



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Δίκτυα Επικοινωνίας Υπολογιστών

Ενότητα 11: Στρώμα μεταφοράς

Μιχαήλ Λογοθέτης

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών  
και Τεχνολογίας Υπολογιστών

**Συνιστώμενο Βιβλίο:**

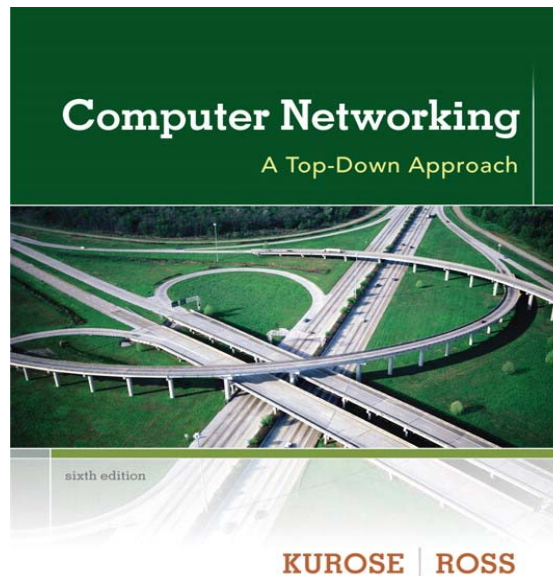
*Δικτύωση Υπολογιστών*  
*Προσέγγιση από Πάνω προς τα Κάτω*  
**KUROSE | ROSS**

**Τίτλος στην Αγγλική:** Computer Networking: A Top-Down Approach

**Επιμέλεια Ελληνικής Μετάφρασης:** Μαυρίδης Ιωάννης - Φουληράς Παναγιώτης

**Εκδόσεις :** Μ. Γκιούρδας

**Έκτη Έκδοση**



Η πλειονότητα των διαφανειών της 11<sup>ης</sup> ενότητας αποτελούν προσαρμογή και απόδοση στα ελληνικά των διαφανειών του 3<sup>ου</sup> κεφαλαίου που συνοδεύουν το βιβλίο «Computer Networking: A Top-Down Approach» J.F Kurose and K.W. Ross, 6/E, Addison-Wesley (**Copyright © Pearson Education Inc**).

Η παρούσα προσαρμογή βασίζεται σε μεγάλο βαθμό σε απόδοση των διαφανειών αυτών στα ελληνικά, την επιμέλεια της οποίας είχε ο καθηγητής του Καποδιστριακού Πανεπιστημίου Αθηνών, κ. Λάζαρος Μεράκος.



# Σκοποί ενότητας

- Κατανόηση των εννοιών της πολύπλεξης και αποπολύπλεξης
- Κατανόηση της ασυνδεσμικής και συνδεσμικής αποπολύπλεξης
- Κατανόηση της αξιόπιστης μεταφοράς δεδομένων



# Περιεχόμενα ενότητας

- Υπηρεσίες και πρωτόκολλα μεταφοράς
- Πρωτόκολλα στρώματος μεταφοράς στο Διαδίκτυο
- Πολύπλεξη/Αποπολύπλεξη
- Ασυνδεσμική αποπολύπλεξη (UDP)
- Συνδεσμική αποπολύπλεξη (TCP)
- User Datagram Protocol
- Αρχές αξιόπιστης μεταφοράς δεδομένων



# Κεφάλαιο 3: Στρώμα Μεταφοράς

## 3.1 Υπηρεσίες επιπέδου μεταφοράς

3.2 Πολύπλεξη και αποπολύπλεξη

3.3 Ασυνδεσμική μεταφορά: UDP

3.4 Αρχές της αξιόπιστης μεταφοράς δεδομένων

3.5 Συνδεσμική μεταφορά: TCP

- Δομή τμήματος
- Αξιόπιστη μεταφορά δεδομένων
- Έλεγχος ροής
- Διαχείριση σύνδεσης

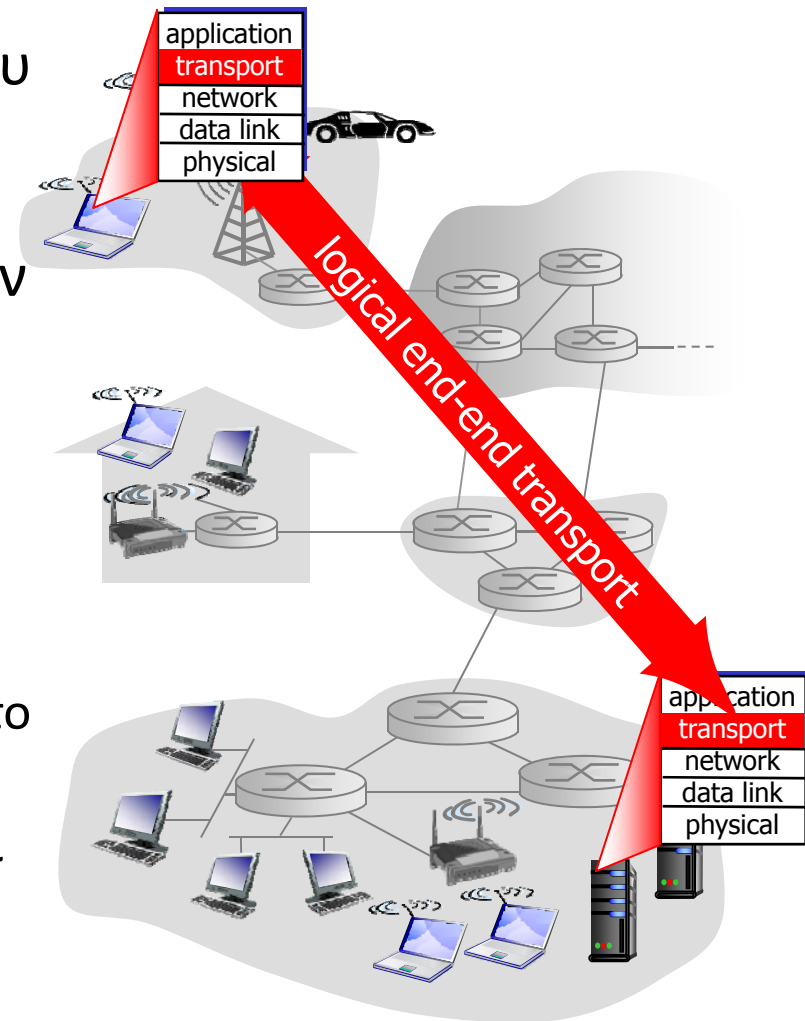
3.6 Αρχές ελέγχου συμφόρησης

3.7 Έλεγχος συμφόρησης του TCP



# Υπηρεσίες και πρωτόκολλα μεταφοράς

- ❖ παρέχουν **λογική επικοινωνία** μεταξύ διεργασιών εφαρμογών που τρέχουν σε διαφορετικά τερματικά συστήματα
- ❖ τα πρωτόκολλα μεταφοράς τρέχουν στα τερματικά συστήματα
  - πλευρά αποστολής: σπάει τα μηνύματα της εφαρμογής σε **τμήματα**, τα περνάει στο επίπεδο δικτύου
  - πλευρά λήψης: ανασύνθεση των τμημάτων σε μηνύματα, τα περνάει στο επίπεδο εφαρμογής
- ❖ Περισσότερα από ένα πρωτόκολλα μεταφοράς διαθέσιμα στις εφαρμογές
  - Διαδίκτυο: TCP και UDP



# Στρώμα Μεταφοράς / Στρώμα Δικτύου

- ❖ **Στρώμα δικτύου:**  
λογική επικοινωνίας  
μεταξύ υπολογιστών
- ❖ **Στρώμα μεταφοράς:**  
λογική επικοινωνία  
μεταξύ διεργασιών
  - στηρίζεται, συμπληρώνει  
τις υπηρεσίες του  
στρώματος δικτύου

## **Αναλογία:**

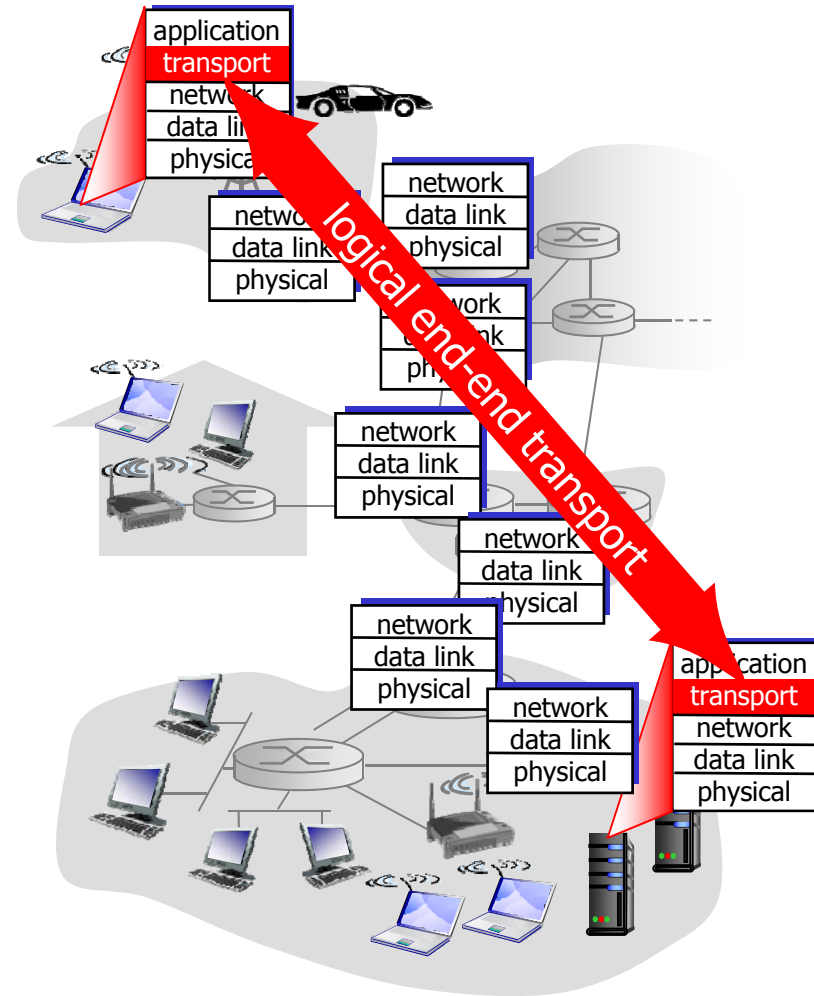
12 παιδιά στο σπίτι της Άννας στέλνουν γράμματα σε 12 παιδιά στο σπίτι του Βασίλη

- διεργασίες = παιδιά
- μηνύματα εφαρμογών = γράμματα σε φακέλους
- υπολογιστές = σπίτια
- πρωτόκολλο μεταφοράς = Άννα και Βασίλης (παιδί που αναλαμβάνει το μείρασμα της αλληλογραφίας σε κάθε σπίτι)
- πρωτόκολλο επιπέδου δικτύου = ταχυδρομική υπηρεσία



# Πρωτόκολλα Στρώματος Μεταφοράς στο Διαδίκτυο

- ❖ αξιόπιστη, σε ορθή σειρά μεταφορά (TCP)
  - έλεγχος συμφόρησης (congestion control)
  - έλεγχος ροής (flow control)
  - εγκαθίδρυση σύνδεσης
- ❖ μη αξιόπιστη, εκτός σειράς παράδοση: UDP
  - 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





# Κεφάλαιο 3: Στρώμα Μεταφοράς

3.1 Υπηρεσίες επιπέδου μεταφοράς

**3.2 Πολύπλεξη και αποπολύπλεξη**

3.3 Ασυνδεσμική μεταφορά: UDP

3.4 Αρχές της αξιόπιστης μεταφοράς δεδομένων

3.5 Συνδεσμική μεταφορά: TCP

- Δομή τμήματος
- Αξιόπιστη μεταφορά δεδομένων
- Έλεγχος ροής
- Διαχείριση σύνδεσης

