



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
Εθνικόν και Καποδιστριακόν
Πανεπιστήμιον Αθηνών

Δίκτυα Επικοινωνιών Ι

Ενότητα 3: Επίπεδο Μεταφοράς

Διδάσκων: Καθ. Ιωάννης Σταυρακάκης

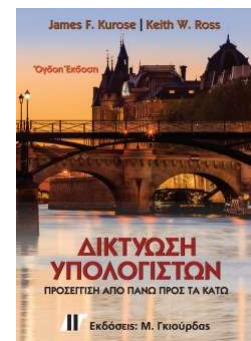
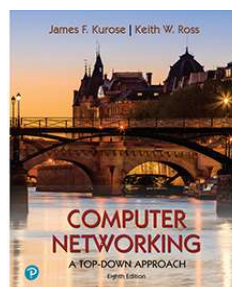
Δίκτυα Επικοινωνιών Ι

Θεματικές Ενότητες (ΘΕ) μαθήματος:

- ΘΕ1: Εισαγωγή**
(Κεφ. 1 του βιβλίου)
- ΘΕ2: Επίπεδο Εφαρμογής**
(Κεφ. 2 του βιβλίου)
- ΘΕ3: Επίπεδο Μεταφοράς**
(Κεφ. 3 του βιβλίου)
- ΘΕ4: Επίπεδο Δικτύου**
Επίπεδο Δεδομένων (Κεφ. 4 του βιβλίου)
Επίπεδο Ελέγχου (Κεφ. 5 του βιβλίου)

Συνιστώμενο Βιβλίο:
Computer Networking: A Top-Down Approach, by Kurose & Ross,
Addison-Wesley, 8^η Έκδοση

Ελληνική Μετάφραση:
Εκδόσεις : Μ. Γκιούρδας



Οι περισσότερες από τις διαφάνειες αυτής της ενότητας αποτελούν προσαρμογή και απόδοση στα ελληνικά των διαφανειών που συνοδεύουν το βιβλίο Computer Networking : A Top-Down Approach, J.F Kurose and K.W. Ross, 8/E, Addison-Wesley.

All material copyright 1996-2021
J.F Kurose and K.W. Ross, All Rights Reserved
Προσαρμογή και επιμέλεια της απόδοσης των πρωτότυπων διαφανειών στα ελληνικά :
Ιωάννης Σταυρακάκης

Κεφάλαιο 3: Επίπεδο Μεταφοράς

Οι στόχοι μας:

- ❑ Κατανόηση των αρχών πίσω από τις υπηρεσίες του επιπέδου μεταφοράς:
 - Πολύπλεξη/αποπολύπλεξη (multiplexing/demultiplexing)
 - Αξιόπιστη μεταφορά δεδομένων
 - Έλεγχος ροής (flow control)
 - Έλεγχος συμφόρησης (congestion control)
- ❑ Να μάθουμε για τα πρωτόκολλα επιπέδου μεταφοράς του Διαδικτύου:
 - UDP: ασυνδεσμική μεταφορά παρέχοντας υπηρεσία βέλτιστης προσπάθειας
 - TCP: αξιόπιστη, συνδεσμική μεταφορά
 - TCP έλεγχο συμφόρησης

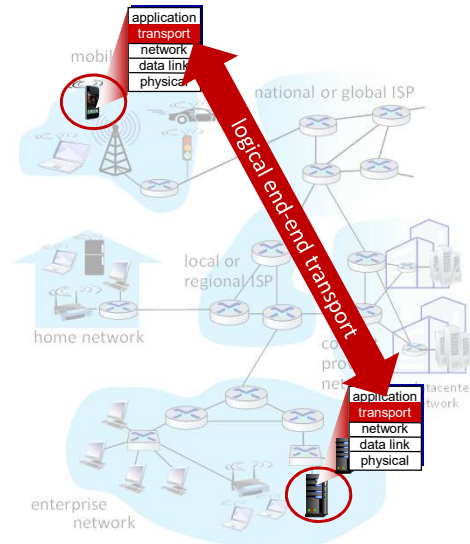
Επίπεδο Μεταφοράς: περίγραμμα

- ❑ **Υπηρεσίες επιπέδου μεταφοράς**
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεσμική μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεσμική μεταφορά: TCP
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP
- ❑ Εξέλιξη του Επιπέδου Μεταφοράς



Υπηρεσίες και πρωτόκολλα μεταφοράς

- παρέχουν **λογική επικοινωνία** μεταξύ διεργασιών εφαρμογών που τρέχουν σε διαφορετικά τερματικά συστήματα
- τα πρωτόκολλα μεταφοράς τρέχουν στα τερματικά συστήματα
 - αποστολέας: σπάει τα μηνύματα της εφαρμογής σε **τμήματα**, τα περνάει στο επίπεδο δικτύου
 - παραλήπτης: ανασύνθεση των τμημάτων σε μηνύματα, τα περνάει στο επίπεδο εφαρμογής
- Δύο πρωτόκολλα μεταφοράς διαθέσιμα στις εφαρμογές Διαδικτύου
 - TCP, UDP



Επίπεδο Μεταφοράς 3-5

Υπηρεσίες επιπέδου Μεταφοράς έναντι υπηρεσιών επιπέδου Δικτύου



© 1982, W.W. EVANS.
THERE was an old woman who lived in a shoe.
She had so many children, she didn't know what to do.
She gave them some milk and nice butter bread,
She kissed them all round and put them to bed.

Αναλογία Νοικοκυριού:

12 παιδιά στο σπίτι της Άννας στέλνουν γράμματα σε 12 παιδιά στο σπίτι του Βασιλή

- υπολογιστές = σπίτια
- διεργασίες = παιδιά
- μηνύματα εφαρμογών = γράμματα σε φακέλους
- πρωτόκολλο μεταφοράς = Άννα και Βασιλίας που αναλαμβάνουν να στείλουν τα γράμματα και να μοιράσουν την αλληλογραφία σε κάθε σπίτι)
- πρωτόκολλο επιπέδου δικτύου = ταχυδρομική υπηρεσία

Επίπεδο Μεταφοράς

3-6

Υπηρεσίες επιπέδου Μεταφοράς έναντι υπηρεσιών επιπέδου Δικτύου

- επίπεδο δικτύου: λογική επικοινωνία μεταξύ **υπολογιστών**
- επίπεδο μεταφοράς: λογική επικοινωνία μεταξύ **διεργασιών**
 - στηρίζεται στις υπηρεσίες του επιπέδου δικτύου, τις οποίες και εμπλουτίζει

Αναλογία Νοικοκυριού:

12 παιδιά στο σπίτι της Άννας στέλνουν γράμματα σε 12 παιδιά στο σπίτι του Βασίλη

- υπολογιστές = σπίτια
- διεργασίες = παιδιά
- μηνύματα εφαρμογών = γράμματα σε φακέλους
- πρωτόκολλο μεταφοράς = Άννα και Βασίλης που αναλαμβάνουν που αναλαμβάνουν να στείλουν τα γράμματα και να μοιράσουν την αλληλογραφία σε κάθε σπίτι)
- πρωτόκολλο επιπέδου δικτύου = ταχυδρομική υπηρεσία

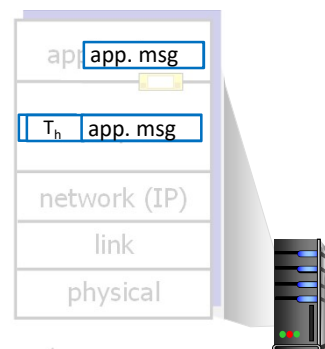
Επίπεδο Μεταφοράς

3-7

Ενέργειες Επιπέδου Μεταφοράς

Αποστολέας:

- Δημιουργείται ένα μήνυμα επιπέδου εφαρμογής προς αποστολή
- Καθορίζονται οι τιμές στα πεδία της κεφαλίδας
- Δημιουργείται ένα τμήμα
- Προωθείται το τμήμα στο επίπεδο IP



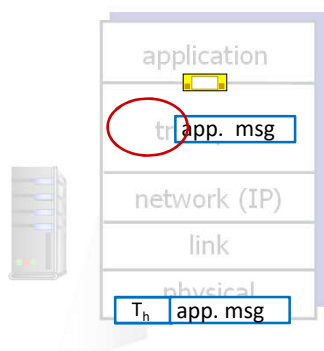
Επίπεδο Μεταφοράς

3-8

Ενέργειες Επιπέδου Μεταφοράς

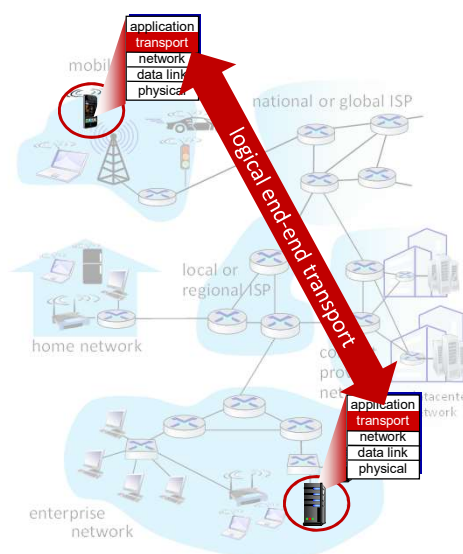
Παραλήπτης

- Λαμβάνει το τμήμα από το επίπεδο IP
- Ελέγχει τις τιμές στην κεφαλίδα
- Απομονώνει το μήνυμα επιπέδου εφαρμογής
- Αποπολυπλέκει το μήνυμα και το προωθεί στην εφαρμογή από το κατάλληλο socket



Δύο βασικά πρωτόκολλα Μεταφοράς στο Διαδίκτυο

- **TCP:** Transmission Control Protocol
 - αξιόπιστη, σε ορθή σειρά μεταφορά
 - έλεγχος συμφόρησης (congestion control)
 - έλεγχος ροής (flow control)
 - εγκαθίδρυση σύνδεσης
- **UDP:** User Datagram Protocol
 - μη αξιόπιστη, εκτός σειράς παράδοση
 - καμία επέκταση της “βέλτιστης προσπάθειας” (best effort) του IP
- υπηρεσίες που δεν είναι διαθέσιμες:
 - εγγυήσεις ως προς την καθυστέρηση
 - εγγυήσεις ως προς το εύρος ζώνης

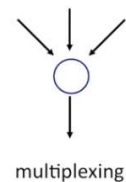
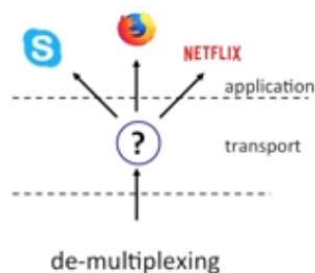


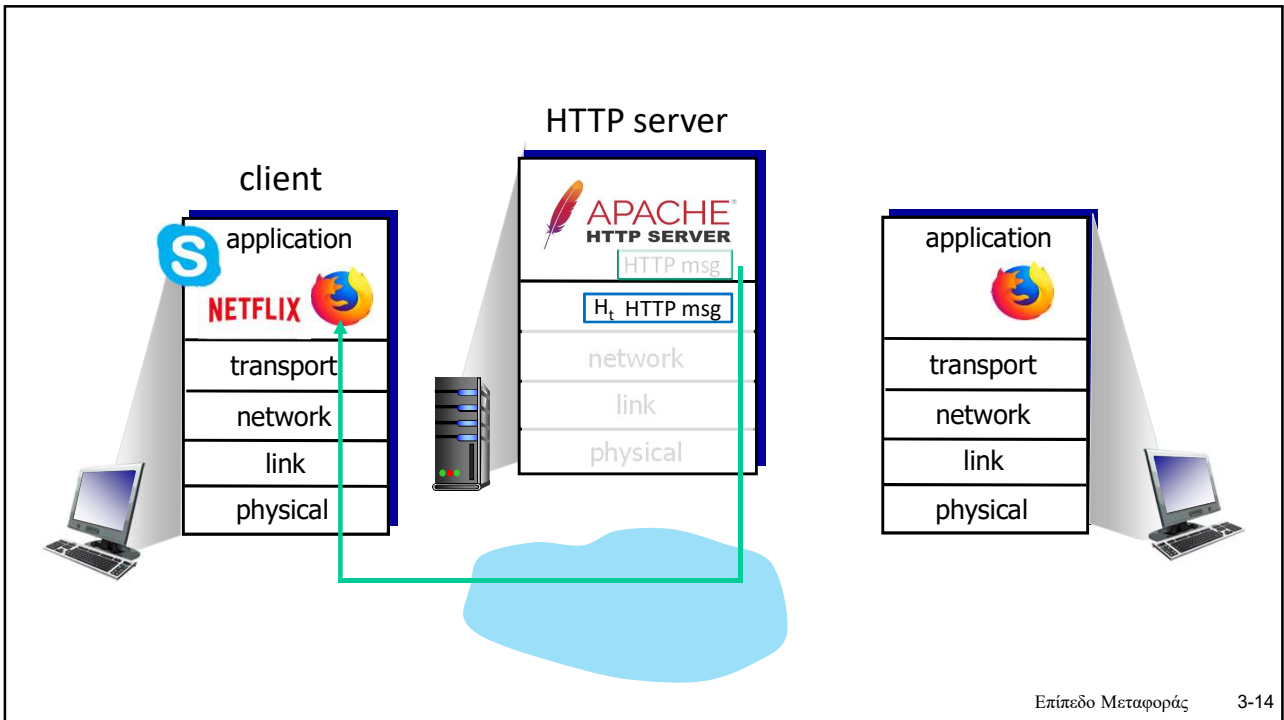
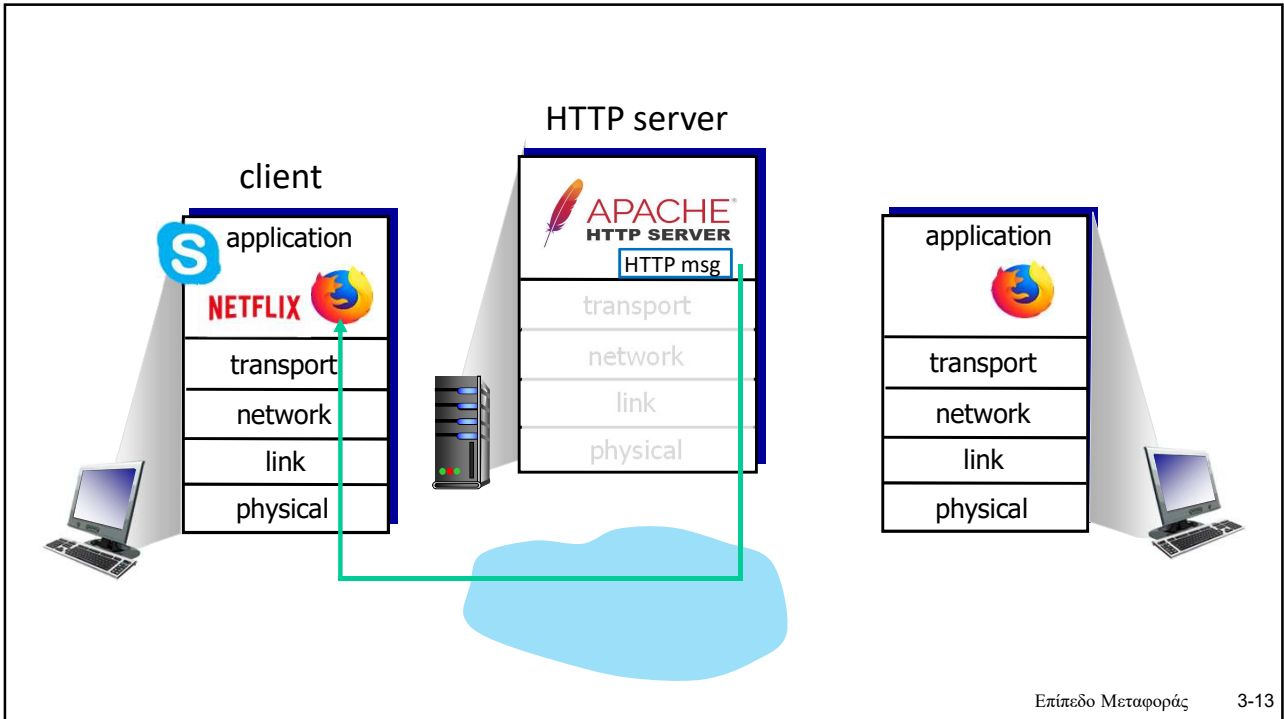
Επίπεδο Μεταφοράς: περίγραμμα

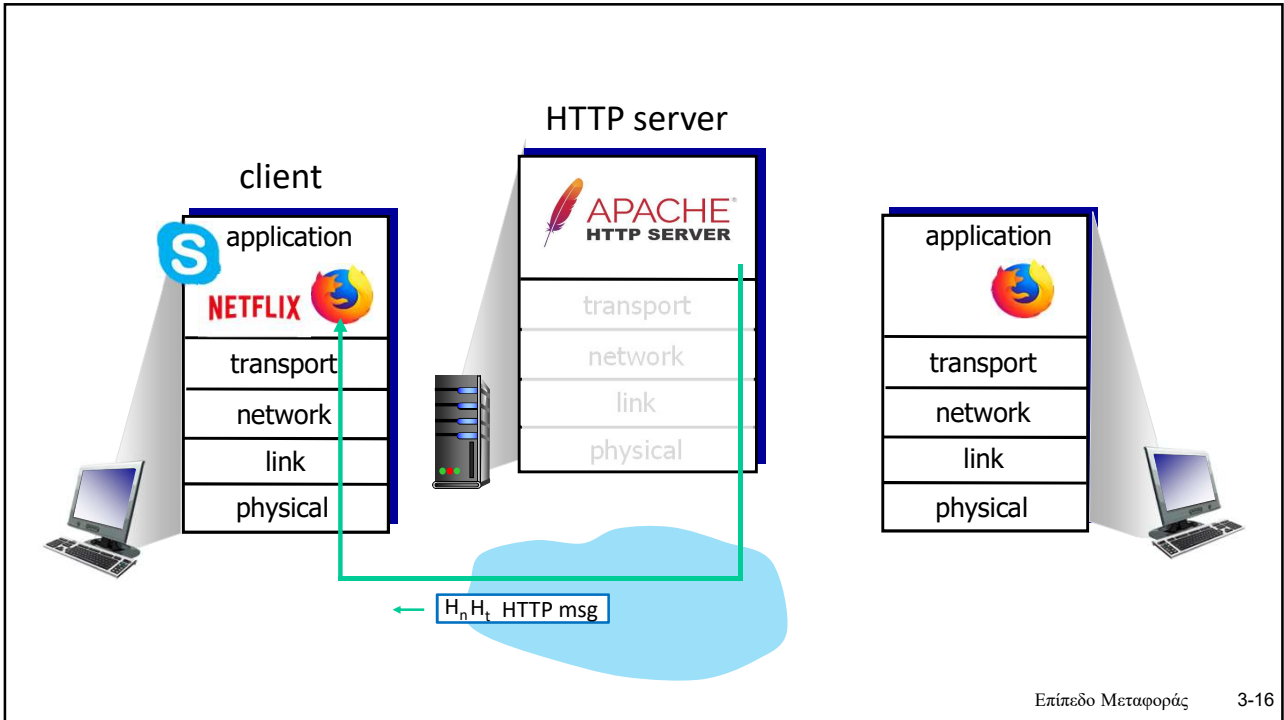
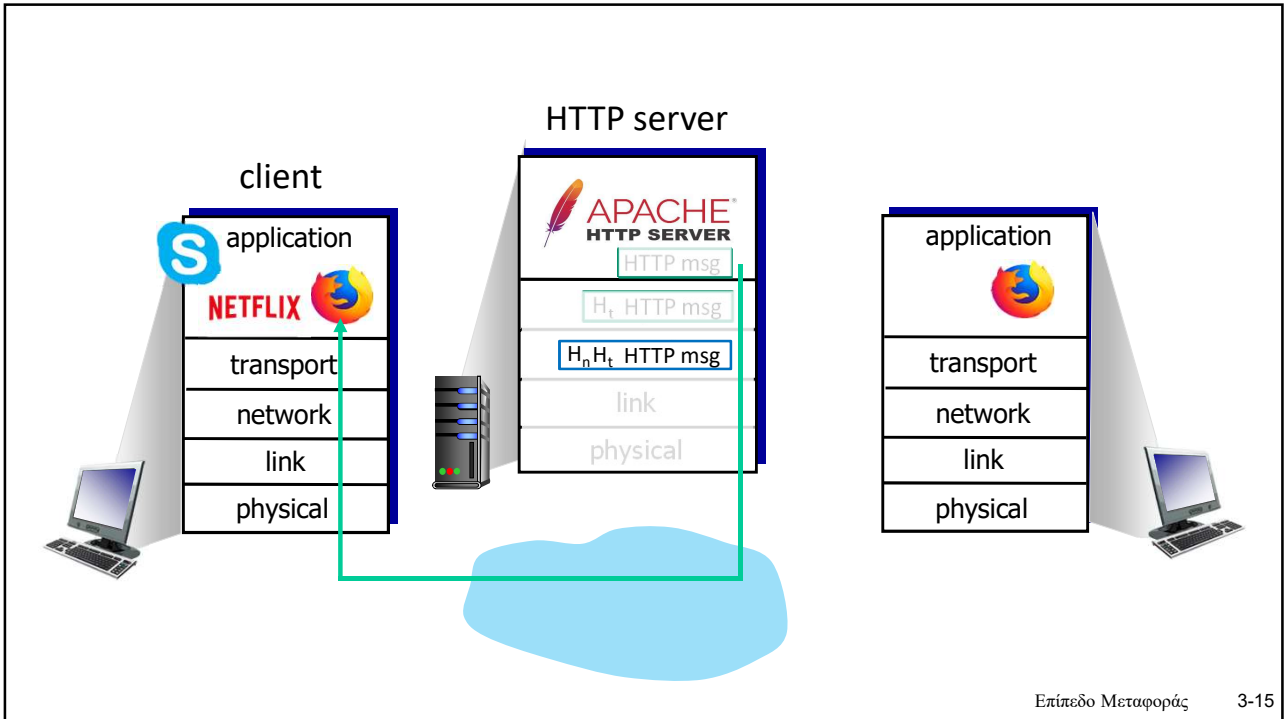
- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεδεσμένη μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεδεσμένη μεταφορά: TCP
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP
- ❑ Εξέλιξη του Επιπέδου Μεταφοράς

