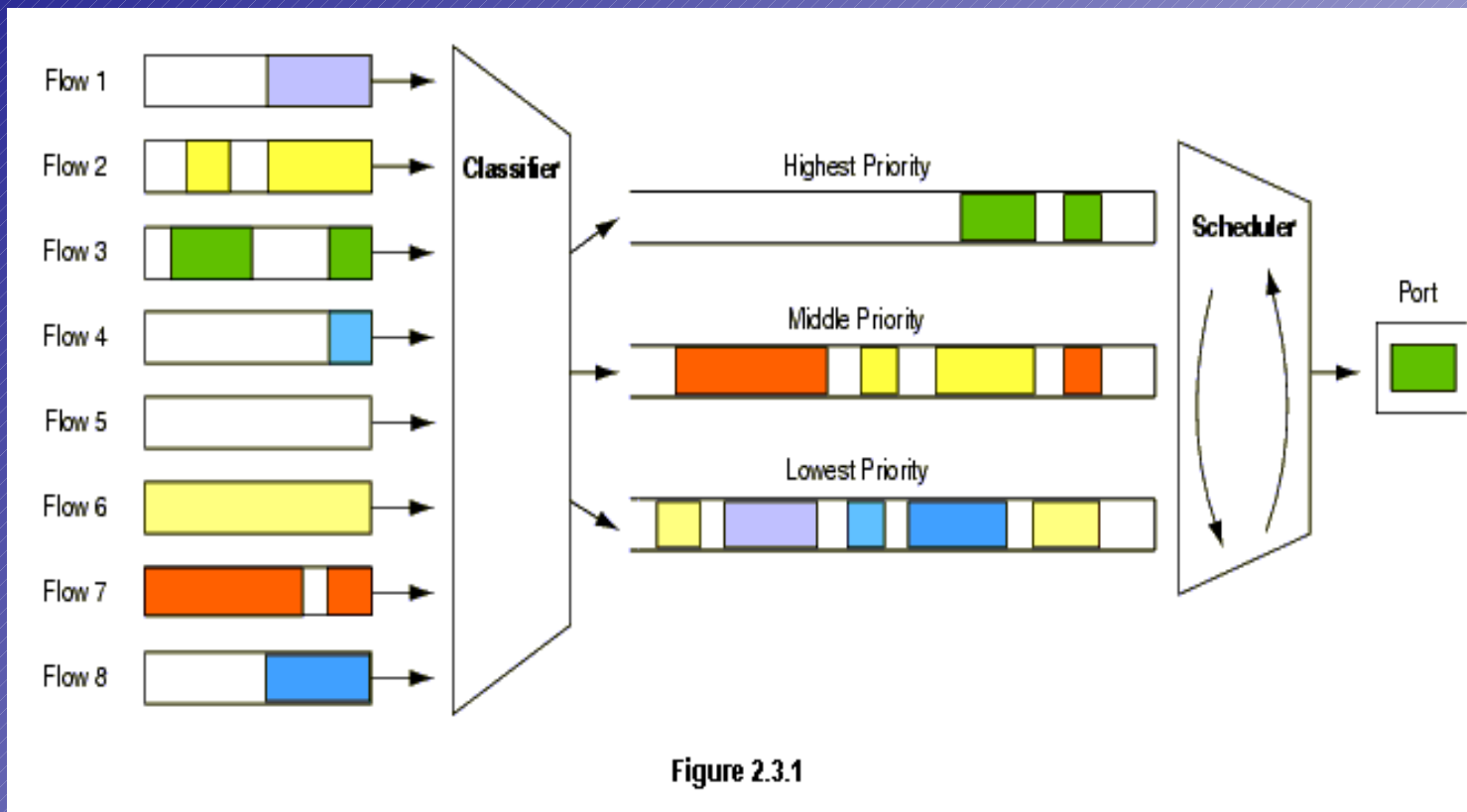
Πηγή: <http://www.opalsoft.net/qos/DS-22.htm>

FIFO (2)

- Χαρακτηριστικά:
 - Είναι γρήγορη.
 - Είναι παθητική.
 - Δεν είναι δίκαιη.
 - Δεν υποστηρίζει κατηγοριοποίηση ροών.
- Ερωτήσεις:
 - Σε περιόδους συμφόρηση τις UDP ή τις TCP ροές αδικεί πιο πολύ μια FIFO;
 - Μια bursty ροή τι μπορεί να προκαλέσει σε άλλες ροές που χρησιμοποιούν την ίδια FIFO;



