



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΠΑΤΡΩΝ  
UNIVERSITY OF PATRAS

ΑΝΟΙΚΤΑ ακαδημαϊκά  
μαθήματα ΠΠ

# Δίκτυα Επικοινωνίας Υπολογιστών

Ενότητα 12: Στρώμα μεταφοράς

Μιχαήλ Λογοθέτης

Πολυτεχνική Σχολή

Τμήμα Ηλεκτρολόγων Μηχανικών  
και Τεχνολογίας Υπολογιστών

**Συνιστώμενο Βιβλίο:**

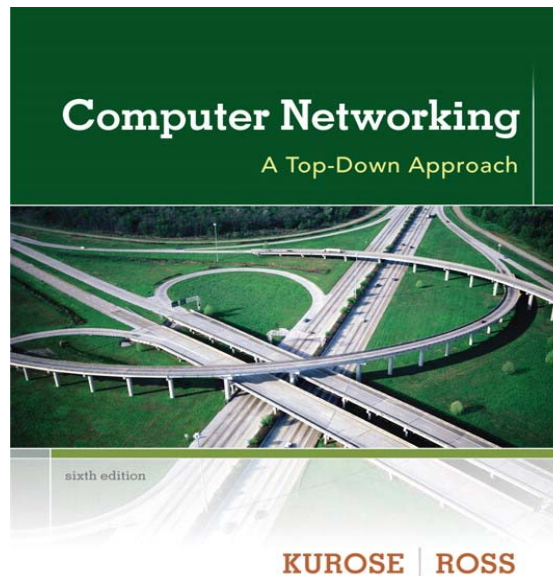
*Δικτύωση Υπολογιστών*  
*Προσέγγιση από Πάνω προς τα Κάτω*  
**KUROSE | ROSS**

**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 12<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 πρωτοκόλλων Go-Back N και selective repeat
- Κατανόηση του πρωτοκόλλου TCP
- Κατανόηση της αξιόπιστης μεταφοράς δεδομένων στο TCP
- Κατανόηση του ελέγχου ροής στο TCP
- Κατανόηση της διαχείρισης σύνδεσης στο TCP



# Περιεχόμενα ενότητας

- Πρωτόκολλα με διοχέτευση
- Go-Back-N
- Επιλεκτική Επανάληψη (Selective Repeat)
- Δομή τμήματος TCP
- Χρόνος Διαδρομής μετ' επιστροφής (Round Trip Time) και Λήξη Χρόνου (Timeout) του TCP
- Αξιόπιστη μεταφορά δεδομένων του TCP
- Έλεγχος ροής του TCP
- Διαχείριση σύνδεσης TCP



