

# Δίκτυα Υπολογιστών I

Δίκτυα Μεταγωγής και Διαδίκτυα: Μέρος Δ'



Ευάγγελος Παπαπέτρου

Τμ. Μηχ. Η/Υ & Πληροφορικής, Παν. Ιωαννίνων

# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP



# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP



# Αναγκαιότητα

- Σε κάθε φυσικό δίκτυο το μέγεθος ενός μεταδιδόμενου πακέτου έχει ένα άνω όριο που καλείται **μέγιστη μονάδα μετάδοσης (maximum transmission unit, MTU)**
- Πρόβλημα 1: Κάθε εφαρμογή μπορεί να επιλέξει ελεύθερα το μέγεθος (ακόμα και  $>MTU$ ) των πακέτων που αυτή δημιουργεί
- Πρόβλημα 2: κάθε τεχνολογία δικτύου επιτρέπει διαφορετικό MTU
  - ▶ π.χ., στα δίκτυα Ethernet το MTU είναι 1500 bytes ενώ στα δίκτυα FDDI 4500 bytes
  - ▶ στο Internet ένα πακέτο μπορεί να διέλθει από φυσικά δίκτυα με διαφορετικό MTU
- Συμπέρασμα: απαιτείται ένας μηχανισμός για τη μετάδοση πακέτων με μέγεθος μεγαλύτερο από MTU



## Παράδειγμα: Κατακερματισμός στο IP (1/3)

- Το πρωτόκολλο IP προσφέρει έναν ευέλικτο μηχανισμό **κατακερματισμού (fragmentation)** και **επανασυναρμολόγησης (reassembly)** των πακέτων
  - ▶ από το αρχικό πακέτο παράγονται ένα ή περισσότερα πακέτα (**θραύσματα, fragments**) με μέγεθος ίσο ή μικρότερο του MTU
  - ▶ ο μηχανισμός ενεργοποιείται μόνο αν απαιτηθεί η μετάδοση σε ένα δίκτυο με MTU μικρότερο από το μέγεθος του πακέτου
- Ένα θραύσμα αποτελεί από μόνο του ένα **αυτόνομο πακέτο IP** και μεταδίδεται ανεξάρτητα από τα υπόλοιπα
  - ▶ όλα τα θραύσματα έχουν την ίδια τιμή στο πεδίο **αναγνωριστικό (Ident)** της κεφαλίδας
  - ▶ το αναγνωριστικό επιλέγεται από τον αποστολέα και είναι μοναδικό για όλα τα μη κατακερματισμένα πακέτα από τον ίδιο αποστολέα



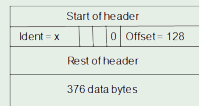
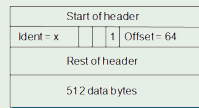
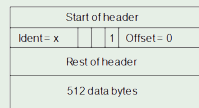
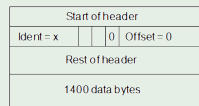
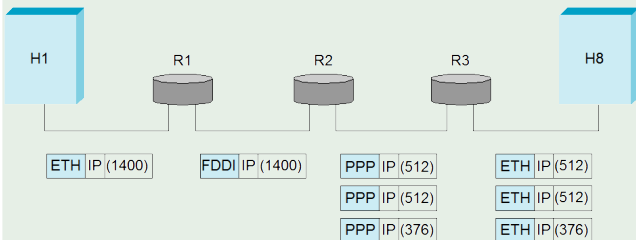
## Παράδειγμα: Κατακερματισμός στο IP (2/3)

- Σε κάθε θραύσμα:
  - ▶ το bit M στο πεδίο **σημαίες (Flags)** έχει την τιμή 1 εκτός από το τελευταίο θραύσμα
  - ▶ το πεδίο **σχετική απόσταση (Offset)** δηλώνει τη θέση των δεδομένων που μεταφέρει το θραύσμα στο αρχικό πακέτο
- Η επανασυναρμολόγηση των θραυσμάτων στο αρχικό πακέτο μπορεί να γίνει:
  - ▶ στον επόμενο δρομολογητή (**διαφανής κατακερματισμός**)
  - ▶ στον παραλήπτη υπολογιστή (**αδιαφανής κατακερματισμός**)
    - ένα θραύσμα μπορεί να υποστεί **ξανά κατακερματισμό**
    - η τεχνική αυτή χρησιμοποιείται στο Διαδίκτυο
- Η επανασυναρμολόγηση αποτυγχάνει αν κάποιο από τα θραύσματα δεν φτάσει στον παραλήπτη
  - ▶ το IP δεν επιχειρεί να ανακτήσει θραύσματα που λείπουν