PRIO (1)



Πηγή: <http://www.opalsoft.net/qos/DS-23.htm>

PRIO (2)

- Οι εισερχόμενες ροές δεδομένων ταξινομούνται σε 3 κατηγορίες διαφορετικής προτεραιότητας (υψηλή-μεσαία-χαμηλή) .
- Υπάρχουν 3 FIFO διαφορετικής προτεραιότητας (υψηλή-μεσαία-χαμηλή) .
- Κάθε κατηγορία εισάγεται στην ουρά με την ίδια προτεραιότητα.
- Εξυπηρετούνται πρώτα οι ροές υψηλότερης προτεραιότητας.
- Εάν δεν υπάρχουν άλλα πακέτα σε μια υψηλότερη προτεραιότητα τότε εξυπηρετούνται τα χαμηλότερης προτεραιότητας.

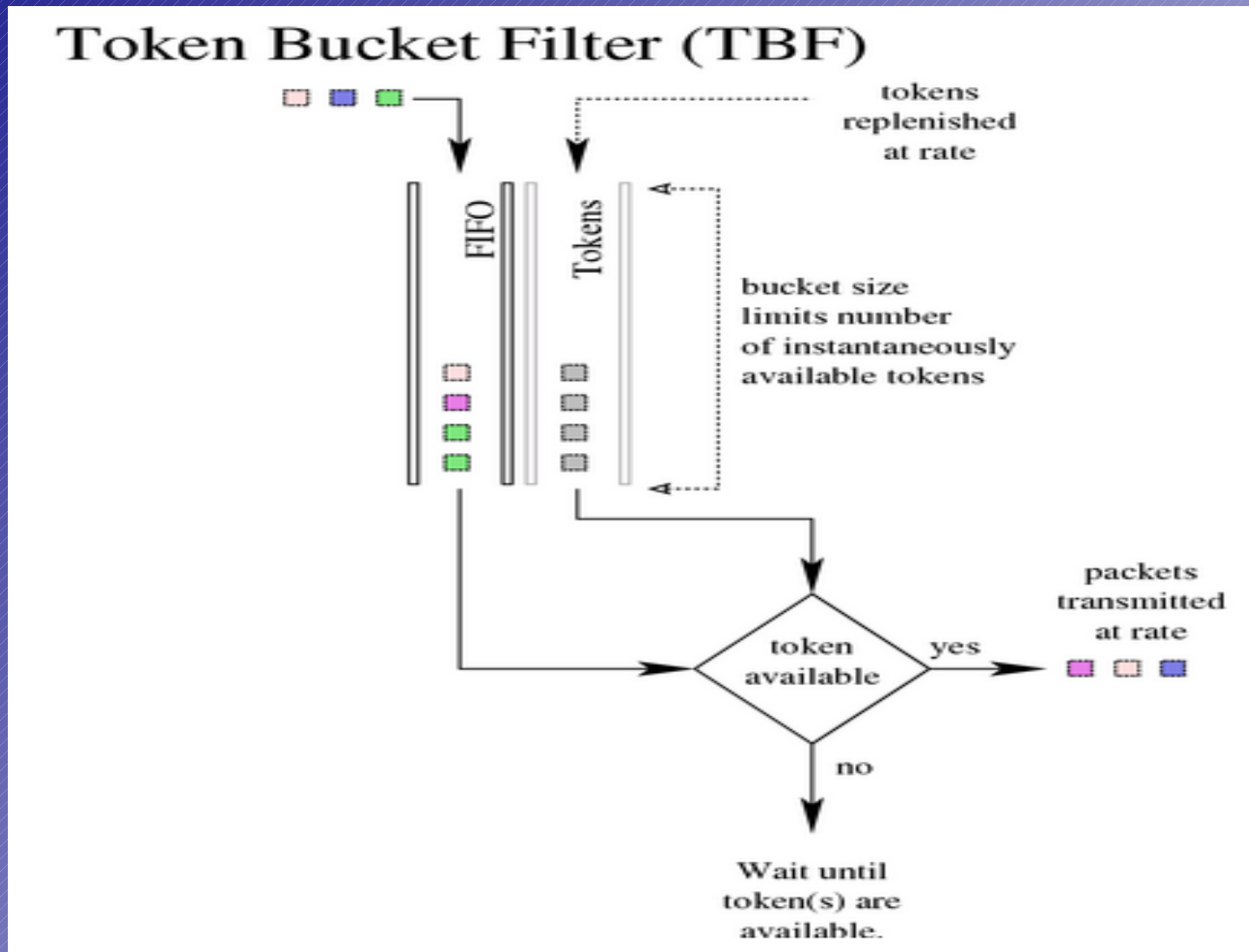


PRIO (3)

