Κατανομή Πόρων και Ποιότητα Υπηρεσιών (Resource Allocation and Quality of Service)

Περίληψη

- Πόροι του δικτύου
- Εφαρμογές και οι απαιτήσεις τους
- Συμφόρηση και μέθοδοι αποσυμφόρησης
  - Στο επίπεδο μεταφοράς
  - Στο επίπεδο δικτύου
  - Στο επίπεδο ζεύξης δεδομένων
- Μέθοδοι παροχής ποιότητας υπηρεσιών
  - Καθοριστικές (Deterministic) και στατιστικές εγγυήσεις
  - Ολοκληρωμένες και Διαφοροποιημένες Υπηρεσίες (Integrated vs. Differentiated services model).
  - Έλεγχος ροών
  - Μέθοδοι «προγραμματισμού» (scheduling) των πακέτων

Πόροι Δικτύου

- Εύρος Ζώνης (bandwidth)
- Ενδιάμεση μνήμη (buffer)
  - Προσωρινή αποθήκευση πακέτων (packet storage)
  - Πίνακες δρομολόγησης (Routing tables)
- Διεκπεραιωτική ικανότητα των μεταγωγέων και δρομολογητών (CPU processing speed of switches and routers)

- **Πρόβλημα**: Πως μπορούν να κατανεμηθούν οι πιο πάνω πόροι έτσι ώστε οι διάφορες εφαρμογές να τρέχουν ικανοποιητικά αλλά και να πετυχαίνεται ικανοποιητική χρήση των πόρων έτσι ώστε το κόστος εγκατάστασης και διαχείρισης του δικτύου να ελαχιστοποιηθεί;

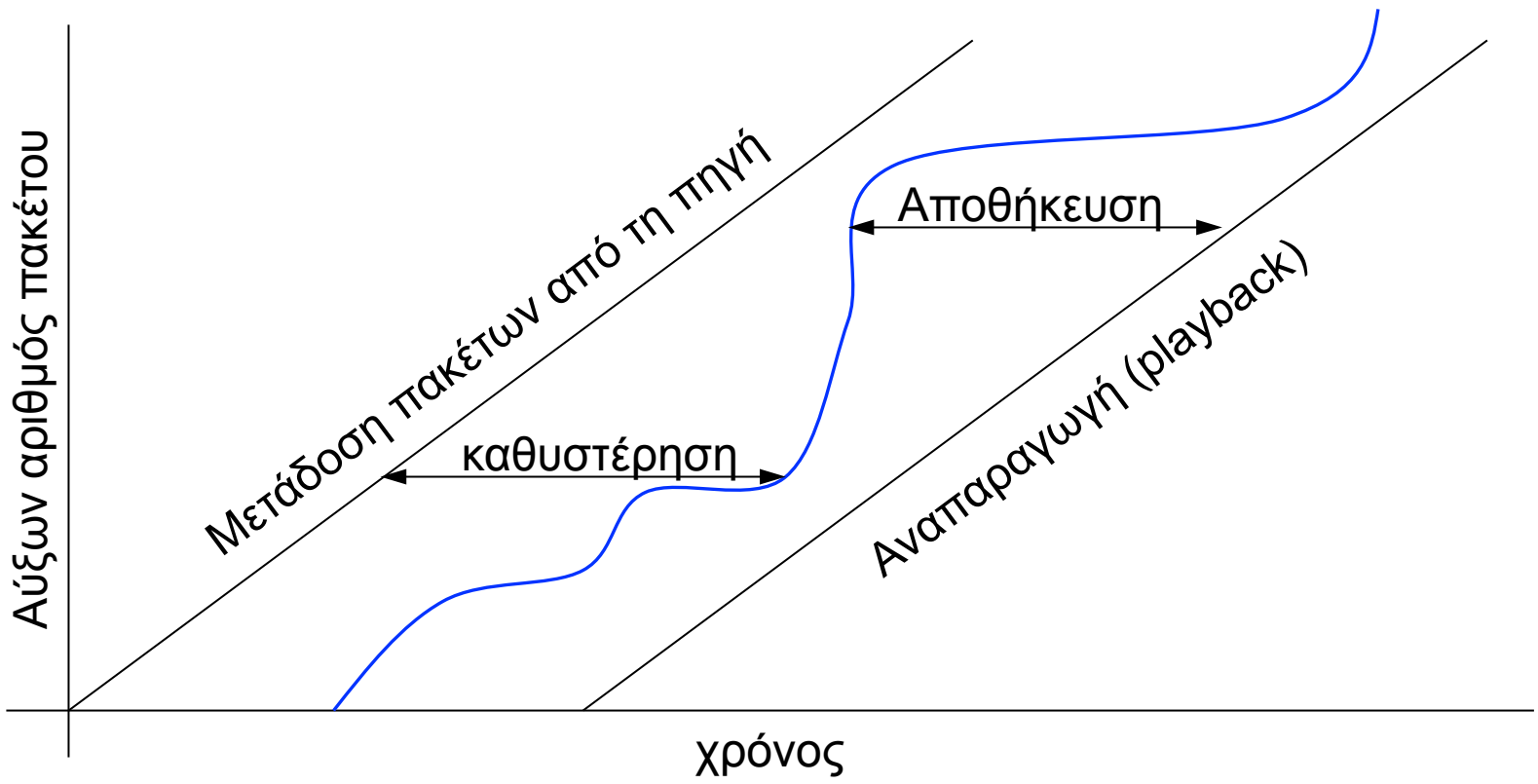
Εφαρμογές και οι απαιτήσεις τους

- Ηλεκτρονικό Ταχυδρομείο (Email)
  - Υψηλή αξιοπιστία
  - Χαμηλή καθυστέρηση, διακύμανση καθυστέρησης και εύρος ζώνης
- Μεταφορά Αρχείων (file transfer)
  - Υψηλή αξιοπιστία
  - Χαμηλή καθυστέρηση και διακύμανση καθυστέρησης
  - Μέτριο εύρος ζώνης
- Πρόσβαση στο Παγκόσμιο Πλέγμα Πληροφοριών (ΠΠΠ) (www)
  - Υψηλή αξιοπιστία
  - Μέτρια καθυστέρηση
  - Χαμηλή διακύμανση καθυστέρησης
  - Μέτριο εύρος ζώνης

Εφαρμογές και οι απαιτήσεις τους

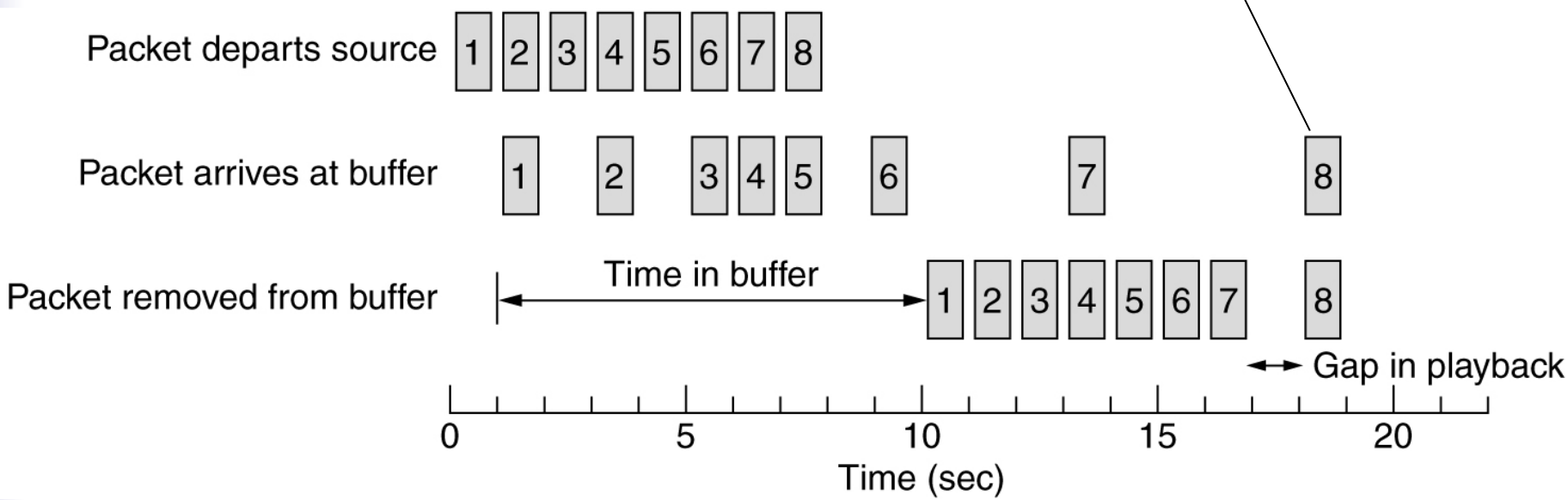
- Σύνδεση εξ αποστάσεως (remote login)
  - Υψηλή αξιοπιστία
  - Μέτρια καθυστέρηση και διακύμανση καθυστέρησης
  - Χαμηλό εύρος ζώνης
- Audio on demand
  - Χαμηλή ή μέτρια αξιοπιστία
  - Χαμηλή ή μέτρια καθυστέρηση
  - Υψηλή διακύμανση καθυστέρηση
  - Μέτριο ή χαμηλό εύρος ζώνης
- Video on demand
  - Χαμηλή ή μέτρια αξιοπιστία
  - Χαμηλή ή μέτρια καθυστέρηση
  - Υψηλή διακύμανση καθυστέρηση
  - Υψηλό εύρος ζώνης

Audio + Video on Demand



Audio + Video on Demand

Η διακύμανση στην καθυστέρηση είναι μεγαλύτερη απ' ότι αναμενόταν με αποτέλεσμα το πακέτο να φτάσει καθυστερημένο



Η αποθήκευση στην ενδιάμεση μνήμη χρησιμοποιείται για τη βελτίωση της ποιότητας της προσφερόμενης υπηρεσίας!