# Παράδειγμα: Κατακερματισμός στο IP (3/3)

## Example



# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP





# Εισαγωγή

- Η λειτουργία ενός δικτύου μεταγωγής θέτει ένα ζήτημα κατανομής πόρων
  - ▶ πακέτα από διαφορετικούς χρήστες **ανταγωνίζονται** για τους πόρους των δρομολογητών
- Δύο προσεγγίσεις για τη διαχείριση των πόρων:
  - ▶ μεταγωγή εικονικού κυκλώματος: **κατανομή ή εκχώρηση πόρων (resource allocation)**
  - ▶ μεταγωγή πακέτου: απουσία κατανομής πόρων
- Και μια ενδιαμέση λύση σε δίκτυα με μεταγωγή πακέτου:
  - ▶ μπορούν να εφαρμοστούν πολιτικές με **συνδεσμωστρεφή λογική** με τη χρήση της έννοιας της **ασυνδεσμικής ροής (flow)**



# Προσεγγίσεις στη διαχείριση πόρων

- Απουσία κατανομής πόρων:
  - ▶ δεν απαιτείται κάποιος μηχανισμός για την κατανομή των πόρων με αποτέλεσμα να μην αποκλείεται η συμφόρηση
  - ▶ η συμφόρηση αντιμετωπίζεται μέσω μηχανισμών **ελέγχου συμφόρησης (congestion control)**
  - ▶ επιτρέπει τον υψηλό βαθμό αξιοποίησης των πόρων αλλά δεν παρέχει εγγυήσεις ποιότητας (**best effort**)
  - ▶ ευέλικτη προσέγγιση με χαμηλή πολυπλοκότητα
- Εκχώρηση πόρων:
  - ▶ απαιτείται μηχανισμός **κράτησης πόρων (resource reservation)** και αποκλείει τη συμφόρηση
  - ▶ αποτελεί την **συνδεσμостρεφή** προσέγγιση γιατί απαιτεί την διατήρηση "μόνιμων" πληροφοριών κατάστασης (**hard state**)
  - ▶ είναι προϋπόθεση για την παροχή **ποιότητας υπηρεσιών (quality of service, QoS)**
  - ▶ έχει σημαντική πολυπλοκότητα



## Ασυνδεσμικές ροές

- Οι αποφάσεις για τη διαχείριση των πόρων πρέπει να λαμβάνονται για ομάδες πακέτων που αποστέλλονται μεταξύ των ίδιων υπολογιστών
- Στα δίκτυα μεταγωγής πακέτων ένας δρομολογητής μπορεί να κατατάξει τα πακέτα σε ροές (flows)
- Ροή: μια ακολουθία πακέτων μεταξύ ενός ζεύγους προέλευσης/προορισμού που ακολουθούν το ίδιο δρομολόγιο
  - ▶ αν και τα πακέτα μεταφέρονται ανεξάρτητα, στην πράξη τις περισσότερες φορές ακολουθούν το ίδιο δρομολόγιο
- Η καταγραφή ροών απαιτεί τη διατήρηση προσωρινών πληροφοριών κατάστασης (soft state)
  - ▶ δεν απαιτείται η δημιουργία και διαγραφή των πληροφοριών με σηματοδότηση

⇒ Μια ροή μπορεί επίσης να οριστεί μεταξύ δύο διεργασιών

# Ταξινόμηση μηχανισμών διαχείρισης πόρων

- Σημαντικότερες ταξινομήσεις μηχανισμών διαχείρισης πόρων:
  - ▶ επικεντρωμένοι στους δρομολογητές vs επικεντρωμένοι στους υπολογιστές υπηρεσίας
    - οι δρομολογητές λαμβάνουν τις αποφάσεις και ενημερώνουν τους υπολογιστές υπηρεσίας vs οι υπολογιστές υπηρεσίας παρατηρούν το δίκτυο και προσαρμόζουν τη συμπεριφορά τους
  - ▶ βασισμένοι σε δεσμεύσεις (reservations) vs βασισμένοι σε ανάδραση (feedback)
    - ο αποστολέας υποβάλλει αίτημα για τη δέσμευση πόρων vs ο αποστολέας προσαρμόζει την ταχύτητα μετάδοσης ανάλογα με την ανάδραση (ρητή ή υπονοούμενη) που λαμβάνει
  - ▶ βασισμένοι σε παράθυρα (window-based) vs βασισμένοι στο ρυθμό (rate-based)
    - ο αποστολέας μπορεί να αποστείλει συγκεκριμένη ποσότητα δεδομένων που ανανεώνεται ανάλογα με την κατάσταση του δικτύου vs ο αποστολέας μπορεί να αποστείλει δεδομένα με συγκεκριμένο ρυθμό