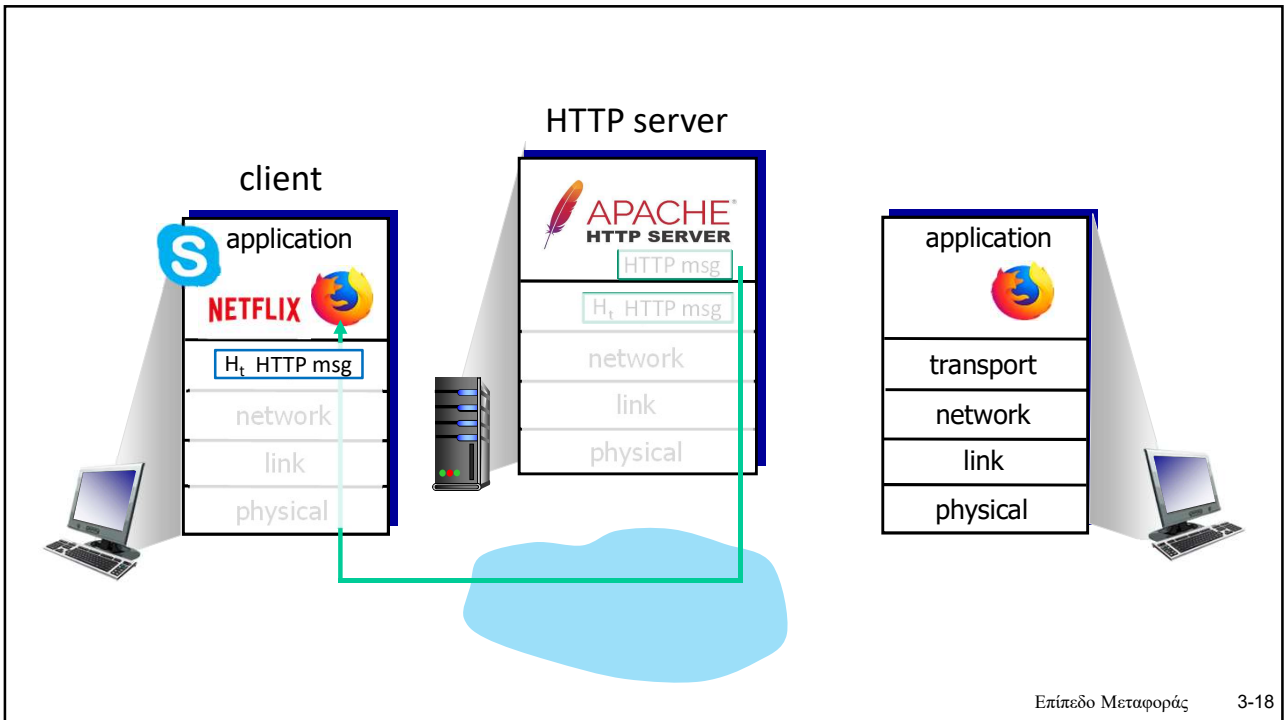
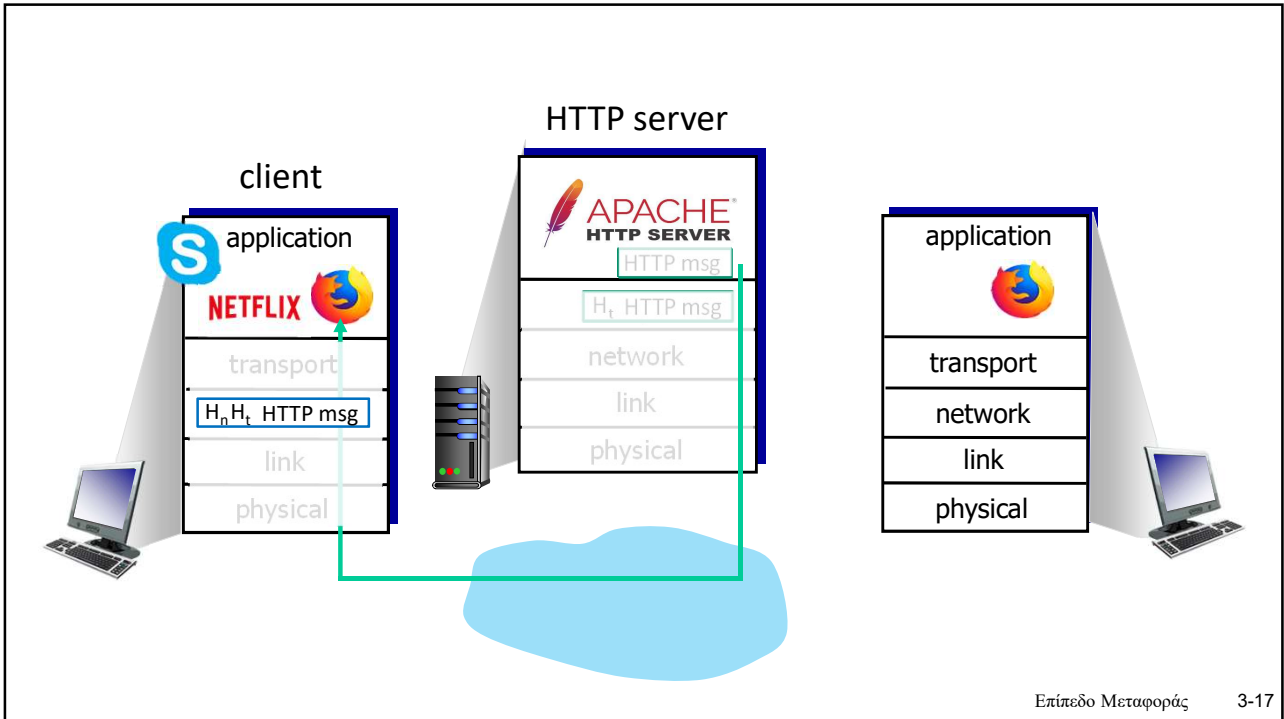


Πολύπλεξη - Αποπολύπλεξη (Multiplexing - Demultiplexing)



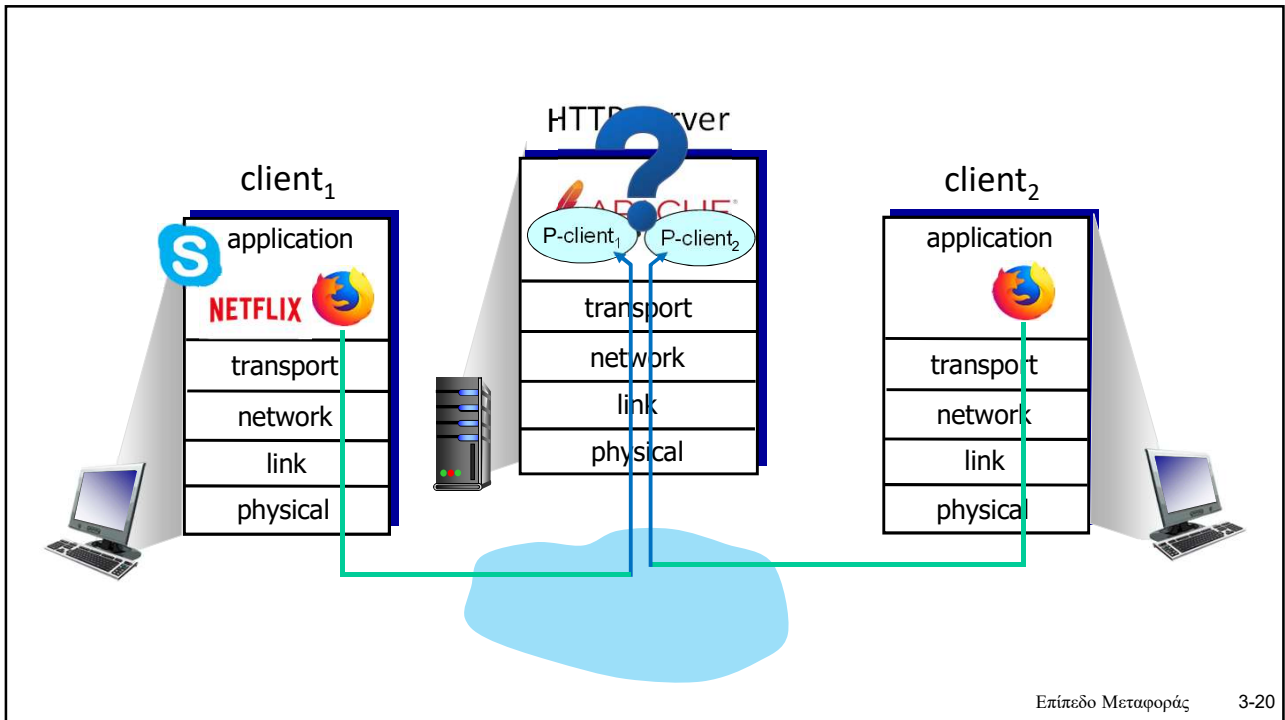
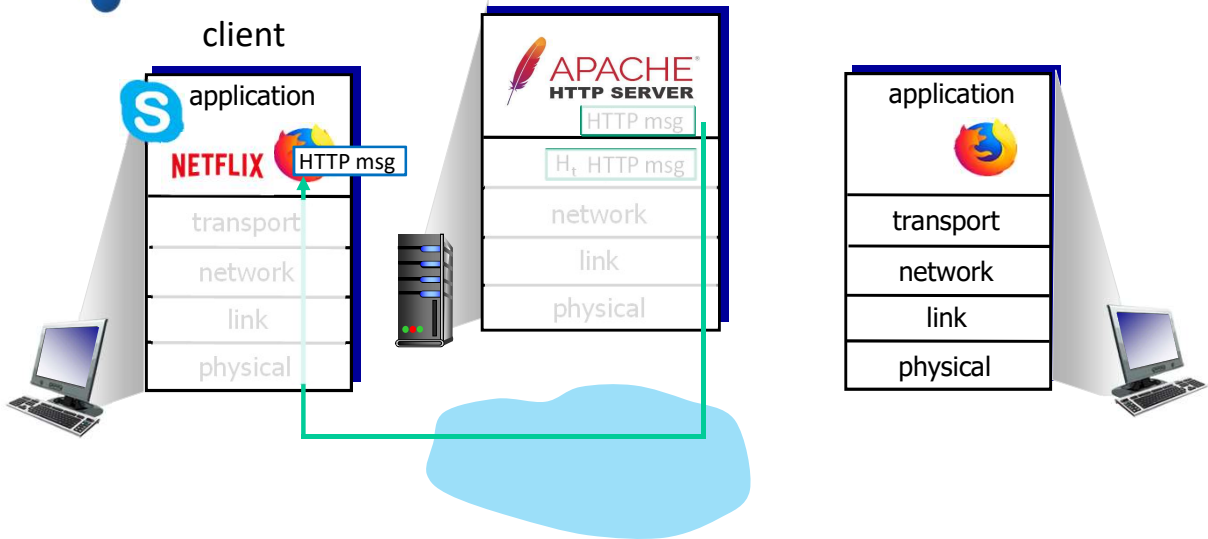








Ε: πως γνωρίζει το επίπεδο μεταφοράς σε ποια διεργασία (Firefox browser ή Netflix ή Skype) να παραδώσει το μήνυμα?



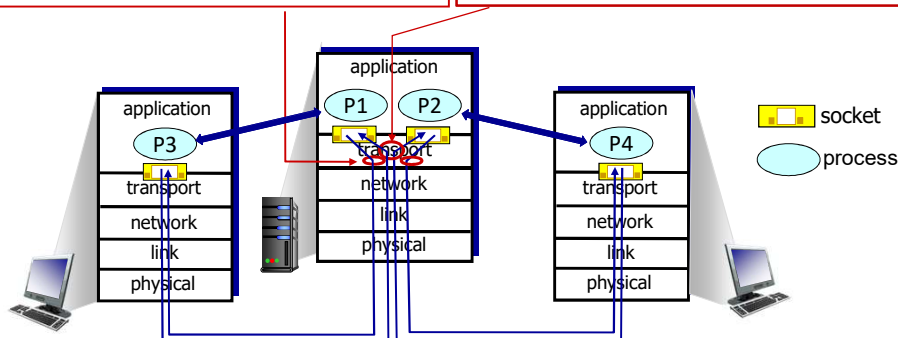
Πολύπλεξη/Αποπολύπλεξη

Πολύπλεξη στον υπολογιστή αποστολής

Διαχείριση δεδομένων από πολλαπλές sockets, προσθήκη κεφαλίδας μεταφοράς (που αργότερα χρησιμοποιείται για αποπολύπλεξη)

Αποπολύπλεξη στον υπολογιστή λήψης :

Χρήση πληροφοριών κεφαλίδας για παράδοση των τμημάτων που λαμβάνονται στη σωστή socket

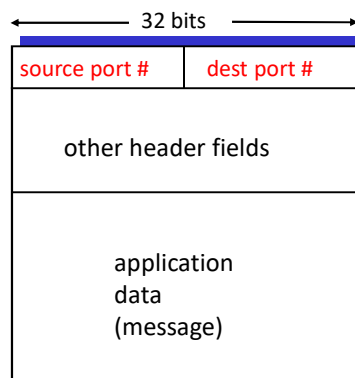


Επίπεδο Μεταφοράς

3-21

Πώς δουλεύει η αποπολύπλεξη

- Ο υπολογιστής λαμβάνει IP datagrams (δεδομενογράμματα)
 - κάθε datagram έχει διεύθυνση IP προέλευσης, διεύθυνση IP προορισμού
 - κάθε datagram μεταφέρει ένα τμήμα επιπέδου μεταφοράς
 - κάθε τμήμα έχει **αριθμό θύρας προέλευσης, προορισμού**
- Ο υπολογιστής χρησιμοποιεί **διευθύνσεις IP & αριθμούς θύρας** για να κατευθύνει το τμήμα στην κατάλληλη socket



TCP/UDP segment format

Επίπεδο Μεταφοράς

3-22

Ασυνδεσμική αποπολύπλεξη (UDP)

- Το επίπεδο μεταφοράς εκχωρεί έναν αριθμό θύρας (**12534**) τοπικά στον υπολογιστή, όταν δημιουργείται ένα UDP socket:

```
DatagramSocket mySocket1 = new
DatagramSocket(12534);
```

- Όταν δημιουργείται το datagram, για να σταλεί στη UDP socket του παραλήπτη πρέπει να καθορισθεί

- IP διεύθυνση προορισμού
- Αριθμός θύρας προορισμού

- Όταν ο υπολογιστής λαμβάνει τμήμα UDP:

- ελέγχει τον αριθμό θύρας προορισμού στο τμήμα
- κατευθύνει το τμήμα UDP στη socket με αυτό τον αριθμό θύρας

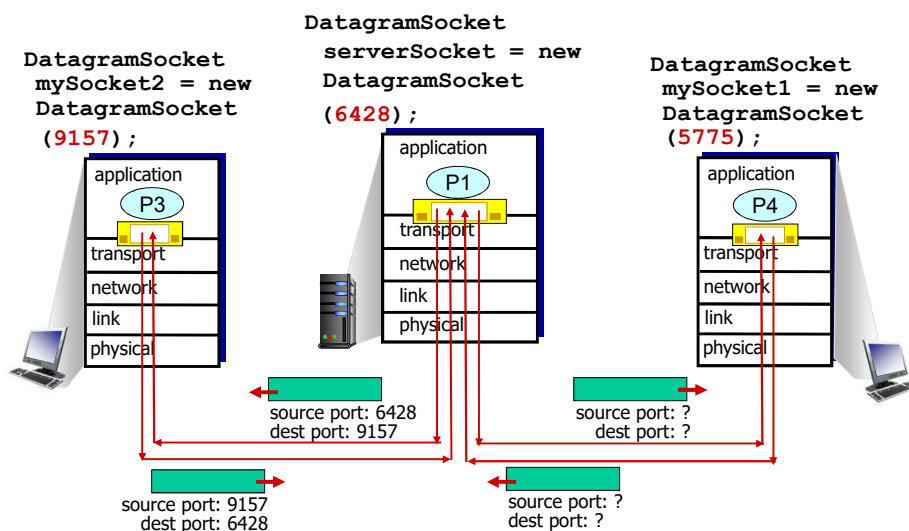


- Τα IP/UDP datagrams με *ίδιο αριθμό θύρας προορισμού*, αλλά διαφορετικές IP διευθύνσεις προέλευσης ή/και αριθμούς θύρας προέλευσης θα κατευθυνθούν προς την *ίδια socket* προορισμού

Επίπεδο Μεταφοράς

3-23

Ασυνδεσμική αποπολύπλεξη (παράδειγμα)



Το Source Port δίνει τη "διεύθυνση επιστροφής"

Επίπεδο Μεταφοράς

3-24

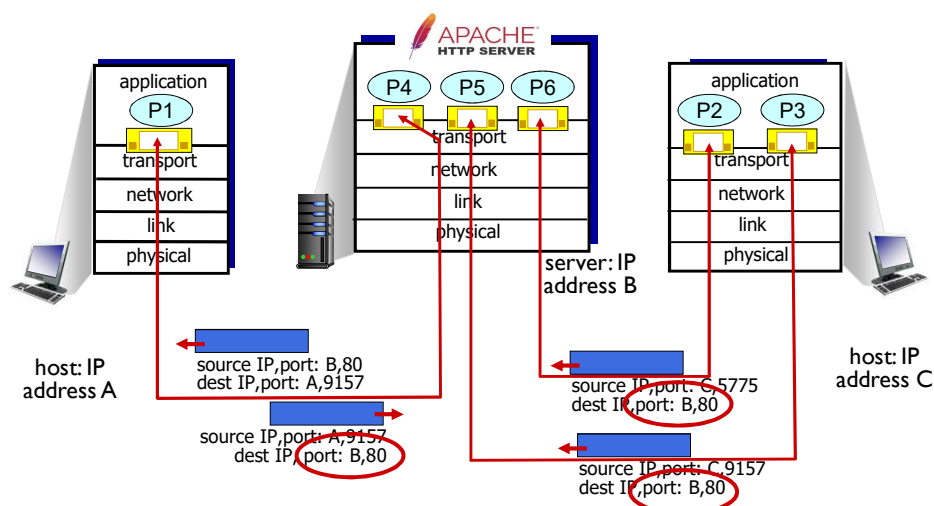
Συνδεσμική αποπολύπλεξη (TCP)

- Η TCP socket ταυτοποιείται από την τετράδα:
 - διεύθυνση IP προέλευσης
 - αριθμός θύρας προέλευσης
 - διεύθυνση IP προορισμού
 - αριθμός θύρας προορισμού
- Αποπολύπλεξη: ο δέκτης χρησιμοποιεί **και τις τέσσερις τιμές** για να κατευθύνει το τμήμα στην κατάλληλη socket
- Ένας εξυπηρετής μπορεί να υποστηρίξει πολλές ταυτόχρονες TCP sockets:
 - κάθε socket αναγνωρίζεται από τη δική της τετράδα
 - κάθε socket σχετίζεται με διαφορετικό πελάτη

Επίπεδο Μεταφοράς

3-25

Συνδεσμική αποπολύπλεξη (παράδειγμα)



3 τμήματα, όλα προορίζονται για IP διεύθυνση: B,
θύρα προορισμού: 80 αποπολυπλέκονται σε **διαφορετικές** sockets

Επίπεδο Μεταφοράς

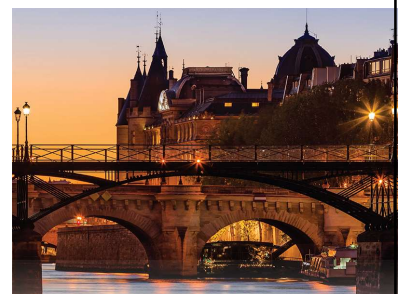
3-26

Περίληψη

- Multiplexing, demultiplexing: βασίζεται στις τιμές στα πεδία των κεφαλίδων του τμήματος, δεδομενογράμματος
- **UDP**: αποπολύπλεξη με χρήση της θύρας προορισμού (μόνο).
- **TCP**: αποπολύπλεξη με χρήση τετράδας: IP διεύθυνση και αριθμός θύρας πηγής και προορισμού
- Multiplexing/demultiplexing λαμβάνει χώρα σε όλα τα επίπεδα

Επίπεδο Μεταφοράς: περίγραμμα

- Υπηρεσίες επιπέδου μεταφοράς
- Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- Ασυνδεσμική μεταφορά: UDP**
- Αρχές της αξιόπιστης μεταφοράς δεδομένων
- Συνδεσμική μεταφορά: TCP
- Αρχές ελέγχου συμφόρησης
- Έλεγχος συμφόρησης του TCP
- Εξέλιξη του Επιπέδου Μεταφοράς