- Χαρακτηριστικά:
 - Είναι γρήγορη.
 - Είναι παθητική.
 - Δεν είναι δίκαιη.
 - Υποστηρίζει κατηγοριοποίηση ροών.
- Ερωτήσεις:
 - Η PRIO λύνει το πρόβλημα που είχαν οι TCP ροές από τις UDP στη FIFO;
 - Πώς κατηγοριοποιούμε τις ροές;



TBF (1)



Πηγή: <http://linux-ip.net/articles/Traffic-Control-HOWTO/classless-qdiscs.html>

TBF (2)

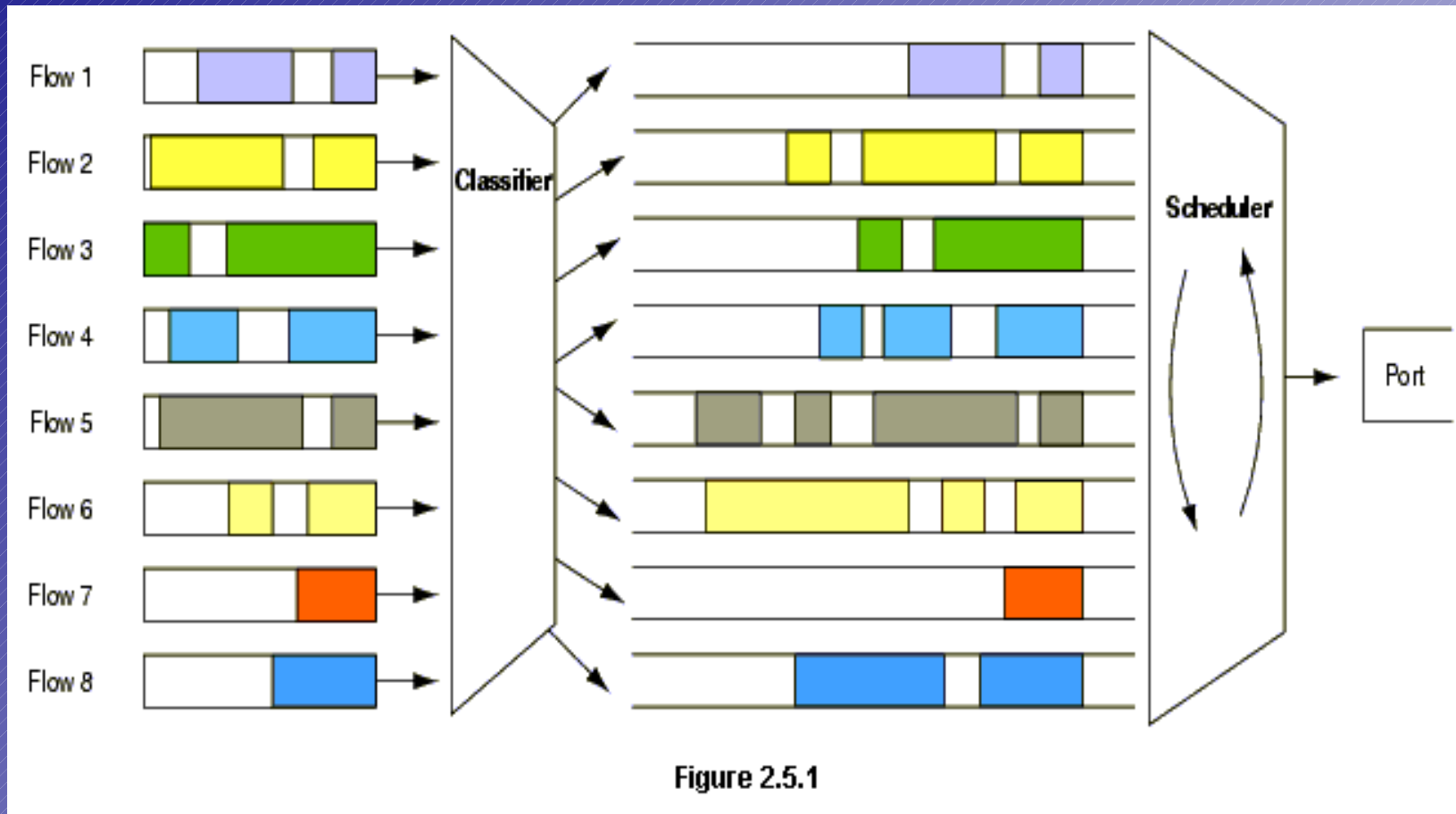
- Υπάρχουν δύο FIFO ουρές.
- Στη μία εισάγονται τα πακέτα ενώ στην άλλη tokens.
- Η FIFO των tokens ονομάζεται bucket.
- Ο ρυθμός εισαγωγής των tokens ορίζεται από εμάς.
- Πακέτα μπορούν να εξαχθούν μόνο εάν υπάρχουν διαθέσιμα tokens.
- Τρεις περιπτώσεις:
 - Ο ρυθμός άφιξης των πακέτων είναι ίσος με το ρυθμό των tokens.
 - Ο ρυθμός άφιξης των πακέτων είναι μικρότερος από το ρυθμό των tokens.
 - Ο ρυθμός άφιξης των πακέτων είναι μεγαλύτερος από το ρυθμό των tokens.