Εφαρμογές και οι απαιτήσεις τους

- Τηλεφωνία
  - Χαμηλή ή μέτρια αξιοπιστία
  - Υψηλή καθυστέρηση
  - Υψηλή διακύμανση καθυστέρηση
  - Χαμηλό εύρος ζώνης
- Videoconferencing
  - Χαμηλή ή μέτρια αξιοπιστία
  - Υψηλή καθυστέρηση
  - Υψηλή διακύμανση καθυστέρηση
  - Υψηλό εύρος ζώνης
- Στις εφαρμογές της τηλεφωνίας και videoconferencing **δεν** μπορούμε να χρησιμοποιήσουμε την αποθήκευση σε ενδιάμεση μνήμη, γιατί;

Χαρακτηριστικά της κυκλοφορίας

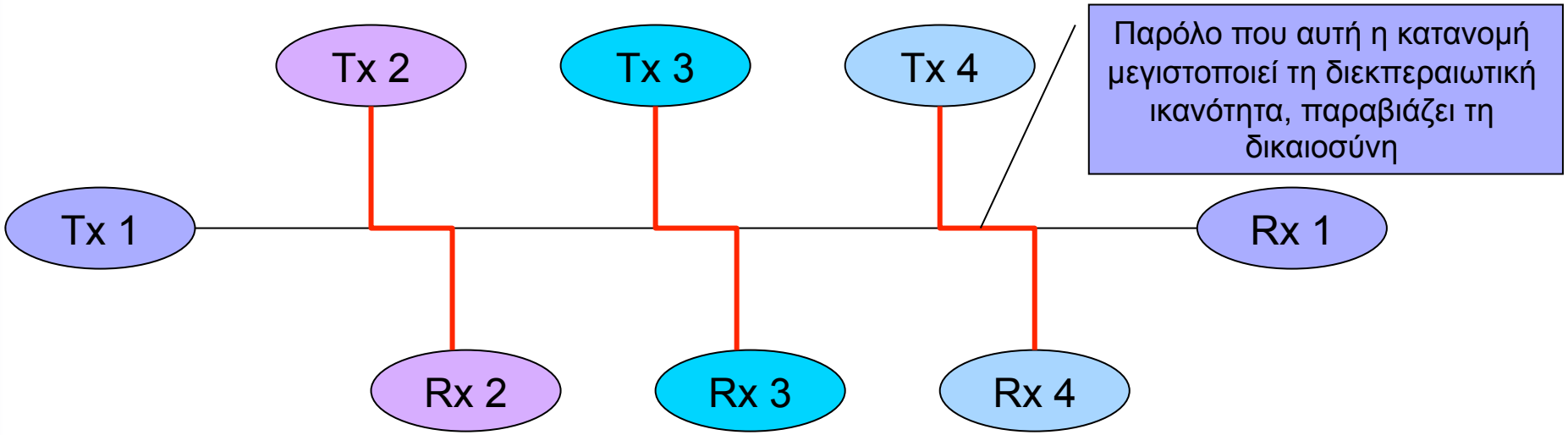
- Κάθε εφαρμογή δημιουργεί πακέτα με το δικό της τρόπο.
  - Σταθερό ρυθμό (constant bit rate)
    - Π.χ., τηλεφωνία
  - Μεταβαλλόμενο ρυθμό (variable bit rate)
    - Π.χ., video
  - Σταθερό ή μεταβαλλόμενο ρυθμό ανάλογα με το διαθέσιμο εύρος ζώνης (available bit rate)
- Κωδικοποίηση των πληροφοριών παίζει επίσης σημαντικό ρόλο
  - Π.χ., συμπιεσμένα αρχεία (compressed documents)
  - Οι αλγόριθμοι συμπίεσης μπορούν να διαχωρίσουν τα πακέτα σε περισσότερο ή λιγότερο σημαντικά.

Μετρικές Απόδοσης

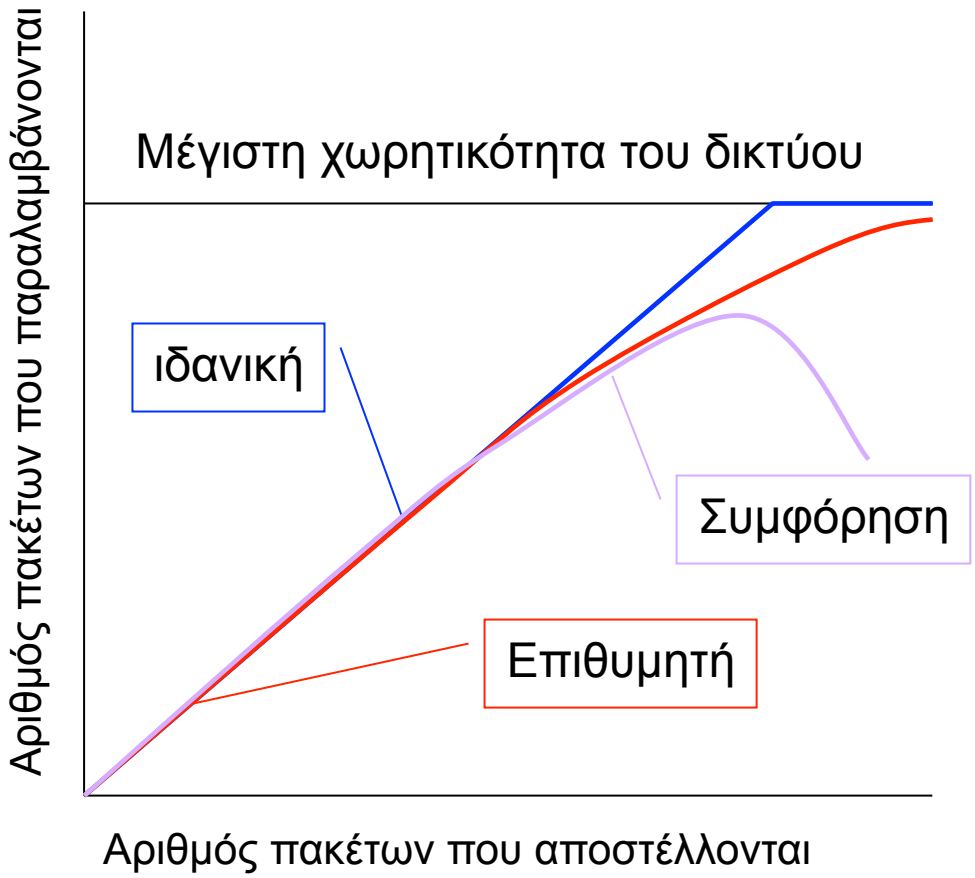
- Αξιοπιστία (reliability) και απώλεια πακέτων (packet loss)
- Καθυστέρηση (delay)
  - Καθυστέρηση στην παραλαβή των πακέτων
  - Διακύμανση στην καθυστέρηση (jitter)
- Εύρος ζώνης (Bandwidth)
- Δικαιοσύνη (fairness)
  - Όλοι οι χρήστες θα πρέπει να μπορούν να χρησιμοποιήσουν το δίκτυο.
- Διεκπεραίωτική ικανότητα (throughput)

Αντιφατικές Μετρικές Απόδοσης

- Για την αύξηση της διεκπεραιωτικής ικανότητας πρέπει να υπάρχει αυξημένη κίνηση που όμως συνεπάγεται περισσότερες απώλειες πακέτων
- Για να περιοριστούν οι απώλειες χρειάζεται μεγάλη ενδιάμεση μνήμη που όμως αυξάνει την καθυστέρηση
- Μεγιστοποίηση της διεκπεραιωτικής ικανότητας δυο δικτύου μπορεί να παραβιάζει τη δικαιοσύνη



Συμφόρηση



- Μεγάλος αριθμός πακέτων συναγωνίζονται για τη χρήση περιορισμένου αριθμού πόρων.
- Μεγάλη ενδιάμεση μνήμη μπορεί να μεγαλώσει το πρόβλημα λόγω πιθανών αχρείαστων επαναμεταδόσεων
- Εναλλακτική δρομολόγηση μπορεί να βοηθήσει αλλά δεν λύνει το πρόβλημα.