3.6 Αρχές ελέγχου συμφόρησης

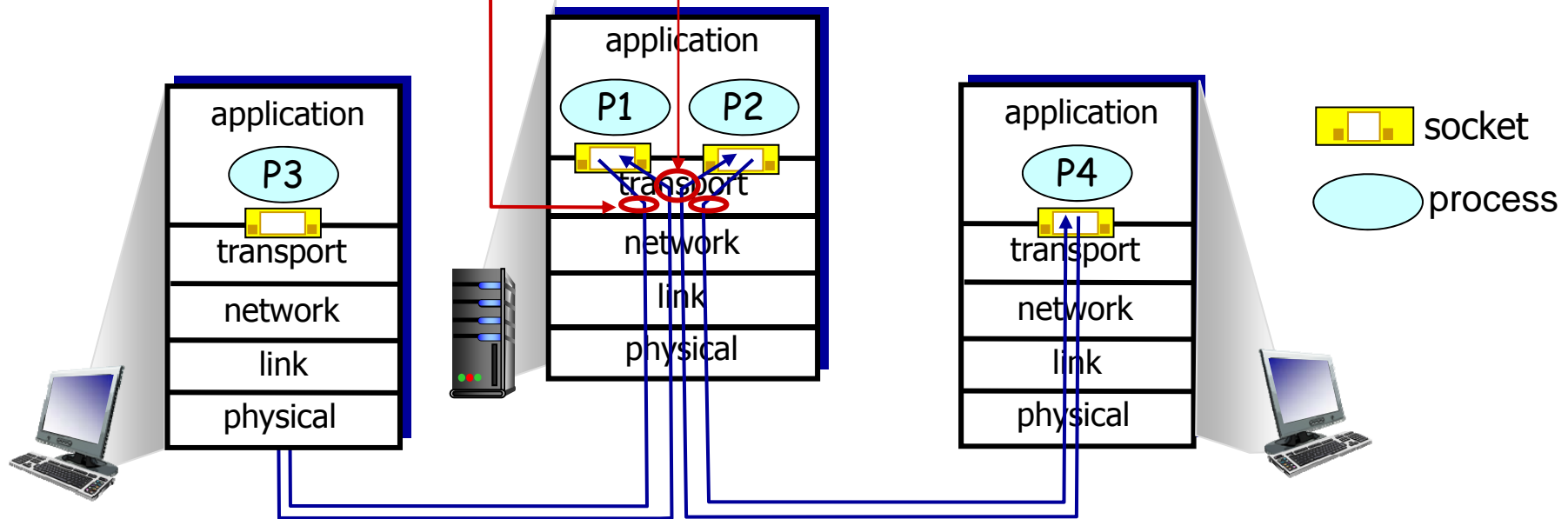
3.7 Έλεγχος συμφόρησης του TCP



# Πολύπλεξη/Αποπολύπλεξη

*Πολύπλεξη στον Η/Υ αποστολής*  
Διαχείριση δεδομένων από πολλαπλές sockets, προσθήκη κεφαλίδας μεταφοράς (που αργότερα χρησιμοποιείται για αποπολύπλεξη)

*Αποπολύπλεξη στον Η/Υ λήψης*  
Χρήση πληροφοριών κεφαλίδας για παράδοση των τμημάτων που λαμβάνονται στη σωστή socket

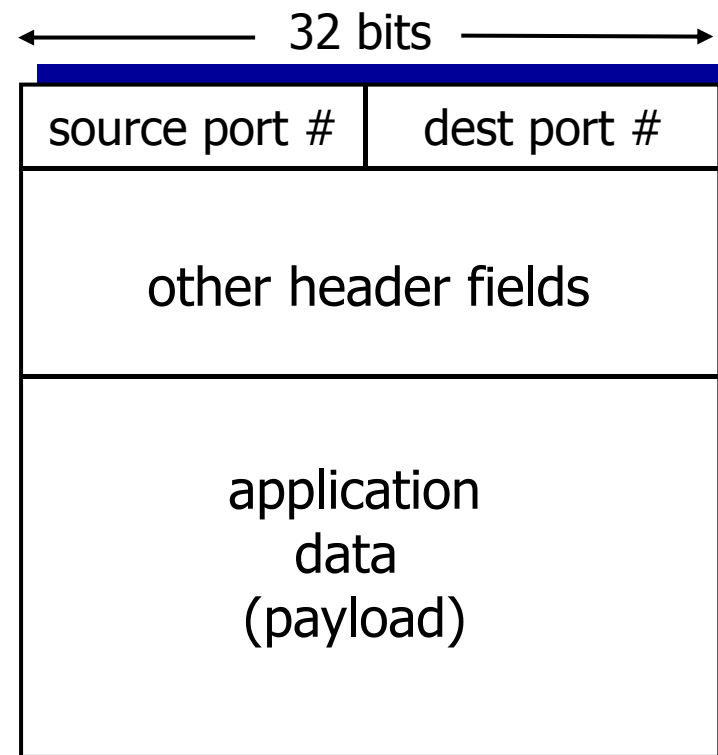


# Πώς δουλεύει η αποπολύπλεξη

## ❖ 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



# Ασυνδεσμική αποπολύπλεξη (UDP)

- ❖ Το δημιουργημένο socket έχει αριθμό θύρας τοπικά στον υπολογιστή:

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

- ❖ Όταν δημιουργείται το *datagram* για να σταλεί στη UDP socket πρέπει να καθοριστεί

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

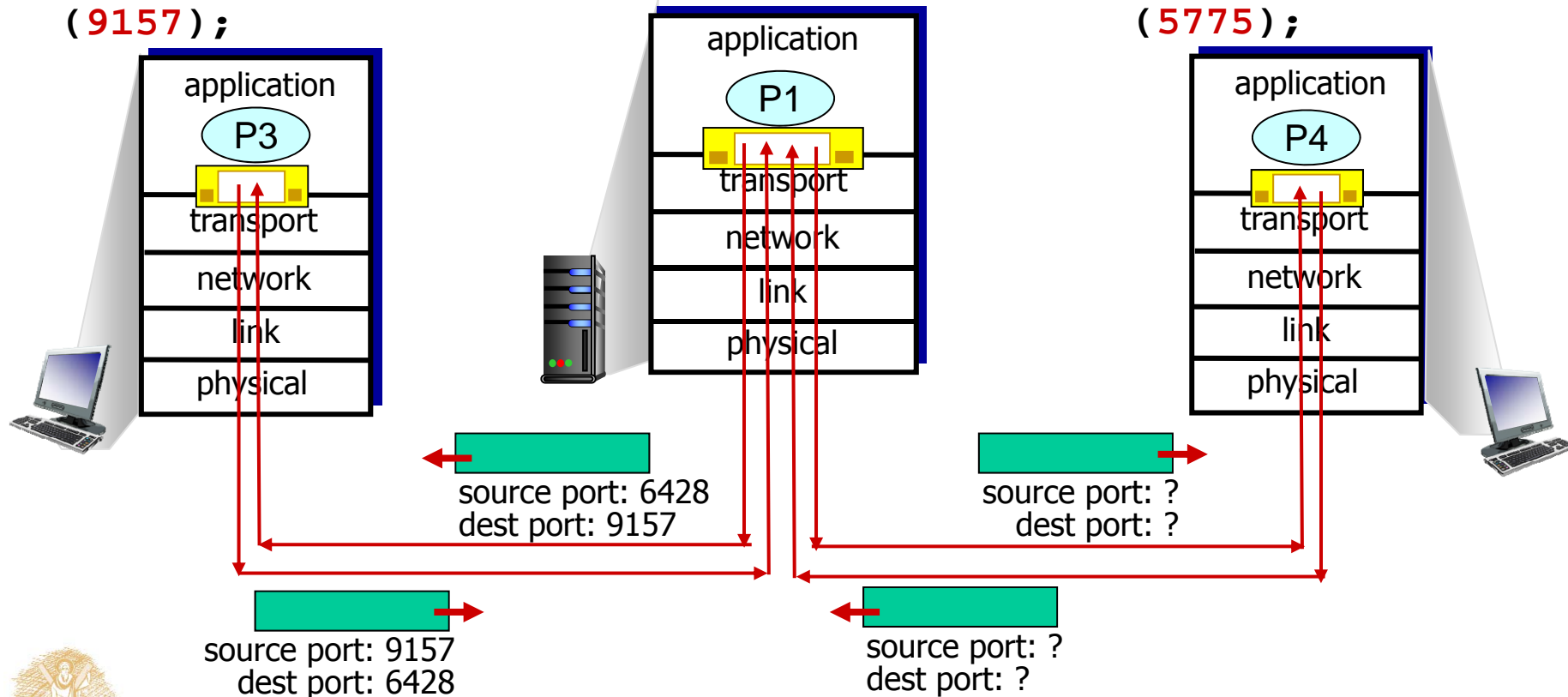


# Ασυνδεσμική αποπολύπλεξη (παράδειγμα)

```
DatagramSocket  
mySocket2 = new  
DatagramSocket  
(9157);
```

```
DatagramSocket  
serverSocket = new  
DatagramSocket  
(6428);
```

```
DatagramSocket  
mySocket1 = new  
DatagramSocket  
(5775);
```

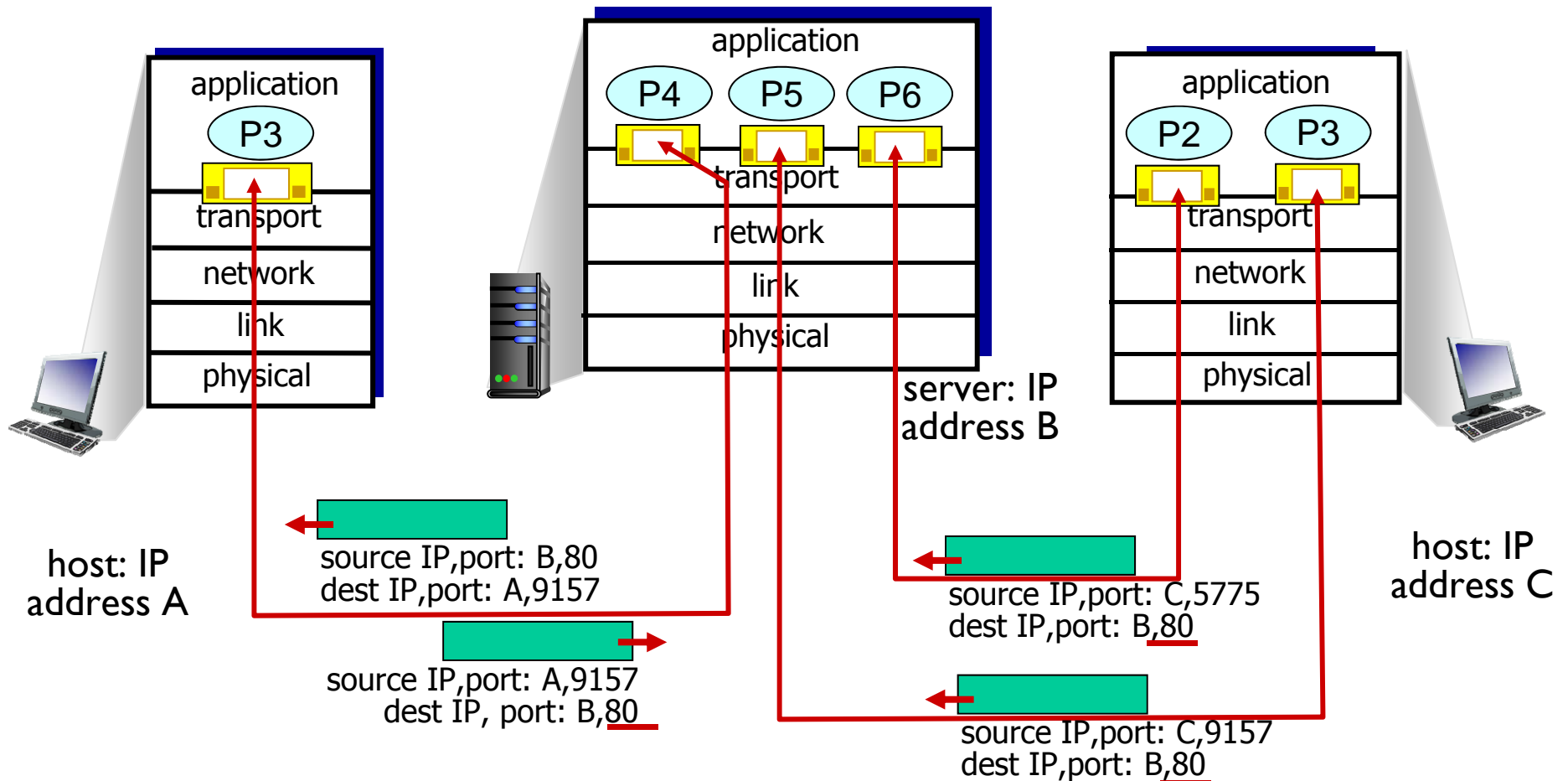


# Συνδεσμική αποπολύπλεξη (TCP)

- ❖ 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 εξυπηρέτης μπορεί να υποστηρίξει πολλές ταυτόχρονες TCP sockets:
  - κάθε socket αναγνωρίζεται από τη δική της τετράδα
- ❖ Οι εξυπηρέτες web έχουν διαφορετικές sockets για κάθε συνδεδεμένο πελάτη
  - το μη-παραμένον HTTP έχει διαφορετική socket για κάθε αίτηση HTTP