# Παραδείγματα μηχανισμών διαχείρισης πόρων

- Μερικοί από τους σημαντικότερους μηχανισμούς που εφαρμόζονται σε δίκτυα είναι:
  - ▶ μηχανισμοί παροχής ποιότητας υπηρεσιών
    - σημαντικές τεχνικές: RSVP, differentiated services, integrated services
  - ▶ μηχανισμοί διαχείρισης ουρών
    - σημαντικές τεχνικές είναι οι FIFO (first-in-first out), FQ (fair queueing) και WFQ (weighted fair queueing)
  - ▶ μηχανισμοί ελέγχου συμφόρησης (congestion control)
    - σημαντικότεροι μηχανισμοί είναι οι DECbit και τυχαίας πρώιμης ανίχνευσης (random early detection, RED)
  - ▶ μηχανισμοί αποφυγής συμφόρησης (congestion avoidance)
    - σημαντικότερη τεχνική: αποβολή φορτίου



# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP



# Εισαγωγή

- Τα πρωτόκολλα δικτύωσης μέχρι και το επίπεδο δικτύου προδιαγράφουν την **επικοινωνία μεταξύ υπολογιστών υπηρεσίας**
- Ζήτημα: η επικοινωνία **λαμβάνει χώρα μεταξύ διεργασιών (εφαρμογών)**
  - ▶ τα δεδομένα παράγονται και καταναλώνονται από τις εφαρμογές
- Το **επίπεδο μεταφοράς (transport layer)** είναι υπεύθυνο για την **επικοινωνία των διεργασιών**

⇒ Ένα πρωτόκολλο μεταφοράς ονομάζεται και **πρωτόκολλο από άκρο σε άκρο** γιατί προδιαγράφει την επικοινωνία μεταξύ των τελικών προγραμμάτων εφαρμογών

## Ρόλος επιπέδου μεταφοράς

Ρόλος του επιπέδου μεταφοράς είναι να **μετατρέψει** την υπηρεσία παράδοσης πακέτων (δεδομένων) μεταξύ υπολογιστών υπηρεσίας (hosts), την οποία παρέχει ένα δίκτυο, σε ένα **κανάλι επικοινωνίας** μεταξύ δύο διεργασιών

## Μηχανισμοί επιπέδου μεταφοράς (1/2)

- Ένα πρωτόκολλο μεταφοράς πρέπει να προσφέρει επικοινωνία των εφαρμογών με τα **χαρακτηριστικά που αυτές επιθυμούν** όπως:
  - ▶ την εγγυημένη παράδοση των μηνυμάτων
  - ▶ την παράδοση των μηνυμάτων με τη σωστή σειρά
  - ▶ την παράδοση ενός μόνο αντιγράφου για κάθε μήνυμα
  - ▶ την παράδοση μηνυμάτων χωρίς περιορισμό μεγέθους
  - ▶ την απόρριψη μηνυμάτων με μεγάλη καθυστέρηση
  - ▶ τον έλεγχο ροής στην επικοινωνία, κα
- Η επικοινωνία των εφαρμογών θα πρέπει να είναι **ανεπηρέαστη από τους περιορισμούς του μοντέλου υπηρεσίας** του υποκείμενου δικτύου
  - ▶ το δίκτυο μπορεί να μην υποστηρίζει συγκεκριμένες υπηρεσίες, π.χ. η παράδοση των πακέτων μπορεί να μην είναι εγγυημένη





## Μηχανισμοί επιπέδου μεταφοράς (2/2)

- Ζήτημα: η ανάπτυξη μηχανισμών που θα μετατρέπουν το **μοντέλο υπηρεσίας** του υποκείμενου δικτύου στο μοντέλο υπηρεσίας που απαιτούν οι εφαρμογές
- Το επίπεδο μεταφοράς θα πρέπει επίσης να επιτρέπει την **από κοινού χρήση** των υπηρεσιών του δικτύου από πολλές διεργασίες
  - ▶ για το σκοπό αυτό απαιτείται: η **διευθυνσιοδότηση των διεργασιών (addressing)** και η **πολύπλεξη/αποπολύπλεξη των μηνυμάτων (multiplexing/demultiplexing)** διαφορετικών διεργασιών

⇒ Σε πολλές περιπτώσεις το πρόβλημα είναι η **αναξιόπιστη λειτουργία** του υποκείμενου δικτύου επομένως θεωρούμε ότι στόχος του επιπέδου μεταφοράς είναι η δημιουργία ενός **αξιόπιστου καναλιού** επικοινωνίας



## Πρωτόκολλα μεταφοράς στο Διαδίκτυο (1/2)

- Ένα πρωτόκολλο μεταφοράς πρέπει να υλοποιεί υποχρεωτικά τη **διευθυνσιοδότηση των διεργασιών** και την **πολύπλεξη/αποπολύπλεξη των δεδομένων**
- Η υλοποίηση επιπλέον μηχανισμών, που βελτιώνουν τα χαρακτηριστικά της επικοινωνίας, **είναι προεραϊκή**
  - ▶ κάθε πρωτόκολλο μεταφοράς επιλέγει τους μηχανισμούς που θα υλοποιήσει και επομένως τις υπηρεσίες που θα προσφέρει
  - ▶ η επιλογή είναι ένας συμβιβασμός μεταξύ των χαρακτηριστικών της **απαιτούμενης υπηρεσίας** και της **πολυπλοκότητας υλοποίησης**
- Ορισμένοι μηχανισμοί υλοποιούνται από τις ίδιες τις εφαρμογές ώστε να μειωθεί η **πολυπλοκότητα** και να ενισχυθεί η **ευελιξία**
  - ▶ ένα απλούστερο πρωτόκολλο μεταφοράς μπορεί να μην ικανοποιεί πλήρως αλλά να είναι συμβατό με περισσότερες εφαρμογές