UDP: User Datagram Protocol

- ❑ Το απλούστερο πρωτόκολλο μεταφοράς του Διαδικτύου
- ❑ Υψηλότερη παράδοση βέλτιστης προσπάθειας (“best effort”), τα τμήματα UDP μπορεί να
 - χαθούν
 - παραδοθούν εκτός σειράς στις εφαρμογές
- ❑ **Ασυνδεσμική:**
 - δεν γίνεται χειραψία μεταξύ του UDP αποστολέα και δέκτη
 - Η διαχείριση κάθε UDP τμήματος γίνεται ξεχωριστά

Γιατί υπάρχει το UDP;

- ❑ χωρίς σύνδεση (που εισάγει καθυστέρηση)
- ❑ απλό: χωρίς κατάσταση σύνδεσης στον αποστολέα, δέκτη
- ❑ μικρή κεφαλίδα τμήματος
- ❑ χωρίς έλεγχο συμφόρησης:
 - το UDP μπορεί να “εκραγεί” όσο γρήγορα θέλουμε
 - μπορεί να λειτουργεί ακόμη και σε συνθήκες συμφόρησης

UDP: User Datagram Protocol

- ❑ Το UDP χρησιμοποιείται:
 - ❑ για εφαρμογές πολυμέσων συνεχούς ροής –streaming (ανοχή ως προς τις απώλειες / ευαισθησία ως προς το ρυθμό)
 - ❑ DNS
 - ❑ SNMP
 - ❑ HTTP/3
- ❑ Εάν απαιτείται αξιόπιστη μεταφορά πάνω από το UDP (e.g., HTTP/3):
 - ❑ προσθήκη αξιοπιστίας στο επίπεδο εφαρμογής
 - ❑ προσθήκη ελέγχου συμφόρησης στο επίπεδο εφαρμογής

UDP: User Datagram Protocol (RFC 768)

INTERNET STANDARD
RFC 768 J. Postel
ISI
28 August 1980

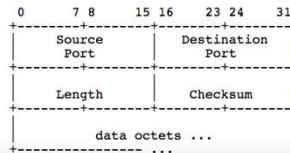
User Datagram Protocol

Introduction

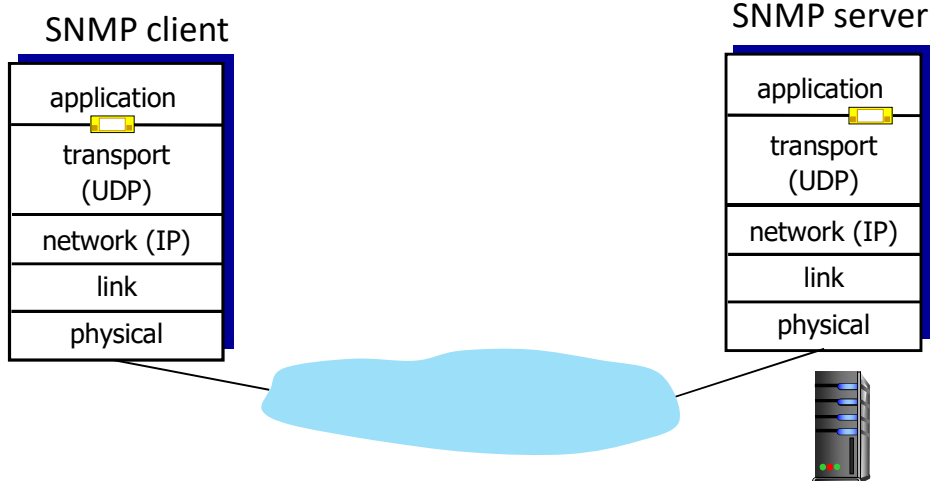
This User Datagram Protocol (UDP) is defined to make available a datagram mode of packet-switched computer communication in the environment of an interconnected set of computer networks. This protocol assumes that the Internet Protocol (IP) [1] is used as the underlying protocol.

This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism. The protocol is transaction oriented, and delivery and duplicate protection are not guaranteed. Applications requiring ordered reliable delivery of streams of data should use the Transmission Control Protocol (TCP) [2].

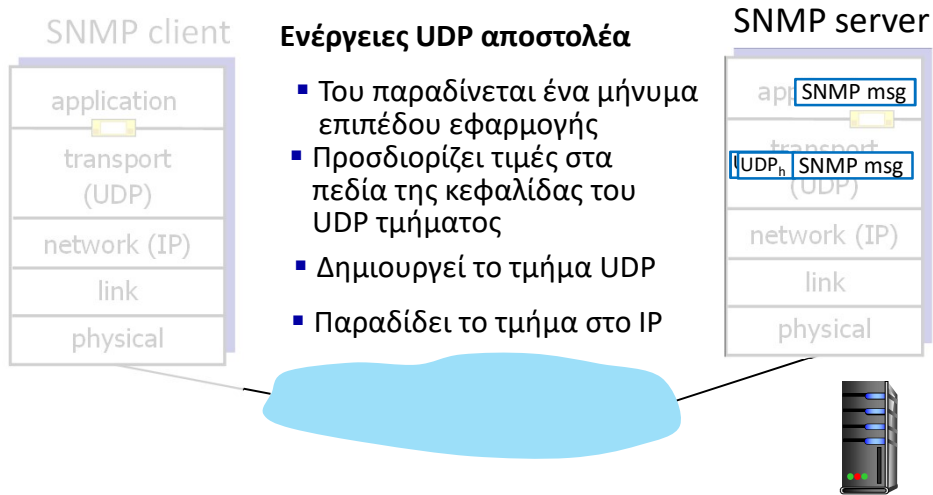
Format



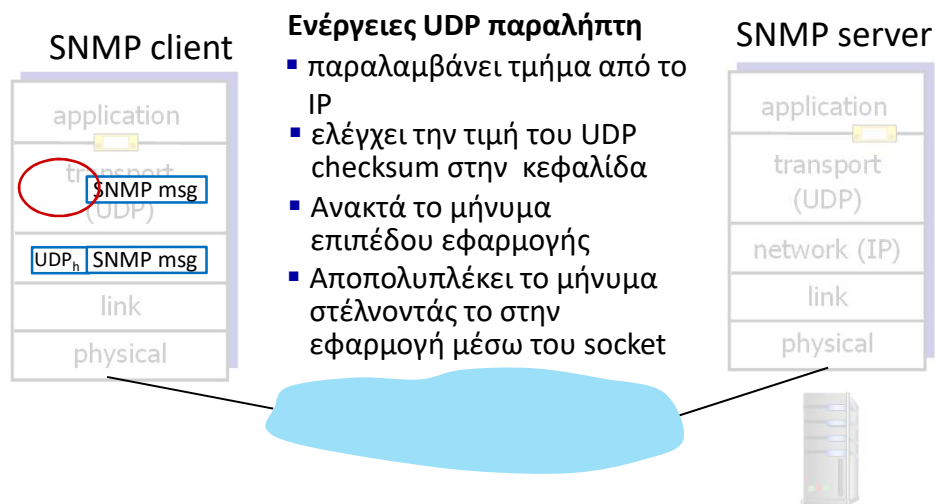
UDP: Ενέργειες Επιπέδου Μεταφοράς



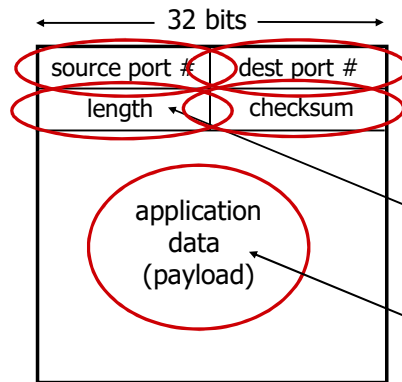
UDP: Ενέργειες Επιπέδου Μεταφοράς



UDP: Ενέργειες Επιπέδου Μεταφοράς



UDP: κεφαλίδα τμήματος



Δομή τμήματος UDP

Μήκος του τμήματος UDP σε bytes, συμπεριλαμβανομένης της κεφαλίδας

data προς/από το επίπεδο εφαρμογής

UDP checksum (άθροισμα ελέγχου)

Σκοπός: ανίχνευση σφαλμάτων (δηλ. ανεστραμμένων bits) στο μεταδιδόμενο τμήμα

	1 st αριθμός	2 nd αριθμός	άθροισμα
αποστολή:	5	6	11



Λήψη:	4	6	11
-------	---	---	----

Άθροισμα-ελέγχου υπολογισμένο από τον παραλήπτη

≠

Άθροισμα-ελέγχου υπολογισμένο από αποστολέα (όπως λαμβάνεται)



UDP checksum (άθροισμα ελέγχου)

Σκοπός: ανίχνευση “σφαλμάτων” (π.χ. ανεστραμμένων bits) στο μεταδιδόμενο τμήμα

Αποστολέας:

- ❑ χειρίζεται το περιεχόμενο του τμήματος (συμπεριλαμβανομένων πεδίων της κεφαλίδας), ως ακολουθία ακεραίων των 16 bits
- ❑ checksum: Πρόσθεση (συμπλήρωμα ως προς το 1 του αθροίσματος) του περιεχομένου του τμήματος
- ❑ ο αποστολέας τοποθετεί την τιμή του checksum στο πεδίο checksum του τμήματος UDP

Δέκτης:

- ❑ υπολογισμός του checksum του λαμβανομένου μηνύματος
- ❑ έλεγχος αν το υπολογισμένο checksum ισούται με την τιμή του πεδίου checksum:
 - Όχι – ανίχνευση σφάλματος
 - Ναι – μη ανίχνευση σφάλματοςΑλλά παρόλα αυτά ενδέχεται να γίνουν λάθη; (περισσότερα στη συνέχεια)

Παράδειγμα Checksum Διαδικτύου

Σημείωση: Κατά την πρόσθεση των αριθμών, το κρατούμενο από την πιο σημαντική θέση πρέπει να προστεθεί στο αποτέλεσμα

Παράδειγμα: πρόσθεση δύο ακεραίων 16-bit

	1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
	1 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

wraparound	① 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1

sum	1 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0
checksum	0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 1

Παράδειγμα Checksum Διαδικτύου

Παράδειγμα: πρόσθεση δύο ακεραίων 16-bit

	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	1	0	1
	1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	1
sum	1	0	1	1	1	0	1	1	1	0	1	1	1	1	0	0	0	0	0
checksum	0	1	0	0	0	1	0	0	0	1	0	0	0	0	0	1	1	1	1

0 1
1 0

Αν και κάποιοι αριθμοί τροποποιήθηκαν (bits ανεστράφησαν), το άθροισμα-ελέγχου **ΔΕΝ** αλλάζει! Αυτά τα σφάλματα δεν ανιχνεύονται (και πολλά άλλα!)

Σύνοψη: UDP

- Πολύ **απλό πρωτόκολλο** μεταφοράς του Διαδικτύου:
 - Τμήματα μπορεί να χαθούν ή να παραδοθούν εκτός σειράς
 - Υψηλότερη «βέλτιστης προσπάθειας»: “αποστολή τμημάτων με ευχές για το καλύτερο
- UDP έχει και τα **θετικά** του:
 - Δεν απαιτείται εγκαθίδρυση σύνδεσης / χειραψία μεταξύ αποστολέας / δέκτη (δεν υπεισέρχεται καθυστέρηση RTT)
 - Μπορεί να λειτουργεί και όταν το δίκτυο εμφανίζει συμφόρηση
 - Παρέχει κάποια αξιοπιστία (checksum)
- Μπορεί να **προστεθεί λειτουργικότητα** πάνω από το UDP στο επίπεδο εφαρμογής (π.χ., HTTP/3)

Επίπεδο Μεταφοράς: περίγραμμα

- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεδεσμένη μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεδεσμένη μεταφορά: TCP
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP
- ❑ Εξέλιξη του Επιπέδου Μεταφοράς



Αρχές αξιόπιστης μεταφοράς δεδομένων

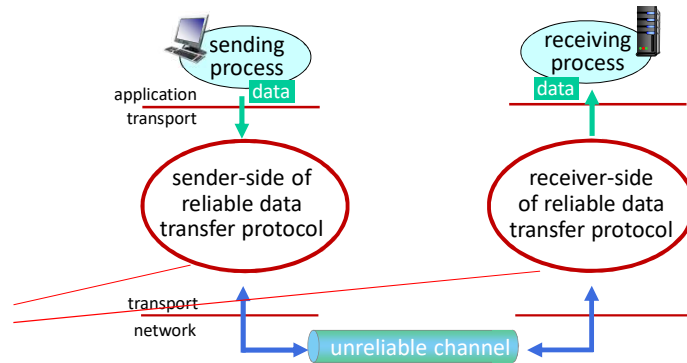
- ❑ Σημαντική στα επίπεδα εφαρμογής, μεταφοράς, ζεύξης
- ❑ Στο top-10 των σημαντικών θεμάτων της δικτύωσης!



Μοντέλο-απεικόνιση
αξιόπιστης μεταφοράς

Αρχές αξιόπιστης μεταφοράς δεδομένων

Η πολυπλοκότητα του πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων θα εξαρτάται από τα χαρακτηριστικά του αναξιόπιστου καναλιού (απώλειες, σφάλματα, εκτός σειράς παράδοση)

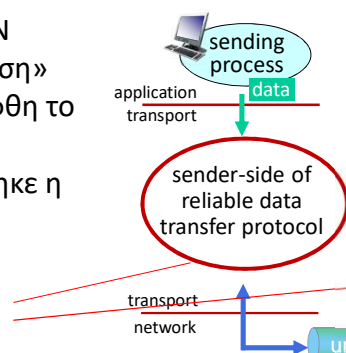


Υλοποίηση υπηρεσίας αξιόπιστης μεταφοράς

Αρχές αξιόπιστης μεταφοράς δεδομένων

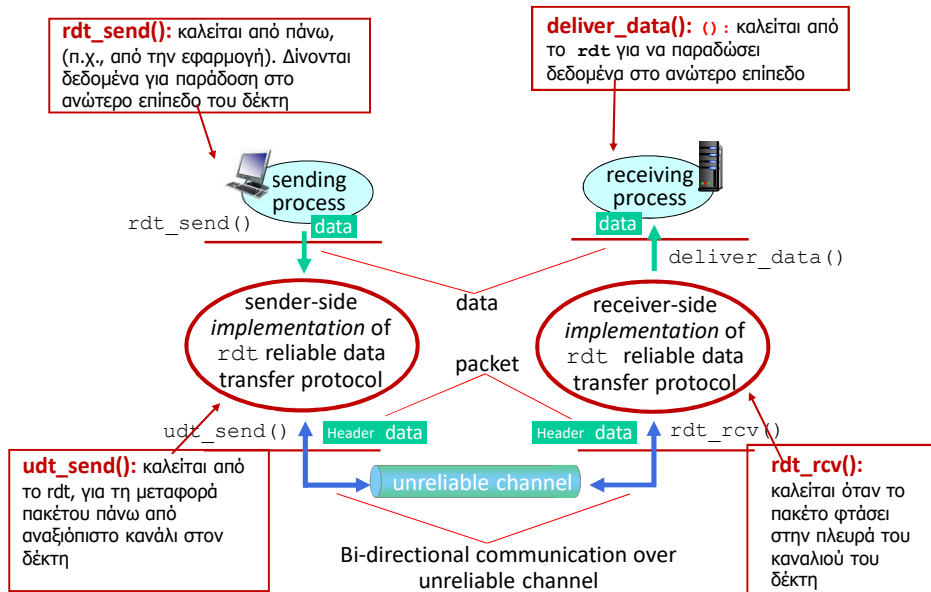
Αποστολέας, δέκτης ΔΕΝ γνωρίζουν την «κατάσταση» του άλλου, π.χ. εάν ελήφθη το μήνυμα.

- Εκτός αν επικοινωνήθηκε η πληροφορία μέσω μηνύματος.



Υλοποίηση υπηρεσίας αξιόπιστης μεταφοράς

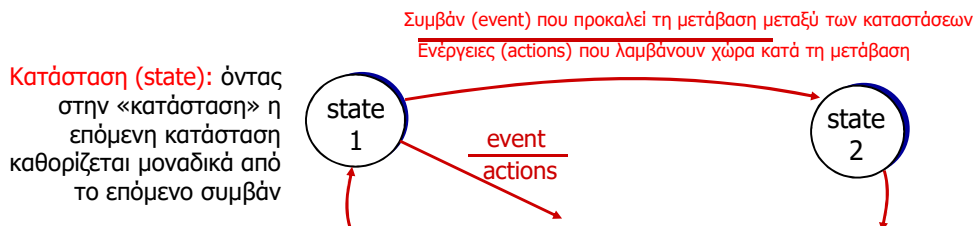
Reliable data transfer protocol (rdt): διεπαφές



Αξιόπιστη μεταφορά δεδομένων: ξεκινώντας

Θα:

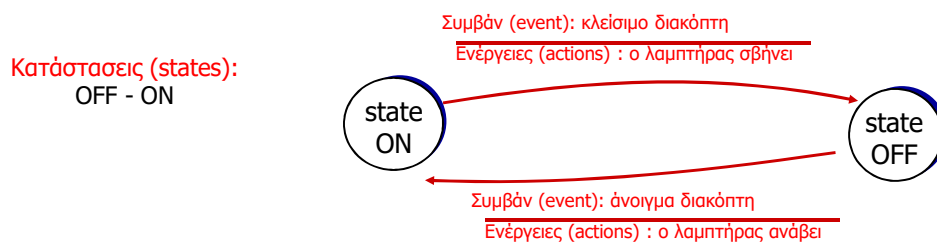
- αναπτύξουμε σταδιακά τις πλευρές του αποστολέα και του δέκτη ενός πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων [reliable **d**ata **t**ransfer (**rdt**) protocol]
- θεωρήσουμε μόνο μονόδρομη μεταφορά δεδομένων
 - Αλλά η κίνηση ελέγχου θα ρέει και προς τις δύο κατευθύνσεις !
- Χρήση μηχανών πεπερασμένων καταστάσεων [finite state machines (FSM)] για την σωστή και ολοκληρωμένη προδιαγραφή / περιγραφή ενός πρωτοκόλλου (π.χ. του rdt που υλοποιούν ο αποστολέας και ο παραλήπτης-δέκτης)



Αξιόπιστη μεταφορά δεδομένων: ξεκινώντας

Παράδειγμα: Σύστημα διακόπτη – λαμπτήρα

- Περιγραφή κανόνων λειτουργίας του συστήματος («πρωτοκόλλου»)
- Χρήση μηχανών πεπερασμένων καταστάσεων [finite state machines (FSM)]

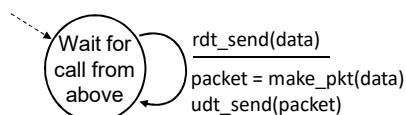


Επίπεδο Μεταφοράς

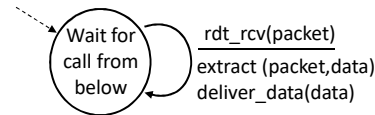
3-47

rdt1.0: αξιόπιστη μεταφορά επάνω σε αξιόπιστο κανάλι

- Το υποκείμενο κανάλι είναι πλήρως αξιόπιστο
 - Χωρίς λάθη bit
 - Χωρίς απώλειες πακέτων
- **Ξεχωριστές** FSMs για τον αποστολέα, δέκτη:



αποστολέας



δέκτης

Επίπεδο Μεταφοράς

3-48

Rdt2.0: κανάλι με σφάλματα bit

- Το υποκείμενο κανάλι ενδέχεται να αναστρέψει bits στο πακέτο
 - Checksum για την ανίχνευση σφαλμάτων bit
- Ερώτημα: πώς να γίνεται η ανάνηψη από λάθη:

Πως ανακάμπτουν οι άνθρωποι μετά από “λάθη”
κατά τη διάρκεια συνομιλίας;

Rdt2.0: κανάλι με σφάλματα bit

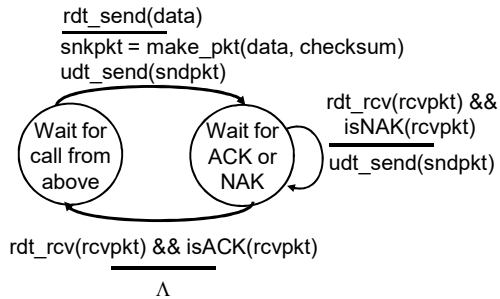
- Το υποκείμενο κανάλι ενδέχεται να αναστρέψει bits στο πακέτο
 - Άθροισμα ελέγχου (Checksum) για την ανίχνευση σφαλμάτων bit
- Ερώτημα: πώς γίνεται η ανάνηψη από λάθη;
 - **Θετικές επιβεβαιώσεις [acknowledgements (ACKs)]**: ο δέκτης λέει ρητά στον αποστολέα ότι το πακέτο λήφθηκε σωστά
 - **Αρνητικές επιβεβαιώσεις [negative acknowledgements (NAKs)]**: ο δέκτης λέει ρητά στον αποστολέα ότι το πακέτο είχε λάθη
 - Ο δέκτης αναμεταδίδει το πακέτο μόλις λάβει NAK

— stop and wait —

Αποστολέας στέλνει ένα πακέτο, μετά περιμένει να ακούσει από τον δέκτη

rdt2.0: FMS περιγραφή

sender



rdt_rcv(rcvpkt) && **corrupt(rcvpkt)**
udt_send(NAK)

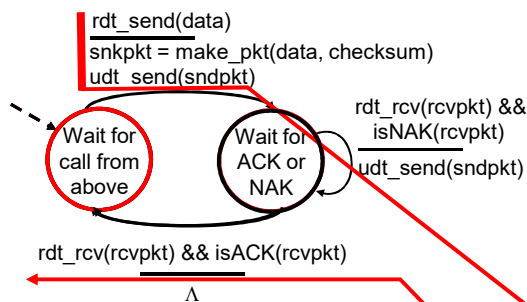


receiver

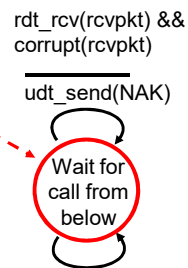
rdt_rcv(rcvpkt) && **notcorrupt(rcvpkt)**
extract(rcvpkt, data)
deliver_data(data)
udt_send(ACK)

Rdt2.0: λειτουργία χωρίς σφάλματα

sender

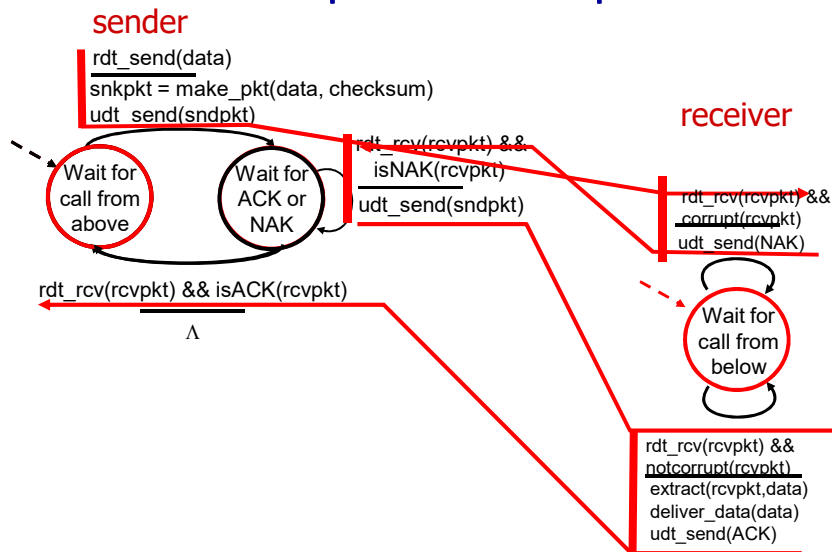


receiver



rdt_rcv(rcvpkt) && **notcorrupt(rcvpkt)**
extract(rcvpkt, data)
deliver_data(data)
udt_send(ACK)

Rdt2.0: σενάριο αλλοιωμένου πακέτου



Το rdt2.0 έχει ένα μοιραίο σφάλμα!