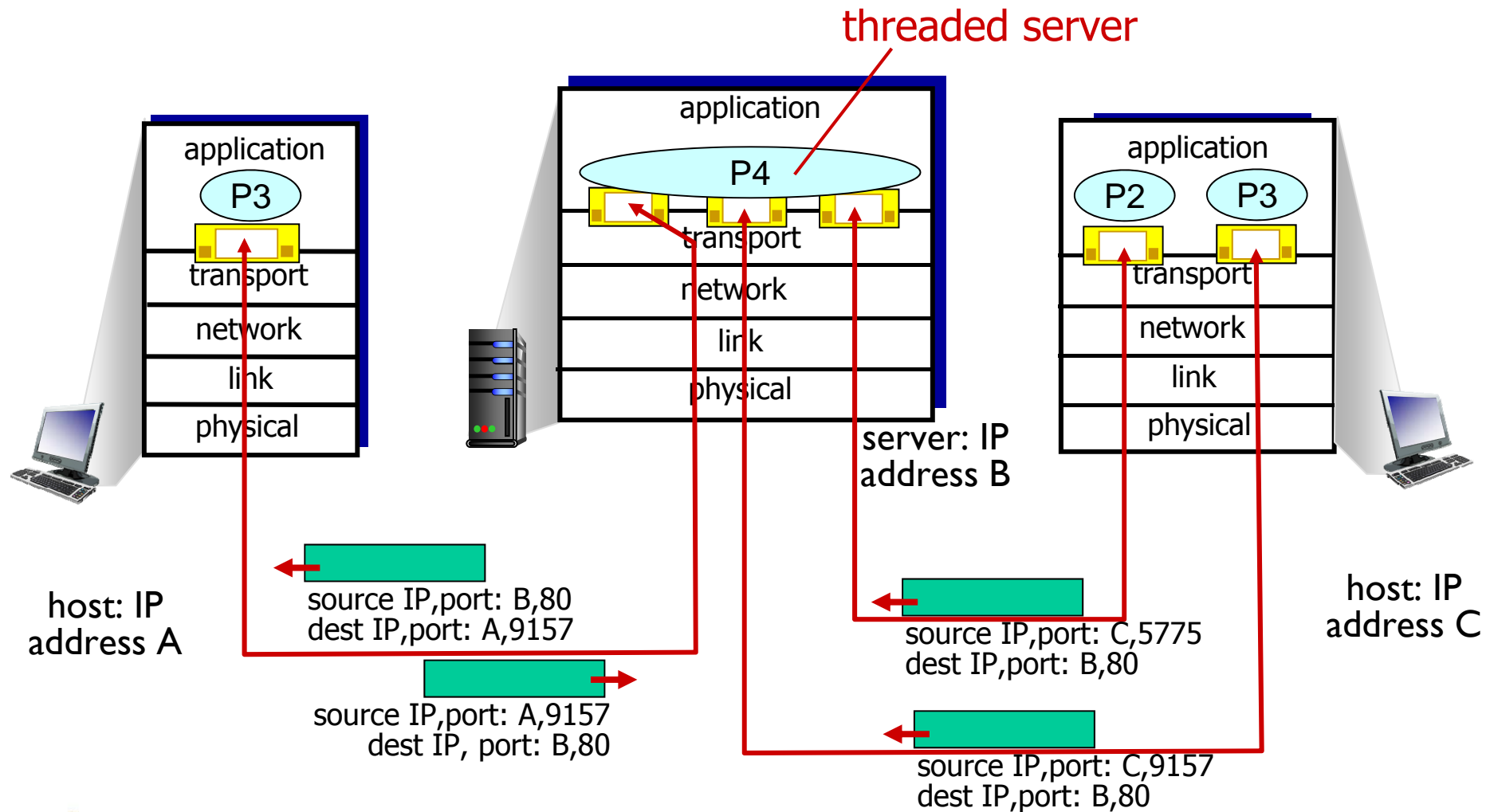
# Συνδεσμική αποπολύπλεξη (παράδειγμα)



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



# Συνδεδεσμένη αποπολύπλεξη : Web Server με νήματα





# Κεφάλαιο 3: Στρώμα Μεταφοράς

3.1 Υπηρεσίες επιπέδου μεταφοράς

3.2 Πολύπλεξη και αποπολύπλεξη

**3.3 Ασυνδεσμική μεταφορά: UDP**

3.4 Αρχές της αξιόπιστης μεταφοράς δεδομένων

3.5 Συνδεσμική μεταφορά: TCP

- Δομή τμήματος
- Αξιόπιστη μεταφορά δεδομένων
- Έλεγχος ροής
- Διαχείριση σύνδεσης

3.6 Αρχές ελέγχου συμφόρησης

3.7 Έλεγχος συμφόρησης του TCP

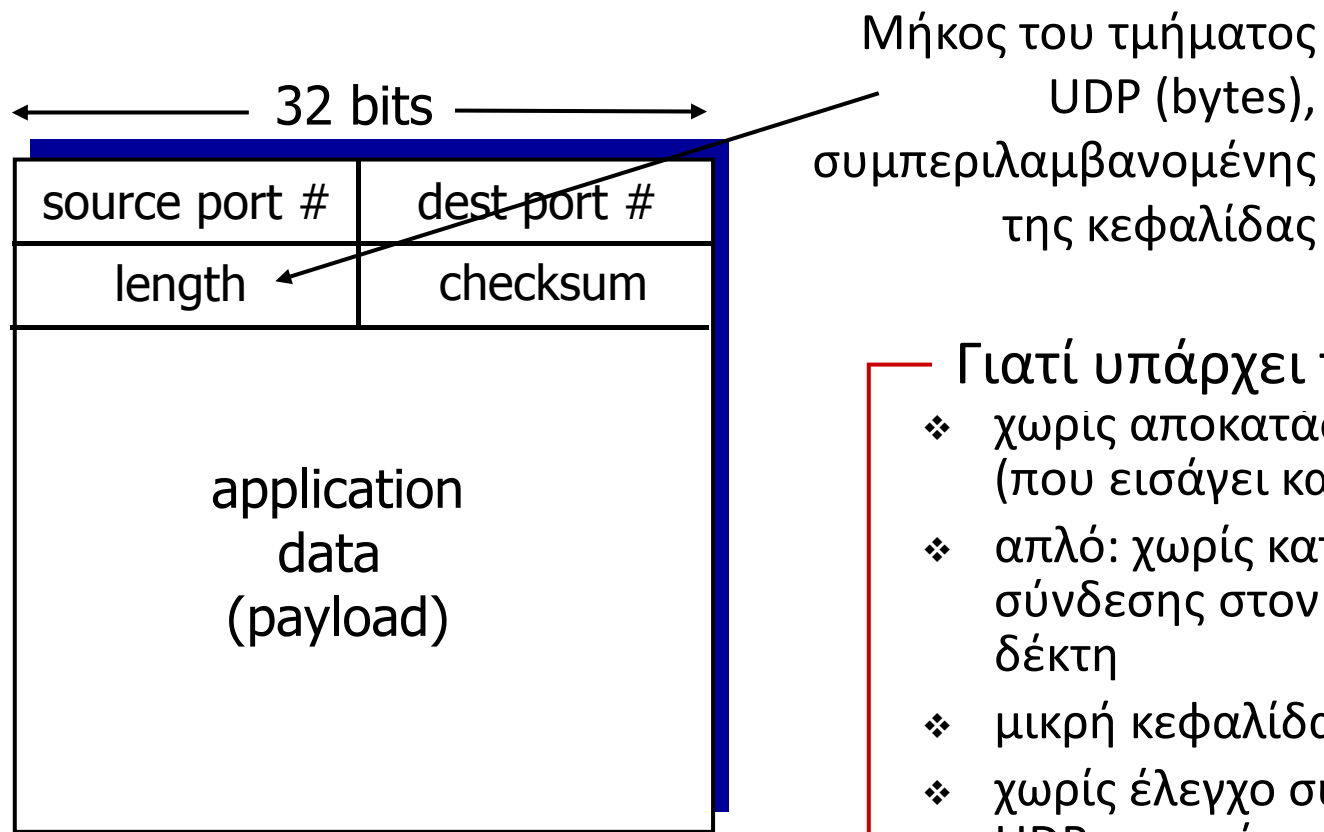


# UDP: User Datagram Protocol [RFC 768]

- ❖ Το απλούστερο πρωτόκολλο μεταφοράς του Διαδικτύου
- ❖ Υπηρεσία παράδοσης 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
- ❖ Το UDP συχνά χρησιμοποιείται για εφαρμογές πολυμέσων συνεχούς ροής (streaming)
  - ανοχή ως προς τις απώλειες
  - ευαισθησία ως προς το ρυθμό
- ❖ άλλες χρήσεις του UDP π.χ.
  - DNS
  - SNMP
- ❖ αξιόπιστη μεταφορά πάνω από το UDP: προσθήκη αξιοπιστίας στο επίπεδο εφαρμογής
  - ανάνηψη από λάθη ανά εφαρμογή



# UDP: κεφαλίδα τμήματος



Δομή τμήματος UDP

## Γιατί υπάρχει το UDP;

- ❖ χωρίς αποκατάσταση σύνδεσης (που εισάγει καθυστέρηση)
- ❖ απλό: χωρίς κατάσταση σύνδεσης στον αποστολέα, δέκτη
- ❖ μικρή κεφαλίδα τμήματος
- ❖ χωρίς έλεγχο συμφόρησης: το UDP μπορεί να “εκραγεί” όσο γρήγορα θέλουμε



# 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: συμπλήρωμα ως προς το 1 του αθροίσματος του περιεχομένου (εκφρασμένου σε 16-bit λέξεις)
- ❖ 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

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	
<hr/>																	
wraparound	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1
<hr/>																	
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	0	1	1

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



# Κεφάλαιο 3: Στρώμα Μεταφοράς

3.1 Υπηρεσίες επιπέδου μεταφοράς

3.2 Πολύπλεξη και αποπολύπλεξη

3.3 Ασυνδεσμική μεταφορά: UDP

**3.4 Αρχές της αξιόπιστης μεταφοράς δεδομένων**

3.5 Συνδεσμική μεταφορά: TCP

- Δομή τμήματος
- Αξιόπιστη μεταφορά δεδομένων
- Έλεγχος ροής
- Διαχείριση σύνδεσης