Έλεγχος Συμφόρησης

- Έλεγχος ανοιχτού βρόγχου (open loop)
  - Το δίκτυο σχεδιάζεται συντηρητικά με περισσότερους πόρους απ' όσους χρειάζεται.
  - Αυξημένο κόστος και μειωμένη χρήση των πόρων
- Κλειστού βρόγχου (closed loop)
  - Παρακολουθούμε το δίκτυο και ανιχνεύουμε την εμφάνιση συμφόρησης
    - Αυξημένη καθυστέρηση, αυξημένες απώλειες, μειωμένο throughout, αυξημένες ουρές, αυξημένη διακύμανση καθυστέρησης.
  - Η πληροφορία για τη συμφόρηση μεταδίδεται σε όλους τους κόμβους που έχουν τη δυνατότητα να περιορίσουν τη συμφόρηση.
    - Ο κόμβος ενημερώνει απευθείας τη πηγή ή τον παραλήπτη ο οποίος ενημερώνει στη συνέχεια τη πηγή.
  - Οι παράμετροι του δικτύου μεταβάλλονται έτσι ώστε να περιοριστεί η συμφόρηση.
    - Μείωση του φορτίου, εναλλακτικά μονοπάτια.

Έλεγχος Συμφόρησης

- Χαρακτηρισμός των μεθόδων αποσυμφόρησης
  - Μέθοδοι βασισμένες στους χρήστες (hosts) ή βασισμένες στους δρομολογητές
  - Μέθοδοι βασισμένες σε προκρατήσεις πόρων (reservation) ή μέθοδοι βασισμένες σε ανάδραση .
  - Έλεγχος παραθύρου ή απευθείας έλεγχος του ρυθμού μετάδοσης πηγής.
- Πώς θα χαρακτηρίζεται τον αλγόριθμο αποσυμφόρησης του TCP;

Πολιτικές Επιπέδων και Έλεγχος Συμφόρησης

- Επίπεδο Μεταφοράς
  - Πολιτική επαναμετάδοσης
  - Πακέτα εκτός σειράς
  - Επαληθεύσεις
  - Έλεγχος ροής
  - Υπολογισμός λήξης χρονομέτρων
- Επίπεδο Δικτύου
  - Εικονικά κυκλώματα ή δίκτυα πακέτων
  - Πολιτικές αποθήκευσης μετάδοσης (buffering and scheduling)
  - Απώλεια πακέτων
  - Αλγόριθμος δρομολόγησης
  - Μέγιστη ζωή πακέτων στο δίκτυο
- Επίπεδο ζεύξης δεδομένων
  - Πολιτική επαναμετάδοσης
  - Πακέτα εκτός σειράς
  - Επαληθεύσεις
  - Έλεγχος ροής

Έλεγχος Συμφόρησης σε Εικονικά Δίκτυα

- Έλεγχος πρόσβαση (admission control)
  - Ο χρήστης «χαρακτηρίζει» τη κίνηση που θα δημιουργεί και ζητά σύνδεση από το δίκτυο.
  - Το δίκτυο ελέγχει κατά πόσο έχει τους πόρους για να παράσχει τις αιτούμενες υπηρεσίες τόσο στο νέο χρήστη όσο και στους υφιστάμενους χρήστες.
    - Εάν ναι, τότε δέχεται το νέο χρήση
    - Εάν όχι, απορρίπτει τη νέα σύνδεση.
    - Π.χ., τηλεφωνικό δίκτυο
- Πώς το δίκτυο αποφασίζει αν έχει αρκετούς πόρους;
- Δρομολόγηση του εικονικού κυκλώματος έτσι ώστε να αποφεύγει σημεία τα οποία υποφέρουν από συμφόρηση.

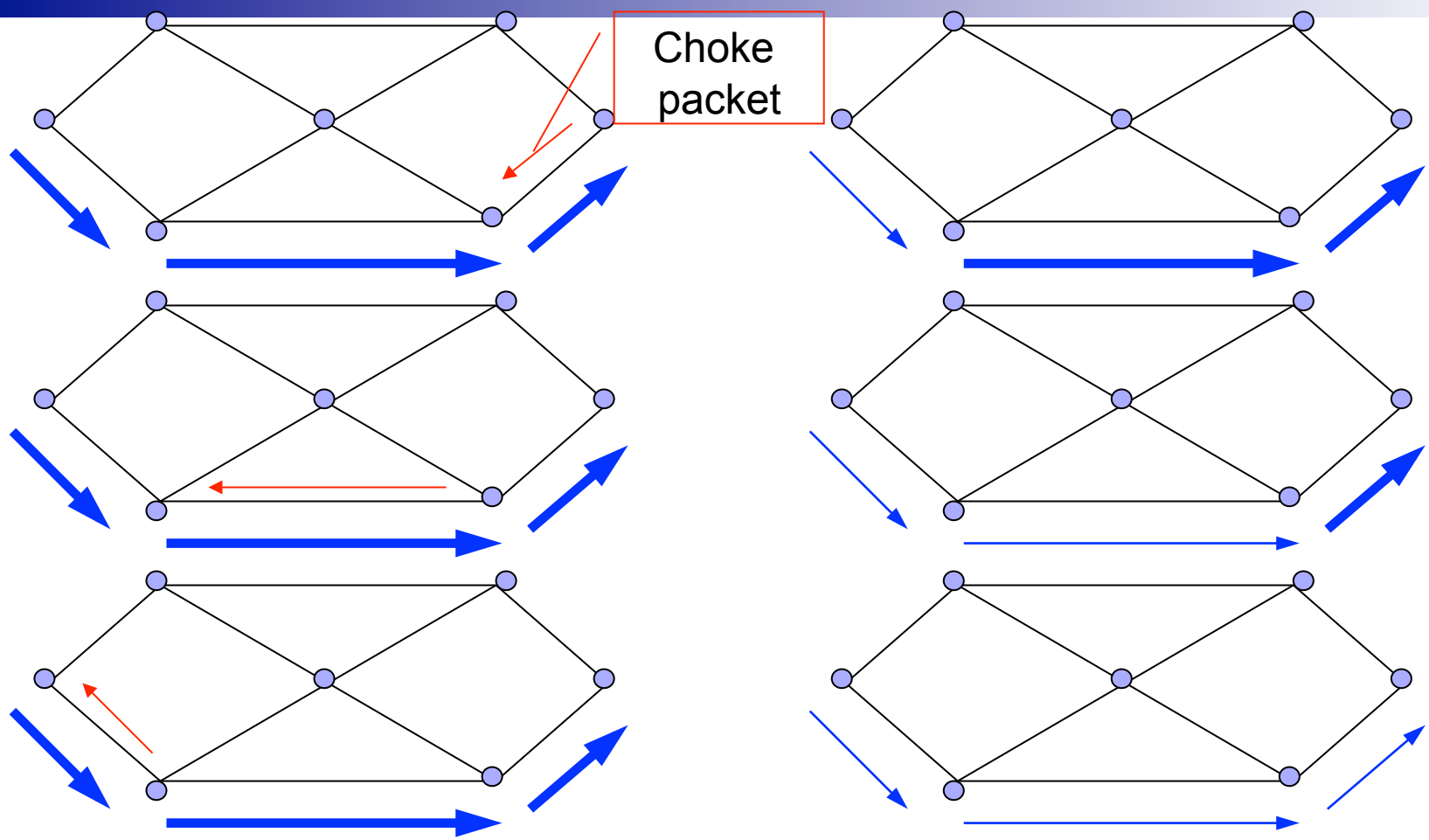
Έλεγχος Συμφόρησης σε Δίκτυα Πακέτων (datagram)