## Πρωτόκολλα μεταφοράς στο Διαδίκτυο (2/2)

- Στο Διαδίκτυο δύο είναι τα σημαντικότερα πρωτόκολλα μεταφοράς
  - ▶ το πρωτόκολλο Αυτοδύναμων Πακέτων Χρήστη (User Datagram Protocol, UDP)
    - προσφέρει την απλούστερη υπηρεσία μεταφοράς δεδομένων με χαμηλή πολυπλοκότητα
  - ▶ το πρωτόκολλο Ελέγχου Μετάδοσης (Transmission Control Protocol, TCP)
    - προσφέρει αξιόπιστη επικοινωνία μεταξύ διεργασιών με αντάλλαγμα την πολυπλοκότητα υλοποίησης

⇒ Στα δίκτυα IP προδιαγράφονται και άλλα πρωτόκολλα που προσφέρουν εξειδικευμένες υπηρεσίες μεταφοράς, όπως τα Real Time Protocol (RTP), Stream Control Transmission Protocol (SCTP), κλπ



# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP



# Εισαγωγή

- Το πρωτόκολλο User Datagram Protocol (UDP) προσφέρει την απλούστερη υπηρεσία μεταφοράς δεδομένων μεταξύ διεργασιών
  - ▶ υλοποιεί μόνο την πολύπλεξη/αποπολύπλεξη δεδομένων
  - ▶ καθιστά διαθέσιμες τις αναξιόπιστες υπηρεσίες ενός δικτύου IP στις εφαρμογές
- Το UDP δεν προσφέρει κανενός είδους αξιοπιστία στη μετάδοση δεδομένων
  - ▶ οι μηχανισμοί για αξιόπιστη μετάδοση πρέπει να υλοποιηθούν από τις ίδιες τις εφαρμογές
- Το UDP χαρακτηρίζεται από χαμηλή πολυπλοκότητα και υψηλή ευελιξία



## Διευθυνσιοδότηση διεργασιών

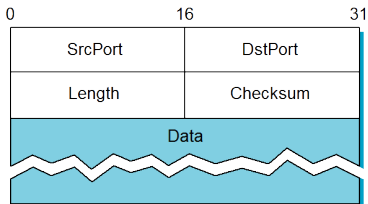
- Κάθε διεργασία σε ένα υπολογιστή υπηρεσίας προσδιορίζεται μοναδικά από έναν ακέραιο αριθμό που ονομάζεται **θύρα (port)**
  - ▶ η θύρα ονομάζεται αλλιώς **γραμματοκιβώτιο (mailbox)**: μια διεργασία μπορεί να στείλει δεδομένα σε μια θύρα και η διεργασία παραλήπτης να διαβάσει τα δεδομένα από τη συγκεκριμένη θύρα
- Οι διεργασίες σε διαφορετικούς υπολογιστές προσδιορίζονται από το ζεύγος τιμών <διεύθυνση IP υπολογιστή, θύρα>
  - ▶ αυτό το ζεύγος τιμών αποτελεί το κλειδί **αποπολύπλεξης (demultiplexing key)**

⇒ Η θύρα είναι ένας αριθμός με μήκος 16 bit, επομένως σε κάθε υπολογιστή υπηρεσίας μπορούν να προσδιοριστούν μονοσήμαντα μέχρι  $\sim 64000$  διεργασίες

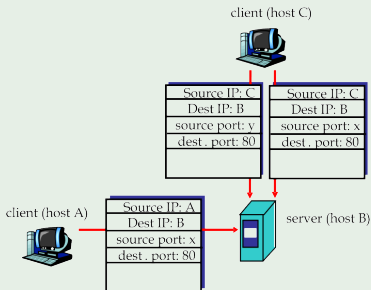


## Μετάδοση πληροφορίας

- Οι θύρες αποστολέα και παραλήπτη μεταφέρονται στην κεφαλίδα του πακέτου UDP ώστε να είναι δυνατή:
  - ▶ η παράδοση του μηνύματος στον παραλήπτη
  - ▶ η απάντηση του παραλήπτη στον αποστολέα
- Το πακέτο UDP μεταφέρει **δεδομένα μεταβλητού μεγέθους** και ενθυλακώνεται σε ένα πακέτο IP
  - ▶ το πεδίο **Μήκος (Length)** στην κεφαλίδα προσδιορίζει το μέγεθος του πακέτου UDP