Τι συμβαίνει αν αλλοιωθεί (έχει σφάλμα) το ACK/NAK?

- ❑ Ο αποστολέας δεν γνωρίζει τι συνέβη στο δέκτη!
- ❑ Δεν μπορεί απλά να αναμεταδώσει (ενδεχομένως διπλά (duplicate) πακέτα - ACK). Ούτε να μην αναμεταδώσει (ενδεχομένως το πακέτο δεν θα ληφθεί ποτέ - NAK).
- ❑ Ο αποστολέας αναμεταδίδει το τρέχον πακέτο αν αλλοιωθεί το ACK/NAK και διαχειρίζεται κατάλληλα ενδεχόμενα διπλά πακέτα

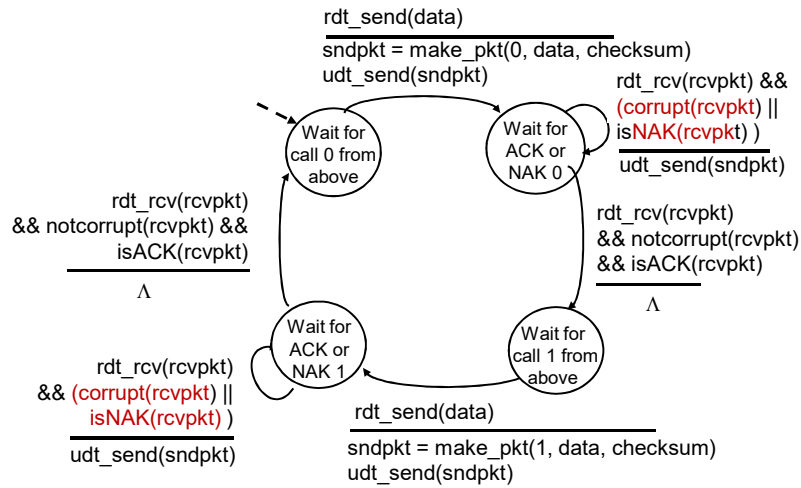
Διαχείριση διπλών πακέτων:

- ❑ Ο αποστολέας αναμεταδίδει το τρέχον πακέτο αν αλλοιωθεί το ACK/NAK
- ❑ Ο αποστολέας προσθέτει *αριθμό ακολουθίας (sequence number)* σε κάθε πακέτο
- ❑ Ο δέκτης απορρίπτει (δεν προωθεί προς τα πάνω) τα διπλά πακέτα

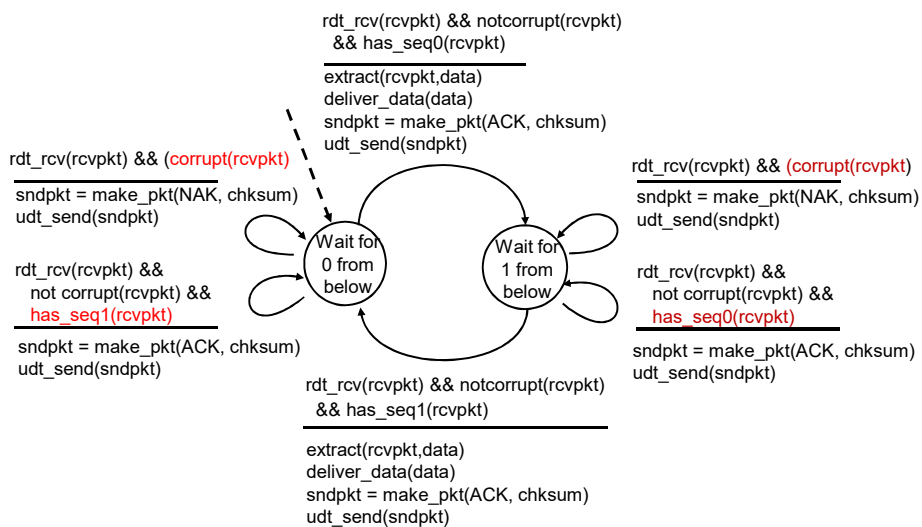
stop and wait

Αποστολέας στέλνει ένα πακέτο, μετά περιμένει να απαντήσει ο δέκτης

rdt2.1: αποστολέας, διαχειρίζεται αλλοιωμένα ACK/NAKs



rdt2.1: δέκτης, διαχειρίζεται αλλοιωμένα ACK/NAKs



rdt2.1: Συζήτηση

Αποστολέας:

- ❑ Στο πακέτο προστίθεται # ακολουθίας
- ❑ Δύο # ακολουθίας (0,1) αρκούν. Γιατί;
- ❑ Πρέπει να ελέγξει αν το ACK/NAK που έλαβε είναι αλλοιωμένο
- ❑ Διπλάσιες καταστάσεις
 - Η κατάσταση πρέπει να «θυμάται» αν το «αναμενόμενο» πακέτο έχει # ακολουθίας 0 ή 1

Δέκτης:

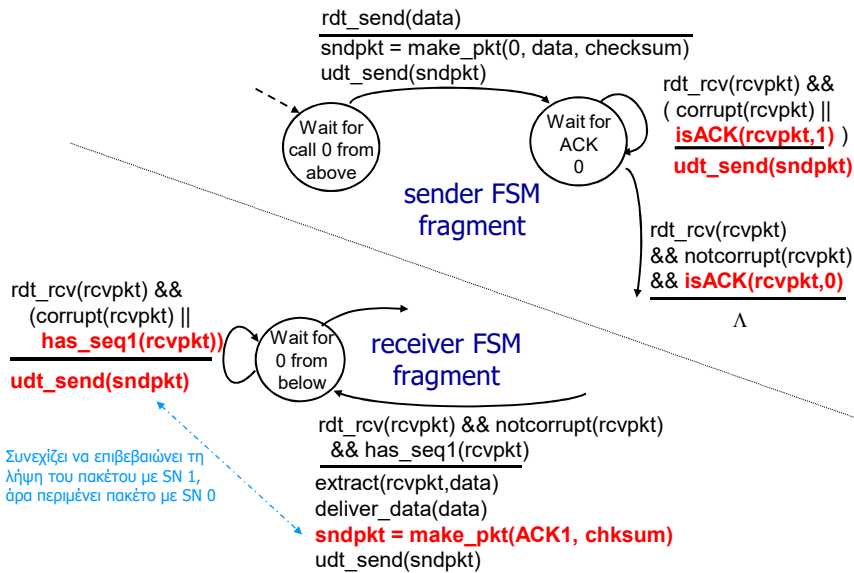
- ❑ Πρέπει να ελέγξει αν το λαμβανόμενο πακέτο είναι διπλό
 - Η κατάσταση υποδεικνύει αν αναμένεται 0 ή 1 ως # ακολουθίας πακέτου
- ❑ **Σημείωση:** ο δέκτης δεν μπορεί να γνωρίζει αν το τελευταίο του ACK/NAK ελήφθη σωστά στον αποστολέα

rdt2.2: Ένα πρωτόκολλο χωρίς NAK

- ❑ Ίδια λειτουργικότητα με το rdt2.1, χρησιμοποιώντας μόνο ACKs
- ❑ Αντί για NAK, ο δέκτης στέλνει ACK για το τελευταίο πακέτο που έλαβε σωστά
 - Ο δέκτης πρέπει ρητά να συμπεριλάβει τον # ακολουθίας του πακέτου για το οποίο γίνεται η θετική επιβεβαίωση (ACK)
- ❑ Διπλό (duplicate) ACK στον αποστολέα έχει σαν αποτέλεσμα την ίδια ενέργεια όπως το NAK: *αναμεταδίδει το τρέχον πακέτο*

Θα δούμε ότι το TCP χρησιμοποιεί αυτή την προσέγγιση (NAK-free)

rdt2.2: Αποσπάσματα αποστολέα, δέκτη



Επίπεδο Μεταφοράς 3-59

rdt3.0: κανάλια με σφάλματα και απώλειες

Νέα υπόθεση: το υποκείμενο κανάλι **μπορεί επίσης να χάσει πακέτα** (δεδομένα ή ACKs)

- Άθροισμα ελέγχου, # ακολουθίας, ACKs, αναμεταδόσεις βοηθούν αλλά δεν αρκούν

Q: Πως διαχειρίζονται οι άνθρωποι το πρόβλημα χαμένων λέξεων σε μια συνομιλία?

Επίπεδο Μεταφοράς 3-60

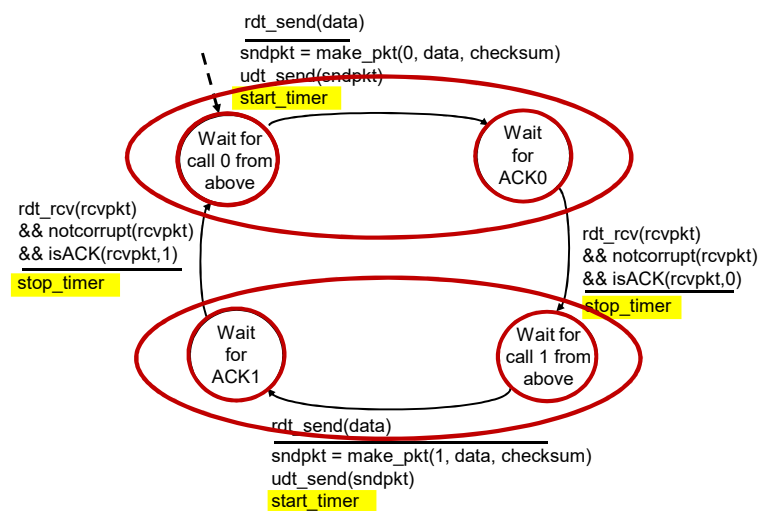
rdt3.0: κανάλια με σφάλματα και απώλειες

Προσέγγιση: ο αποστολέας περιμένει για «εύλογο» χρονικό διάστημα για το ACK

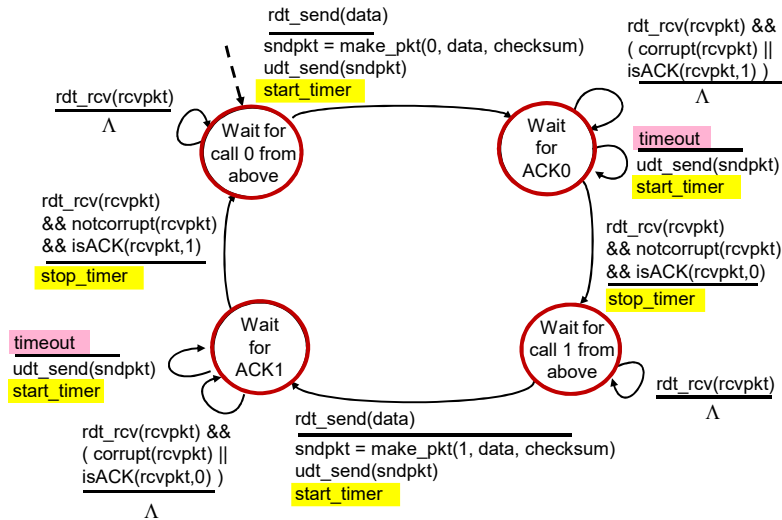
- Αναμεταδίδει αν δεν ληφθεί ACK σε αυτό το διάστημα
- Εάν πακέτο (ή ACK) απλά καθυστέρησαν (δεν χάθηκαν):
 - Επαναμετάδοση θα οδηγήσει σε διπλό (duplicate) πακέτο, αλλά οι # ακολουθίας το αντιμετωπίζουν αυτό
 - Ο δέκτης θα πρέπει να προσδιορίζει # ακολουθίας του πακέτου που επιβεβαιώνεται.
- Χρήση χρονομετρητή αντίστροφης μέτρησης για να διακόψει μετά από «εύλογο» διάστημα



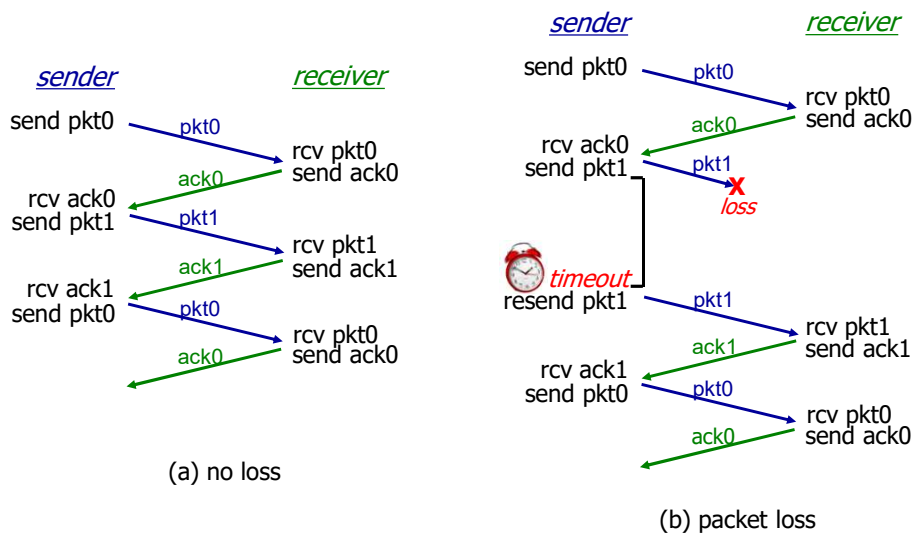
rdt3.0 αποστολέας



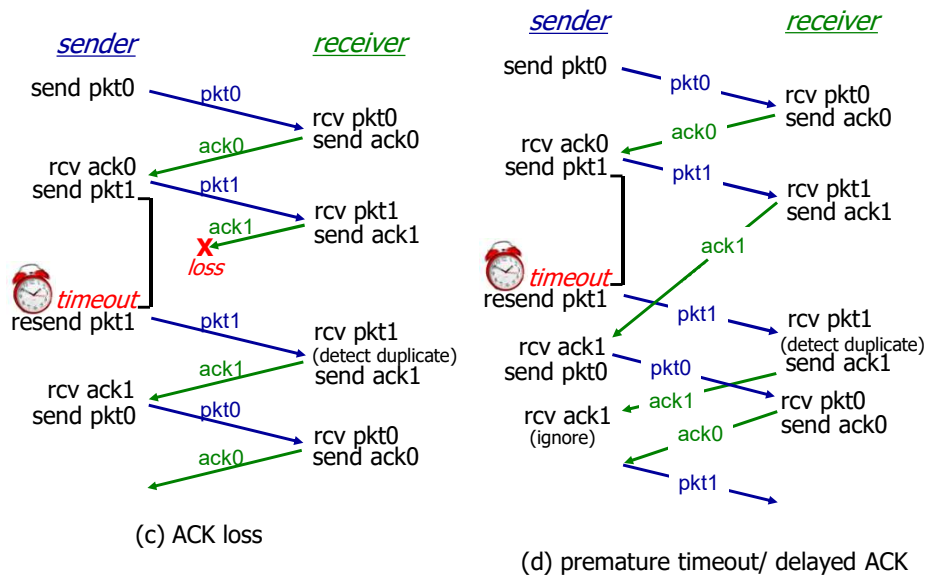
rdt3.0 αποστολέας



rdt3.0 σε δράση



rdt3.0 σε δράση

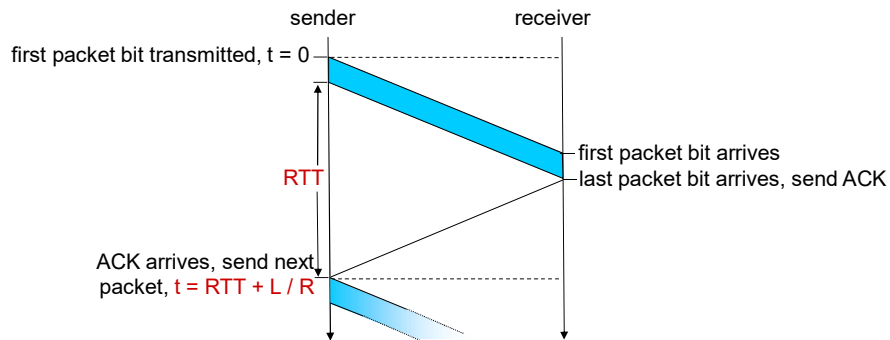


Απόδοση του rdt3.0 (stop-and-wait)

- Αποστολέας: βαθμός χρήσης (**utilization**) – ποσοστό του χρόνου που ο αποστολέας είναι απασχολημένος στέλνοντας
- Π.χ.: ζεύξη 1 Gbps, καθυστέρηση **διάδοσης** 15 ms, πακέτο 8000 bit:
 - Χρόνος για την **μετάδοση** του πακέτου στο κανάλι:

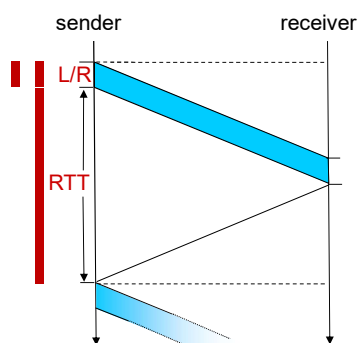
$$D_{trans} = \frac{L}{R} = \frac{8000 \text{ bits}}{10^9 \text{ bits/sec}} = 8 \text{ microsecs}$$

rdt3.0 stop-and-wait λειτουργία



rdt3.0 stop-and-wait λειτουργία

$$\begin{aligned} U_{\text{sender}} &= \frac{L/R}{RTT + L/R} \\ &= \frac{.008}{30.008} \\ &= 0.00027 \end{aligned}$$

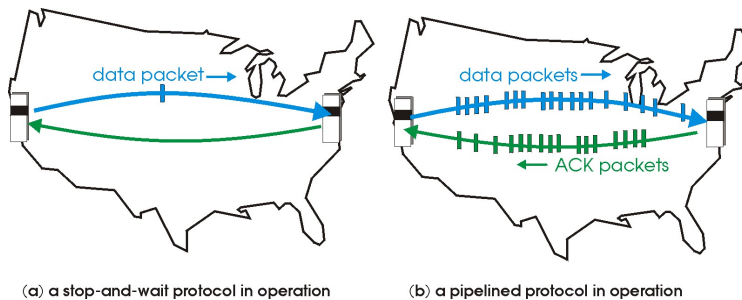


- Η απόδοση του rdt 3.0 πρωτόκολλου είναι πολύ μικρή!
- Το πρωτόκολλο περιορίζει την απόδοση της υποκείμενης υποδομής (κανάλι)