- Μέτρηση της χρήσης ($0 \le u \le 1$) κάθε ζεύξης, f={0, 1}

$$u_{new} = au_{old} + (1-a)f$$

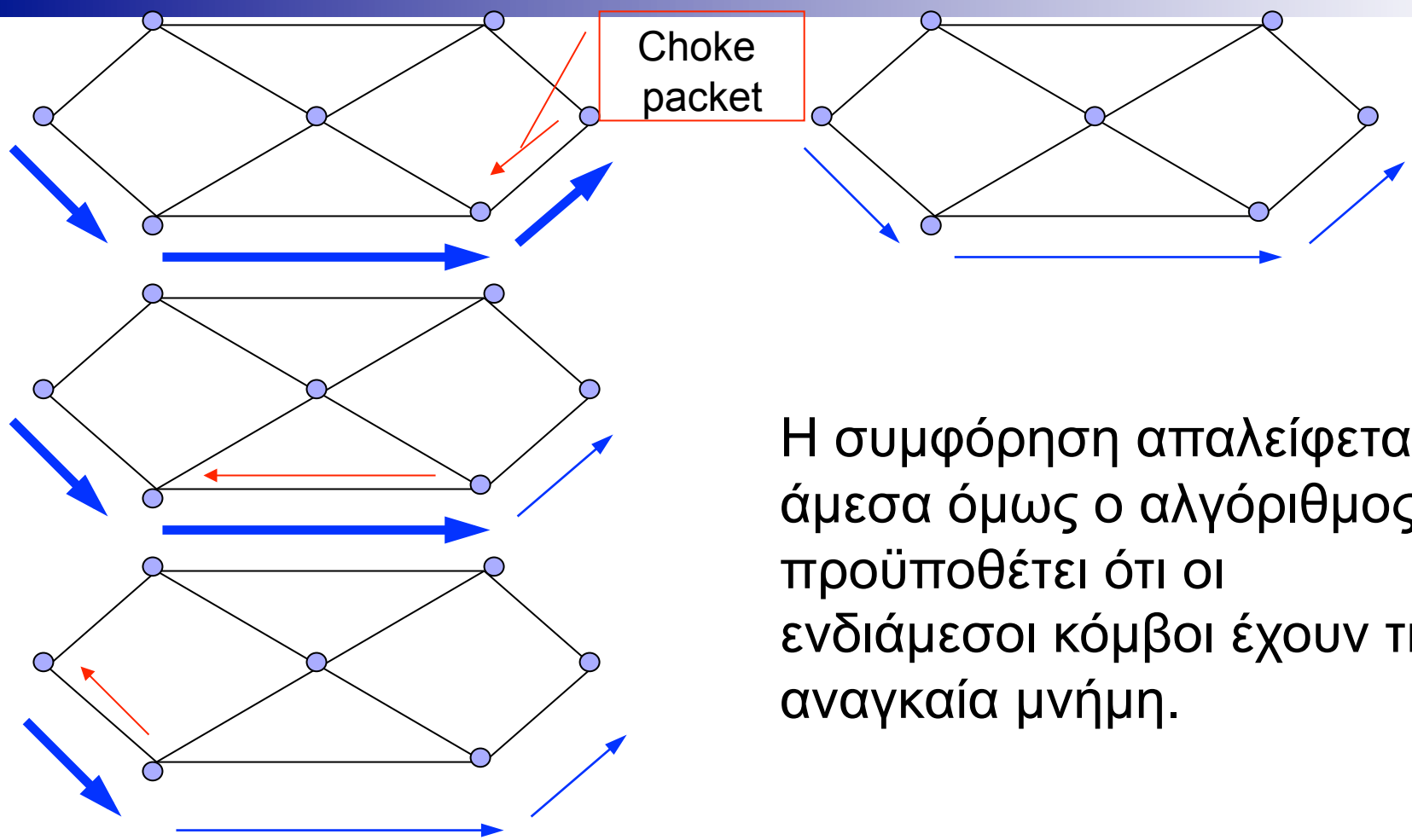
- Εάν η χρήση υπερβεί κάποιο όριο u_{max} τότε ο δρομολογητής μπορεί να λάβει κάποια μέτρα για να αποφύγει τη συμφόρηση.
- Προειδοποιητικό bit
- Αποστολή choke packets
  - Σε τέτοια περίπτωση, ο δρομολογητής αποστέλλει ένα πακέτο τη πηγή η οποία μόλις το παραλάβει περιορίζει το ρυθμό μετάδοσης.
    - Choke packets to source
    - Hop-by-hop choke packets

Choke Packets



Πόσος χρόνος πέρνα μέχρι να απαλειφθεί η συμφόρηση; RTT

Hop-by-hop Choke Packets



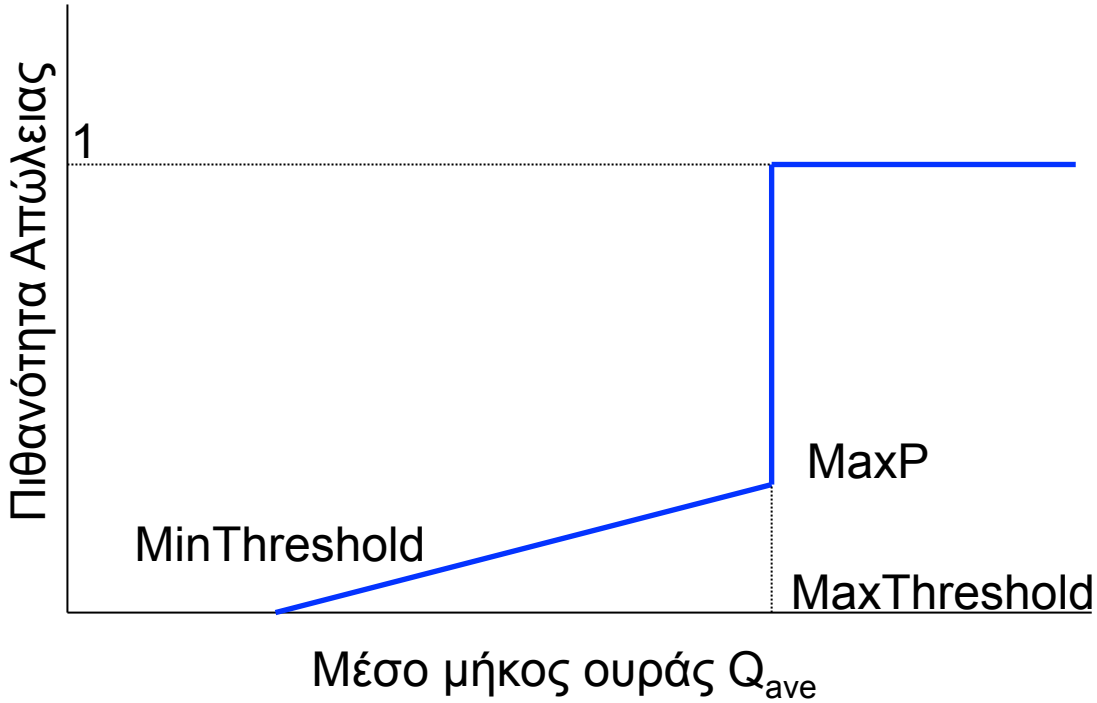
Η συμφόρηση απαλείφεται άμεσα όμως ο αλγόριθμος προϋποθέτει ότι οι ενδιάμεσοι κόμβοι έχουν την αναγκαία μνήμη.

Απόρριψη Φορτίου

- Όταν ένας δρομολογητής υποφέρει από σοβαρή συμφόρηση τότε αρχίζει να χάνει πακέτα στην τύχη. Στην πράξη όμως πιθανόν ένας δρομολογητής να μπορεί να επιλέξει πια πακέτα να απορρίψει έτσι ώστε να ελαχιστοποιηθεί κάποιο κόστος
  - Σε μια εφαρμογή μεταφοράς αρχείων, εάν απορριφθεί το πακέτο 12, τότε όλα τα επόμενα θα πρέπει να ξανασταλθούν…
  - Σε μια εφαρμογή πολυμέσων είναι ίσως καλύτερα να απολεσθούν παλαιότερα πακέτα παρά νέα πακέτα.
- Πώς όμως ο δρομολογητής θα ξεχωρίσει πια πακέτα να απορρίψει;
- Χρειάζεται βοήθεια από τον χρήστη ο οποίος πρέπει να σημαδέψει πια πακέτα είναι πιο σημαντικά και πια πακέτα λιγότερο σημαντικά.
  - Πρόβλημα: ένας χρήστης θα σημαδεύει όλα τα πακέτα του ως άκρως σημαντικά…
  - Κοστολόγηση (pricing)…

Random Early Detection (RED)

- Μόλις ανιχνευθεί ότι υπάρχει περίπτωση συμφόρησης, τότε ο δρομολογητής αρχίζει να χάνει πακέτα στην τύχη, πριν ακόμα εμφανιστεί το πρόβλημα της συμφόρησης