### Example



## Μηχανισμοί ανάθεσης θυρών

- Προϋπόθεση για έναρξη επικοινωνίας: η διεργασία που ενεργοποιεί την επικοινωνία (**πελάτης, client**) να μάθει τη θύρα της διεργασίας αποδέκτη της επικοινωνίας (**διακομιστής, server**)
  - ▶ η διεργασία διακομιστής μπορεί να μάθει τη θύρα της διεργασίας πελάτη καθώς αυτή περιέχεται στα μηνύματα από τον πελάτη
- Υπάρχουν τρεις διαφορετικές προσεγγίσεις:
  - ▶ ο διακομιστής δέχεται μηνύματα σε μια προκαθορισμένη θύρα (**ευρέως γνωστή θύρα, well known port**)
    - υπάρχουν προκαθορισμένες θύρες για συγκεκριμένες εφαρμογές, π.χ. θύρα 25 για μηνύματα αλληλογραφίας
  - ▶ ο διακομιστής δέχεται μηνύματα σε μια ευρέως γνωστή θύρα και στη συνέχεια η επικοινωνία γίνεται σε νέα θύρα που συμφωνείται
  - ▶ μια διεργασία **αντιστοιχιστή θυρών (port mapper)** δέχεται μηνύματα σε μια μοναδική θύρα και προσδιορίζει τη θύρα επικοινωνίας ανάλογα με την εφαρμογή

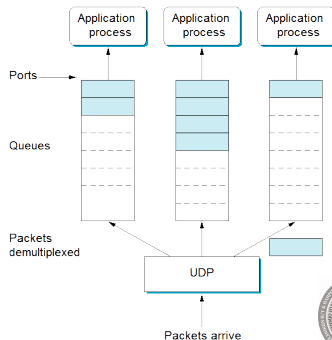




## Αποπολύπλεξη δεδομένων

- Κάθε υπολογιστής διατηρεί μια **ουρά** για κάθε διεργασία που επικοινωνεί μέσα από το δίκτυο
  - ▶ στην ουρά αποθηκεύονται τα **λαμβανόμενα πακέτα** ώστε να είναι διαθέσιμα για ανάγνωση από τη διεργασία
  - ▶ η ανάγνωση των πακέτων γίνεται με την τεχνική FIFO

- Τα λαμβανόμενα πακέτα τοποθετούνται στην κατάλληλη ουρά ανάλογα με τη **θύρα προορισμού**
- Δεν υπάρχει κάποιος μηχανισμός **ελέγχου ροής**
  - ▶ πακέτα μπορεί να χαθούν λόγω υπερχειλίσισης των ουρών αποθήκευσης



## Έλεγχος ορθής παράδοσης

- Το UDP παρέχει τη δυνατότητα ελέγχου τη ορθότητας ενός πακέτου που παραδόθηκε μέσω της χρήσης του **αθροίσματος ελέγχου (checksum)**
- Το άθροισμα ελέγχου υπολογίζεται με βάση:
  - ▶ την κεφαλίδα του UDP
  - ▶ τα δεδομένα του πακέτου UDP
  - ▶ ένα στοιχείο που αποκαλείται **ψευδοκεφαλίδα**
- Η ψευδοκεφαλίδα αποτελείται από:
  - ▶ τον αριθμό πρωτοκόλλου της κεφαλίδας IP
  - ▶ τις διευθύνσεις IP αποστολέα και παραλήπτη
  - ▶ το πεδίο μήκους του UDP
- Η χρήση της ψευδοκεφαλίδας στο άθροισμα ελέγχου έχει στόχο να πιστοποιήσει ότι το πακέτο παραδόθηκε μεταξύ των σωστών άκρων επικοινωνίας



# Διάρθρωση

- 1 Κατακερματισμός
- 2 Κατανομή πόρων και Έλεγχος συμφόρησης
- 3 Επικοινωνία από άκρο σε άκρο
  - Το πρωτόκολλο UDP
  - Το πρωτόκολλο TCP



## Εισαγωγή (1/2)

- Το πρωτόκολλο Transmission Control Protocol (TCP) παρέχει μια αξιόπιστη, συνδεοστρεφή (connection-oriented) επικοινωνία μεταξύ διεργασιών
- Η συνδεοστρεφής προσέγγιση είναι απαραίτητη ώστε πέρα από τη αξιόπιστη μετάδοση να εξασφαλιστούν:
  - ▶ η εν σειρά παράδοση των δεδομένων
  - ▶ η αποφυγή παράδοσης διπλότυπων μηνυμάτων
- Το TCP είναι πλήρως αμφίδρομο (full-duplex) πρωτόκολλο
  - ▶ κάθε σύνδεση υποστηρίζει την ταυτόχρονη μετάδοση δεδομένων και προς τις δύο κατευθύνσεις