# 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

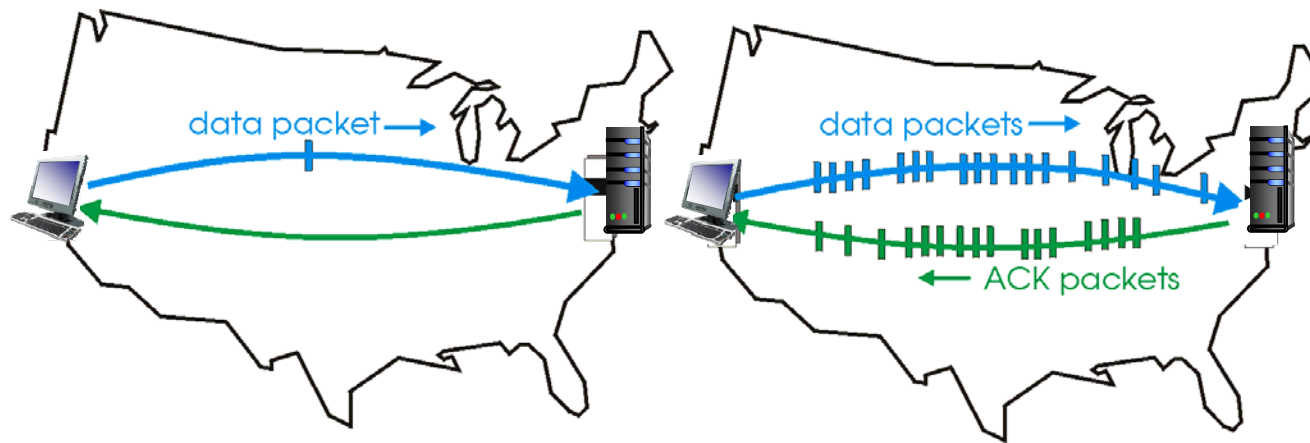
3.7 Έλεγχος συμφόρησης του TCP



# 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



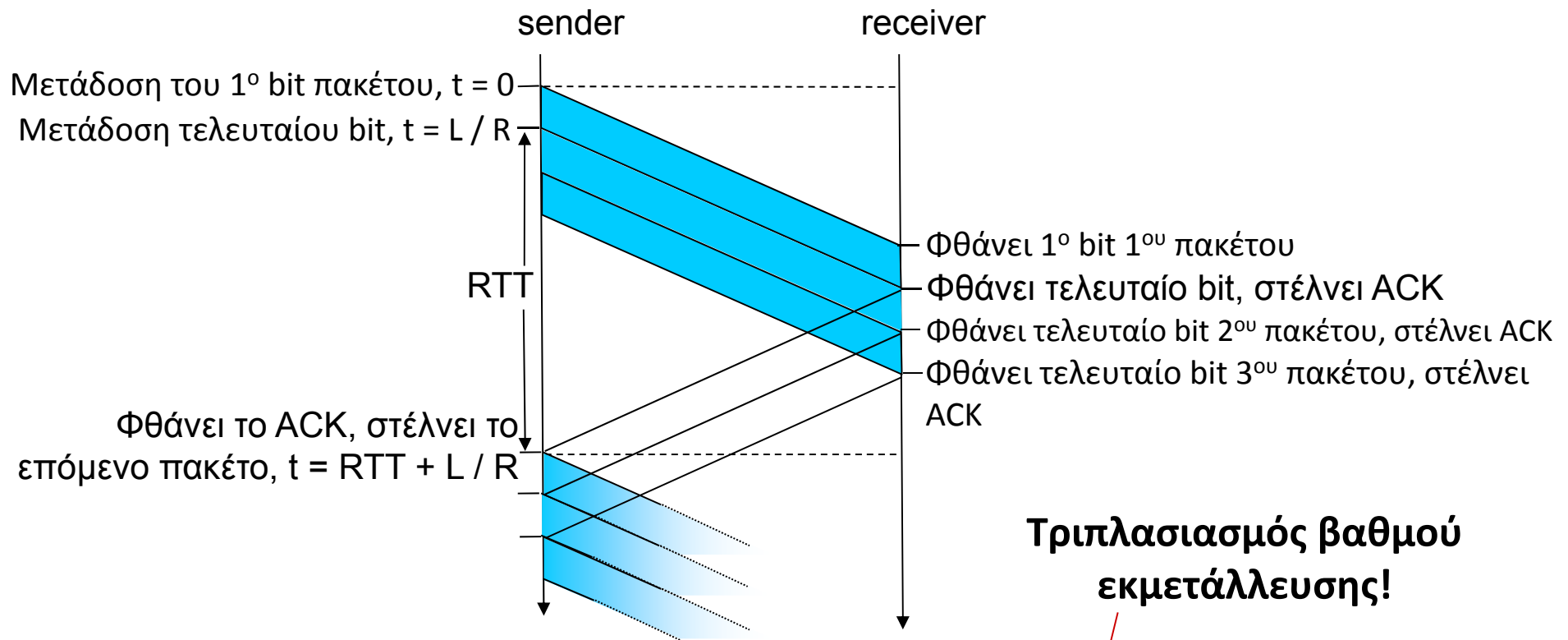
(a) a stop-and-wait protocol in operation

(b) a pipelined protocol in operation

Δύο γενικές μορφές των πρωτοκόλλων διοχέτευσης:  
**go-Back-N, επιλεκτική επανάληψη (selective repeat)**



# Διοχέτευση (pipelining): αύξηση βαθμού χρήσης (utilization)



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



# Πρωτόκολλα διοχέτευσης

## Go-back-N: γενική εικόνα

- ❖ Ο αποστολέας μπορεί να έχει έως και  $N$  μη επιβεβαιωμένα πακέτα στη διοχέτευση
- ❖ Ο δέκτης μόνο στέλνει σωρευτικά acks
  - Δεν επιβεβαιώνει πακέτο αν υπάρχει κάποιο κενό
- ❖ Ο αποστολέας έχει χρονομετρητή για το πιο παλιό μη επιβεβαιωμένο πακέτο
  - Αν ο χρονομετρητής λήξει, αναμεταδίδει όλα τα μη επιβεβαιωμένα πακέτα