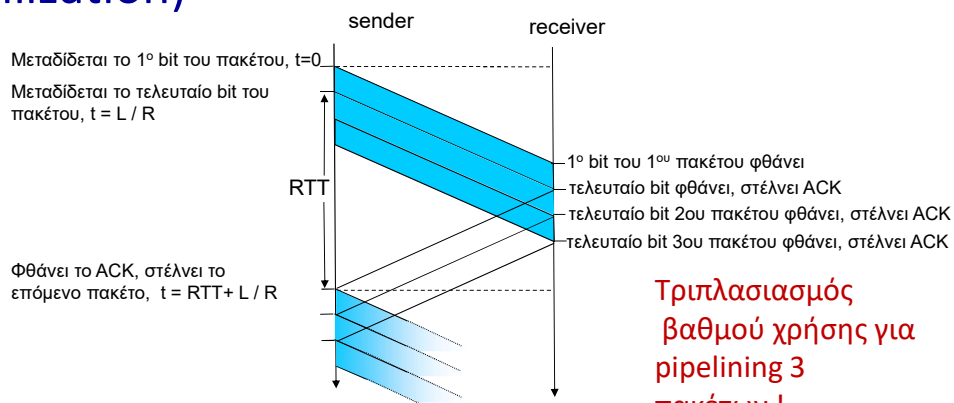
Rdt3.0: Πρωτόκολλα με διοχέτευση

Διοχέτευση (Pipelining): ο αποστολέας επιτρέπει πολλαπλά, “εν πτήση”, προς επιβεβαίωση πακέτα

- Το εύρος του αριθμού ακολουθίας πρέπει να αυξηθεί
- Ενταμίευση (buffering) στον αποστολέα ή/και στο δέκτη



Διοχέτευση (pipelining): αύξηση βαθμού χρήσης (utilization)



$$U_{\text{sender}} = \frac{3L / R}{RTT + L / R} = \frac{.0024}{30.008} = 0.00081$$

Go-Back-N: αποστολέας

- Αποστολέας: Επιτρέπεται «παράθυρο» (“window”) έως και N , συνεχόμενων μη επιβεβαιωμένων πακέτων
 - k -bit # ακολουθίας στην κεφαλίδα του πακέτου



- «**συσσωρευτικό ACK**» (“cumulative ACK”): $ACK(n)$: επιβεβαιώνει όλα τα πακέτα έως και αυτό με # ακολουθίας n
 - Με την λήψη του $ACK(n)$, το παράθυρο μετακινείται μπροστά ώστε να ξεκινά από το $n+1$
- χρονομετρητής για το αρχαιότερο «εν πτήση» πακέτο
- $timeout(n)$ (Λήξη χρόνου(n)): αναμεταδίδει το πακέτο n και όλα τα πακέτα με υψηλότερο # ακολουθίας στο παράθυρο

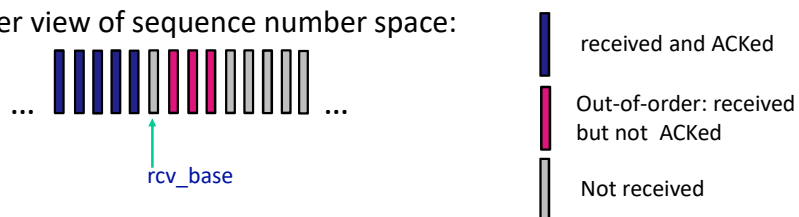
Επίπεδο Μεταφοράς

3-71

Go-Back-N: δέκτης

- ACK-only: στέλνει ACK για το πακέτο που έχει ληφθεί σωστά και φέρει τον μεγαλύτερο # ακολουθίας, υπό την προϋπόθεση ότι όλα τα πακέτα με μικρότερους # ακολουθίας έχουν ληφθεί σωστά (highest *in-order* seq #)
 - Μπορεί να δημιουργήσει διπλά ACKs
 - Χρειάζεται να θυμάται μόνο το rcv_base
- Εάν λάβει πακέτο εκτός σειράς:
 - Μπορεί να το απορρίψει ή να το ενταμιεύσει (θέμα υλοποίησης)
 - Επανα-επιβεβαιώνει το πακέτο με τον μεγαλύτερο in-order seq #

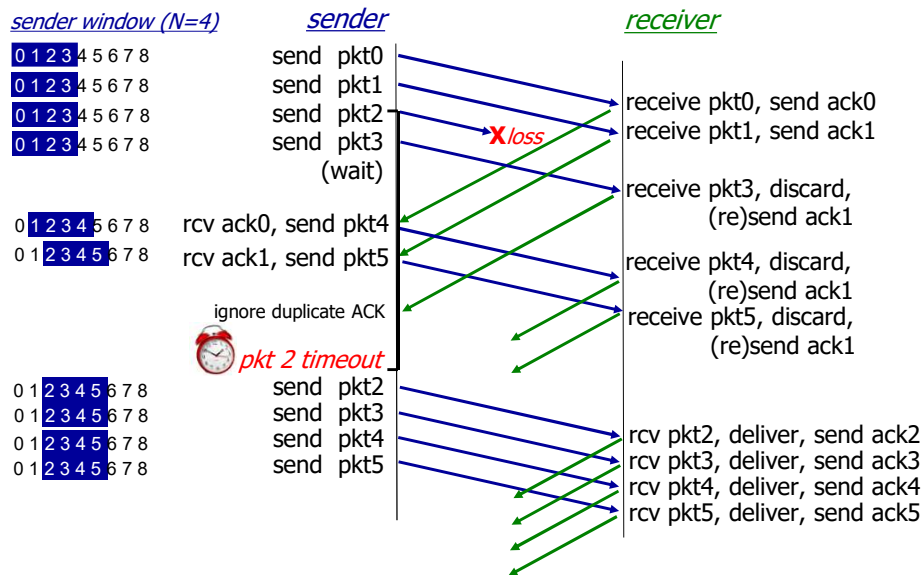
Receiver view of sequence number space:



Επίπεδο Μεταφοράς

3-72

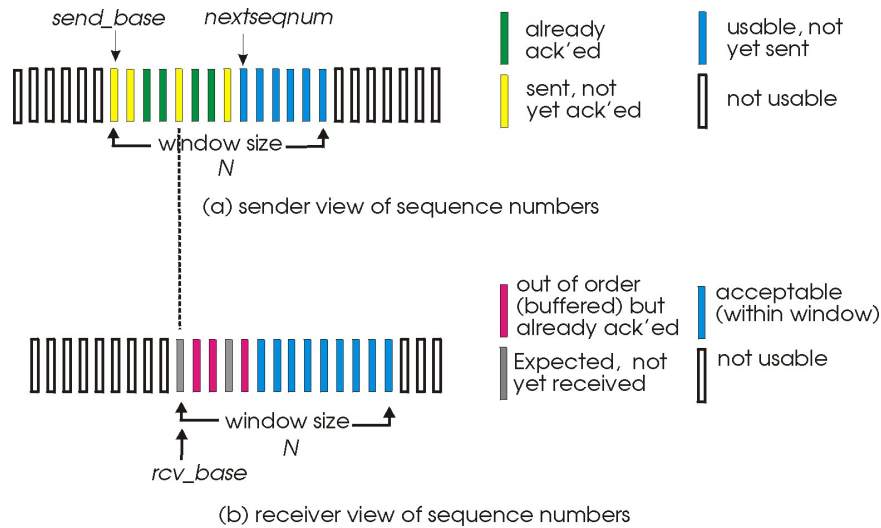
Go-Back-N: Λειτουργία



Επιλεκτική Επανάληψη (Selective Repeat)

- Ο δέκτης επιβεβαιώνει **μεμονωμένα** όλα τα σωστά ληφθέντα πακέτα
 - Ενταμιεύει πακέτα, αν χρειάζεται, ώστε τελικά να παραδώσει τα πακέτα σε σωστή σειρά στο ανώτερο επίπεδο
- Ο αποστολέας times-out/ξαναστέλνει μόνο τα πακέτα για τα οποία δεν έχει ληφθεί ACK
 - Χρονομετρητής στον αποστολέα για κάθε μη επιβεβαιωμένο πακέτο
- Παράθυρο αποστολέα
 - N συνεχόμενοι # ακολουθίας
 - Περιορίζει τους # ακολουθίας των σταλμένων, μη επιβεβαιωμένων πακέτων

Επιλεκτική επανάληψη: παράθυρα αποστολέα και δέκτη



Επίπεδο Μεταφοράς 3-75

Επιλεκτική επανάληψη: αποστολέας, δέκτης

αποστολέας

Δεδομένα από πάνω:

- Αν είναι διαθέσιμος ο επόμενος #ακολουθίας στο παράθυρο, στείλε πακέτο

timeout(n)(Λήξη χρόνου (n)):

- Ξαναστείλε το πακέτο n, επανεκκίνησε το χρονομετρητή

ACK(n) σε [sendbase, sendbase+N]:

- Σημείωσε το πακέτο n ως ληφθέν
- Αν το n είναι το μικρότερο μη επιβεβαιωμένο πακέτο, μετακίνησε τη βάση του παραθύρου στον επόμενο μη επιβεβαιωμένο # ακολουθίας

δέκτης

πακέτο n στο [rcvbase, rcvbase+N-1]

- στείλε ACK(n)
- Εκτός σειράς: ενταμίευσε
- Σε σειρά: παράδωσε (επίσης παράδωσε τα ενταμιευμένα, σε σειρά πακέτα), μετακίνησε το παράθυρο στο επόμενο πακέτο που δεν έχει ληφθεί ακόμα

πακέτο n στο [rcvbase-N, rcvbase-1]

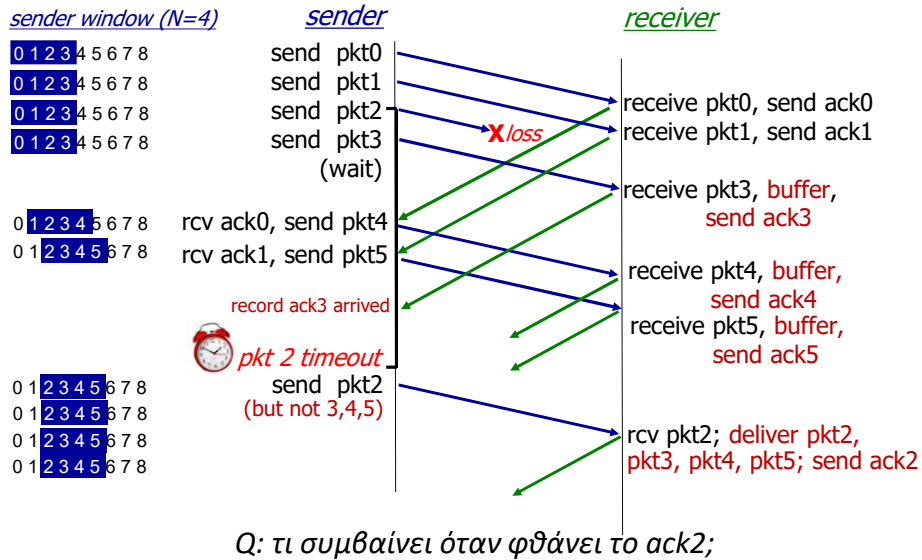
- ACK(n)

διαφορετικά:

- αγνόησε

Επίπεδο Μεταφοράς 3-76

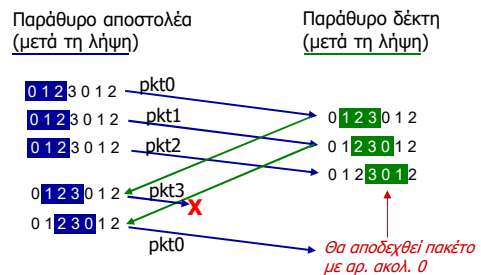
Επιλεκτική επανάληψη «εν δράσει»



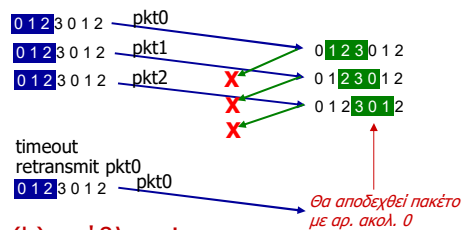
Επιλεκτική επανάληψη: δίλημμα

παράδειγμα:

- seq #s: 0, 1, 2, 3 (base 4 counting)
- window size=3



(a) Κανένα πρόβλημα!
 Λήψη του 5ου πακέτου



(b) πρόβλημα!
 Επανάμετάδοση 1ου όχι 5ου

Επιλεκτική επανάληψη: δίλημμα

παράδειγμα:

- seq #s: 0, 1, 2, 3 (base 4 counting)
- window size=3

E: Ποια πρέπει να είναι η σχέση μεταξύ sequence # size και window size για να μην παρουσιάζεται το πρόβλημα του σεναρίου?

A: Window size μέχρι το μισό του εύρους τιμών των αριθμών ακολουθίας

Παράθυρο αποστολέα
(μετά τη λήψη)

0 1 2 3
0 1 2 3
0 1 2 3

Παράθυρο δέκτη
(μετά τη λήψη)

0 1 2 3 0 1 2
0 1 2 3 0 1 2
0 1 2 3 0 1 2

- Ο δέκτης δεν μπορεί να δει τι γίνεται στην πλευρά του αποστολέα

- Ο δέκτης συμπεριφέρεται πανομοιότυπα και στις 2 περιπτώσεις!

- Υπάρχει κάποιο πρόβλημα!

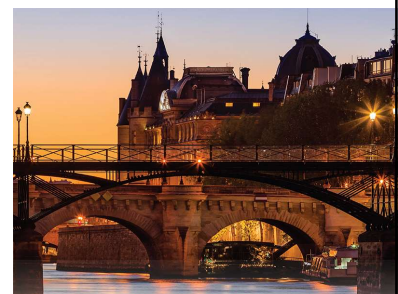
Θα δεχθεί το πακέτο με αριθμό ακολουθίας 0

Θα δεχθεί το πακέτο με αριθμό ακολουθίας 0

Επίπεδο Μεταφοράς 3-79

Επίπεδο Μεταφοράς: περίγραμμα

- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεσμική μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ **Συνδεσμική μεταφορά: TCP**
 - Δομή τμήματος
 - Αξιόπιστη μεταφορά δεδομένων
 - Έλεγχος ροής
 - Διαχείριση σύνδεσης
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP



Επίπεδο Μεταφοράς 3-80

TCP: Επισκόπηση RFCs: 793, 1122, 2018, 5681, 7323

□ Από σημείο προς σημείο:

- Ένας αποστολέας, ένας δέκτης

□ Αξιόπιστη, σε σειρά ροή από bytes:

- Χωρίς “όρια μηνυμάτων”

□ Πλήρως αμφίδρομα δεδομένα:

- Δικατευθυντική ροή δεδομένων στην ίδια σύνδεση
- MSS: maximum segment size (μέγιστο μέγεθος τμήματος, συνήθως 1460 bytes)

□ Συσσωρευτικά (cumulative) ACKs)

□ Με διοχέτευση:

- Ο έλεγχος συμφόρησης και ροής του TCP καθορίζουν το μέγεθος του παραθύρου

□ Συνδεσμική:

- Η χειραψία (handshaking) (ανταλλαγή μηνυμάτων ελέγχου) αρχικοποιεί την κατάσταση του αποστολέα και του δέκτη πριν την ανταλλαγή δεδομένων

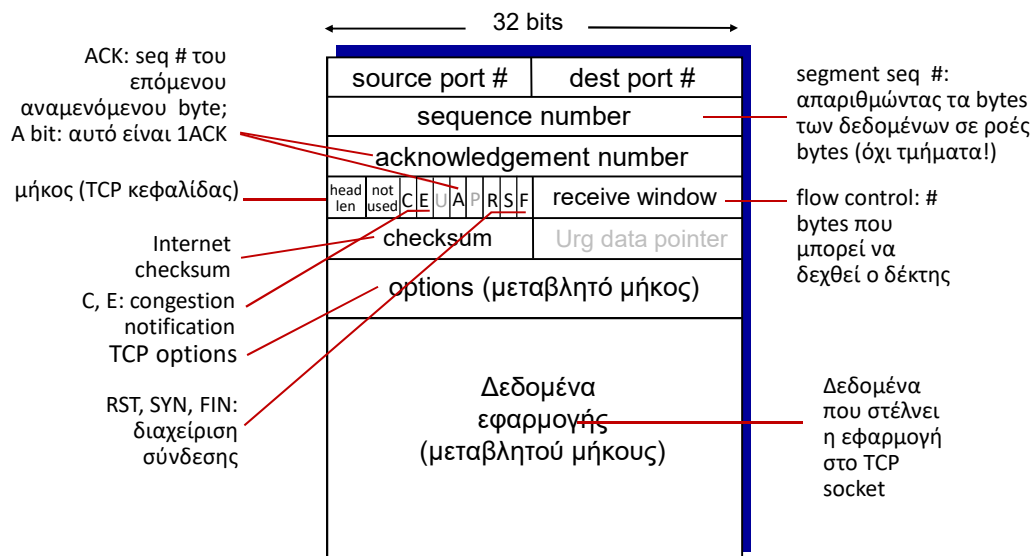
□ Ροή υπό έλεγχο:

- Ο αποστολέας δεν θα υπερφορτώσει το δέκτη

Επίπεδο Μεταφοράς

3-81

Δομή τμήματος TCP



Επίπεδο Μεταφοράς

3-82

TCP: αριθμοί ακολουθίας και ACKs

Αριθμοί ακολουθίας (Seq. #'s):

- Αριθμός του πρώτου byte των δεδομένων του τμήματος

ACKs:

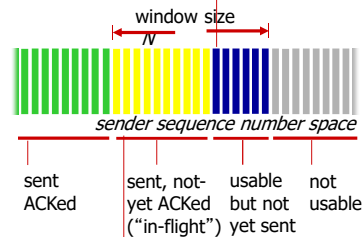
- seq # του επόμενου byte που αναμένεται από την άλλη πλευρά
- συσσωρευτικά ACK

E: πώς διαχειρίζεται ο δέκτης τα τμήματα εκτός σειράς;

- A:** η προδιαγραφή του TCP δεν καθορίζει (εναπόκειται στην υλοποίηση)

outgoing segment from sender

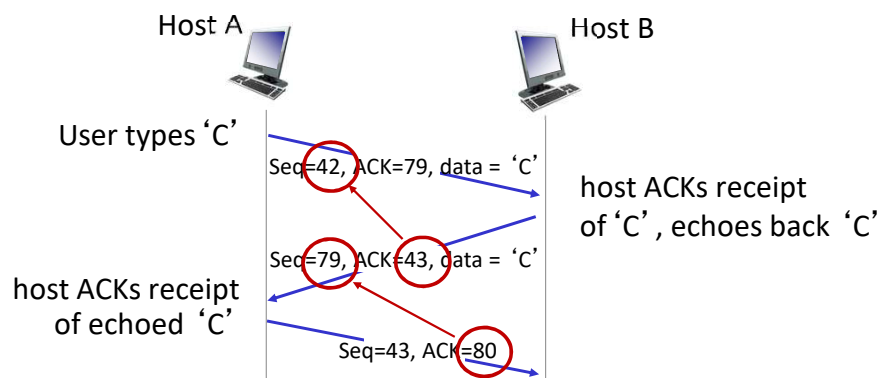
source port #	dest port #
sequence number	
acknowledgement number	
	rwnd
checksum	urg pointer



Outgoing segment from receiver

source port #	dest port #
sequence number	
acknowledgement number	
A	rwnd
checksum	urg pointer

TCP αριθμοί ακολουθίας, ACKs



simple telnet scenario

Χρόνος Διαδρομής μετ' επιστροφής (Round Trip Time) και Λήξη Χρόνου (Timeout) του TCP

E: Πώς καθορίζεται η τιμή του timeout (λήξη χρόνου) του TCP;

- ❑ Μεγαλύτερο από RTT, αλλά το RTT μεταβάλλεται !
- ❑ **πολύ σύντομο**: πρώιμο timeout, μη απαραίτητες αναμεταδόσεις
- ❑ **μεγάλης διάρκειας**: αργή αντίδραση σε απώλεια τμήματος

E: Πώς εκτιμάται το RTT;

- ❑ **SampleRTT**: χρόνος που μετριέται από τη μετάδοση του τμήματος ως την παραλαβή του ACK
 - αγνοούνται οι αναμεταδόσεις
- ❑ **To SampleRTT** θα μεταβάλλεται, θέλουμε το εκτιμώμενο RTT πιο "ομαλό"
 - μέσος όρος αρκετών πρόσφατων μετρήσεων, όχι μόνο του τρέχοντος SampleRTT

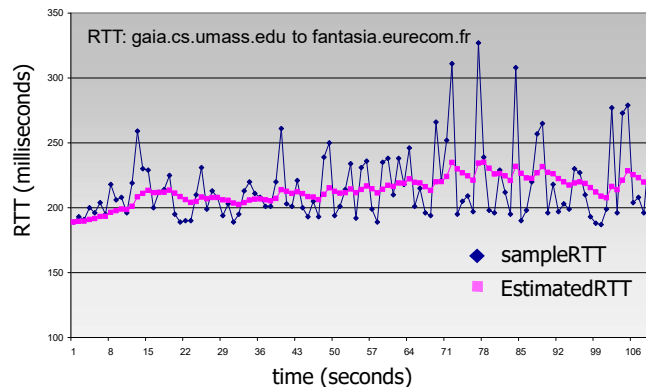
Επίπεδο Μεταφοράς

3-85

Χρόνος Διαδρομής Μετ' επιστροφής (Round Trip Time) και Λήξη Χρόνου (Timeout) του TCP

$$\text{EstimatedRTT} = (1 - \alpha) * \text{EstimatedRTT} + \alpha * \text{SampleRTT}$$

- ❑ Εκθετική σταθμισμένη κινητή μέση τιμή (Exponentially Weighted Moving Average - EWMA)
- ❑ η επίδραση των παλαιών δειγμάτων μειώνεται εκθετικά
- ❑ τυπική τιμή: $\alpha = 0.125$



Επίπεδο Μεταφοράς

3-86

Χρόνος Διαδρομής Μετ' επιστροφής (Round Trip Time) και Λήξη Χρόνου (Timeout) του TCP

Καθορισμός του timeout: EstimatedRTT συν “περιθώριο ασφαλείας”

- ο μεγάλη μεταβολή στο EstimatedRTT -> μεγαλύτερο περιθώριο ασφαλείας

$$\text{TimeoutInterval} = \underset{\substack{\uparrow \\ \text{estimated RTT}}}{\text{EstimatedRTT}} + 4 * \underset{\substack{\uparrow \\ \text{“safety margin”}}}{\text{DevRTT}}$$



DevRTT: EWMA για εκτίμηση απόκλισης SampleRTT από EstimatedRTT:

$$\text{DevRTT} = (1-\beta) * \text{DevRTT} + \beta * |\text{SampleRTT} - \text{EstimatedRTT}|$$

(τυπικά, $\beta = 0.25$)

Αποστολέας TCP (απλοποιημένος):

Συμβάν: Λήψη δεδομένων από εφαρμογή:

- δημιουργία τμήματος με # ακολουθίας
- # ακολουθίας είναι ο αριθμός του πρώτου byte δεδομένων στο τμήμα
- εκκίνηση χρονομετρητή αν δεν τρέχει ήδη
 - ο χρονομετρητής είναι σαν το χρονομετρητή του πιο παλιού μη επιβεβαιωμένου τμήματος)
 - διάστημα λήξης: TimeoutInterval

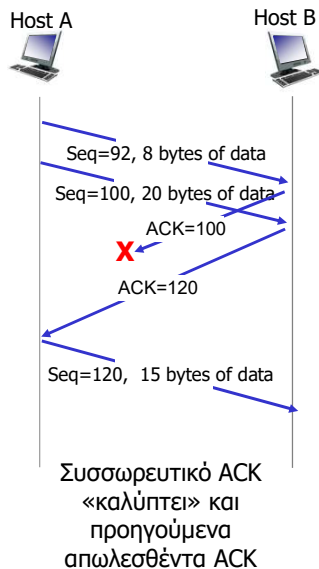
Συμβάν: Λήξη χρόνου (timeout):

- επαναμετάδοση του τμήματος που προκάλεσε το timeout
- επανεκκίνηση χρονομετρητή

Συμβάν: Λήψη ACK:

- αν επιβεβαιώνει τμήματα που δεν έχουν ήδη επιβεβαιωθεί
 - ενημέρωση του τί έχει επιβεβαιωθεί
 - εκκίνηση χρονομετρητή αν εξακολουθούν να υπάρχουν τμήματα μη επιβεβαιωμένα.