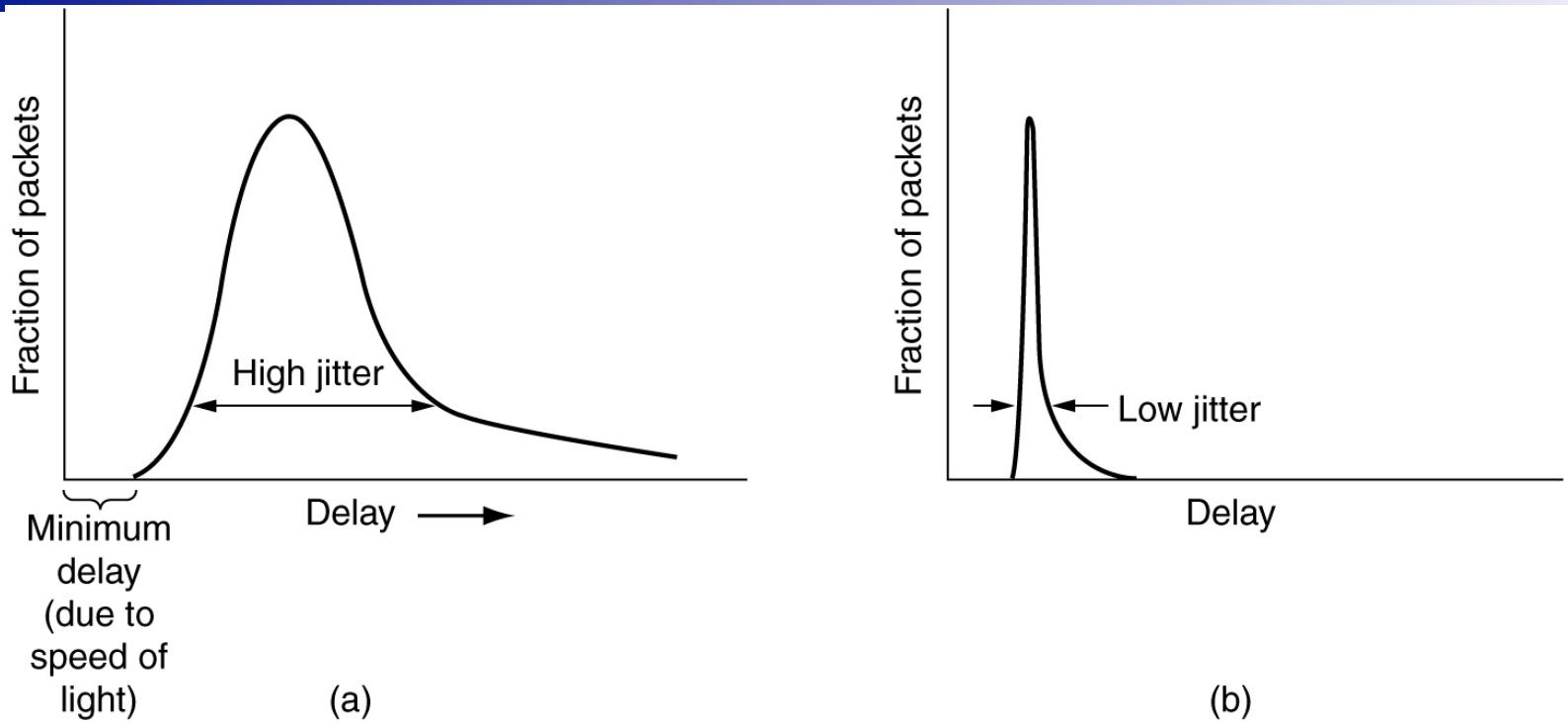


$$Q_{ave} = wQ_{ave} + (1-w)q$$

συντελεστής

Νέα μέτρηση

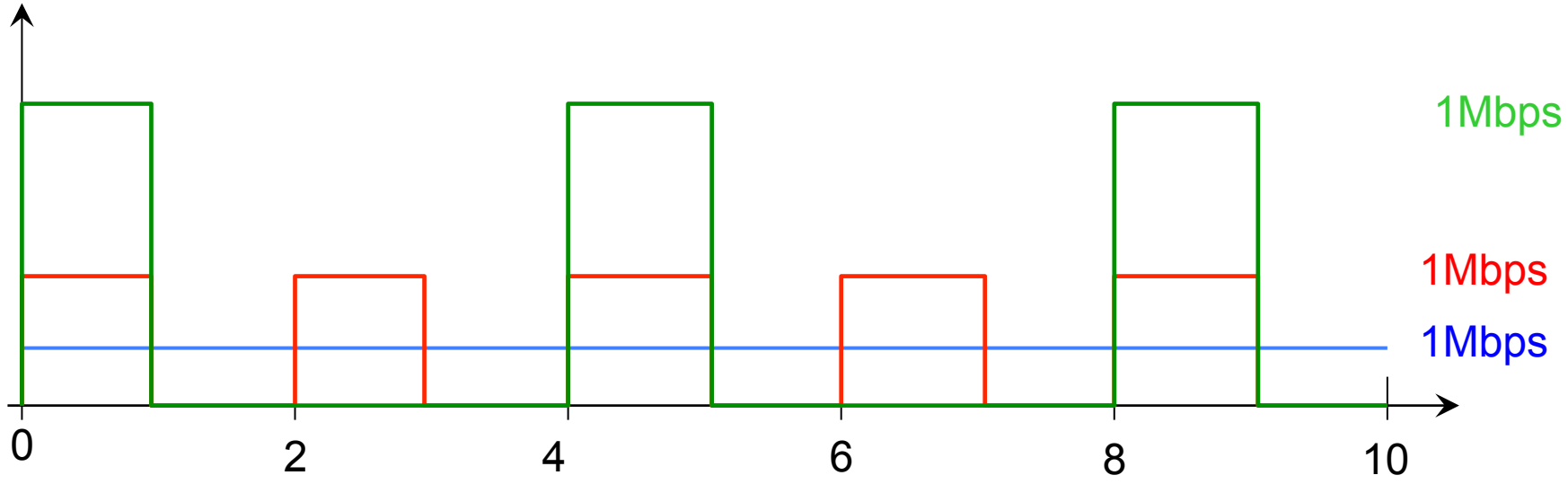
Έλεγχος Διακύμανσης Καθυστέρησης



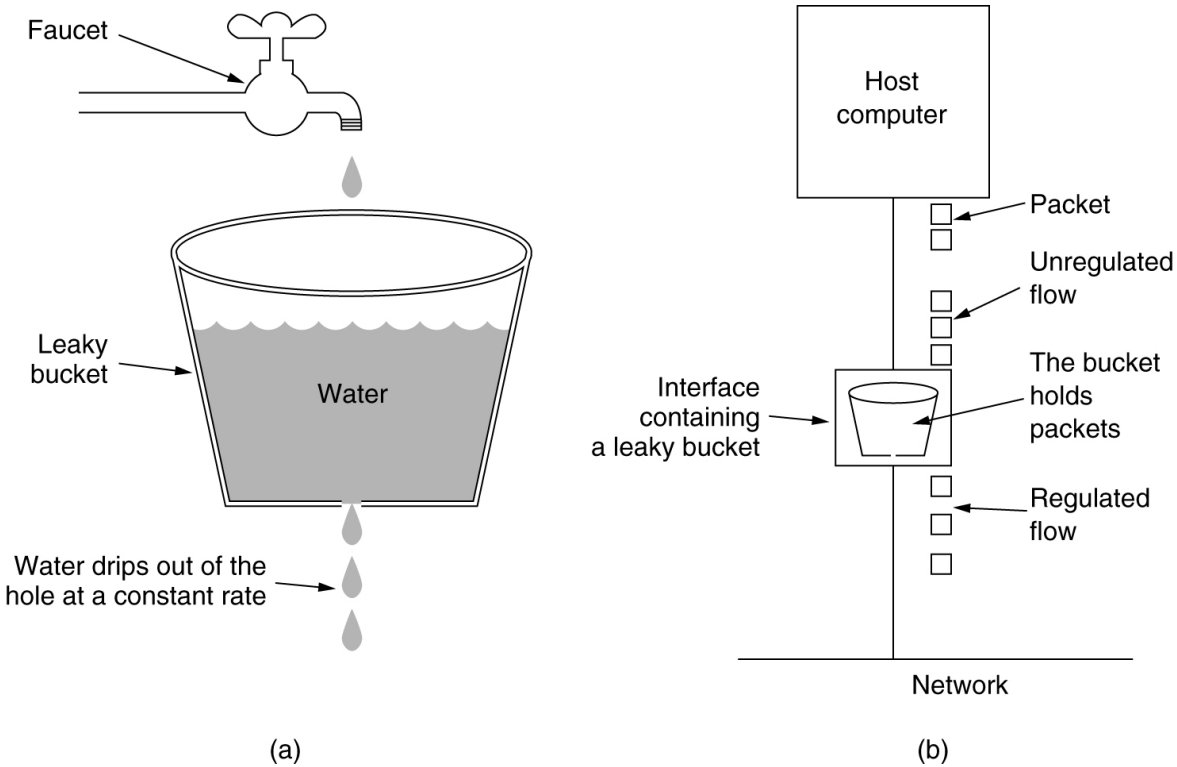
- Ένας δρομολογητής μπορεί να κατηγοριοποιήσει τα πακέτα με βάση την ώρα που πρέπει να παραδοθούν
  - Πακέτα που έχουν αρκετό χρόνο
  - Πακέτα που έχουν καθυστερήσει κάπως
  - Πακέτα που έχουν καθυστερήσει υπερβολικά

Μορφοποίηση Κυκλοφορίας (Traffic Shaping)

- Εάν όλες οι πηγές στέλνουν πακέτα με σταθερό ρυθμό, τότε το πρόβλημα της κατανομής πόρων και αποσυμφόρησης είναι γενικά εύκολα!
- Μη ομοιομορφία της κυκλοφορίας (burstiness) κάνει το πρόβλημα της αποσυμφόρησης και της κατανομής πόρων πολύ δύσκολα προβλήματα.
- Η παροχή ποιότητας υπηρεσίας είναι δύσκολο πρόβλημα όταν η κάθε πηγή μπορεί να στείλει όσο γρήγορα θέλει.