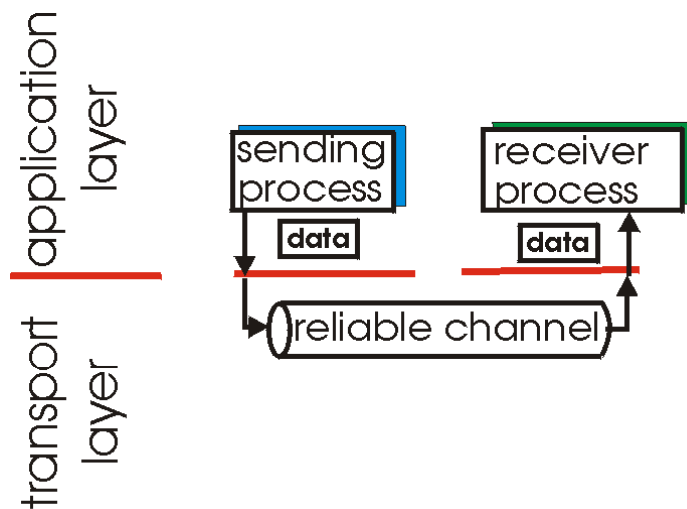
3.6 Αρχές ελέγχου συμφόρησης

3.7 Έλεγχος συμφόρησης του TCP



# Αρχές αξιόπιστης μεταφοράς δεδομένων

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



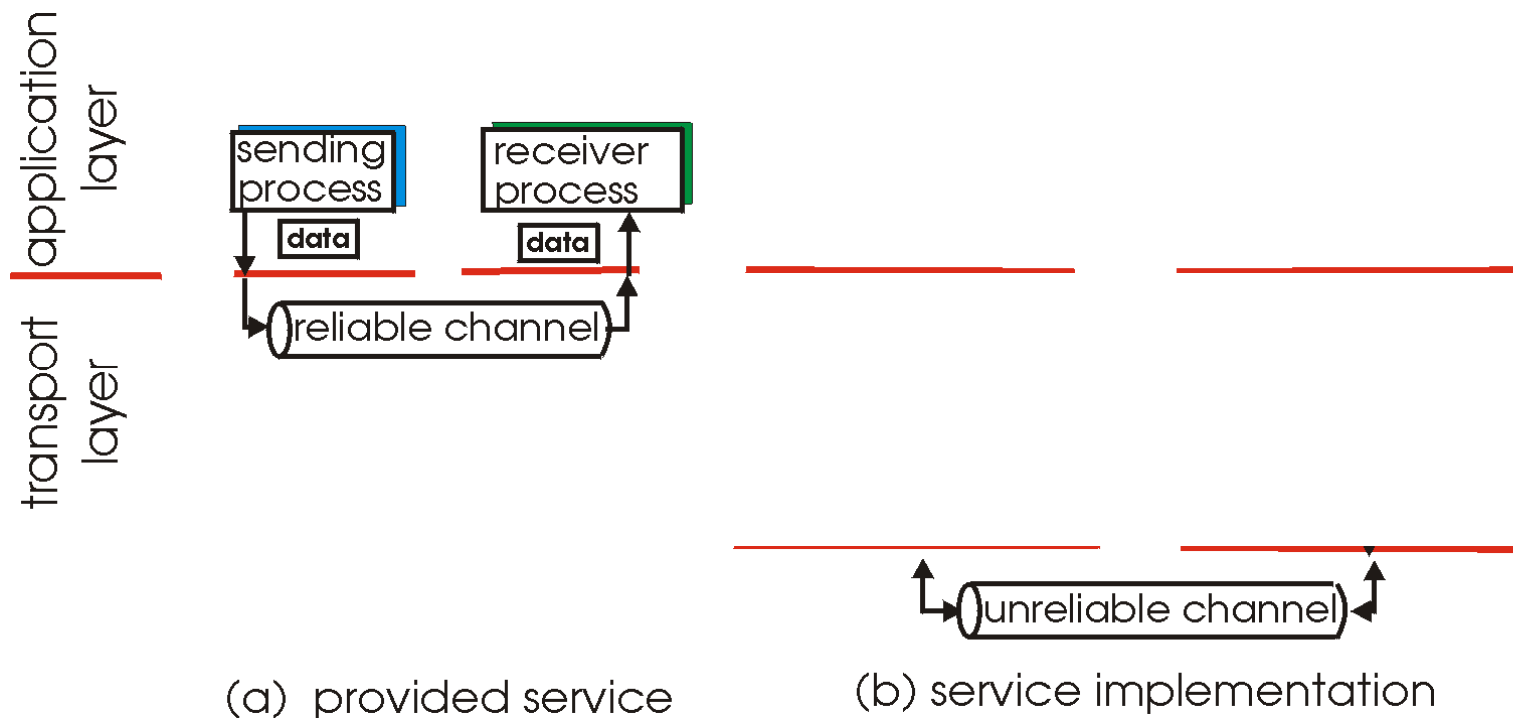
(a) provided service

Τα χαρακτηριστικά του αναξιόπιστου καναλιού θα καθορίσουν την πολυπλοκότητα του πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων (reliable data transfer protocol- rdt)



# Αρχές αξιόπιστης μεταφοράς δεδομένων

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



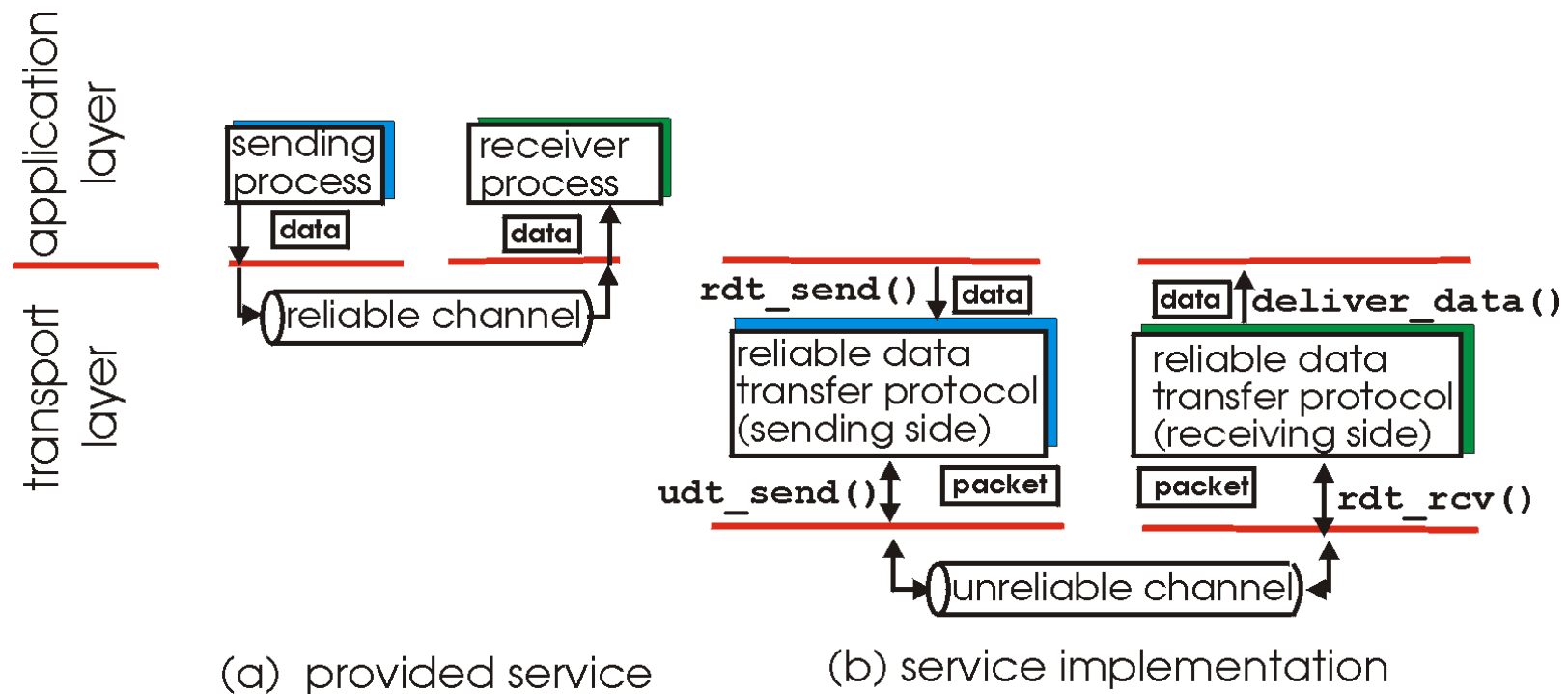
Τα χαρακτηριστικά του αναξιόπιστου καναλιού θα καθορίσουν την πολυπλοκότητα του πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων (reliable data transfer protocol- rdt)





# Αρχές αξιόπιστης μεταφοράς δεδομένων

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



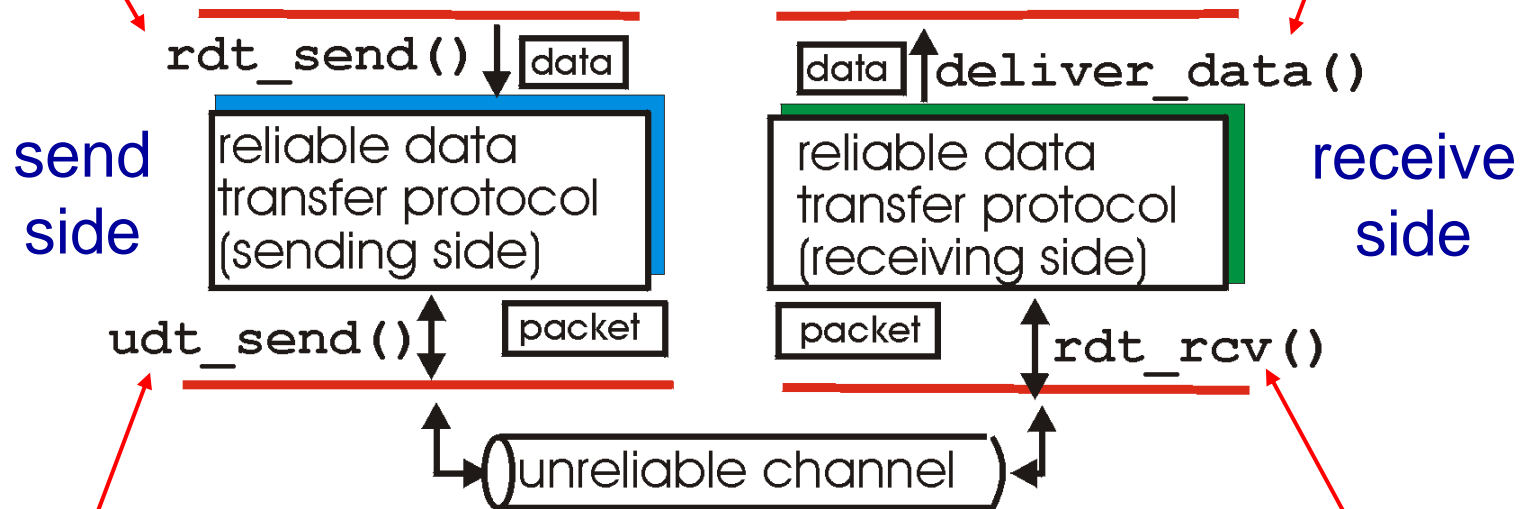
Τα χαρακτηριστικά του αναξιόπιστου καναλιού θα καθορίσουν την πολυπλοκότητα του πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων (reliable data transfer protocol- rdt)