TCP: σενάριο επαναμετάδοσης (συν.)



Επίπεδο Μεταφοράς

3-91

TCP fast retransmit (γρήγορη επαναμετάδοση)

TCP fast retransmit

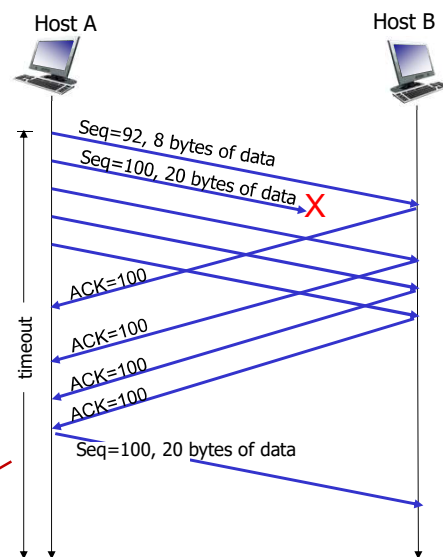
□ TCP ταχεία αναμετάδοση:

Εάν ο αποστολέας λάβει 3 duplicate ACK για τα ίδια δεδομένα, ξαναστέλνει το μη επιβεβαιωμένο τμήμα με το μικρότερο αριθμό ακολουθίας.

- Πιθανότατα το μη επιβεβαιωμένο πακέτο έχει χαθεί, οπότε μην περιμένεις τη λήξη του χρονομετρητή



Λήψη 3 duplicate ACKs υποδηλώνει ότι 3 τμήματα ελήφθησαν μετά από ένα μάλλον χαμένο πακέτο. Επομένως, αναμετάδοση!



Επίπεδο Μεταφοράς

3-92

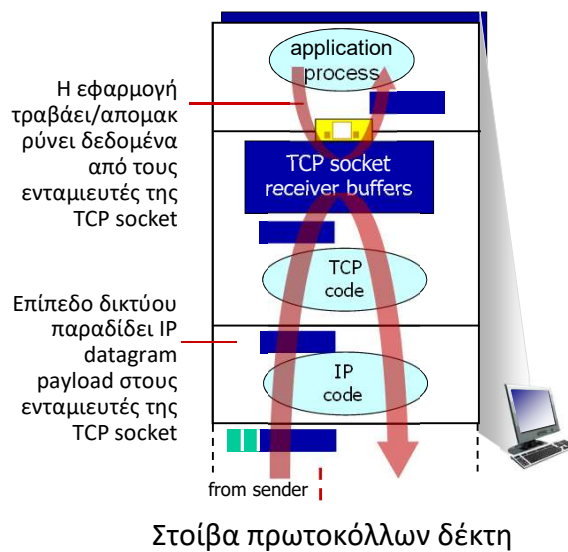
Επίπεδο Μεταφοράς: περίγραμμα

- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεδεσμένη μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεδεσμένη μεταφορά: TCP
 - Δομή τμήματος
 - Αξιόπιστη μεταφορά δεδομένων
 - Έλεγχος ροής
 - Διαχείριση σύνδεσης
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP



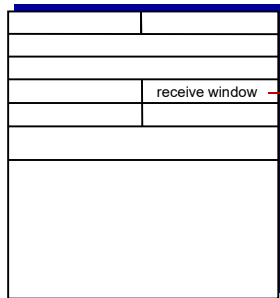
Έλεγχος ροής TCP

Q: Τι συμβαίνει όταν το επίπεδο δικτύου παραδίδει δεδομένα γρηγορότερα από ότι μπορεί να τραβήξει (διαβάσει) η διεργασία επιπέδου εφαρμογής, από τους ενταμιευτές της socket?

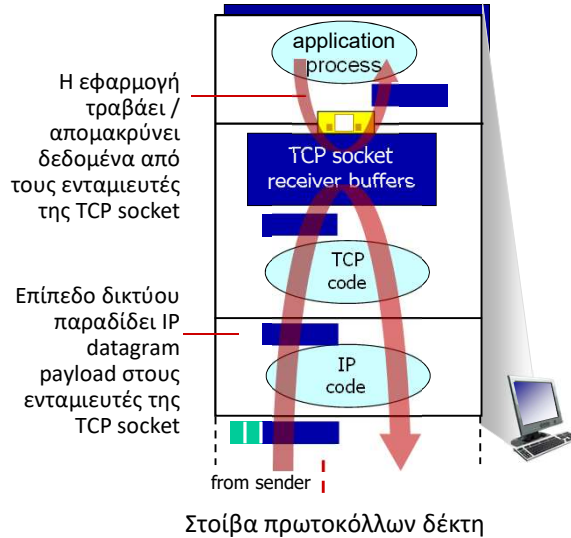


Έλεγχος ροής TCP

Q: Τι συμβαίνει όταν το επίπεδο εφαρμογής παραδίδει δεδομένα γρηγορότερα από το ρυθμό απομάκρυνσης τους από τους ενταμιευτές της socket στο επίπεδο εφαρμογής?



Έλεγχος ροής:
bytes που μπορεί να δεχθεί ο δέκτης



Έλεγχος ροής TCP

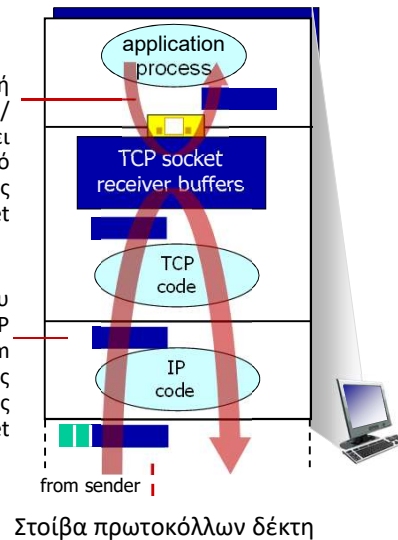
Q: Τι συμβαίνει όταν το επίπεδο εφαρμογής παραδίδει δεδομένα γρηγορότερα από το ρυθμό απομάκρυνσης τους από τους ενταμιευτές της socket στο επίπεδο εφαρμογής?

Η εφαρμογή τραβάει / απομακρύνει δεδομένα από τους ενταμιευτές της TCP socket

Επίπεδο δικτύου παραδίδει IP datagram payload στους ενταμιευτές της TCP socket

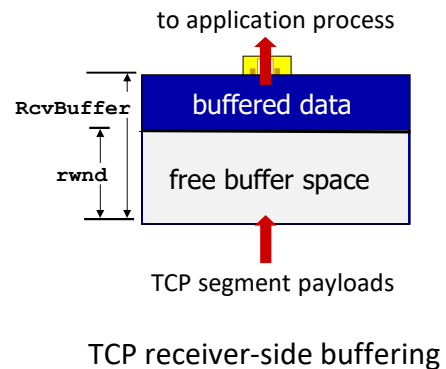
Έλεγχος ροής

Ο δέκτης ελέγχει τον αποστολέα, ώστε να μην υπερχειλίζει τους ενταμιευτές του δέκτη αποστέλλοντας πολύ γρήγορα πολλά δεδομένα



Έλεγχος ροής TCP

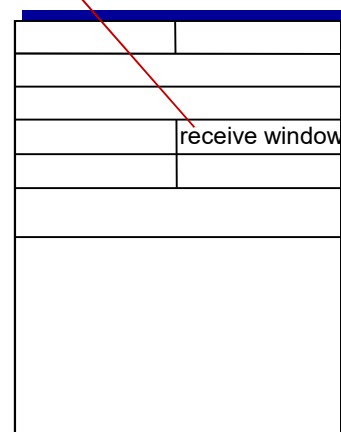
- Ο δέκτης TCP «κοινοποιεί» τον ελεύθερο χώρο του ενταμιευτή του στο πεδίο **rwnd** της TCP κεφαλίδας
 - το μέγεθος του **RcvBuffer** ορίζεται μέσω των επιλογών του socket (προκαθορισμένη τιμή 4096 bytes)
 - πολλά λειτουργικά συστήματα ρυθμίζουν αυτόματα το **RcvBuffer**
- Ο αποστολέας περιορίζει τα μη επιβεβαιωμένα («καθ' οδόν») δεδομένα στην τιμή **rwnd** του δέκτη
- εγγυάται ότι ο ενταμιευτής στον δέκτη δεν υπερχειλίζει



Έλεγχος ροής TCP

- Ο δέκτης TCP «κοινοποιεί» τον ελεύθερο χώρο του ενταμιευτή του στο πεδίο **rwnd** της TCP κεφαλίδας
 - το μέγεθος του **RcvBuffer** ορίζεται μέσω των επιλογών του socket (προκαθορισμένη τιμή 4096 bytes)
 - πολλά λειτουργικά συστήματα ρυθμίζουν αυτόματα το **RcvBuffer**
- Ο αποστολέας περιορίζει τα μη επιβεβαιωμένα («καθ' οδόν») δεδομένα στην τιμή **rwnd** του δέκτη
- εγγυάται ότι ο ενταμιευτής στον δέκτη δεν υπερχειλίζει

Έλεγχος ροής: # bytes που μπορεί να δεχθεί ο δέκτης

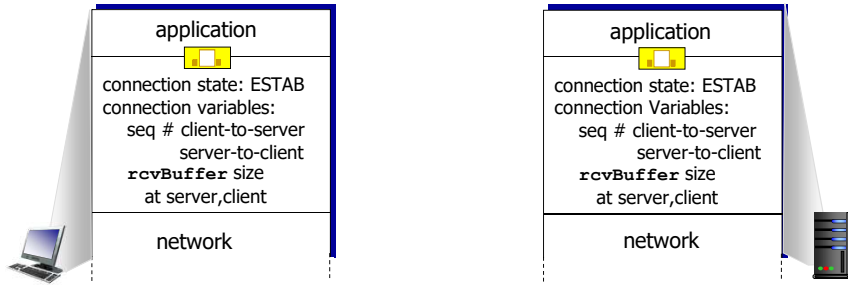


πεδία TCP τμήματος

Διαχείριση σύνδεσης TCP

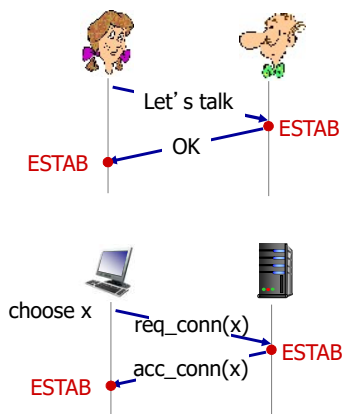
Πριν γίνει ανταλλαγή δεδομένων, ο αποστολέας κι ο δέκτης κάνουν “χειραψία”:

- ❑ συμφωνούν για τη δημιουργία σύνδεσης (ο καθένας να γνωρίζει ότι ο άλλος είναι πρόθυμος να δημιουργήσει τη σύνδεση)
- ❑ συμφωνούν στις παραμέτρους της σύνδεσης



Συμφωνία για δημιουργία σύνδεσης

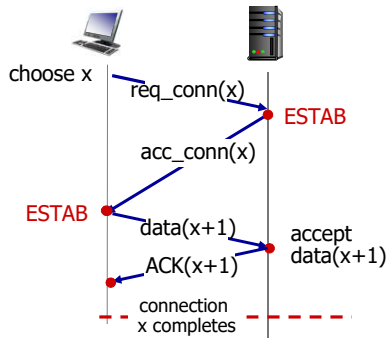
Διμερής χειραψία (2-way handshake):



E: η διμερής χειραψία δουλεύει πάντα στο Δίκτυο;

- ❑ μεταβλητές καθυστερήσεις
- ❑ επαναμεταδιδόμενα μηνύματα (π.χ. req_conn(x)) λόγω απωλειών μηνυμάτων
- ❑ αναδιάταξη (re-ordering) μηνύματος
- ❑ δεν μπορεί να δει την “άλλη” πλευρά

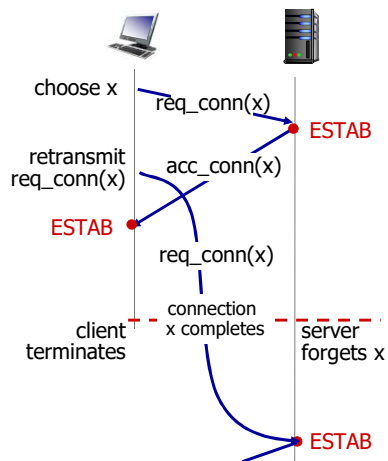
Σενάρια Διμερούς χειραψίας



No problem!

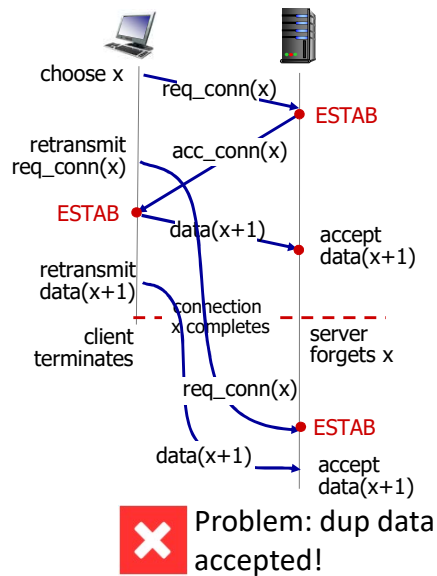


Σενάρια Διμερούς χειραψίας

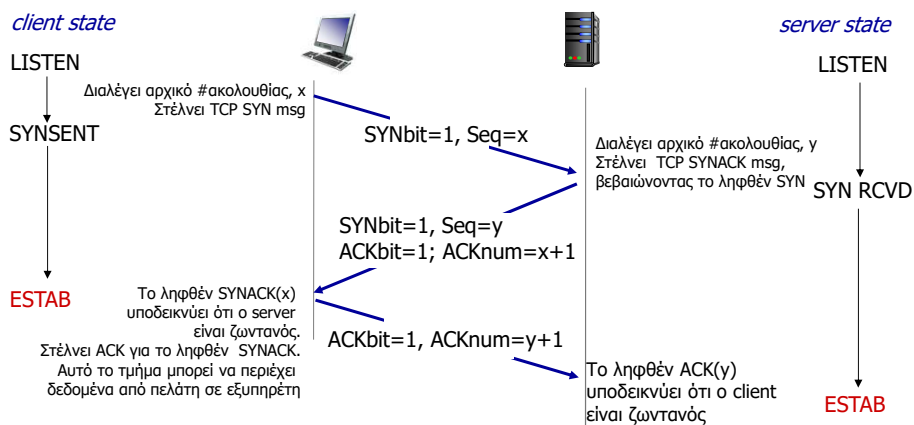


Problem: half open connection! (no client)

Σενάρια Διμερούς χειραψίας



TCP 3-μερής χειραψία



TCP: Κλείσιμο σύνδεσης

- Πελάτης και εξυπηρέτης, ο καθένας κλείνει την σύνδεση από μεριάς τους.
 - Αποστολή TCP τμήματος με FIN bit = 1 (μήνυμα FIN)
- Απαντούν στο λαμβανόμενο FIN με ACK
 - Μετά τη λήψη του FIN, η αποστολή του ACK μπορεί να συνδυαστεί με το FIN της πλευράς που επιβεβαιώνει
- Ταυτόχρονες αποστολές FIN exchanges είναι διαχειρίσιμες

Επίπεδο Μεταφοράς: περίγραμμα

- Υπηρεσίες επιπέδου μεταφοράς
- Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- Ασυνδεσμική μεταφορά: UDP
- Αρχές της αξιόπιστης μεταφοράς δεδομένων
- Συνδεσμική μεταφορά: TCP
- Αρχές ελέγχου συμφόρησης
- Έλεγχος συμφόρησης του TCP
- Εξέλιξη του Επιπέδου Μεταφοράς



Αρχές του Ελέγχου Συμφόρησης

Συμφόρηση:

- Με απλά λόγια: “πολλές πηγές που στέλνουν πολλά δεδομένα πολύ γρήγορα για να τα χειριστεί το δίκτυο”
- συμπτώματα:
 - μεγάλες καθυστερήσεις (αναμονή στους ενταμιευτές των δρομολογητών)
 - χαμένα πακέτα (υπερχειλίση ενταμιευτών στους δρομολογητές)
- διαφορετικό από τον έλεγχο ροής!
- σημαντικό πρόβλημα!



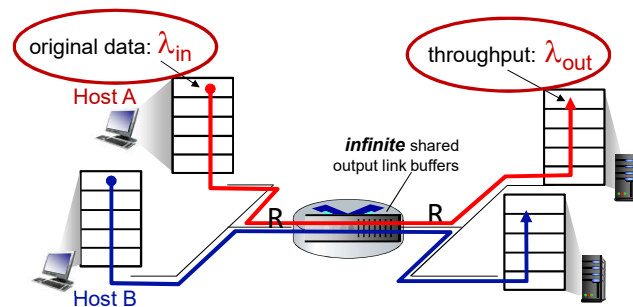
congestion control:
πολλοί στέλνουν πολύ γρήγορα

flow control: ένας αποστολέας στέλνει πολύ γρήγορα για έναν δέκτη

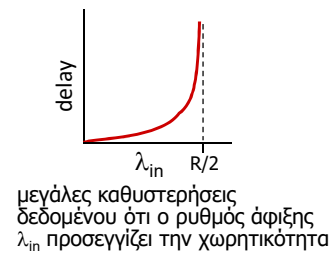
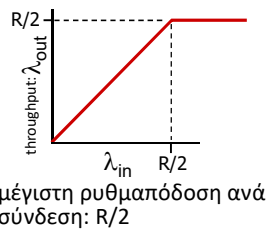
Αίτια/κόστη συμφόρησης: σενάριο 1

Απλό σενάριο:

- Ένας δρομολογητής, *άπειρους* ενταμιευτές
- χωρητικότητα εξερχόμενης / εισερχόμενης ζεύξης: R
- Δύο ροές
- Χωρίς αναμεταδόσεις

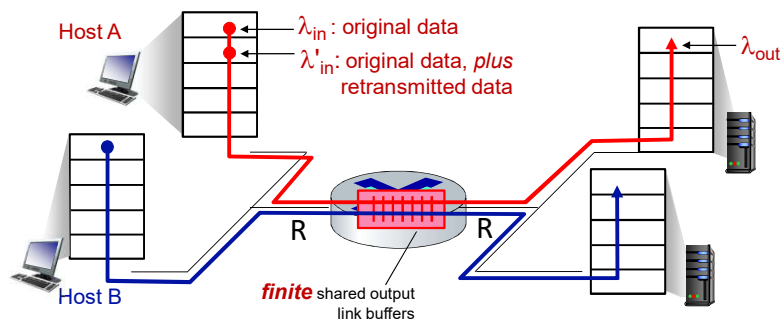


Q: Τι συμβαίνει όταν οι ρυθμοί άφιξης λ_{in} προσεγγίζουν $R/2$?