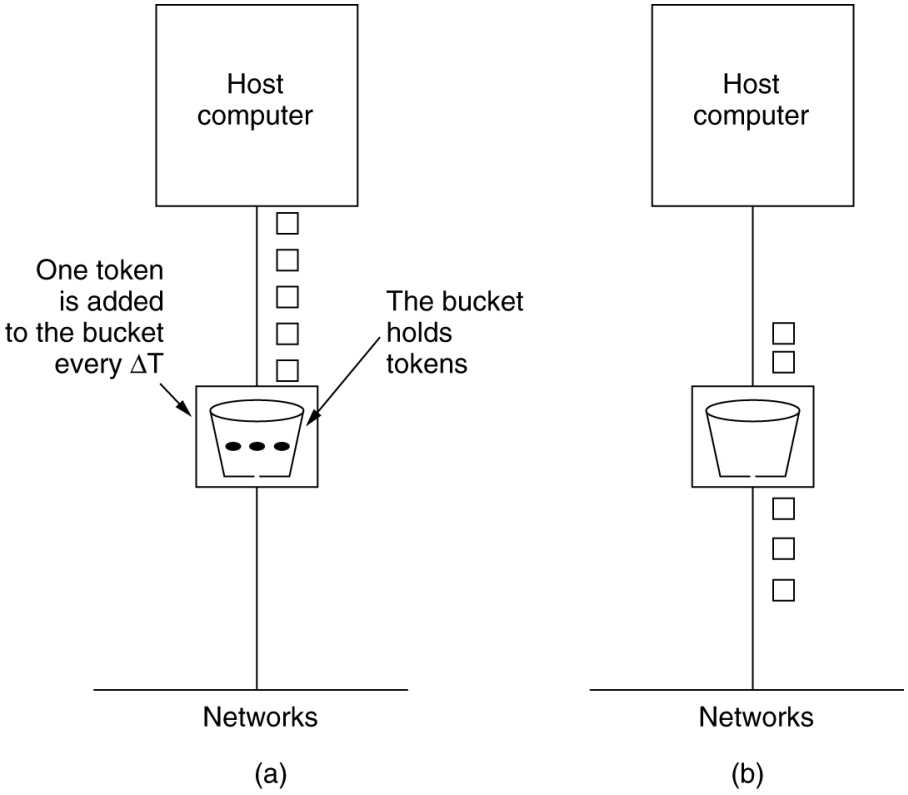


Leaky Bucket Algorithm



- Η πηγή δημιουργεί πακέτα με ακανόνιστο ρυθμό, αλλά βγαίνουν από αυτό το μηχανισμό με σταθερό ρυθμό.
- Εάν η μνήμη υπερχειλίσει, τότε τα πακέτα χάνονται.

Token Bucket Algorithm

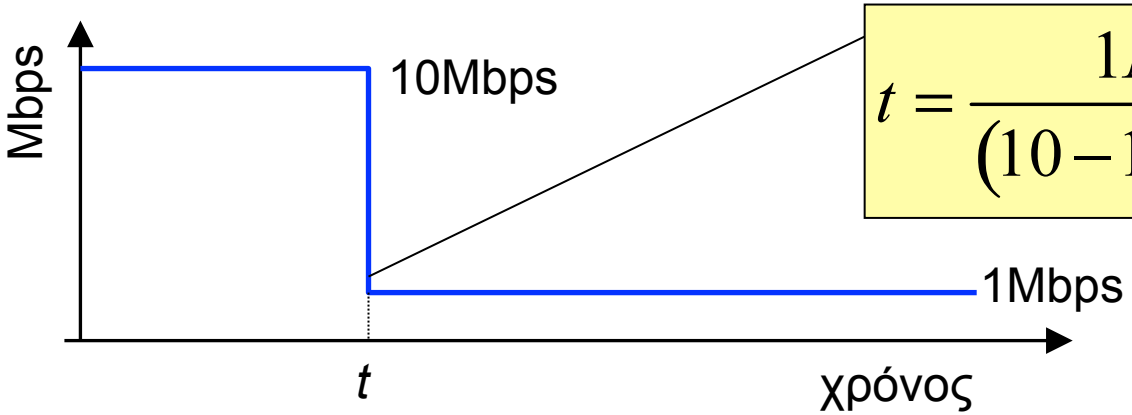


- Ένα πακέτο μεταδίδεται εφόσον υπάρχει σκυτάλη στον κουβά με τις σκυτάλες.
- Κάθε πακέτο που μεταδίδεται, καταστρέφει μία σκυτάλη.
- Σκυτάλες παράγονται με ρυθμό *r* σκυτάλες το δευτερόλεπτο
- Ο κουβάς μπορεί να αποθηκεύσει μέχρι *b* σκυτάλες
- Σε μια περίοδο *t*, η πηγή δεν μπορεί να αποστείλει περισσότερα πακέτα από

$$b + rt$$

Token Bucket Algorithm

- Υποθέτουμε πως ο μέγιστος ρυθμός μετάδοσης μιας γραμμής είναι 10Mbps. Επίσης υποθέτουμε πως μια πηγή ελέγχεται από τον token bucket algorithm ο οποίος για κάθε byte που μεταδίδει πρέπει να έχει μία σκυτάλη και έχει επίσης τις εξής παραμέτρους
  - Ρυθμός δημιουργίας σκυταλών *r* = 1Mbps
  - Μέγιστη χωρητικότητα του κουβά: b= 1Mb
- Για κάθε χρονική στιγμή, υπολογίστε το ρυθμό μετάδοσης στην έξοδο του token bucket υποθέτοντας πως η πηγή έχει άπειρα πακέτα να στείλει και ότι αρχικά ο κουβάς είναι γεμάτος.



$$t = \frac{1Mb}{(10-1)Mbps} = \frac{1}{9} = 1.11\sec$$

Παροχή Ποιότητα Υπηρεσίας (Quality of Service (QoS) Provisioning)

- Καθοριστικές και στατιστικές εγγυήσεις (deterministic or stochastic guarantees)
- Σχεδιασμός για τη χειρότερη δυνατή περίπτωση
- Ολοκληρωμένες Υπηρεσίες (Integrated Services)
- Διαφοροποιημένες Υπηρεσίες (Differentiated Services)
- Πολιτικές σχηματισμού ουράς και χρονοπρογραμματισμός (packet queueing and scheduling)

Καθοριστικές και Στατιστικές Εγγυήσεις

- Ας υποθέσουμε πως έχουμε *N* πηγές οι οποίες μοιράζονται μία ζεύξη και σε κάθε χρονική περίοδο είτε μεταδίδουν με πιθανότητα *p* και ρυθμό 1Mbps, είτε με πιθανότητα *q*=1-*p* δεν μεταδίδουν. Πως θα υπολογίζεται τη χωρητικότητα της ζεύξης;
- Με καθοριστικές εγγυήσεις η ζεύξη θα πρέπει να έχει χωρητικότητα τουλάχιστον *N* Mbps
- Με στατιστικές εγγυήσεις, η πιθανότητα να έχουμε *k* πηγές να προσπαθούν να μεταδώσουν ταυτόχρονα είναι

$$\Pr(k) = \binom{N}{k} p^k (1-p)^{N-k} = \frac{N!}{(N-k)!k!} p^k (1-p)^{N-k}$$

- Μέση χρήση (utilization) της ζεύξης η οποία έχει χωρητικότητα C Mbps

$$U = \frac{1}{C}\sum_{k=0}^{N} k \Pr(k) = \frac{1}{C}\sum_{k=0}^{N} k \frac{N!}{(N-k)!k!} p^k (1-p)^{N-k} = \frac{Np}{C}$$

- Πιθανότητα κάποια πηγή να μην έχει καθόλου εύρος ζώνης:

$$\Pr(F) = \sum_{k=C+1}^{N} \Pr(k)$$

Καθοριστικές και Στατιστικές Εγγυήσεις

- Ας υποθέσουμε πως N=100 και p=0.1, q=0.9
- Με καθοριστικές εγγυήσεις η ζεύξη πρέπει να έχει χωρητικότητα N=100 Mbp

Χωρητικότητα Ζεύξης C	Χρήση U	Πιθανότητα Pr[F]
100	0.1	0
80	0.125	2 [illegible] 10^{-62}
60	0.17	2 [illegible] 10^{-35}
40	0.25	5 [illegible] 10^{-16}
20	0.5	8 [illegible] 10^{-4}
10	1	0.42

Σχεδιασμός για τη Χειρότερη Δυνατή Περίπτωση (Worst Case Design)

- Εάν το δίκτυο έχει περισσότερους πόρους απ' όσους χρειάζεται, τότε μπορεί να εγγυηθεί την παροχή ποιότητας υπηρεσιών
  - Π.χ., το τηλεφωνικό δίκτυο
    - Εφόσον κάποιος πάρει γραμμή τότε η ποιότητα της φωνής είναι πάντα πολύ καλή
    - Η πιθανότητα να μην βρίσκει κάποιος γραμμή είναι σχετικά μικρή
    - Παρόλο του το τηλεφωνικό δίκτυο έχει σχεδιαστεί συντηρητικά, η πιθανότητα μη εύρεσης γραμμής είναι μεν μικρή, αλλά όχι μηδέν!
- Το κόστος ενός τέτοιου δικτύου είναι πολύ ψηλό.
- Σε ένα τέτοιο δίκτυο το ποσοστό χρήση (utilization) είναι πολύ μικρό.