TBF (3)

- Χαρακτηριστικά:
 - Είναι γρήγορη.
 - Είναι παθητική.
 - Δεν είναι δίκαιη.
 - Δεν υποστηρίζει κατηγοριοποίηση ρών.
- Ερωτήσεις:
 - Για πιο λόγο να χρησιμοποιήσουμε μια TBF;
 - Πότε υπάρχει περίπτωση να συσσωρεύονται tokens; Πώς πιστεύεται ότι χρησιμοποιούνται;

SFQ



Πηγή: <http://www.opalsoft.net/qos/DS-25.htm>

SFQ (2)

- Υπάρχουν ένας πεπερασμένος αριθμός από FIFO ουρές.
- Ένας hash αλγόριθμος τοποθετεί κάθε ροή σε μια FIFO.
- Κάθε FIFO εξυπηρετείται με τη σειρά της και για το ίδιο χρονικό διάστημα.
- Ο hash αλγόριθμος αλλάζει ανά κάποιο χρονικό διάστημα;
 - Γιατί;



SFQ (3)

- Χαρακτηριστικά:
 - Είναι γρήγορη.
 - Είναι παθητική.
 - Είναι δίκαιη.
 - Δεν υποστηρίζει κατηγοριοποίηση ροών.



RED (1)

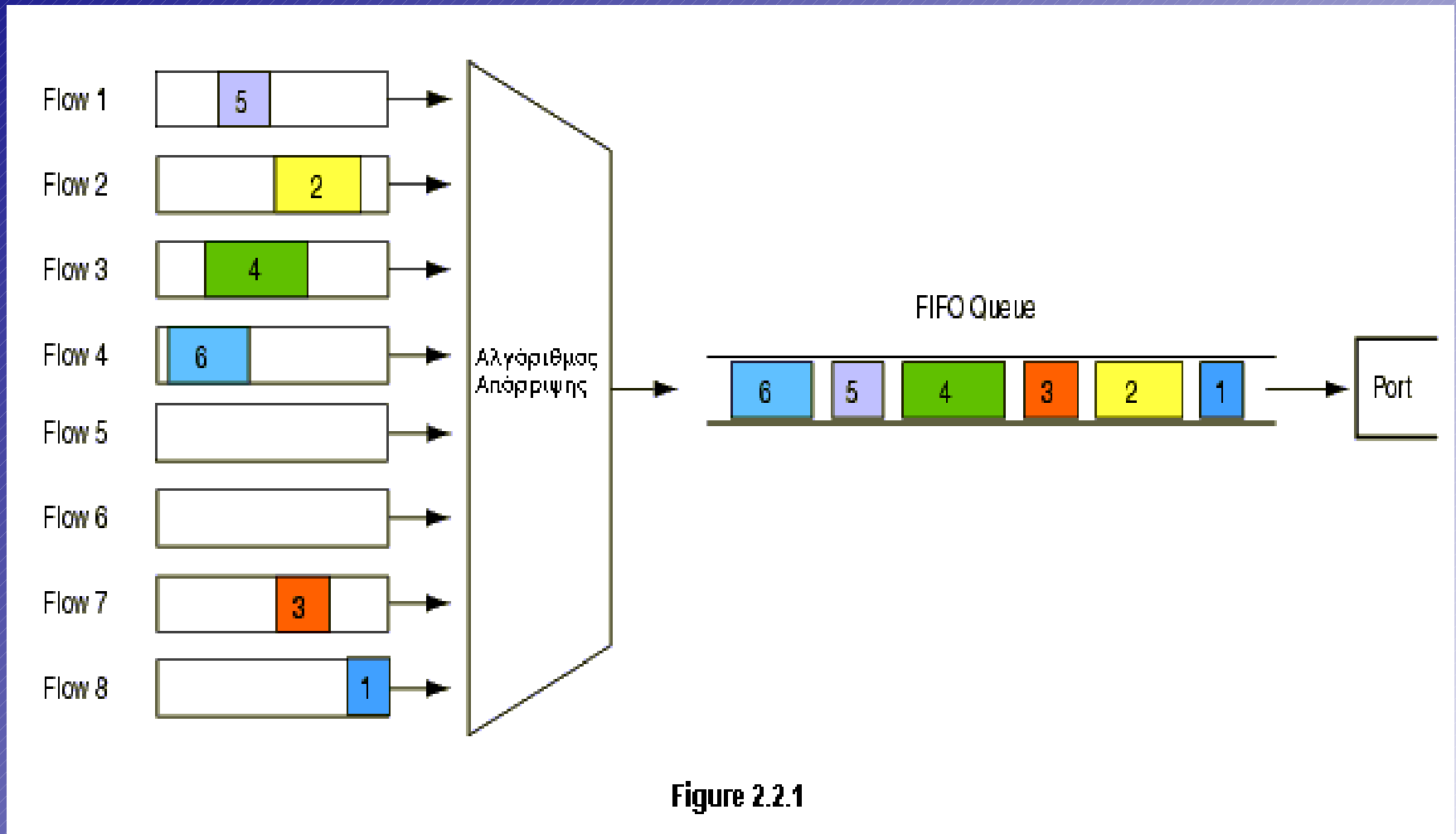


Figure 2.2.1

Πηγή: <http://www.opalsoft.net/qos/DS-26.htm> (τροποποιημένη)

RED (2)

- Υπάρχουν μια FIFO και ένας αλγόριθμος απόρριψης πακέτων.
- Ο αλγόριθμος απόρριψης δουλεύει ως εξής:
 - Εάν τα πακέτα μέσα σε μια ουρά είναι κάτω από ένα όριο (min_nq) τότε η πιθανότητα να απορριφτεί ένα πακέτο είναι 0.
 - Εάν τα πακέτα μέσα σε μια ουρά είναι πάνω από ένα όριο (max_nq) τότε η πιθανότητα να απορριφτεί ένα πακέτο είναι 1.
 - Εάν τα πακέτα μέσα σε μια ουρά είναι μεταξύ max_nq και min_nq τότε η πιθανότητα να απορριφτεί ένα πακέτο είναι μεταξύ του ανοιχτού διαστήματος $(0,1)$.