Αίτια/κόστη συμφόρησης: σενάριο 2

- Ένας δρομολογητής, *πεπερασμένοι* ενταμιευτές
- ο αποστολέας *αναμεταδίδει* τα χαμένα, καθυστερημένα (timed-out) πακέτα
 - Είσοδος επιπέδου εφαρμογής = έξοδος επιπέδου εφαρμογής: $\lambda_{in} = \lambda_{out}$
 - Είσοδος επιπέδου μεταφοράς περιλαμβάνει αναμεταδόσεις: $\lambda'_{in} \geq \lambda_{in}$



Επίπεδο Μεταφοράς 3-109

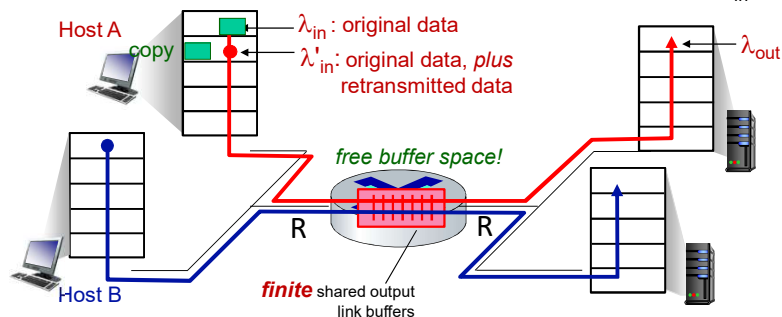
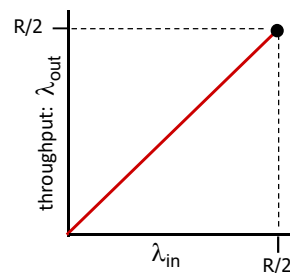
Αίτια/κόστη συμφόρησης: σενάριο 2

Εξιδανίκευση : τέλεια γνώση :

- Αποστολέας στέλνει μόνο όταν υπάρχει χώρος στους ενταμιευτές του δρομολογητή.
(Μηδενικές απώλειες και επαναμεταδόσεις $\rightarrow \lambda'_{in} = \lambda_{in}$)

$$\lambda_{in} = \lambda_{out}$$

$$\lambda'_{in} = \lambda_{in}$$



Επίπεδο Μεταφοράς 3-110

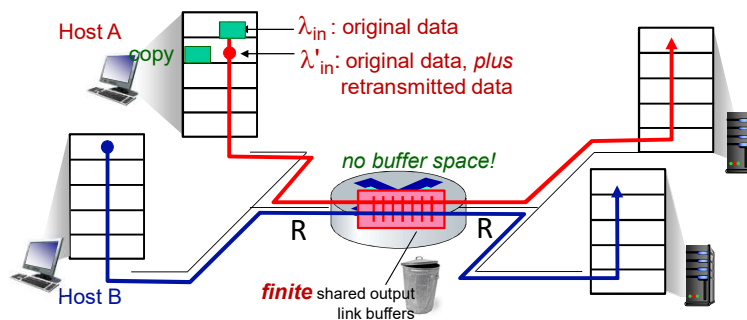
Αίτια/κόστη συμφόρησης: σενάριο 2

$$\lambda_{in} = \lambda_{out}$$

$$\lambda'_{in} \geq \lambda_{in}$$

Εξιδανίκευση: μερική τέλεια γνώση

- Πακέτα μπορεί να χαθούν (απορριφθούν στον δρομολογητή) λόγω γεμάτων ενταμιευτών
- Αποστολέας γνωρίζει πότε απορρίπτεται πακέτο: το επαναμεταδίδει μόνο εάν το πακέτο είναι γνωστό ότι χάθηκε.



Επίπεδο Μεταφοράς 3-111

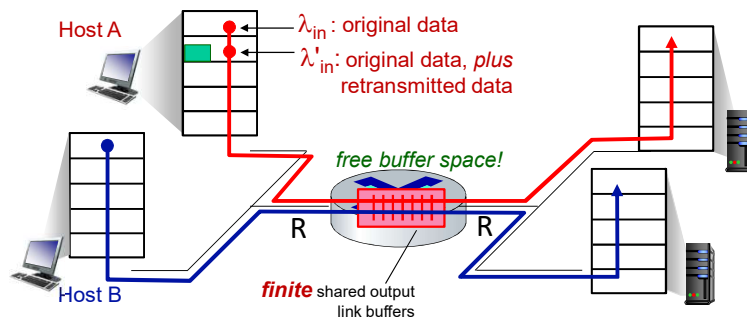
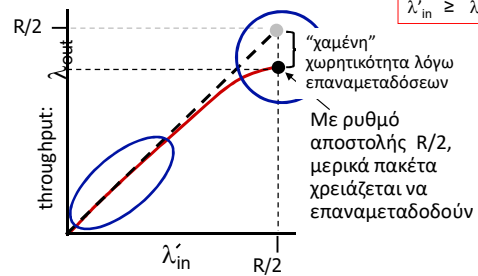
Αίτια/κόστη συμφόρησης: σενάριο 2

$$\lambda_{in} = \lambda_{out}$$

$$\lambda'_{in} \geq \lambda_{in}$$

Εξιδανίκευση: μερική τέλεια γνώση

- Πακέτα μπορεί να χαθούν (απορριφθούν στον δρομολογητή) λόγω γεμάτων ενταμιευτών
- Αποστολέας **γνωρίζει** πότε απορρίπτεται πακέτο: το επαναμεταδίδει μόνο εάν το πακέτο είναι γνωστό ότι χάθηκε.

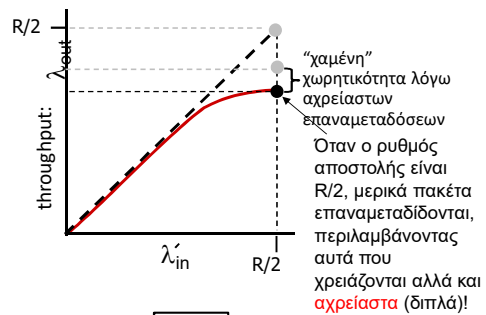
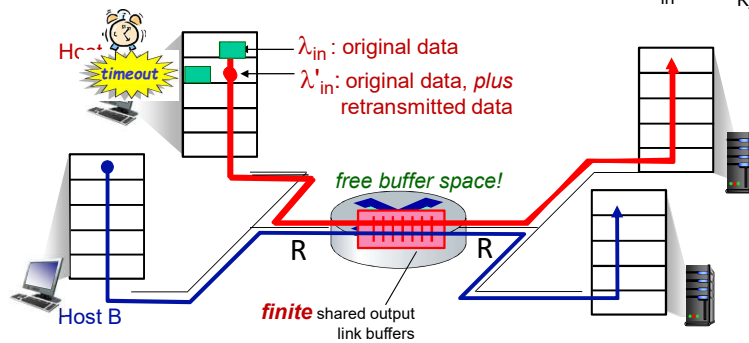


Επίπεδο Μεταφοράς 3-112

Αίτια/κόστη συμφόρησης: σενάριο 2

Ρεαλιστικό σενάριο: **αχρείαστα «διπλά»**

- Πακέτα μπορεί να **χαθούν** (απορριφθούν στους δρομολογητές) – χρειάζονται επαναμετάδοση
- Αλλά μπορεί ο χρονομετρητής αποστολέα να λήξει πρώωρα, και να **αποσταλούν δύο αντίγραφα** που θα ληφθούν και τα δύο.



Αίτια/κόστη συμφόρησης: σενάριο 2

“κόστος” συμφόρησης:

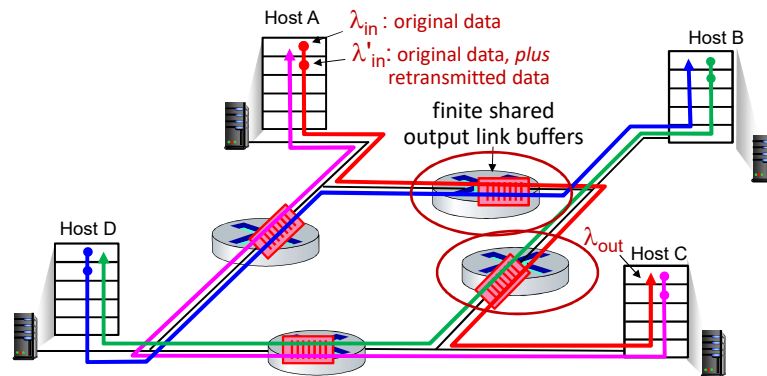
- περισσότερη δουλειά (επαναμεταδόσεις) για την ίδια ρυθμαπόδοση στο δέκτη
- Αχρείαστες επαναμεταδόσεις: η ζεύξη μεταφέρει πολλαπλά αντίγραφα ενός πακέτου
 - Μειώνοντας τη μέγιστη ρυθμαπόδοση που μπορεί να επιτευχθεί

Αίτια/κόστη συμφόρησης: σενάριο 3

- Τέσσερις αποστολές
- Διαδρομές πολλών τμημάτων
- Λήξη χρόνου / επαναμετάδοση

Q: τι συμβαίνει όταν αυξάνονται τα λ_{in} and λ_{in}' ?

A: καθώς το κόκκινο λ_{in}' αυξάνει, όλα τα μπλε πακέτα που φτάνουν στην τελευταία ουρά, απορρίπτονται και η μπλε ρυθμαπόδοση $\rightarrow 0$



Επίπεδο Μεταφοράς 3-115

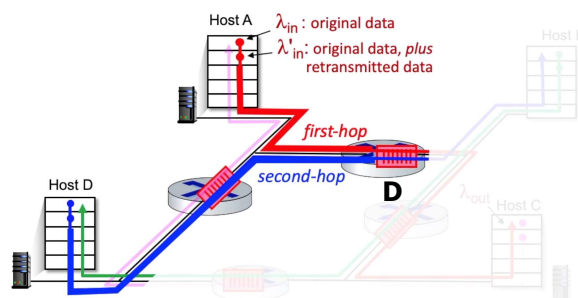
Αίτια/κόστη συμφόρησης: σενάριο 3

- Τέσσερις αποστολές
- Διαδρομές πολλών τμημάτων
- Λήξη χρόνου / επαναμετάδοση

Q: τι συμβαίνει όταν αυξάνονται τα λ_{in} and λ_{in}' ?

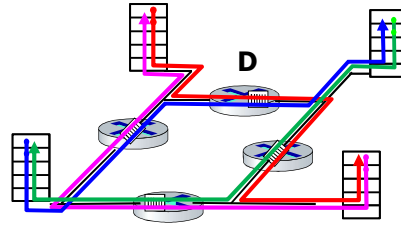
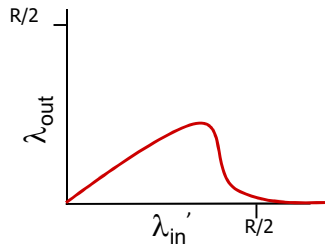
A: καθώς το κόκκινο λ_{in}' αυξάνει πάνω από $R/2$, στον δρομολογητή D η ροή σε «first-hop» (κόκκινη) θα επικρατήσει πλήρως της ροής σε «second-hop» (μπλε) η οποία θα λάβει ασυμπτωτικά μηδενική χωρητικότητα.

▪ timeout/retransmit



Επίπεδο Μεταφοράς 3-116

Αίτια/κόστη συμφόρησης: σενάριο 3

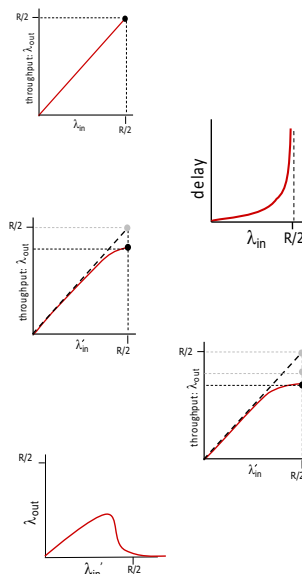


Ένα άλλο «κόστος» της συμφόρησης:

- Όταν απορρίπτεται ένα πακέτο, η χωρητικότητα μετάδοσης και αποθήκευσης που χρησιμοποιήθηκε σε προηγούμενους (upstream) δρομολογητές για το συγκεκριμένο πακέτο, πάει χαμένη!

Αίτια/κόστη συμφόρησης: επισκόπηση

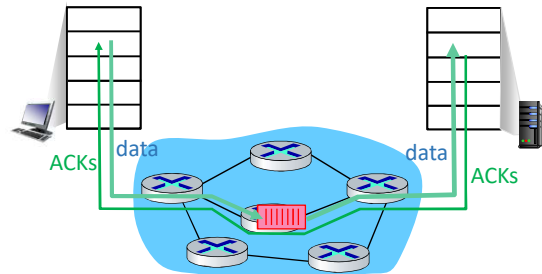
- Η ρυθμαπόδοση δεν ξεπερνά ποτέ τη χωρητικότητα
- Η καθυστέρηση αυξάνεται πολύ όταν προσεγγίζεται η χωρητικότητα.
- Απώλειες / επαναμεταδόσεις μειώνουν την ρυθμαπόδοση
- Αχρειαστα διπλά πακέτα μειώνουν περαιτέρω την ρυθμαπόδοση
- upstream χωρητικότητα μετάδοσης / αποθήκευσης πάνε χαμένες αναφορικά με πακέτα που τελικά χάνονται downstream



Προσεγγίσεις για τον έλεγχο συμφόρησης

Έλεγχος συμφόρησης «απ' άκρη σε άκρη» (end-end):

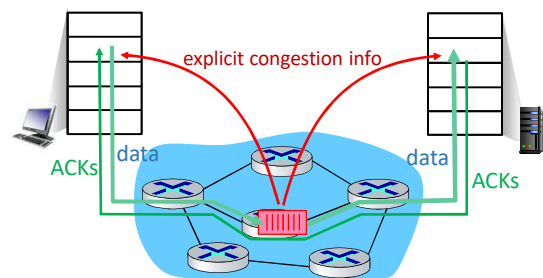
- ❑ Χωρίς άμεση ανάδραση από το δίκτυο
- ❑ Συμφόρηση «**συμπεραίνεται**» από τις παρατηρούμενες απώλειες / καθυστερήσεις
- ❑ είναι η προσέγγιση που ακολουθεί το TCP



Προσεγγίσεις για τον έλεγχο συμφόρησης

Έλεγχος συμφόρησης υποβοηθούμενος από το δίκτυο (network-assisted):

- ❑ Οι δρομολογητές παρέχουν **άμεση** ανάδραση (direct feedback) στα τερματικά (αποστολέα/δέκτη) όταν η ροή τους περνάει από δρομολογητές με συμφόρηση
- ❑ Μπορεί να αναφέρεται συγκεκριμένο επίπεδο συμφόρησης ή και να υποδειχνεται ο ρυθμός με τον οποίο πρέπει να μεταδίδει ο αποστολέας.
- ❑ Πρωτόκολλα TCP ECN, ATM, DECBbit



Επίπεδο Μεταφοράς: περίγραμμα

- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεδεσμένη μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεδεσμένη μεταφορά: TCP
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP
- ❑ Εξέλιξη του Επιπέδου Μεταφοράς



Έλεγχος συμφόρησης TCP - επισκόπηση

«classic» TCP : βασισμένο σε απώλειες, από άκρη σε άκρη

- Προσθετική αύξηση, πολλαπλασιαστική μείωση
- «αργή εκκίνηση»
- CUBIC

Εμπλουτισμένοι μηχανισμοί TCP

- TCP έλεγχος συμφόρησης βασισμένος στην καθυστέρηση
- Ρητή ειδοποίηση συμφόρησης (explicit congestion notification)

Δικαιοσύνη TCP

Έλεγχος συμφόρησης TCP: προσθετική αύξηση, πολλαπλασιαστική μείωση (AIMD)

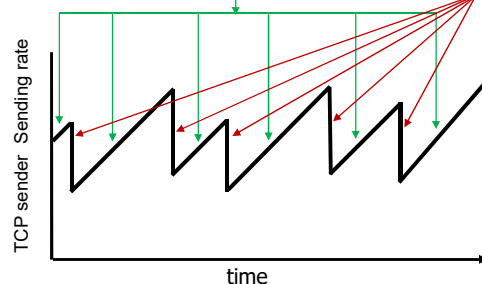
- **Προσέγγιση:** αποστολές μπορούν να αυξάνουν το ρυθμό αποστολής μέχρι να υπάρξει απώλεια πακέτου (συμφόρηση), τότε μειώνουν τον ρυθμό.

Additive Increase

Αύξηση ρυθμού αποστολής κατά 1 μέγιστο μήκος τμήματος κάθε RTT μέχρι να ανιχνευθεί απώλεια

Multiplicative Decrease

Μείωση ρυθμού αποστολής στο μισό κάθε φορά που ανιχνεύεται απώλεια



AIMD «πριονωτή» συμπεριφορά: βολιδοσκοπώντας (*probing*) τη διαθέσιμη χωρητικότητα

Επίπεδο Μεταφοράς

3-123

TCP AIMD: περισσότερα

πολλαπλασιαστική μείωση: ο ρυθμός αποστολής

- Μειώνεται στο μισό όταν ανιχνευθεί απώλεια μέσω τριπλού ACK (TCP Reno)
- Μειώνεται στο 1 MSS (maximum segment size) όταν ανιχνευθεί απώλεια μέσω timeout (TCP Tahoe)

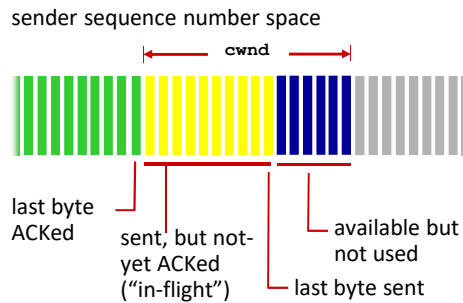
Γιατί **AIMD**?

- AIMD – καταναμημένος, ασύγχρονος αλγόριθμος – ο οποίος:
 - Βελτιστοποιεί τους ρυθμούς των ροών σε συνθήκες συμφόρησης σε όλο το δίκτυο (network wide)!
 - Έχει επιθυμητές ιδιότητες ευστάθειας.

Επίπεδο Μεταφοράς

3-124

Έλεγχος συμφόρησης TCP: λεπτομέρειες



Συμπεριφορά TCP αποστολέα:

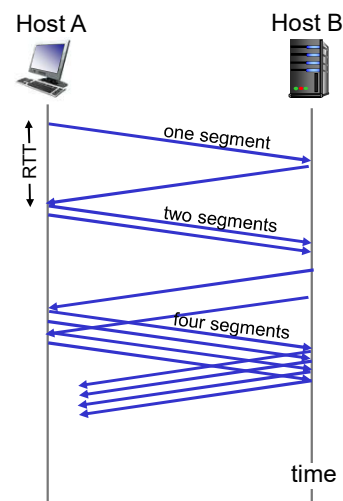
- *περίπου*: αποστολή $cwnd$ bytes, αναμονή RTT για τα ACKS, και μετά αποστολή περισσότερων bytes

$$\text{TCP ρυθμός} \approx \frac{cwnd}{RTT} \text{ bytes/sec}$$

- Αποστολέας περιορίζει τη μετάδοση: $\text{LastByteSent} - \text{LastByteAcked} \leq cwnd$
- $cwnd$ προσαρμόζεται δυναμικά, σε συνάρτηση με την παρατηρούμενη συμφόρηση δικτύου (υλοποιώντας τον έλεγχο συμφόρησης του TCP)

Αργή εκκίνηση του TCP (slow start)

- Όταν ξεκινά η σύνδεση, ο **ρυθμός αυξάνεται εκθετικά** μέχρι την πρώτη απώλεια:
 - αρχικά $cwnd = 1$ MSS
 - το $cwnd$ **αυξάνεται κατά 1** με κάθε ACK που λαμβάνεται
 - → το $cwnd$ **διπλασιάζεται** σε κάθε RTT
- **Σύνοψη**: αρχικός ρυθμός αργός, αλλά ανεβαίνει εκθετικά γρήγορα



Σημείωση: **Γραμμική αύξηση** ($cwnd + 1$ κάθε RTT)
Εκθετική αύξηση ($cwnd + 1$ με κάθε ACK)

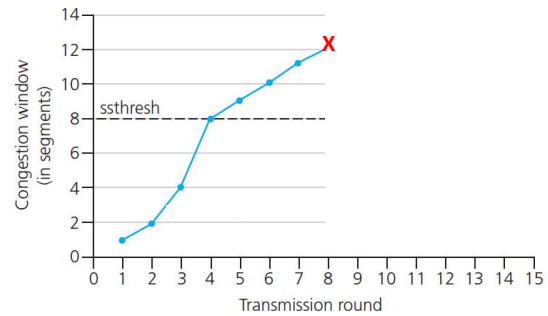
TCP: Μετάβαση από αργή εκκίνηση σε αποφυγή συμφόρησης (Slow Start to Congestion Avoidance)

Q: Πότε θα πρέπει να γίνει η αλλαγή από εκθετική σε γραμμική αύξηση (+1 κάθε RTT);

A: Όταν το **cwnd** γίνει το 1/2 της τιμής του πριν το timeout.