## Επιλεκτική επανάληψη (Selective Repeat): γενική εικόνα

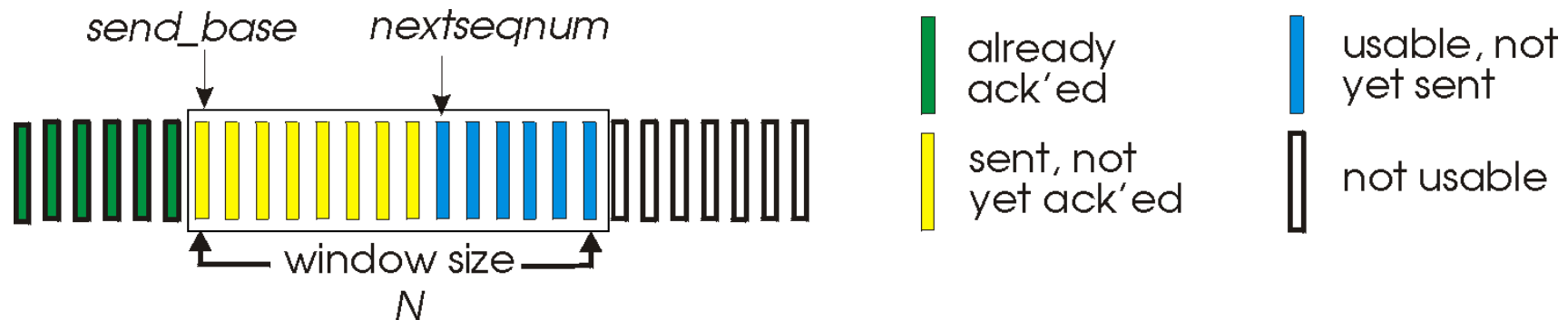
- ❖ Ο αποστολέας μπορεί να έχει έως και  $N$  μη επιβεβαιωμένα πακέτα στη διοχέτευση
- ❖ Ο δέκτης επιβεβαιώνει μεμονωμένα πακέτα
- ❖ Ο αποστολέας διατηρεί χρονομετρητή για κάθε μη επιβεβαιωμένο πακέτο
  - Όταν ο χρονομετρητής λήξει, αναμεταδίδει μόνο το μη επιβεβαιωμένο πακέτο