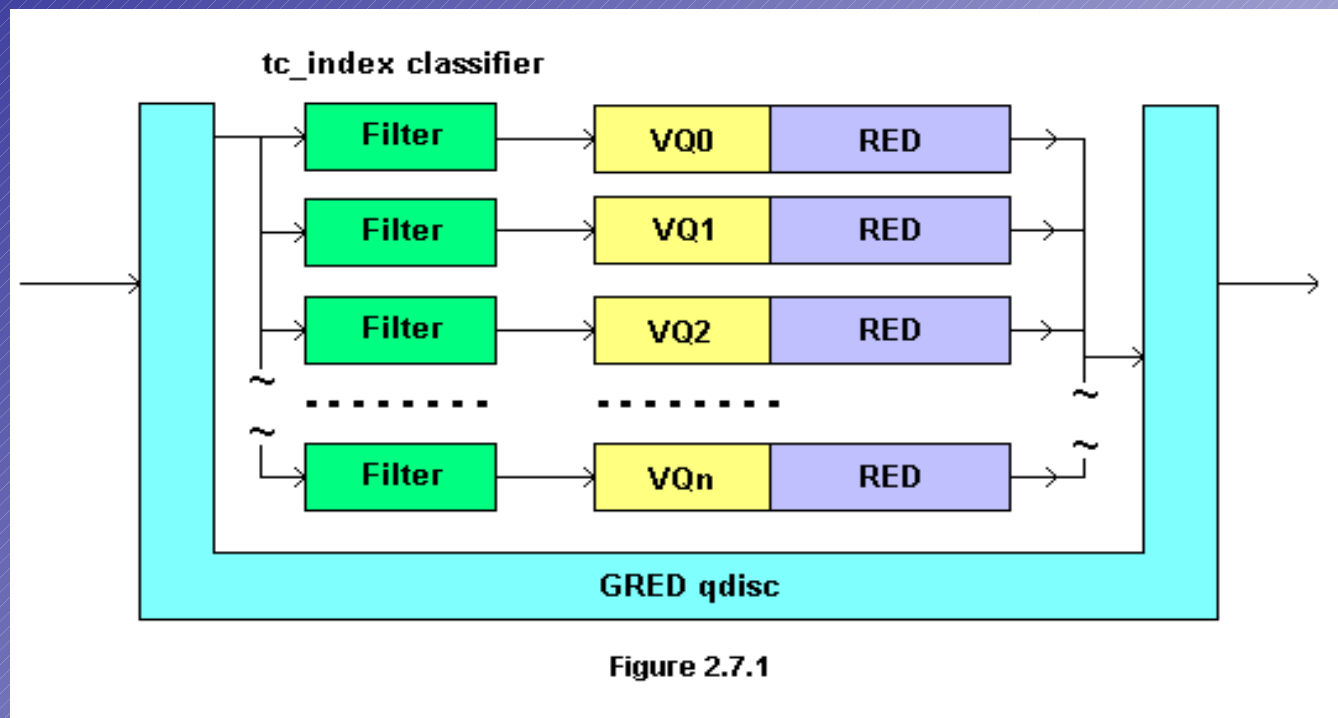
RED (3)

- Χαρακτηριστικά:
 - Είναι γρήγορη.
 - Είναι ενεργητική.
 - Δεν είναι δίκαιη.
 - Δεν υποστηρίζει κατηγοριοποίηση ροών.
- Ερωτήσεις:
 - Για πιο λόγο να χρησιμοποιήσουμε μια RED;
 - Αποφυγή φαινομένου συγχρονισμού ροών.
 - Άλλο;
 - Είναι ιδανική για TCP ή UDP ροές;