# Αξιόπιστη μεταφορά δεδομένων: ξεκινώντας

**rdt\_send():** καλείται από πάνω (π.χ., από την εφαρμογή). Δίνονται δεδομένα για παράδοση στο ανώτερο στρώμα του δέκτη

**deliver\_data():** καλείται από το rdt για να παραδώσει δεδομένα στο ανώτερο στρώμα



**udt\_send():** καλείται από το rdt, Για τη μεταφορά πακέτου πάνω από αναξιόπιστο κανάλι στον δέκτη

**rdt\_rcv():** καλείται όταν το πακέτο φτάσει στην πλευρά του καναλιού του δέκτη



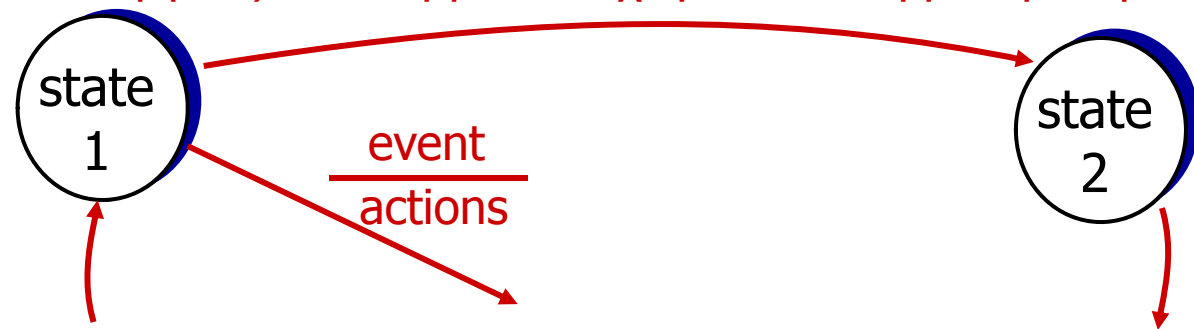
# Αξιόπιστη μεταφορά δεδομένων: ξεκινώντας

**Θα:**

- ❖ αναπτύξουμε σταδιακά τις πλευρές του αποστολέα και του δέκτη ενός πρωτοκόλλου αξιόπιστης μεταφοράς δεδομένων [reliable data transfer (**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 περιγραφή του αποστολέα και του δέκτη

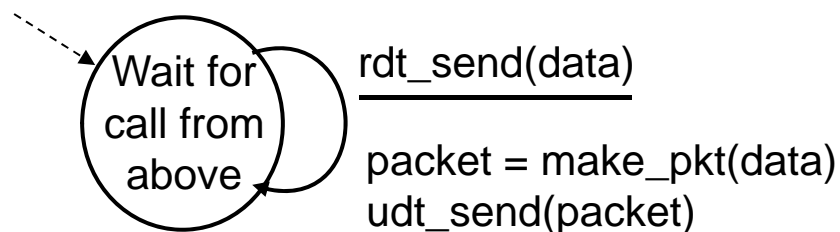
Συμβάν που προκαλεί τη μετάβαση μεταξύ των καταστάσεων  
Ενέργειες που λαμβάνουν χώρα κατά τη μετάβαση

«Κατάσταση»: όντας στην «κατάσταση» η επόμενη κατάσταση καθορίζεται μοναδικά από το επόμενο συμβάν

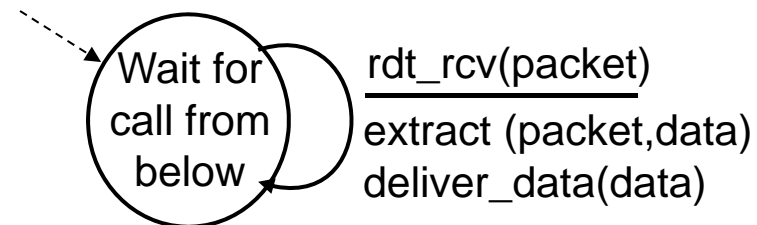


# Rdt1.0: αξιόπιστη μεταφορά επάνω σε αξιόπιστο κανάλι

- ❖ Το υποκείμενο κανάλι είναι πλήρως αξιόπιστο
  - Χωρίς λάθη bit
  - Χωρίς απώλειες πακέτων
- ❖ Ξεχωριστές FSMs για τον αποστολέα, δέκτη:
  - Ο αποστολέας στέλνει δεδομένα στο υποκείμενο κανάλι
  - Ο δέκτης διαβάζει δεδομένα από το υποκείμενο κανάλι



**Αποστολέας (sender)**



**Δέκτης (receiver)**



# Rdt2.0: κανάλι με σφάλματα bit

- ❖ Το υποκείμενο κανάλι ενδέχεται να αντιστρέψει 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;

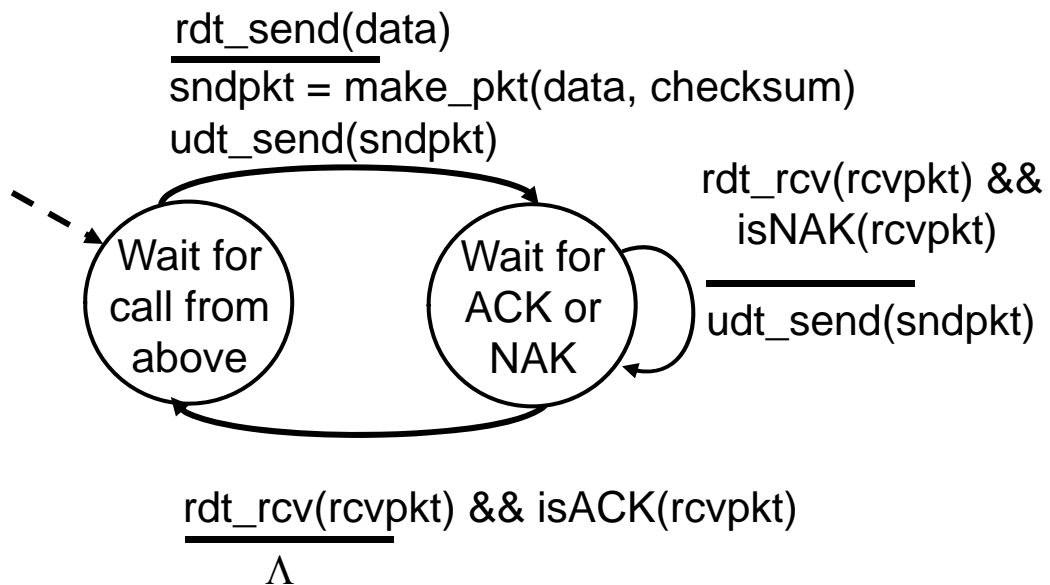


# Rdt2.0: κανάλι με σφάλματα bit

- ❖ Το υποκείμενο κανάλι δεν χάνει ποτέ πακέτα (δεδομένα ή ACKs)
- ❖ 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
- ❖ Νέοι μηχανισμοί στο **rdt2.0** (πέρα από το **rdt1.0**):
  - Ανίχνευση σφάλματος (error detection)
  - Ανάδραση δέκτη (receiver feedback): μηνύματα ελέγχου (ACK, NAK) δέκτης->αποστολέας