# Go-Back-N: Αποστολέας

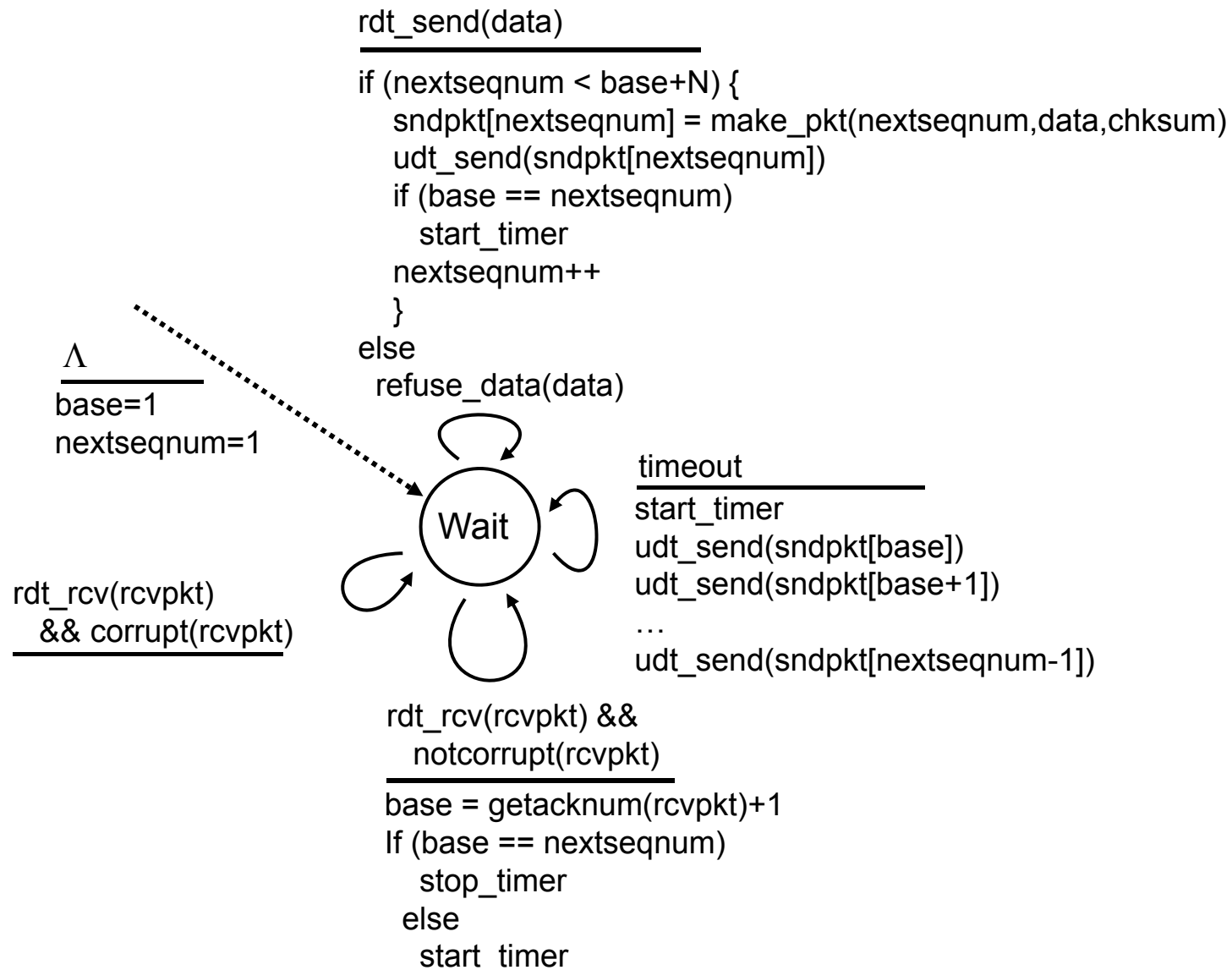
- ❖ k-bit # ακολουθίας στην κεφαλίδα του πακέτου
- ❖ Επιτρέπεται «παράθυρο» (window) έως και N, συνεχόμενων μη επιβεβαιωμένων πακέτων



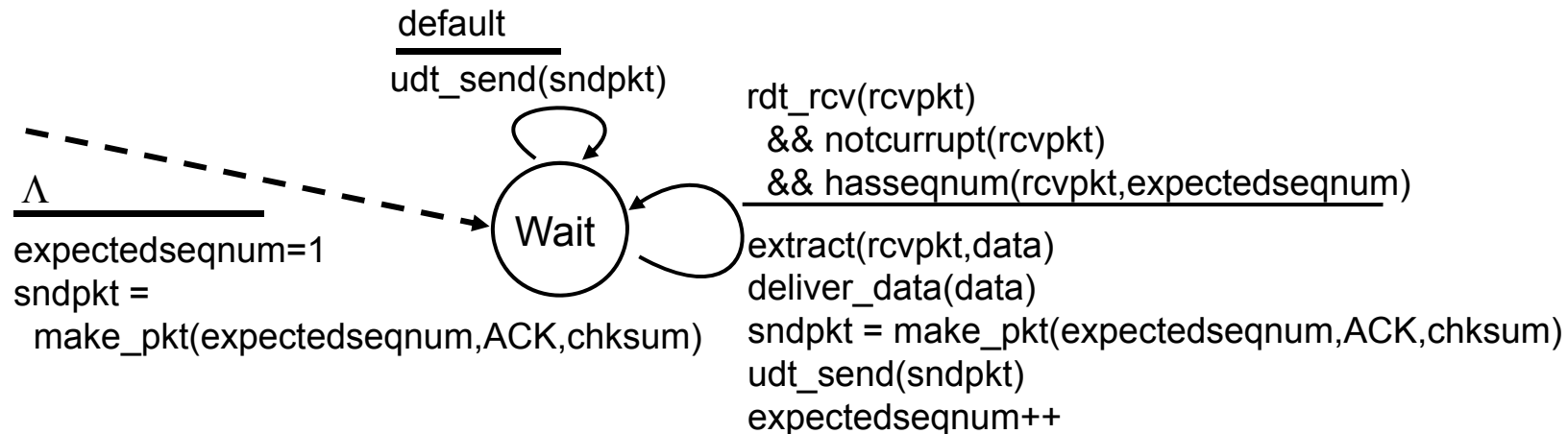
- ❖ ACK(n): επιβεβαιώνει όλα τα πακέτα έως και αυτό με # ακολουθίας n – «συσσωρευτικό ACK» (“**cumulative ACK**”)
  - ενδέχεται να λάβει διπλά ACKs (βλέπε δέκτη)
- ❖ χρονομετρητής για το αρχαιότερο «εν πτήση» πακέτο
- ❖ timeout(n) (Λήξη χρόνου(n)): αναμεταδίδει το πακέτο n και όλα τα πακέτα με υψηλότερο # ακολουθίας στο παράθυρο