Υλοποίηση:

- Μεταβλητό κατώφλι-Threshold (ssthresh -SlowStartThreshold)
- Σε γεγονός απώλειας (X), το Threshold τίθεται στο 1/2 του **cwnd** πριν το γεγονός της απώλειας



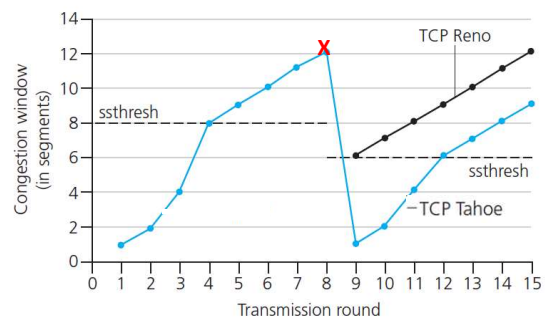
TCP: Μετάβαση από αργή εκκίνηση σε αποφυγή συμφόρησης (Slow Start to Congestion Avoidance)

Q: Πότε θα πρέπει να γίνει η αλλαγή από εκθετική σε γραμμική αύξηση (+1 κάθε RTT);

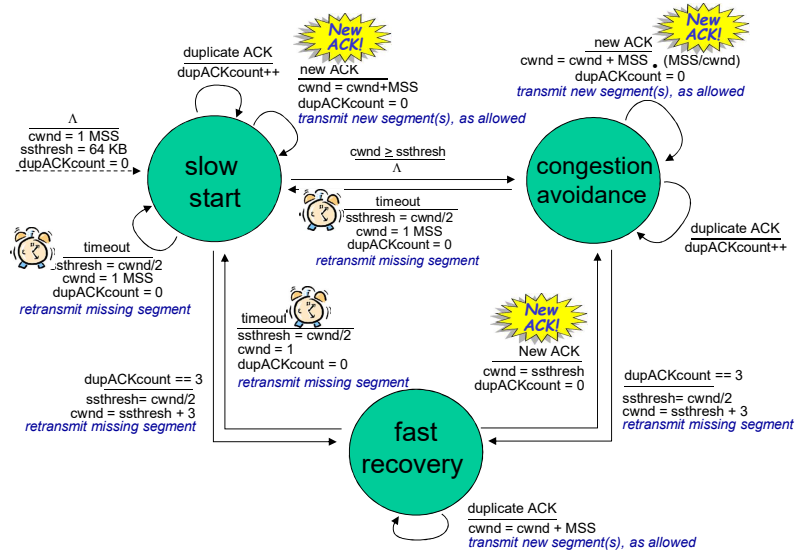
A: Όταν το **cwnd** γίνει το 1/2 της τιμής του πριν το timeout.

Υλοποίηση:

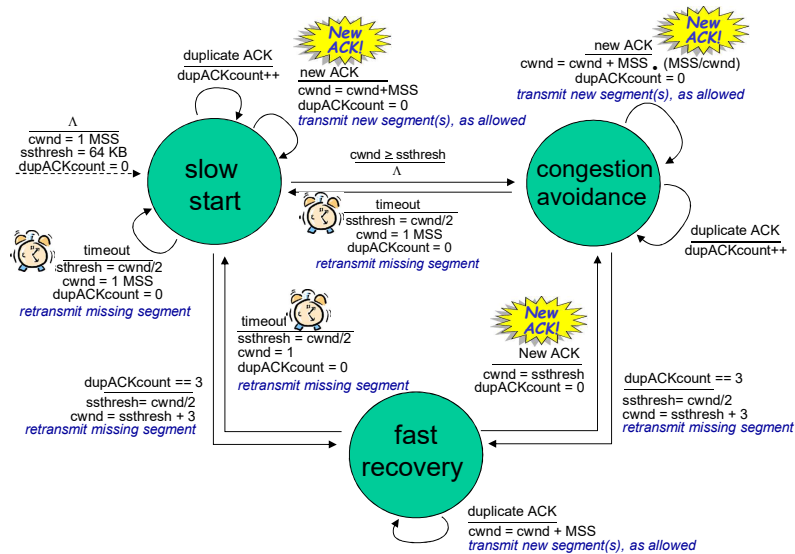
- Μεταβλητό κατώφλι-Threshold (ssthresh -SlowStartThreshold)
- Σε γεγονός απώλειας (X), το Threshold τίθεται στο 1/2 του **cwnd** πριν το γεγονός της απώλειας



Σύνοψη: Έλεγχος συμφόρησης TCP

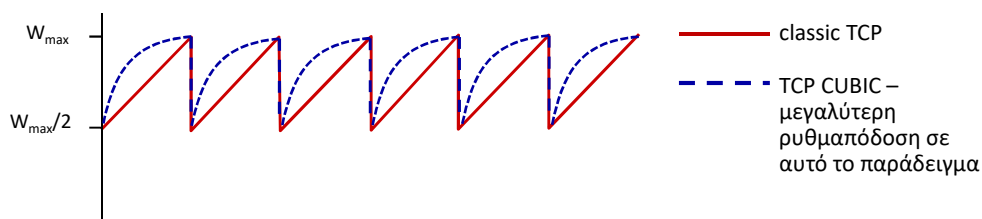


Σύνοψη: Έλεγχος συμφόρησης TCP



TCP CUBIC

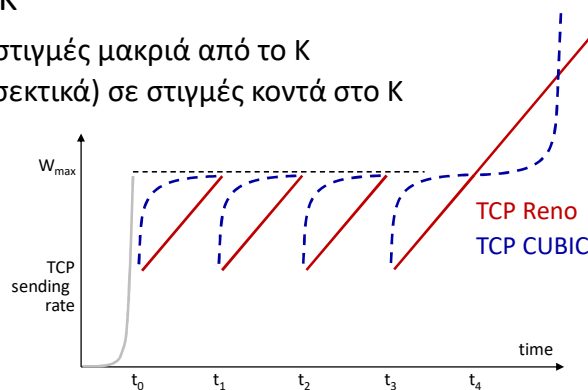
- Υπάρχει καλύτερος τρόπος να «βολιδοσκοπίσουμε» για διαθέσιμη χωρητικότητα?
 - Με βάση τη διαίσθηση / κατανόηση:
 - W_{max} : ρυθμός μετάδοσης όταν ανιχνεύεται απώλεια λόγω συμφόρησης
 - Η κατάσταση της συμφορημένης ζεύξης πιθανόν (?) δεν έχει αλλάξει ιδιαίτερα
 - Μετά τον υποδιπλασιασμό του ρυθμού / παραθύρου λόγω απώλειας, αρχικά προσέγγισε το W_{max} *γρηγορότερα*, αλλά στη συνέχεια με *αργότερο* ρυθμό



Επίπεδο Μεταφοράς 3-131

TCP CUBIC

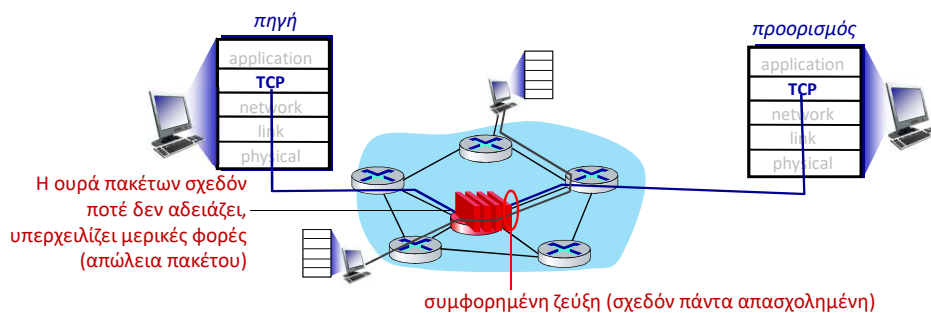
- K : χρονική στιγμή που το TCP παράθυρο προσεγγίζει το W_{max}
 - K μπορεί να ρυθμιστεί
- Αύξησε το W με τον *κύβο* της απόστασης μεταξύ της τρέχουσας χρονικής στιγμής και του K
 - μεγαλύτερες αυξήσεις σε στιγμές μακριά από το K
 - μικρότερες αυξήσεις (προσεκτικά) σε στιγμές κοντά στο K
- TCP CUBIC είναι στάνταρ επιλογή σε Linux, πλέον δημοφιλές TCP για δημοφιλείς Web servers



Επίπεδο Μεταφοράς 3-132

TCP και η «συμφορημένη ζεύξη (bottleneck link)»

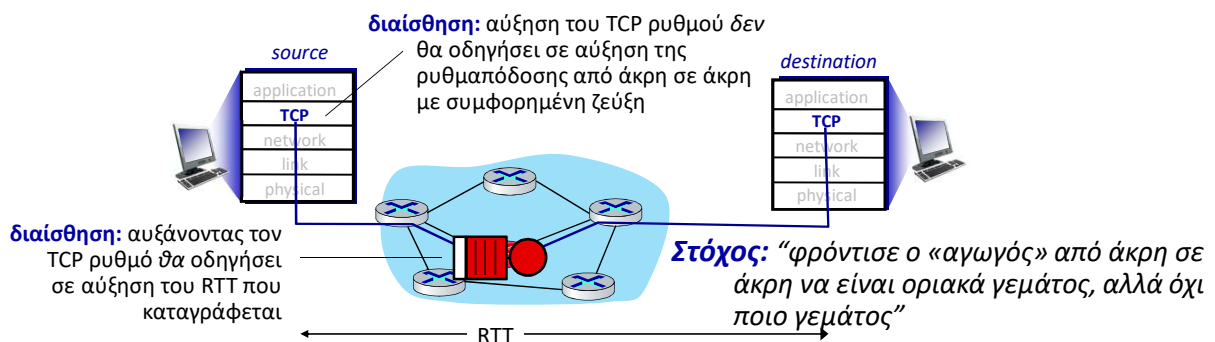
- TCP (classic, CUBIC) αυξάνει το ρυθμό μέχρι να συμβεί απώλεια πακέτου στην έξοδο κάποιου δρομολογητή: τη **συμφορημένη ζεύξη**



Επίπεδο Μεταφοράς 3-133

TCP και η «συμφορημένη ζεύξη (bottleneck link)»

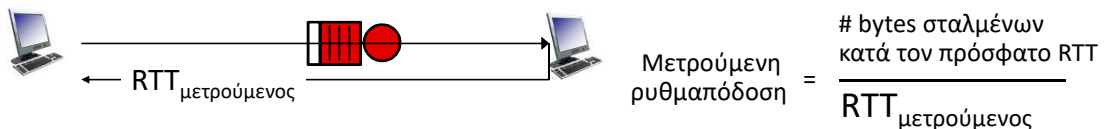
- TCP (classic, CUBIC) αυξάνει το ρυθμό μέχρι να συμβεί απώλεια πακέτου στην έξοδο κάποιου δρομολογητή: τη **συμφορημένη ζεύξη**
- Κατανοώντας συμφόρηση: χρήσιμο να εστιάσουμε στη συμφορημένη ζεύξη



Επίπεδο Μεταφοράς 3-134

Έλεγχος συμφόρησης TCP βασισμένος στην καθυστέρηση

Διατηρώντας τον αγωγό από αποστολέα σε παραλήπτη «οριακά γεμάτο αλλά όχι ποιο γεμάτο»: η συμφορημένη ζεύξη μεταδίδει συνέχεια, αποφεύγοντας μεγάλες καθυστερήσεις / αποθήκευση



Προσέγγιση βασισμένη στην (μετρούμενη) καθυστέρηση:

- RTT_{\min} – ελάχιστη τιμή του RTT (μονοπάτι με ζεύξεις χωρίς συμφόρηση)
- Ρυθμαπόδοση όταν δεν υπάρχει συμφόρηση και το παράθυρο έχει τιμή cwnd ισούται με $\text{cwnd}/\text{RTT}_{\min}$

Εάν η μετρούμενη ρυθμαπόδοση είναι «πολύ κοντά» σε αυτήν χωρίς συμφόρηση

αύξησε γραμμικά το cwnd /* μη συμφορημένο μονοπάτι */

Διαφορετικά, εάν η μετρούμενη ρυθμαπόδοση είναι «πολύ κάτω» από αυτήν χωρίς συμφόρηση

μείωσε γραμμικά το cwnd /* συμφορημένο μονοπάτι */

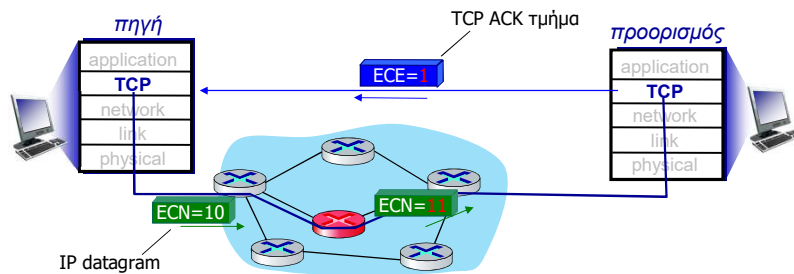
Έλεγχος συμφόρησης TCP βασισμένος στην καθυστέρηση

- Έλεγχος συμφόρησης χωρίς να προκαλεί / εξαναγκάζει σε απώλειες
- Μεγιστοποιεί τη ρυθμαπόδοση («διατηρώντας τον αγωγό οριακά γεμάτο,») ενώ διατηρείται η καθυστέρηση σε χαμηλά επίπεδα («...αλλά όχι ποιο γεμάτο»)
- Αρκετές υλοποιήσεις TCP ακολουθούν την προσέγγιση βασισμένη στην καθυστέρηση
 - BBR (Bottleneck Bandwidth and Round-trip propagation time) έχει αναπτυχθεί στο εσωτερικό δίκτυο κορμού της Google

Explicit congestion notification (ECN)

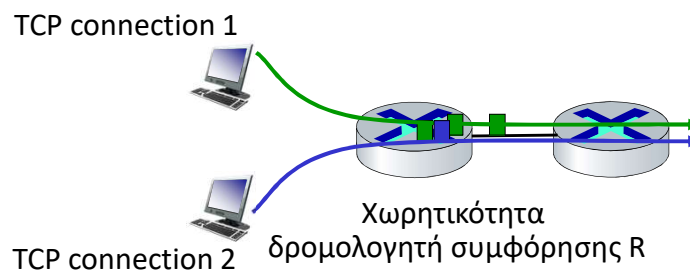
Υλοποιήσεις TCP συχνά βασίζονται σε *υποβοηθούμενο από το δίκτυο (network-assisted) έλεγχο συμφόρησης*:

- 2 bits (ECN) στην κεφαλίδα IP προσδιορίζονται από *δρομολογητή δικτύου* ανάλογα τη συμφόρηση
 - Η πολιτική προσδιορισμού των bits επιλέγεται από τον πάροχο δικτύου
- Η ένδειξη συμφόρησης λαμβάνεται από τον προορισμό
- Ο προορισμός προσδιορίζει το ECE bit στο τμήμα ACK, ενημερώνοντας έτσι για τη συμφόρηση την πηγή
- Εμπλέκονται και το επίπεδο IP (IP κεφαλίδα, προσδιορισμός του ECN bit) and το TCP (TCP κεφαλίδα, προσδιορισμός C,E bit)



Δικαιοσύνη του TCP

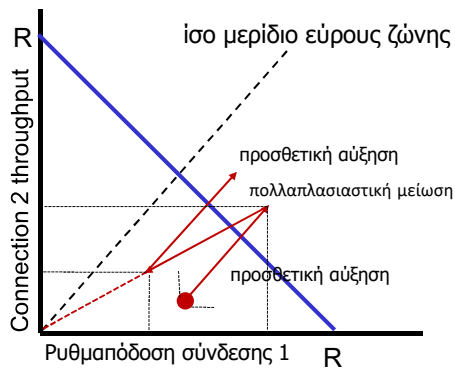
Στόχος : αν K συνδέσεις TCP μοιράζονται την ίδια μπουτλιαρισμένη ζεύξη (bottleneck link) εύρους ζώνης R , καθεμία θα έπρεπε να έχει μέσο ρυθμό R/K



Είναι το TCP δίκαιο;

Δύο ανταγωνιζόμενες συνδέσεις:

- Η προσθετική αύξηση δίνει κλίση 1, καθώς αυξάνει η ρυθμαπόδοση
- Η πολλαπλασιαστική μείωση μειώνει αναλογικά τη ρυθμαπόδοση

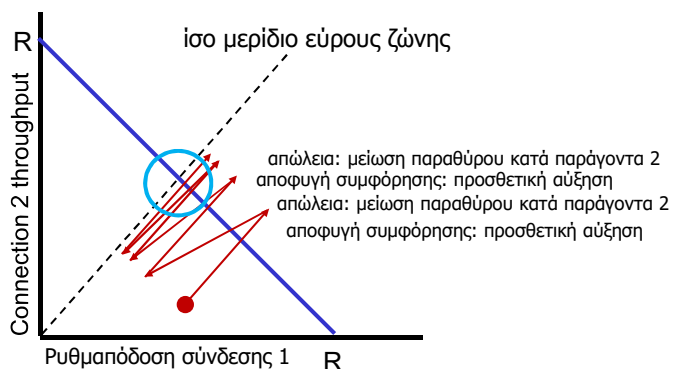


Επίπεδο Μεταφοράς 3-139

Είναι το TCP δίκαιο;

Δύο ανταγωνιζόμενες συνδέσεις:

- Η προσθετική αύξηση δίνει κλίση 1, καθώς αυξάνει η ρυθμαπόδοση
- Η πολλαπλασιαστική μείωση μειώνει αναλογικά τη ρυθμαπόδοση



Είναι δίκαιο το TCP?

A: Ναι, κάτω από ιδανικές συνθήκες:

- Ίδια RTT
- Σταθερός αριθμός ροών μόνο στην κατάσταση αποφυγής συμφόρησης

Επίπεδο Μεταφοράς 3-140

Δικαιοσύνη: Θα πρέπει όλες οι δικτυακές εφαρμογές να είναι «δίκαιες»?

Fairness and UDP

- ❑ Οι εφαρμογές πολυμέσων συχνά δε χρησιμοποιούν TCP
 - Δεν θέλουν έλεγχο του ρυθμού από τον έλεγχο συμφόρησης
- ❑ Αντί αυτού, UDP:
 - αποστολή audio/video με σταθερό ρυθμό, ανοχή σε απώλειες πακέτων
- ❑ Δεν υπάρχει «Αστυνομία Διαδικτύου» για αστυνόμευση της εφαρμογής ελέγχου συμφόρησης

Δικαιοσύνη και παράλληλες συνδέσεις TCP

- Μία εφαρμογή μπορεί να ανοίξει πολλαπλές παράλληλες συνδέσεις μεταξύ δύο υπολογιστών
- Οι web browsers το κάνουν, π.χ. ζεύξη ρυθμού R με 9 υπάρχουσες συνδέσεις:
 - Νέα εφαρμογή ζητά 1 σύνδεση TCP, παίρνει ρυθμό $1/10 * R$
 - Νέα εφαρμογή ζητά 11 συνδέσεις TCP, παίρνει ρυθμό $11/20 * R \sim 1/2 * R$!

Επίπεδο Μεταφοράς: περίγραμμα

- ❑ Υπηρεσίες επιπέδου μεταφοράς
- ❑ Πολύπλεξη και αποπολύπλεξη (Multiplexing and demultiplexing)
- ❑ Ασυνδεσμική μεταφορά: UDP
- ❑ Αρχές της αξιόπιστης μεταφοράς δεδομένων
- ❑ Συνδεσμική μεταφορά: TCP
- ❑ Αρχές ελέγχου συμφόρησης
- ❑ Έλεγχος συμφόρησης του TCP
- ❑ Εξέλιξη του Επιπέδου Μεταφοράς



Εξελίσσοντας λειτουργίες επιπέδου Μεταφοράς

- TCP, UDP: βασικά πρωτόκολλα μεταφοράς για 40 χρόνια
- Αναπτύχθηκαν διαφορετικές «παραλλαγές» του TCP για ειδικά σενάρια - συνθήκες:

Scenario	Challenges
Long, fat pipes (large data transfers)	Πολλά πακέτα «σε πτήση» - οι απώλειες σταματούν την διοχέτευση
Wireless networks	Απώλειες οφείλονται στα θορυβώδη ασύρματα κανάλια και κινητικότητα. Το TCP τις αποδίδει εσφαλμένα σε συμφόρηση
Long-delay links	Πολύ μεγάλες τιμές RTTs
Data center networks	Ευαίσθητα σε καθυστερήσεις
Background traffic flows	“background” TCP ροές χαμηλής προτεραιότητας

- Ανάθεση λειτουργιών επιπέδου Μεταφοράς στο επίπεδο Εφαρμογής, πάνω από UDP
 - QUICK: application-layer protocol, on top of UDP
 - HTTP/3: QUIC

Κεφάλαιο 3: Σύνοψη

- Υπηρεσίες του επιπέδου μεταφοράς:
 - Πολύπλεξη, αποπολύπλεξη
 - Αξιοπίστη μεταφορά δεδομένων
 - Έλεγχος ροής
 - Έλεγχος συμφόρησης
 - Πραγμάτωση και υλοποίηση στο Διαδίκτυο
 - UDP
 - TCP
- Στη συνέχεια:**
- Αφήνοντας το “άκρο” του δικτύου (επίπεδα εφαρμογής, μεταφοράς)
 - προχωράμε στον “πυρήνα” του δικτύου
 - Δύο κεφάλαια για το επίπεδο Δικτύου:
 - Επίπεδο δεδομένων
 - Επίπεδο ελέγχου