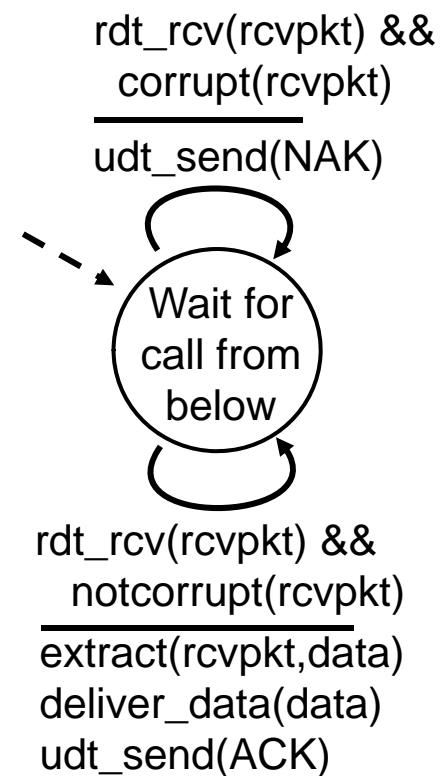


# rdt2.0: περιγραφή FSM

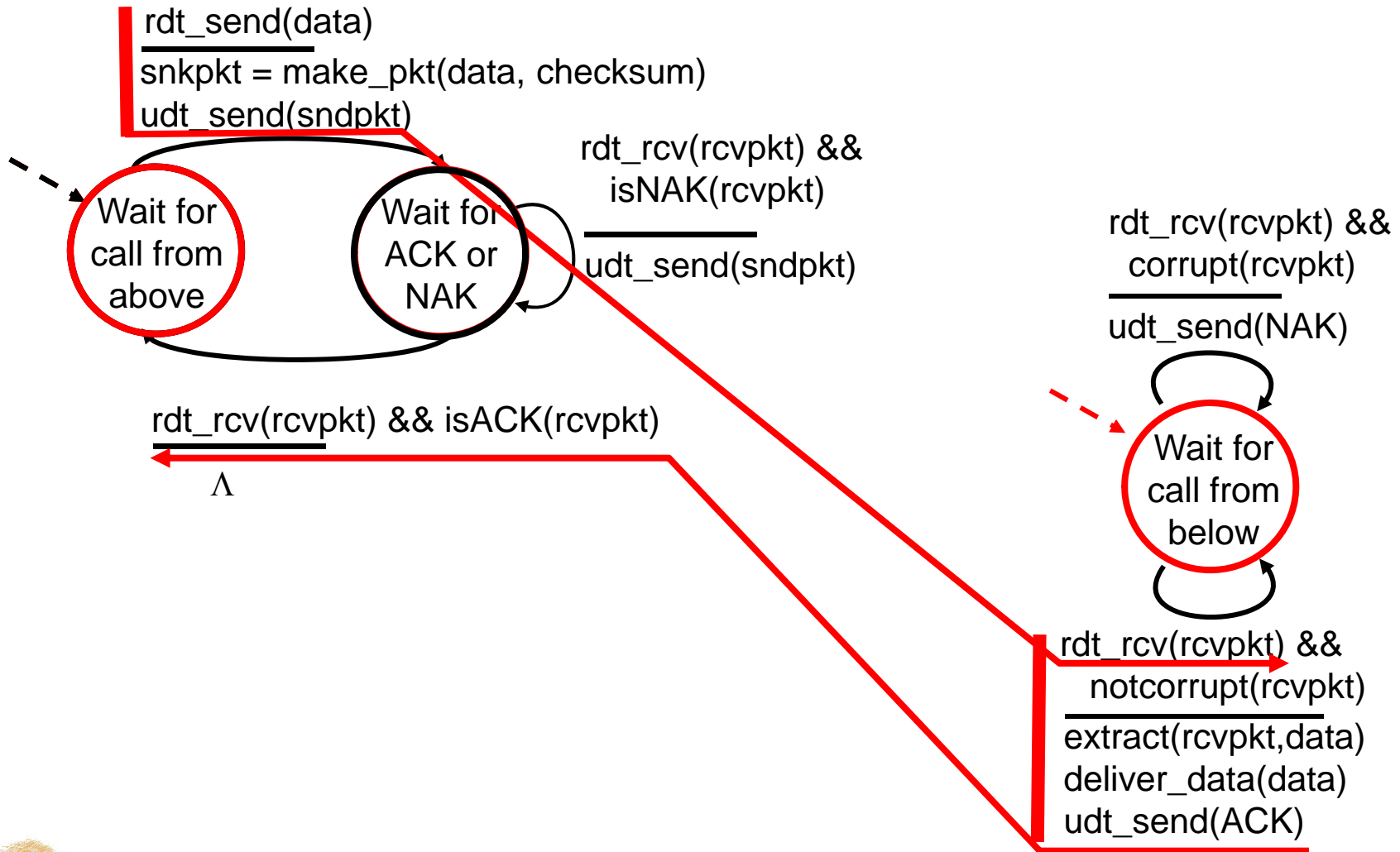


## Αποστολέας (sender)

## Δέκτης (receiver)

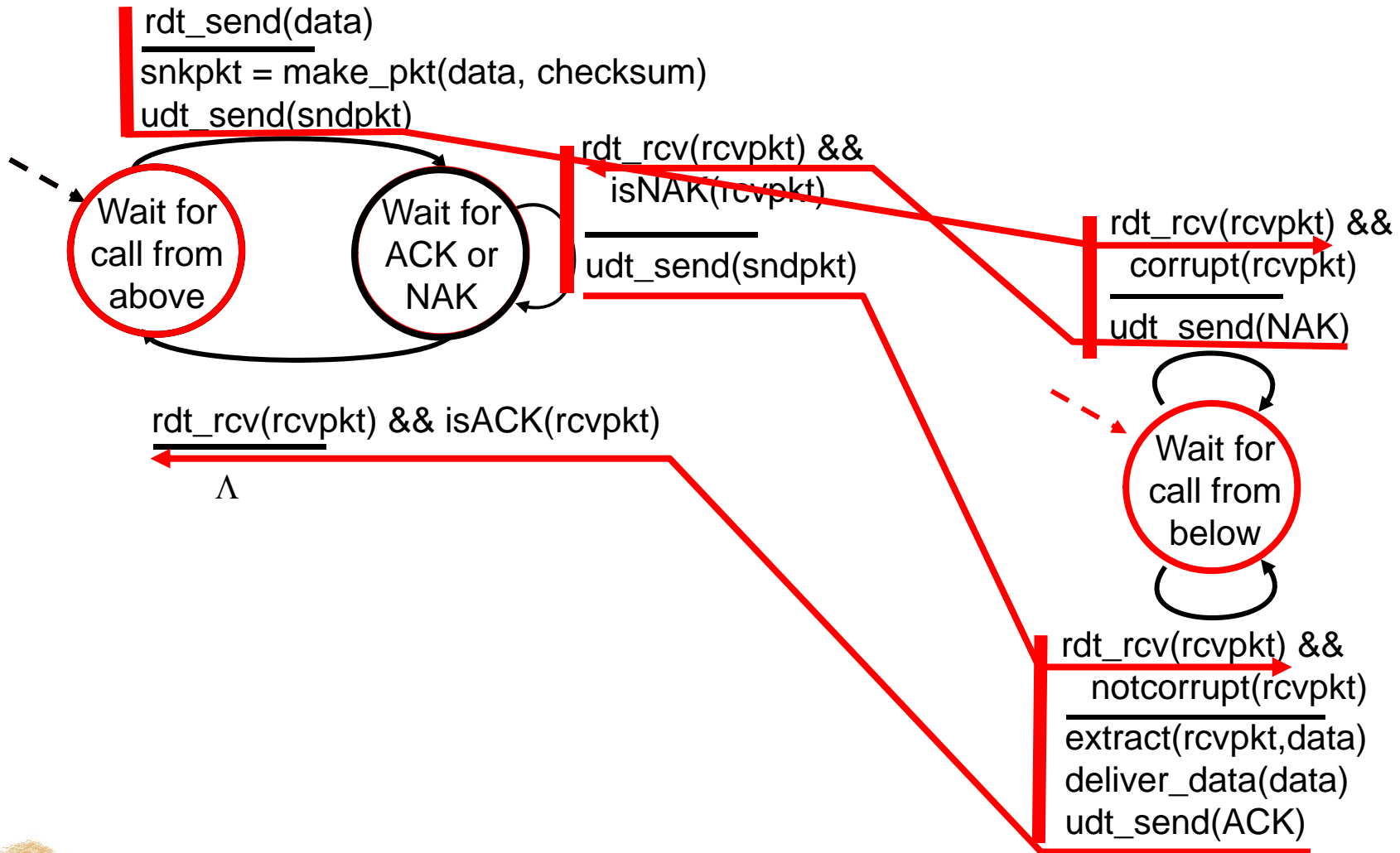


# rdt2.0: λειτουργία χωρίς σφάλματα





# rdt2.0: σενάριο σφάλματος



# Το rdt2.0 έχει ένα μοιραίο σφάλμα!

**Τι συμβαίνει αν καταστραφεί το ACK/NAK;**

- ❖ Ο αποστολέας δεν γνωρίζει τι συνέβη στο δέκτη!
- ❖ Δεν μπορεί απλά να αναμεταδώσει: ενδεχομένως διπλά (duplicate) πακέτα

**Διαχείριση διπλών πακέτων:**

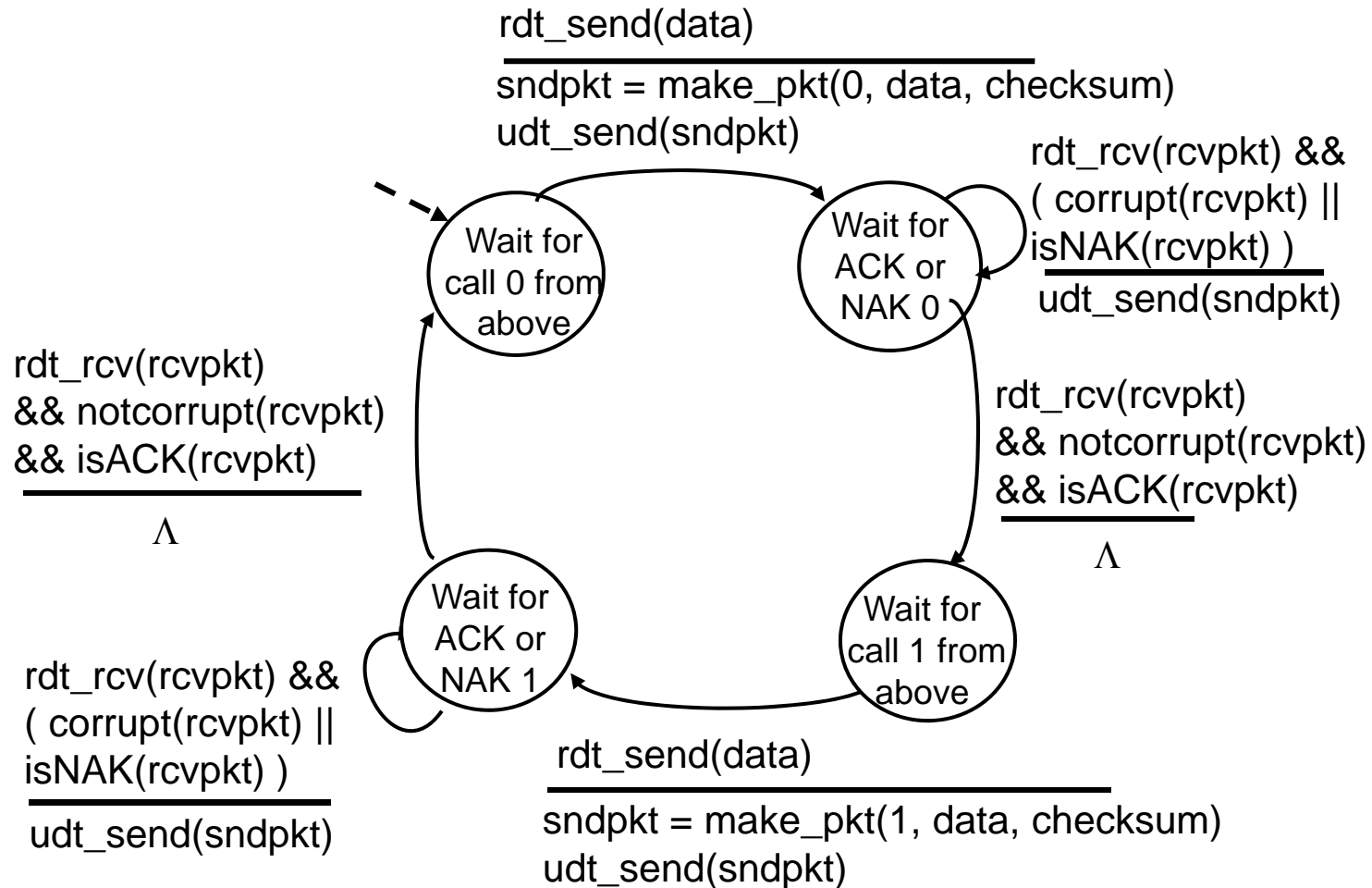
- Ο αποστολέας αναμεταδίδει το τρέχον πακέτο αν αλλοιωθεί το ACK/NAK
- Ο αποστολέας προσθέτει **αριθμό ακολουθίας (sequence number)** σε κάθε πακέτο
- Ο δέκτης απορρίπτει (δεν προωθεί προς τα πάνω) τα διπλά πακέτα

Διακοπή και αναμονή (stop and wait)

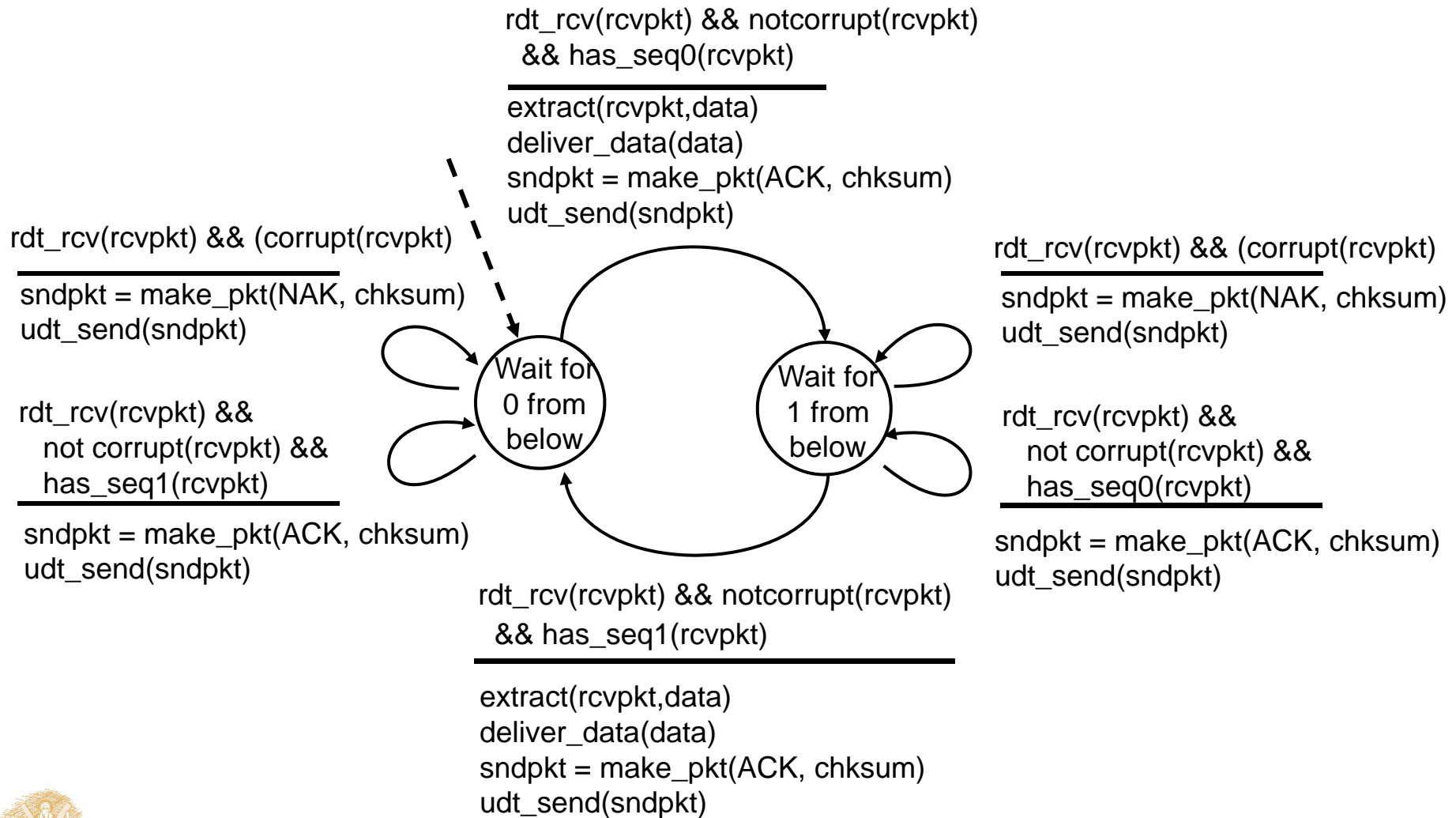
Ο αποστολέας στέλνει ένα πακέτο και μετά περιμένει για την απόκριση του δέκτη



# 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 κατεστραμμένο
- ❖ 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