# GBN: Επεκτεταμένη FSM αποστολέα



# GBN: Επεκτεταμένη FSM δέκτη



- ❖ Μόνο ACK: πάντα στέλνει ACK για το πακέτο με το μέγιστο σε (ορθή) σειρά # ακολουθίας που έχει ληφθεί σωστά
  - Ενδεχομένως να δημιουργήσει διπλά ACKs
  - Χρειάζεται να θυμάται μόνο το **expectedseqnum**
- ❖ Εκτός σειράς πακέτα:
  - Απόρριψε (μην ενταμιεύεις) -> **χωρίς ενταμίευση δέκτη!**
  - Επανα-επιβεβαίωσε το πακέτο με το μέγιστο σε σειρά # ακολουθίας



# GBN «εν δράσει»

sender window (N=4)

0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8

0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8

0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8  
 0 1 2 3 4 5 6 7 8

sender

send pkt0  
 send pkt1  
 send pkt2  
 send pkt3  
 (wait)

rcv ack0, send pkt4  
 rcv ack1, send pkt5

ignore duplicate ACK



*pkt 2 timeout*

send pkt2  
 send pkt3  
 send pkt4  
 send pkt5

receiver

receive pkt0, send ack0  
 receive pkt1, send ack1

receive pkt3, discard,  
 (re)send ack1

receive pkt4, discard,  
 (re)send ack1

receive pkt5, discard,  
 (re)send ack1

rcv pkt2, deliver, send ack2  
 rcv pkt3, deliver, send ack3  
 rcv pkt4, deliver, send ack4  
 rcv pkt5, deliver, send ack5