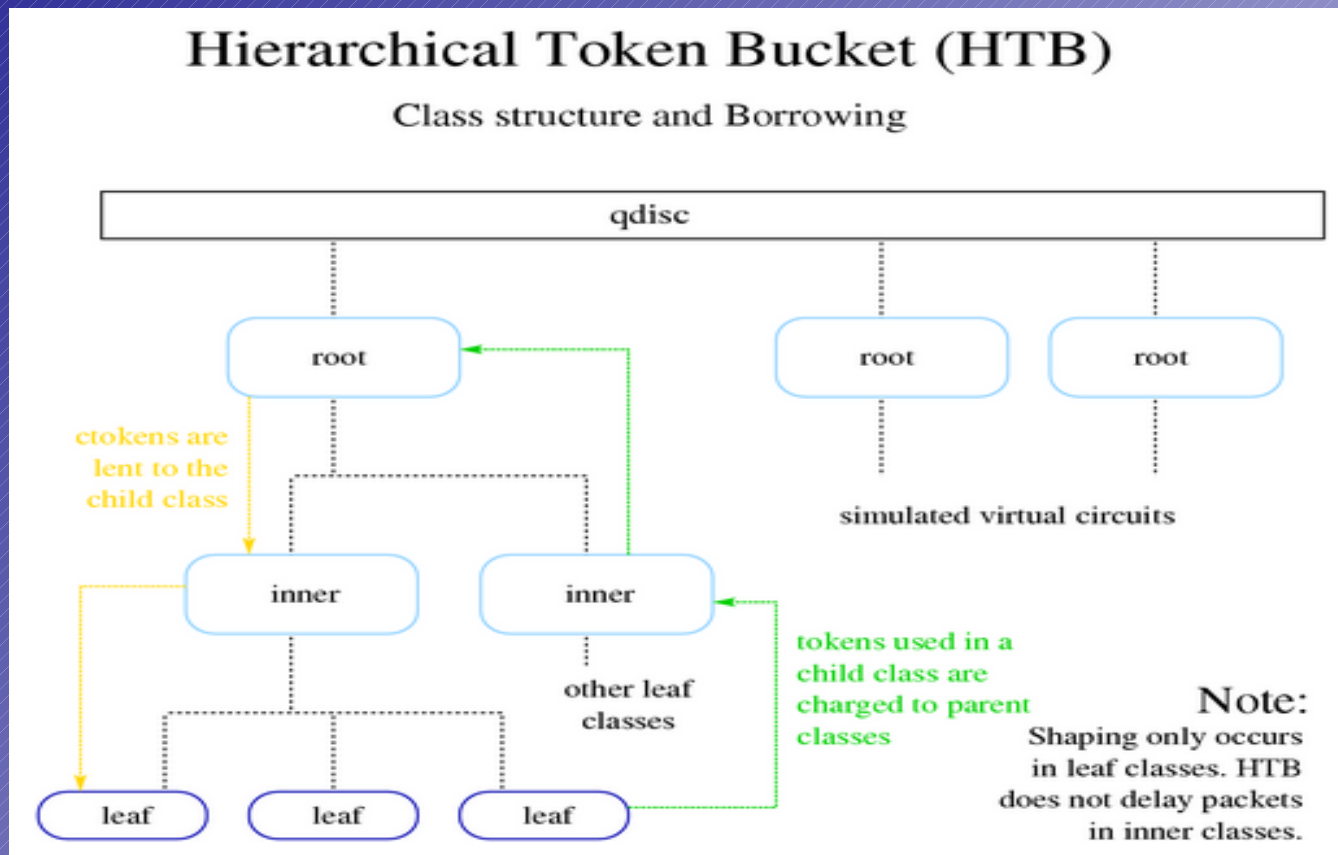
URED

- Η RED με δυνατότητα κατηγοριοποίησης.
- Ερώτηση:
 - Πιο PHB του DiffServ μπορεί να υλοποιηθεί;



Πηγή: <http://www.opalsoft.net/qos/DS-27.htm>

HTB (1)



Πηγή: <http://linux-ip.net/articles/Traffic-Control-HOWTO/classful-qdiscs.html>

HTB (2)

- Βασίζεται στην ιδέα του TBF.
- Οι ροές μπορούν να χωριστούν σε κλάσεις.
- Κάθε κλάση μπορεί να χωριστεί σε υποκλάσεις.
- Σε κάθε κλάση κατανέμεται ένα μέρος του συνολικού εύρους ζώνης με τη χρήση tokens.
- Το τελευταίο παιδί κάθε κλάσης μπορεί να είναι μια FIFO, SFQ, RED, PRIO, GRED.
- Η κάθε κλάση μπορεί να δανειστεί πόρους από τον γονέα της.



HTB (3)

- Χαρακτηριστικά:
 - Είναι ενεργητική και παθητική.
 - Είναι δίκαιη και άδικη.
 - Υποστηρίζει κατηγοριοποίηση ροών.
- Ερωτήσεις:
 - Χρησιμοποιήστε τη HTB για να κατανέμετε τους πόρους της 100Mbit κάρτα του router σας κατά τον εξής τρόπο:
 - www (25%)
 - Mail (25%)
 - IP video audio (50%)
 - Video (90%) (οι ροές video να εξυπηρετούνται δίκαια).
 - Audio (10%)

ΒΙΒΛΙΟΓΡΑΦΙΑ

- "Δίκτυα Υπολογιστών", A.S. Tanenbaum, Εκδόσεις Παπασωτηρίου, 3η έκδοση.
- "Differentiated Service on Linux HOWTO", Leonardo Balliache,
<http://www.opalsoft.net/qos/DS.htm>,
τελευταία πρόσβαση 13/05/2010.
- "Traffic Control HOWTO", Martin A. Brown,
<http://linux-ip.net/articles/Traffic-Control-HOWTO/index.html>,
τελευταία πρόσβαση 13/05/2010.