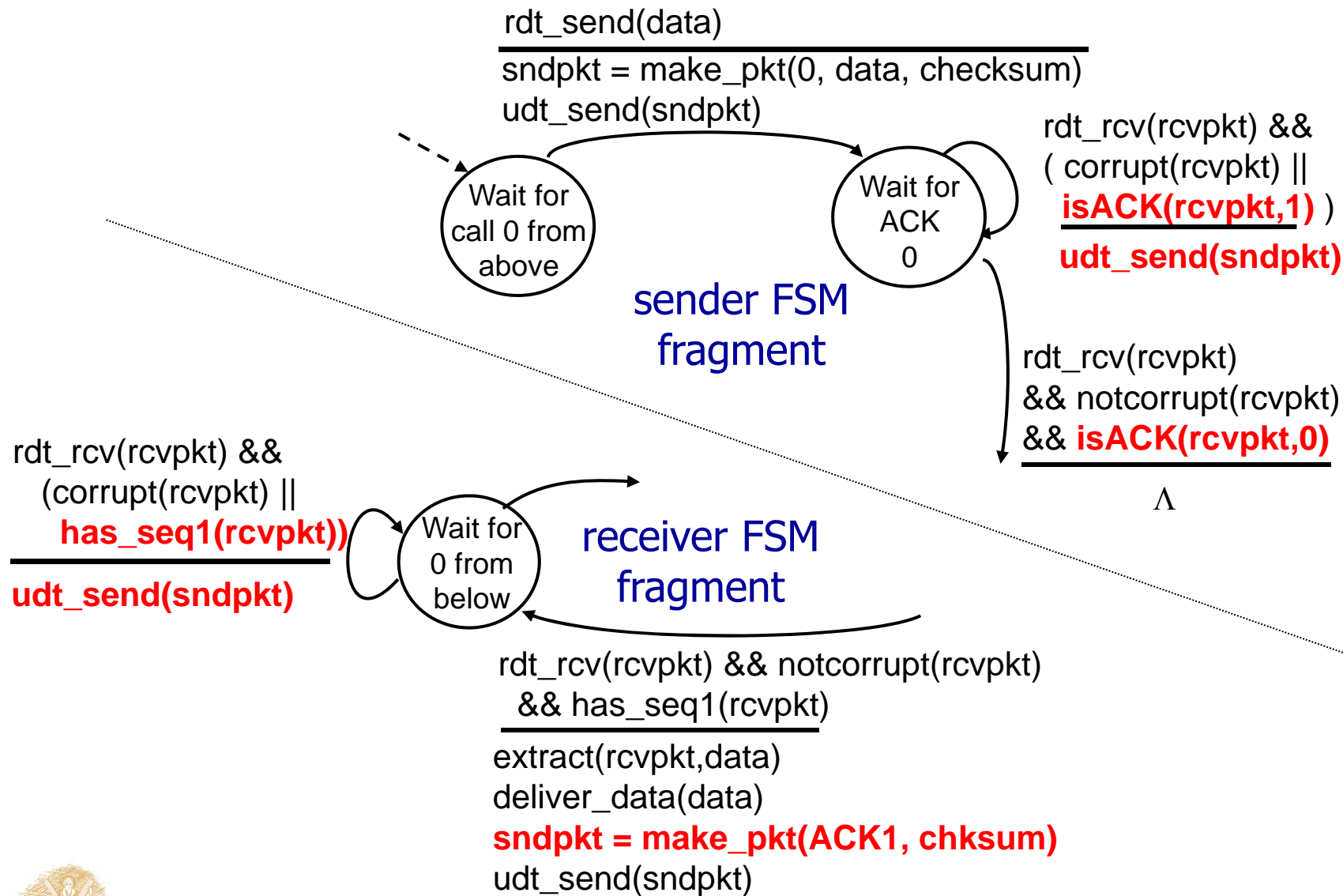


# rdt2.2: Ένα πρωτόκολλο χωρίς NAK

- ❖ 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



# rdt2.2: Αποσπάσματα αποστολέα, δέκτη



# rdt3.0: κανάλια με λάθη και απώλειες

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

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

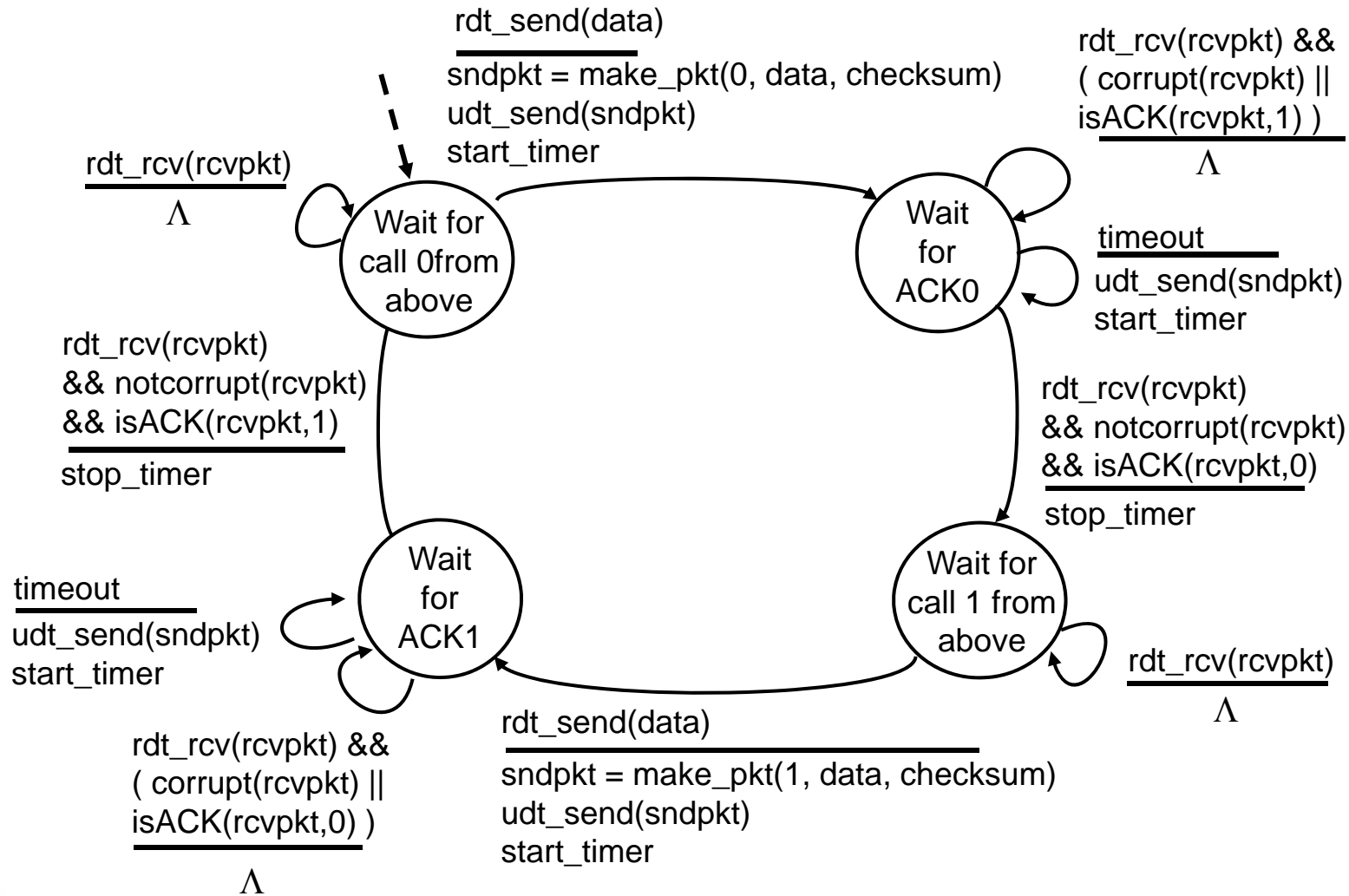
**Προσέγγιση:** ο αποστολέας περιμένει για «εύλογο» χρονικό διάστημα για το ACK

- ❖ Αναμεταδίδει αν δεν ληφθεί ACK σε αυτό το διάστημα
- ❖ Αν το πακέτο (ή το ACK) απλά καθυστέρησε (δεν χάθηκε):
  - Οι αναμεταδόσεις θα είναι διπλές (duplicate), αλλά η χρήση # ακολουθίας το αντιμετωπίζει ήδη αυτό
  - Ο δέκτης πρέπει να καθορίσει # ακολουθίας του πακέτου για το οποίο είναι το ACK
- ❖ Απαιτεί χρονομετρητή αντίστροφης μέτρησης