Προκράτηση Πόρων (Resource Reservation)

- Οι χρήστες προκρατούν τους αναγκαίους πόρους για τη διάρκεια της επικοινωνίας
- Αφού τελειώσει η επικοινωνίας, τότε ελευθερώνουν τους πόρους έτσι ώστε να μπορούν άλλοι χρήστες να τους χρησιμοποιήσουν.
- Για την προκράτηση πόρων χρειάζεται δίκτυο με μηχανισμούς παρόμοιούς με τα εικονικά κυκλώματα.
- Εάν δεν υπάρχουν αρκετοί πόροι τότε η νέα συνδιάλεξη δεν γίνεται αποδεκτή (admission control).
  - Πόροι: Εύρος ζώνης, μνήμη και υπολογιστική ικανότητα (CPU)
  - Πως υπολογίζουμε κατά πόσο το δίκτυο έχει αρκετούς πόρους για να μεταφέρει τη νέα συνδιάλεξη;
    - Καθοριστικές και στατιστικές εγγυήσεις.
    - Ελαχιστοποίηση της καθυστέρησης ή της απώλειας πακέτων
    - Μεγιστοποίηση του throughput

Προκράτηση Πόρων (Resource Reservation)

- Υποθέτουμε πως ένας δρομολογητή χρειάζεται κατά μέσο όρο $1/\mu$ =1 μsec για να επεξεργαστεί κάθε πακέτο.
- Επίσης υποθέτουμε πώς τα πακέτα φτάνουν με ρυθμό λ=900,000 πακέτα το δευτερόλεπτο (Poisson)
- Πια η μέση καθυστέρηση του κάθε πακέτου;
  - Τα πακέτα φτάνουν τυχαία με αποτέλεσμα πολλές φορές να μην υπάρχουν πακέτα έτσι κάποια από την διεκπεραιωτική ικανότητα του επεξεργαστή να χάνεται (idle) και πολλές φορές να φτάνουν περισσότερα από ένα με αποτέλεσμα κάποια να περιμένουν στην ουρά

$$T = \frac{1}{\mu} \times \frac{1}{1 - \lambda / \mu} = 10 \mu Sec$$

- Το δίκτυο θα μπορούσε να προβλέψει πολύ καλύτερα τις απαιτήσεις από το δίκτυο εάν όλες οι πηγές είχαν σταθερό ρυθμό μετάδοσης!

Συμφωνία Επιπέδου Ποιότητας (Service Level Agreement)

- Όταν ένας χρήστης θέλει να χρησιμοποιήσει το δίκτυο διαπραγματεύεται το επίπεδο της αναμενόμενης ποιότητας υπηρεσίας.
- Η πηγή πρέπει να χαρακτηρίσει την κυκλοφορία (traffic) την οποία θα δημιουργεί. Χρησιμοποιεί κάποιο από τους αλγορίθμους μορφοποίησης
  - Leaky bucket algorithm
  - Token bucket algorithm

- Παράμετροι κυκλοφορίας:

Παράμετρος	**Μονάδες**
Token bucket rate *r*	Bytes/sec
Token bucket size *b*	Bytes
Peak data rate	Bytes/sec
Minimum packet size	Bytes
Maximum packet size	Bytes

Συμφωνία Επιπέδου Ποιότητας (Service Level Agreement)

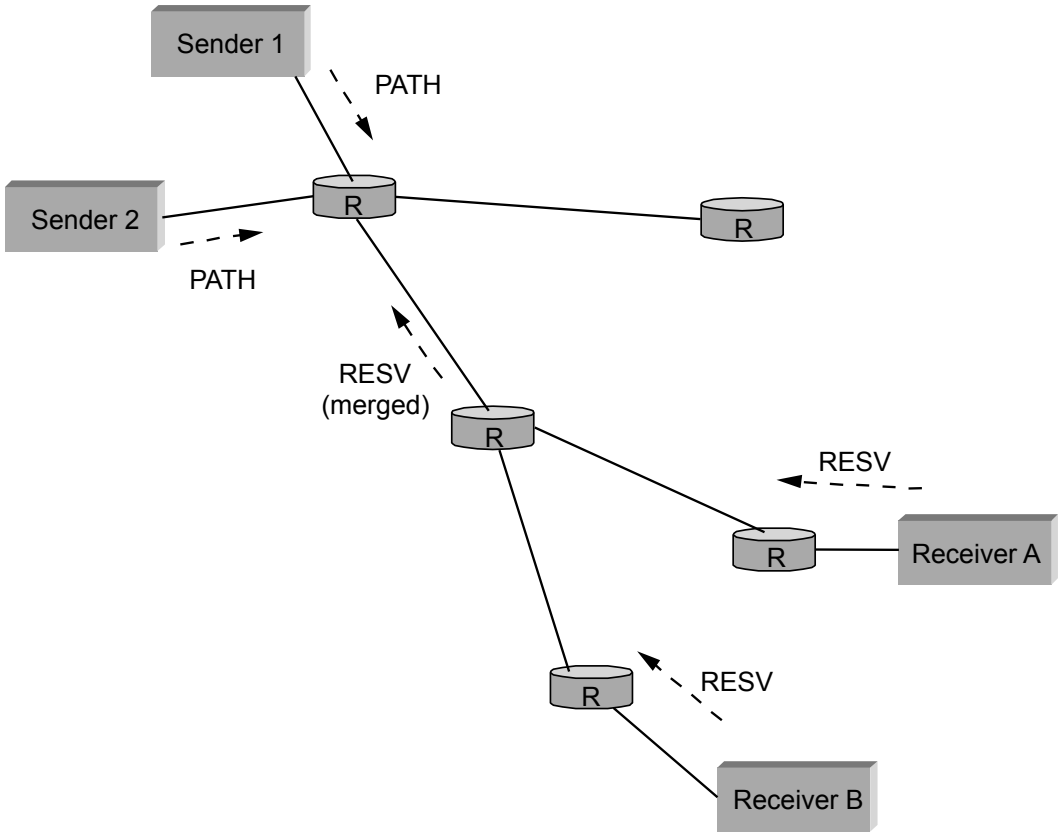
- Το μήνυμα με το αίτημα για προκράτηση πόρων περνά από όλους του κόμβους οι οποίοι αποφασίζουν κατά πόσο έχουν τους απαιτούμενους πόρους
  - Εάν όλοι οι κόμβοι έχουν τους απαραίτητους πόρους, η σύνδεση γίνεται δεκτή
  - Εάν όχι, η σύνδεση απορρίπτεται.
- Το πρόβλημα δεν σταματά εδώ
  - Τι θα συμβεί αν κάποιος κόμβος ζητήσει πολύ περισσότερους πόρους απ' όσους χρειάζεται για να είναι σίγουρος ότι θα απολαμβάνει υψηλή ποιότητα υπηρεσίας
  - Τι θα συμβεί εάν κάποιος αρχίσει να στέλνει με ρυθμό μεγαλύτερο από αυτόν που όρισε αρχικά.

Ολοκληρωμένες Υπηρεσίες (Integrated Services)

- Αρχιτεκτονική στη οποία κάθε σύνδεση προκρατεί τους πόρους που χρειάζεται σε κάθε κόμβο.
- Χρησιμοποιεί το Πρωτόκολλο Προκράτησης (Resource reSerVation protocol RSVP)
  - Το πρωτόκολλο υποστηρίζει πολλαπλούς δέκτες
  - Οι προκρατήσεις ξεκινούν από το δέκτη (receiver based).
  - Το πρωτόκολλο χρησιμοποιεί δύο είδη μηνυμάτων PATH και RESV.
  - Κάθε κόμβος διατηρεί την κατάσταση και κράτηση για κάθε ροή ξεχωριστά
    - Περιοδικά (περίπου κάθε 30 δευτερόλεπτα) οι εμπλεκόμενοι κόμβοι ανταλλάσσουν τα μηνύματα PATH και RESV.