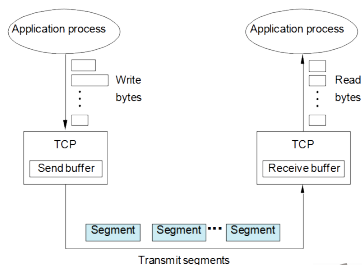
## Εισαγωγή (2/2)

- Κεντρική ιδέα: χρήση ενός μηχανισμού **κυλιόμενου παραθύρου** για την αξιόπιστη μετάδοση δεδομένων πάνω από το αναξιόπιστο διαδίκτυο
- Η πολυπλοκότητα ενός διαδικτύου επιβάλλει επίσης την ενσωμάτωση στο TCP:
  - ▶ ενός μηχανισμού **ελέγχου ροής** (flow control)
  - ▶ ενός μηχανισμού **ελέγχου συμφόρησης** (congestion control)
- Χρησιμοποιείται ο ίδιος μηχανισμός αποπολύπλεξης όπως και στο UDP



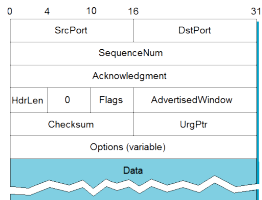
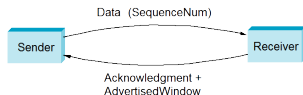
# Ροή πληροφορίας και Τμήματα

- Το πρωτόκολλο TCP είναι ένα πρωτόκολλο **προσανατολισμένο σε byte** και επομένως μεταφέρει τα δεδομένα ως ένα **ρεύμα byte (byte stream)**
  - ▶ η διεργασία αποστολέας αποστέλλει μια συνεχόμενη σειρά από bytes και η διεργασία παραλήπτης διαβάζει την ίδια σειρά από bytes
  - ▶ το πρωτόκολλο TCP είναι υπεύθυνο για την τμηματοποίηση της πληροφορίας σε πακέτα και την ανασύνθεση του αρχικού ρεύματος byte
- Τα πακέτα TCP ονομάζονται **τμήματα (segments)**
  - ▶ κάθε πακέτο μεταφέρει ένα τμήμα του αρχικού ρεύματος byte



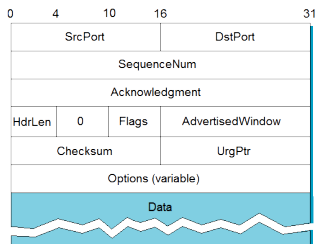
## Μορφή τμήματος (1/2)

- Τα πεδία **θύρα προέλευσης (SrcPort)** και **θύρα προορισμού (DstPort)** καθορίζουν τα άκρα της επικοινωνίας
- Ο αλγόριθμος κυλιόμενου παραθύρου χρησιμοποιεί τα πεδία:
  - ▶ **αριθμός ακολουθίας (SequenceNum)**: περιέχει τον αριθμό ακολουθίας του πρώτου byte που μεταφέρεται
    - στο TCP κάθε byte έχει έναν ακολουθιακό αριθμό
  - ▶ **επιβεβαίωση (Acknowledgment)**: περιέχει τον αριθμό ακολουθίας στον οποίο αναφέρεται μια επιβεβαίωση
  - ▶ **αναγγελθέν παράθυρο (AdvertisedWindow)**
- Ο αριθμός ακολουθίας χρησιμοποιείται στην κατεύθυνση μετάδοσης των δεδομένων
  - ▶ η επιβεβαίωση και το αναγγελθέν παράθυρο χρησιμοποιούνται στην αντίθετη κατεύθυνση



## Μορφή τμήματος (2/2)

- Το πεδίο **σημαίες (Flags)** προσδιορίζει πακέτα ελέγχου του TCP
  - ▶ SYN (synchronize): χρησιμοποιείται στην εγκαθίδρυση μιας σύνδεσης
  - ▶ FIN (finish): χρησιμοποιείται στον τερματισμό μιας σύνδεσης
  - ▶ RESET: χρησιμοποιείται για την επανεκκίνηση μιας σύνδεσης μετά από περιπτώσεις σφαλμάτων
  - ▶ ACK: χρησιμοποιείται για να δηλώσει ένα πακέτο επιβεβαίωσης
  - ▶ PUSH: δηλώνει ότι τα δεδομένα στάλθηκαν από τον αποστολέα με τη λειτουργία *ώθησης (push)*
- **Άθροισμα ελέγχου (Checksum):**
  - ▶ χρησιμοποιείται για τον έλεγχο ορθότητας ενός ληφθέντος πακέτου όπως και στο UDP
- **Μήκος κεφαλίδας (HdrLen):**
  - ▶ δηλώνει το μέγεθος της κεφαλίδας σε λέξεις των 32 bit