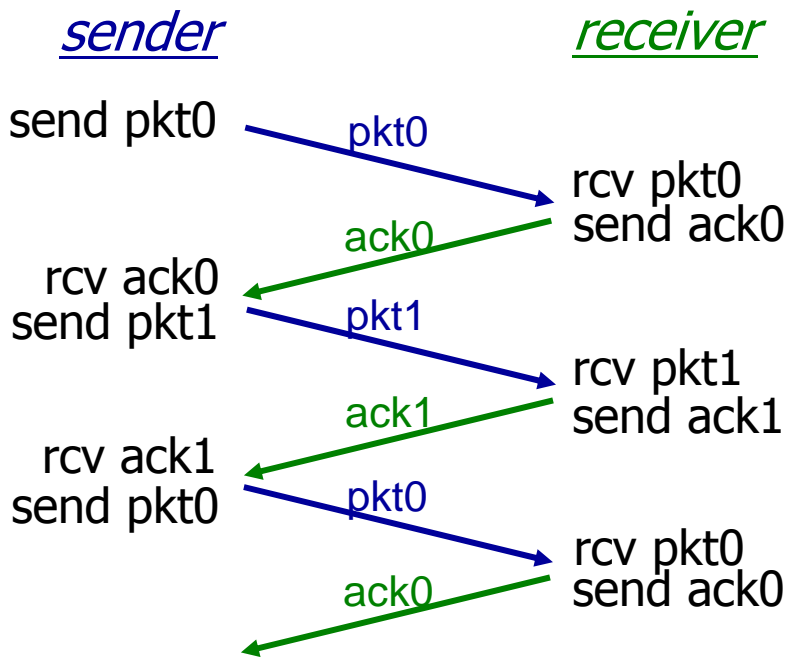




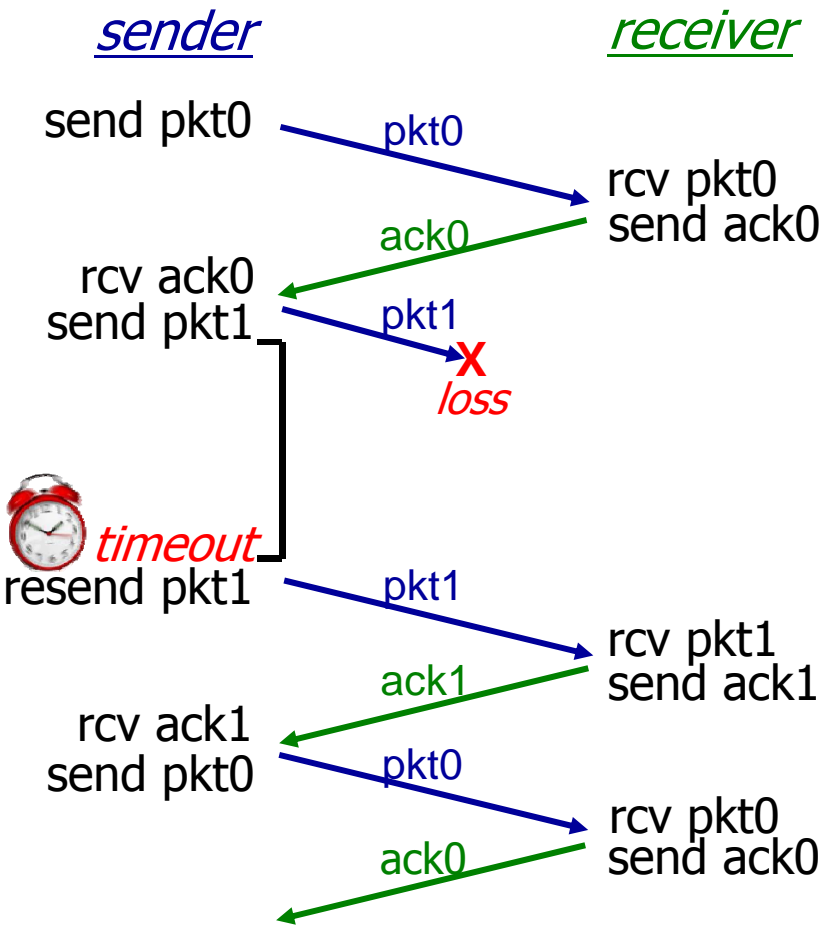
# rdt3.0 αποστολέας



# rdt3.0 εν δράσει



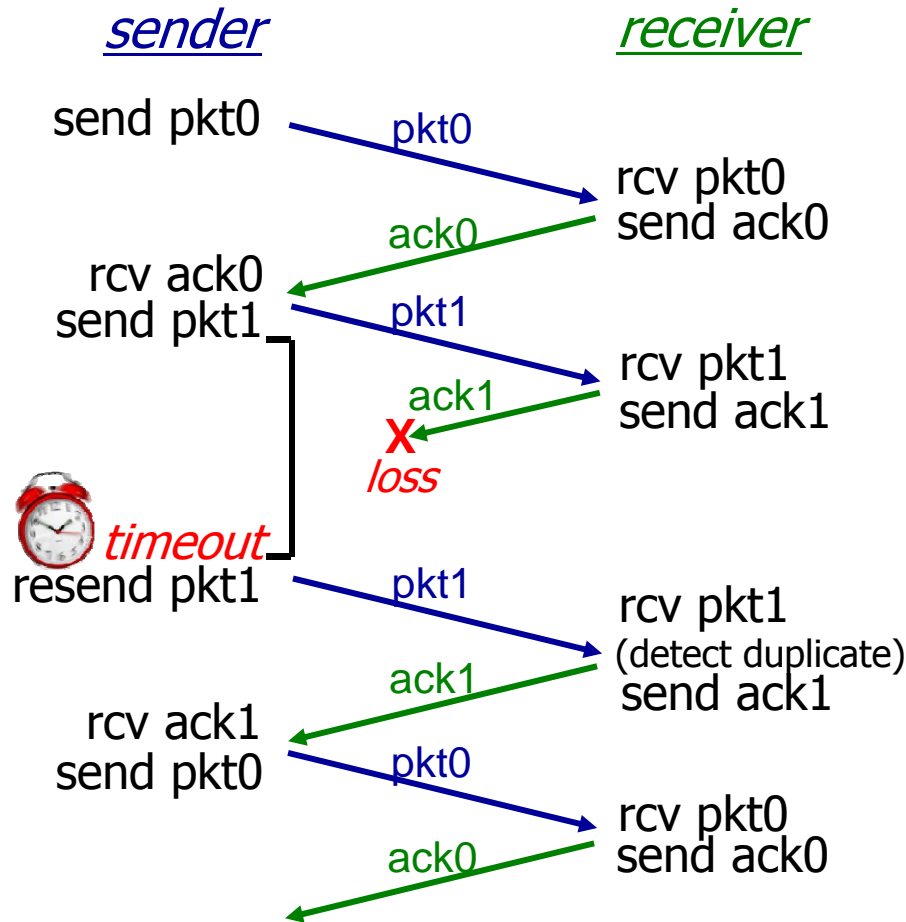
(a) no loss



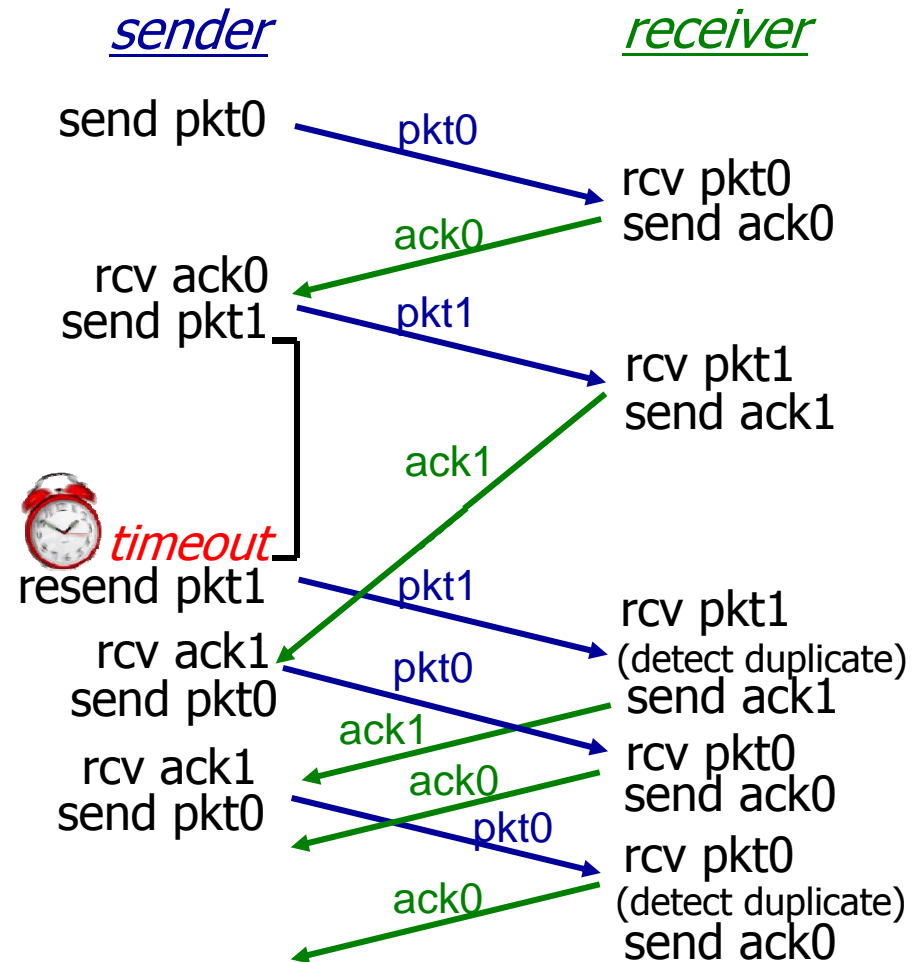
(b) packet loss



# rdt3.0 εν δράσει



(c) ACK loss



(d) premature timeout/ delayed ACK



# Απόδοση του rdt3.0

- ❖ Το rdt3.0 δουλεύει, αλλά η απόδοση του είναι χαμηλή
- ❖ Π.χ.: ζεύξη 1 Gbps, καθυστ. διάδοσης 15 ms, πακέτο 8000 bit:

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

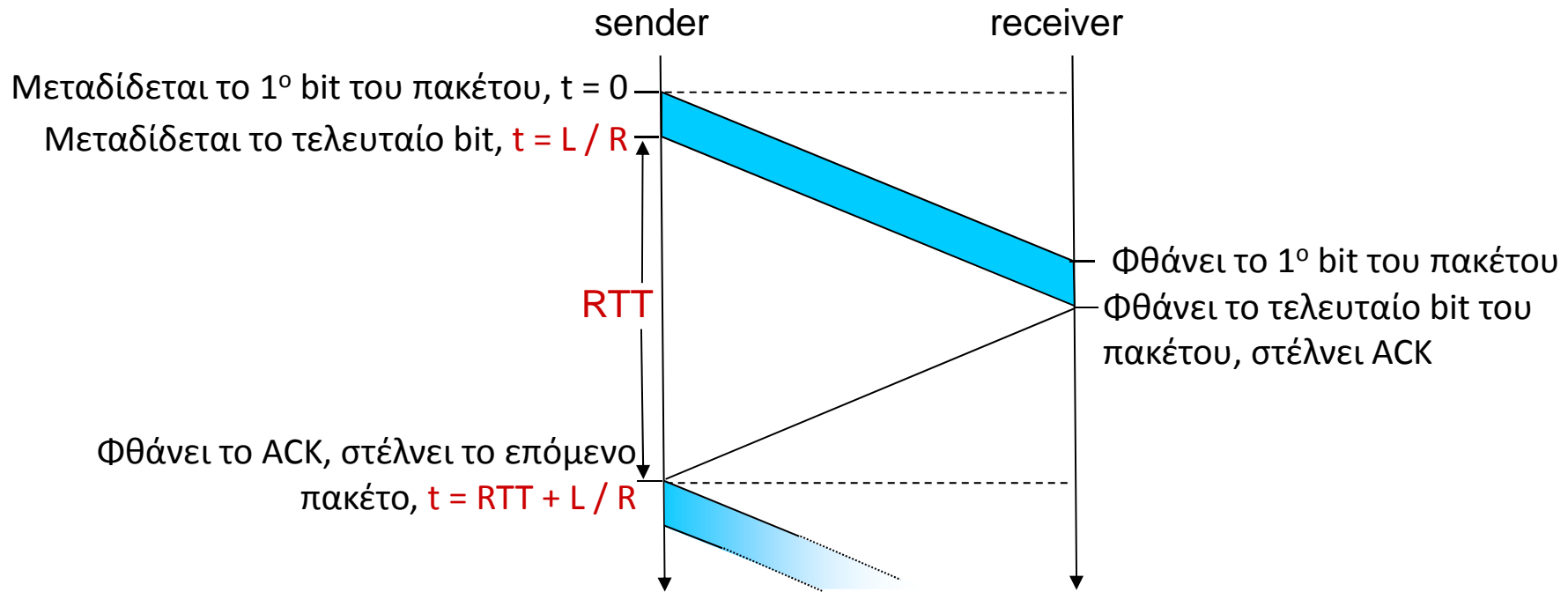
- $U_{sender}$ : **βαθμός εκμετάλλευσης (utilization)** – ποσοστό του χρόνου που ο αποστολέας είναι απασχολημένος στέλνοντας

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$

- Αν το RTT=30 msec, πακέτο 1KB pkt κάθε 30 msec: 33kB/sec throughput σε ζεύξη 1 Gbps
  - ❖ Το δικτυακό πρωτόκολλο περιορίζει τη χρήση των φυσικών πόρων!



# rdt3.0: Λειτουργία διακοπής και αναμονής (stop-and-wait)



$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{.008}{30.008} = 0.00027$$



Τέλος Ενότητας

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στο πλαίσιο του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα Πανεπιστημίου Πατρών**» έχει χρηματοδοτήσει μόνο την αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



# Σημειώματα





# Σημείωμα Ιστορικού Εκδόσεων Έργου

Το παρόν έργο αποτελεί την έκδοση 1.00.



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Πατρών, Μιχαήλ Λογοθέτης. «Δίκτυα Επικοινωνίας Υπολογιστών. Στρώμα μεταφοράς». Έκδοση: 1.0. Πάτρα 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <https://eclass.upatras.gr/courses/EE604/>



# Σημείωμα Αδειοδότησης

Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση Παρόμοια Διανομή 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-sa/4.0/>

Ως **Μη Εμπορική** ορίζεται η χρήση:

- που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
- που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
- που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο

Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



# Διατήρηση Σημειωμάτων

Οποιαδήποτε αναπαραγωγή ή διασκευή του υλικού θα πρέπει να συμπεριλαμβάνει:

- το Σημείωμα Αναφοράς
- το Σημείωμα Αδειοδότησης
- τη δήλωση Διατήρησης Σημειωμάτων
- το Σημείωμα Χρήσης Έργων Τρίτων (εφόσον υπάρχει)

μαζί με τους συνοδευόμενους υπερσυνδέσμους.



# Σημείωμα Χρήσης Έργων Τρίτων

Το Έργο αυτό κάνει χρήση του ακόλουθου έργου:

**Εικόνες/Σχήματα/Διαγράμματα/Φωτογραφίες/Πίνακες**

[1] J. Kurose and K. Ross, Δικτύωση Υπολογιστών – Προσέγγιση από Πάνω προς τα Κάτω, 6<sup>η</sup> έκδοση, Γκιούρδας, 2013