Πρωτόκολλο RSVP

- Ο αποστολέας στέλνει το PATH στο οποίο χαρακτηρίζει την κυκλοφορία την οποία θα δημιουργεί
- Ο παραλήπτης στέλνει το RESV το οποίο χρησιμοποιείται για να γίνουν οι προκρατήσεις σε κάθε ενδιάμεσο κόμβο.
- Τα πακέτα RESV δρομολογούνται χρησιμοποιώντας το αλγόριθμο αντίστροφου μονοπατιού (Reverse Path Algorithm)



Διαφοροποιημένες Υπηρεσίες (Differentiated Services)

- Οι Ολοκληρωμένες υπηρεσίες είναι δύσκολο να εφαρμοστούν όταν το δίκτυο είναι μεγάλο (scalability problem)
- Differentiated Services: Αρχιτεκτονική στη οποία υπάρχει ένας περιορισμένος αριθμός ομάδων οι οποίες έχουν παρόμοιες απαιτήσεις.
- Όταν υπάρξει αίτημα για νέα σύνδεση, αυτή κατηγοριοποιείται, και όλα της τα πακέτα σημαδεύονται ανάλογα.
- Σε όλους τους ενδιάμεσους κόμβους τα πακέτα λαμβάνουν υπηρεσίες ανάλογες με την κατηγορία στην οποία ανήκουν λύνοντας έτσι το πρόβλημα της επεκτασιμότητας (scalability)
- Όμως δεν μπορούν να δοθούν εγγυήσεις σε συγκεκριμένη ροή (δίνονται εγγυήσεις μόνο για μια κατηγορία)

Διαφοροποιημένες Υπηρεσίες (Differentiated Services)

- Η αρχιτεκτονική ορίζει τις ακόλουθες κατηγορίες
  - Expedited Forwarding (EF)
    - Υπηρεσία υψηλής ποιότητας με χαμηλή καθυστέρηση και μικρή πιθανότητα απώλειας πακέτων.
  - Assured Forwarding (AF)
    - 4 διαφορετικές κλάσεις AF
    - Κάθε μια από τις κλάσεις υποδιαιρείται σε τρεις υποκατηγορίες η κάθε μια με διαφορετική προτεραιότητα απώλειας πακέτων.
    - Σύνολο 12 διαφορετικές κατηγορίες AF.
  - Best Effort
- Τα πακέτα σημαδεύονται είτε από την εφαρμογή, είτε από κάποιο «ρυθμιστή» στην άκρη του δικτύου ο οποίος εποπτεύει ότι οι διάφορες πηγές δεν στέλνουν περισσότερα πακέτα απ’ όσα έχουν υποσχεθεί.

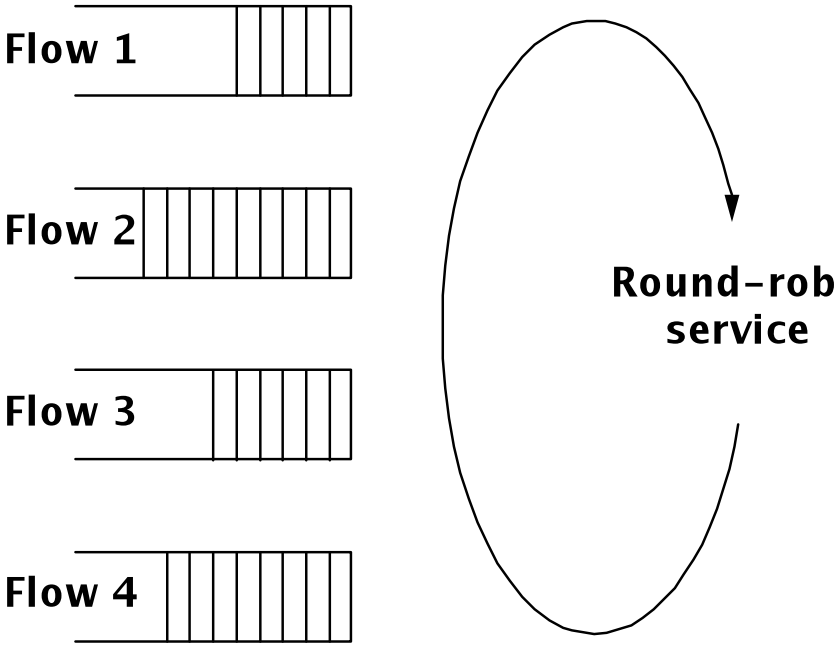
Στρατηγικές Προσωρινής Αποθήκευσης σε Ουρές

- Μία FIFO (First In First Out) ουρά
- Παράλληλες ουρές υποστηριζόμενες από Round-Robin στρατηγική
- Fair Queueing
- Weighted Fair Queueing

FIFO Queue

- Ενόσω υπάρχει διαθέσιμος χώρος στην ουρά τα πακέτα αποθηκεύονται
- Πακέτα τα οποία φτάνουν όταν η ουρά είναι γεμάτη χάνονται.
- Διάφορες FIFO ουρές μπορούν να εφαρμοστούν παράλληλα, η κάθε μια με τη δική της προτεραιότητα
  - **Πρόβλημα**: Ποιος καθορίζει την προτεραιότητα κάθε πακέτου;
- Απλή μέθοδος η οποία έχει εφαρμοστεί πολύ σε διάφορα δίκτυα.

Round-Robin



- Κάθε ουρά αντιπροσωπεύει είτε μια ροή είτε μια κλάση ροών.
- Ο εξυπηρετητής, σε κάθε βήμα επιλέγει ένα πακέτο από κάθε ουρά για να το προωθήσει
- Οι ουρές επισκέπτονται με τη σειρά
- Εάν μια ουρά δεν έχει πακέτο για να στείλει, τότε ο εξυπηρετητής πηδά στην επόμενη ουρά.
- **Πρόβλημα**: Μεγάλα πακέτα παίρνουν μεγαλύτερο μερίδιο του εύρους ζώνης!

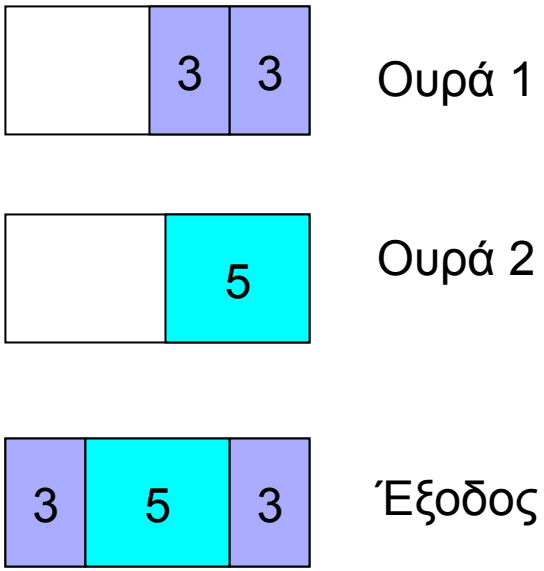
Fair Queueing

- Ο Εξυπηρετητής προσομοιώνει ένα σύστημα στο οποίο σε κάθε βήμα προωθεί ένα byte από κάθε ουρά (round-robin)
- Υπολογίζει το χρόνο στον οποίο τα τελείωνε η μετάδοση του κάθε πακέτου της κάθε ουράς
- Προωθεί το πακέτο το οποίο θα τέλειωνε πιο γρήγορα.
- Σε μια απλή ουρά, αν το πακέτο *i* φτάσει τη χρονική στιγμή A_i, και χρειάζεται S_i χρονικές μονάδες να μεταδοθεί, τότε ο χρόνος στον οποίο θα τελειώσει η μετάδοση του πακέτου F_i, δίνεται από

$$F_i = \max\{F_{i-1}, A_i\} + S_i$$

- Όταν υπάρχουν πολλαπλές ουρές, τότε πρέπει να υπολογίσουμε επίσης και το χρόνο τον οποίο θα αποστέλλονται τα bytes από τις άλλες ουρές

Fair Queueing



- Παίρνοντας ένα byte από κάθε ουρά, το πρώτο πακέτο της ουράς 1 θα τελείωνε στο χρόνο 5, ενώ το δεύτερο στο χρόνο 11. Το πακέτο της ουράς 2, θα τελείωνε στο χρόνο 10, οπόταν η σειρά με την οποία τα στέλνει το FQ φαίνεται στην «έξοδο».

Weighted Fair Queueing

- Ο αλγόριθμος Fair Queueing είναι δίκαιος όταν όλοι οι πελάτες έχουν τις ίδιες ανάγκες και πληρώνουν το ίδιο.
- Όταν θέλουμε να διαφοροποιήσουμε τις προσφερόμενες υπηρεσίες σε κάποιους χρήστες ή κάποιες κλάσεις χρηστών, τότε χρησιμοποιούμε το Weighted Fair Queueing
- Το Weighted Fair Queueing σε κάθε γύρο επιτρέπει σε μερικές ουρές να στείλουν περισσότερα από ένα bytes.
- Πόσα περισσότερα, εξαρτάται από διάφορους παράγοντες.
  - Κόστος
  - Πολιτική
  - Κλπ.