## Μηχανισμός κυλιόμενου παραθύρου: Γενικές αρχές

- Ο μηχανισμός κυλιόμενου παραθύρου του TCP εξασφαλίζει την **αξιόπιστη** και **εν σειρά** παράδοση των δεδομένων
- Επιπλέον, **ενσωματώνει** και ένα μηχανισμό **ελέγχου ροής**
  - ▶ αντί της χρήσης ενός σταθερού μεγέθους παραθύρου, ο **παραλήπτης αναγγέλλει στον αποστολέα το μέγεθος παραθύρου**
  - ▶ η αναγγελία του παραθύρου γίνεται μέσω του πεδίου **αναγγελθέν παράθυρο**
  - ▶ ο αποστολέας προσαρμόζεται ώστε τα μη επιβεβαιωμένα δεδομένα να μην έχουν μέγεθος μεγαλύτερο από το μέγεθος του παραθύρου
- Ο παραλήπτης επιλέγει το μέγεθος παραθύρου με βάση το διαθέσιμο αποθηκευτικό χώρο

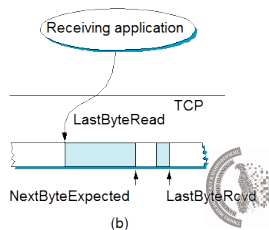
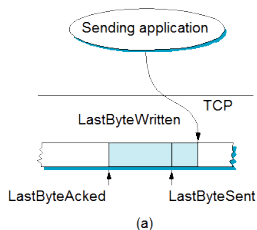


## Αξιόπιστη και διατεταγμένη παράδοση (1/2)

- Ο αποστολέας διατηρεί ένα χώρο προσωρινής αποθήκευσης για δεδομένα που:
  - ▶ έχουν σταλεί αλλά δεν έχουν επιβεβαιωθεί
  - ▶ έχουν εγγραφεί από την εφαρμογή αλλά δεν έχουν αποσταλεί
- Ο αποστολέας διατηρεί τρεις μετρητές:
  - ▶ LastByteAcked: τελευταίο επιβεβαιωμένο byte
  - ▶ LastByteSent: τελευταίο σταλθέν byte
  - ▶ LastByteWritten: τελευταίο εγγεγραμμένο byte

$$\text{LastByteAcked} \leq \text{LastByteSent}$$

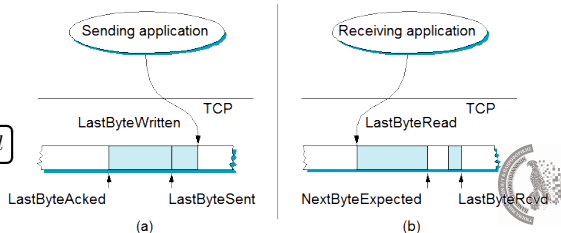
$$\text{LastByteSent} \leq \text{LastByteWritten}$$



## Αξιόπιστη και διατεταγμένη παράδοση (2/2)

- Ο παραλήπτης διατηρεί ένα χώρο προσωρινής αποθήκευσης για δεδομένα που:
  - ▶ λαμβάνονται χωρίς σωστή σειρά
  - ▶ δεδομένα που είναι σε σωστή σειρά (δηλαδή δεν λείπουν byte στο ρεύμα byte που προηγείται) αλλά η εφαρμογή δεν τα έχει διαβάσει
- Ο παραλήπτης διατηρεί επίσης τρεις μετρητές:
  - ▶ LastByteRead: τελευταίο διαβασμένο byte
  - ▶ NextByteExpected: επόμενο αναμενόμενο byte
  - ▶ LastByteRcvd: τελευταίο ληφθέν byte

$$\text{LastByteRead} \leq \text{NextByteExpected}$$



## Έλεγχος ροής (1/3)

- Στο πρωτόκολλο κυλιόμενου παραθύρου το μέγεθος του παραθύρου καθορίζει το μέγιστο ρυθμό του αποστολέα
  - ▶ αναπαριστά τα δεδομένα που μπορεί να στείλει ο αποστολέας χωρίς επιβεβαίωση
  - ▶ το μέγιστο παράθυρο καθορίζεται από το μέγιστο αποθηκευτικό χώρο του αποστολέα (*MaxSendBuffer*) και του παραλήπτη (*MaxRcvBuffer*)
- Ο παραλήπτης μπορεί να ελέγξει το ρυθμό του αποστολέα *αναγγέλοντας* μέσω μιας επιβεβαίωσης ένα παράθυρο ανάλογο με την αποθηκευτική του ικανότητα  
( $LastByteRcvd - LastByteRead \leq MaxRcvBuffer$ )

$$AdvertisedWindow = MaxRcvBuffer - ((NextByteExpected - 1) - LastByteRead)$$

- Το παράθυρο αναπροσαρμόζεται ανάλογο με την ικανότητα του παραλήπτη να διαβάζει δεδομένα
  - ▶ το παράθυρο λαμβάνει τιμές από 0 μέχρι *MaxRcvBuffer*



## Έλεγχος ροής (2/3)

- Ο αποστολέας πρέπει να τηρήσει το αναγγελθέν παράθυρο ( $LastByteSent - LastByteAcked \leq AdvertisedWindow$ ) υπολογίζοντας ένα **δραστικό παράθυρο**:

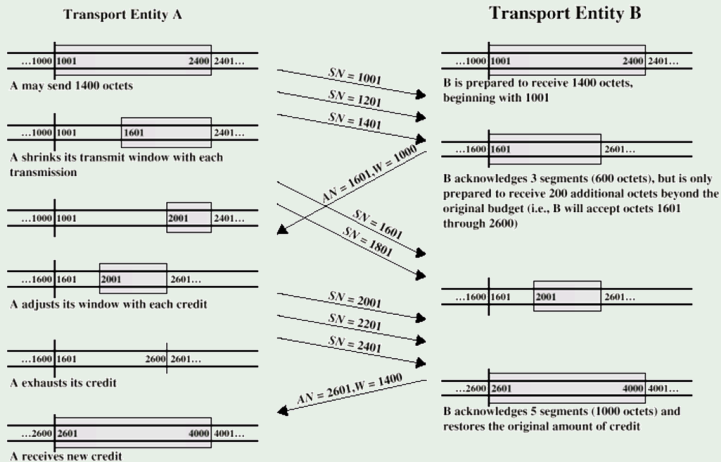
$$EffectiveWindow = AdvertisedWindow - (LastByteSent - LastByteAcked)$$

- ▶ παράλληλα πρέπει να εξασφαλίσει ότι η εφαρμογή δεν θα προκαλέσει υπερχείλιση του αποθηκευτικού χώρου ( $LastByteWritten - LastByteAcked \leq MaxSendBuffer$ )
- Ο αποστολέας σταματά την αποστολή δεδομένων αν το δραστικό παράθυρο γίνει μηδέν
  - ▶ πρόβλημα: η μη αποστολή δεδομένων δεν παράγει επιβεβαιώσεις ώστε να ενημερωθεί ο αποστολέας για μεταβολές του παραθύρου του παραλήπτη
  - ▶ λύση: το TCP εξακολουθεί να στέλνει ένα byte δεδομένων ακόμα και αν το δραστικό παράθυρο είναι μηδέν



## Έλεγχος ροής (3/3)

## Example



## Προσαρμοστική αναμετάδοση

- Καθορισμός βέλτιστου χρόνου αναμονής (timeout): ιδιαίτερα σημαντικός για την απόδοση του αλγόριθμου κυλιόμενου παραθύρου
  - ▶ ο προσδιορισμός του σχετίζεται με το **χρόνο RTT**
  - ▶ ο χρόνος RTT είναι **μεταβαλλόμενος** σε ένα διαδίκτυο και η μέτρησή του είναι δύσκολη
- Το TCP καθορίζει ένα μηχανισμό εκτίμησης του χρόνου RTT
  - ▶ για κάθε αποστολή πακέτου και λήψη μιας επιβεβαίωσης ένας κόμβος πραγματοποιεί μια μέτρηση του χρόνου RTT (**SampleRTT**)
  - ▶ ο **εκτιμώμενος χρόνος RTT (EstimatedRTT)** υπολογίζεται ως:  
$$EstimatedRTT = \alpha EstimatedRTT + (1 - \alpha) SampleRTT$$
  - ▶ η τιμή  $\alpha$  έχει σκοπό να **εξομαλύνει** τις απότομες διακυμάνσεις του χρόνου RTT
- Το TCP καθορίζει  $timeout = 2EstimatedRTT$

⇒ Έχουν προταθεί περισσότεροι αποδοτικοί αλγόριθμοι για τον υπολογισμό του χρόνου RTT όπως οι αλγόριθμοι των Karn/Partridge και Jacobson/Karels

## Εγκαθίδρυση και τερματισμός σύνδεσης

- Η ανταλλαγή δεδομένων στο TCP μπορεί να γίνει μετά την εγκαθίδρυση της σύνδεσης (connection setup)
- Σκοπός: τα δύο άκρα επικοινωνίας πρέπει να συμφωνήσουν σε ένα σύνολο παραμέτρων
  - ▶ στην περίπτωση του TCP οι παράμετροι είναι οι **εναρκτήριοι αριθμοί ακολουθίας**
- Η εγκαθίδρυση της σύνδεσης γίνεται με τον αλγόριθμο **τριπλής χειραψίας (threeway handshake)**
  - ▶ η διαδικασία ξεκινά από τον πελάτη (ασύμμετρη διαδικασία)
- Η διαδικασία τερματισμού της σύνδεσης πρέπει να ξεκινήσει από κάθε άκρο ξεχωριστά (συμμετρική διαδικασία)
  - ▶ σκοπός: η αποδέσμευση των πόρων του συστήματος
  - ▶ κάθε άκρο τερματίζει τη δική του κατεύθυνση της σύνδεσης (δεν επιθυμεί να στείλει άλλα δεδομένα)