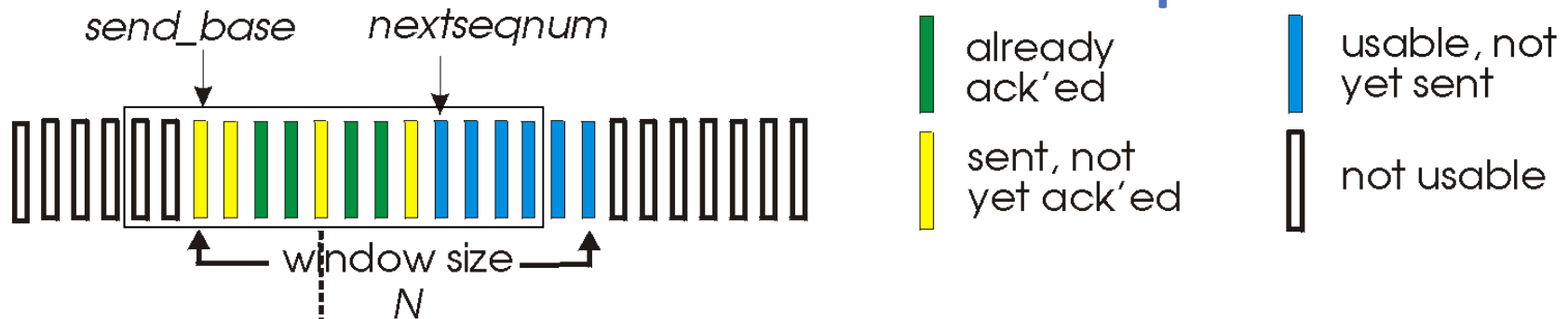


# Επιλεκτική Επανάληψη (Selective Repeat)

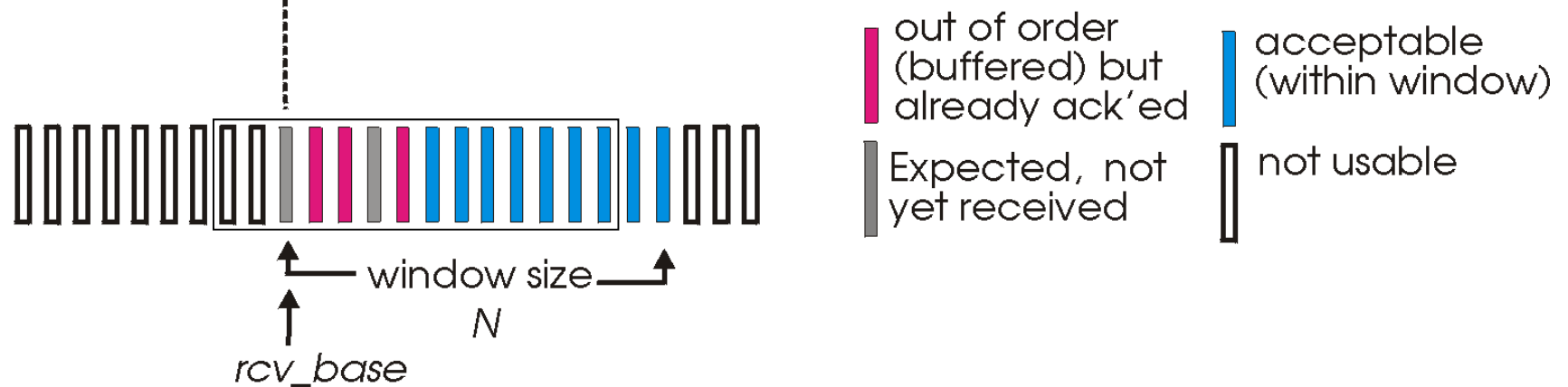
- ❖ Ο δέκτης επιβεβαιώνει μεμονωμένα όλα τα σωστά ληφθέντα πακέτα
  - Ενταμιεύει πακέτα, αν χρειάζεται, ώστε τελικά να παραδώσει τα πακέτα σε σωστή σειρά στο ανώτερο επίπεδο
- ❖ Ο αποστολέας ξαναστέλνει μόνο τα πακέτα για τα οποία δεν έχει ληφθεί ACK
  - Χρονομετρητής στον αποστολέα για κάθε μη επιβεβαιωμένο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



# Επιλεκτική επανάληψη: παράθυρα αποστολέα και δέκτη



(a) sender view of sequence numbers



(b) receiver view of sequence numbers



# Επιλεκτική επανάληψη

## 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

### pkt n in [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)

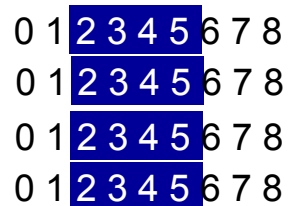
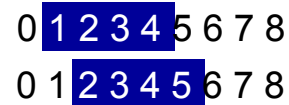
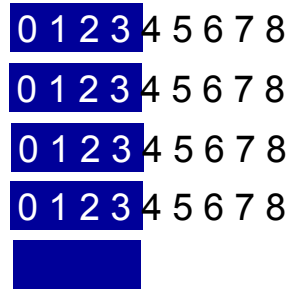
### διαφορετικά:

- ❖ αγνόησε



# Επιλεκτική επανάληψη «εν δράσει»

sender window (N=4)



sender

send pkt0  
 send pkt1  
 send pkt2  
 send pkt3  
 (wait)

rcv ack0, send pkt4  
 rcv ack1, send pkt5

record ack3 arrived



*pkt 2 timeout*

send pkt2

record ack4 arrived

record ack5 arrived

receiver

receive pkt0, send ack0

receive pkt1, send ack1

receive pkt3, buffer,  
 send ack3

receive pkt4, buffer,  
 send ack4

receive pkt5, buffer,  
 send ack5

rcv pkt2; deliver pkt2,  
 pkt3, pkt4, pkt5;  
 send ack2

*X loss*

*Ε: τι συμβαίνει όταν φθάνει το ack2;*



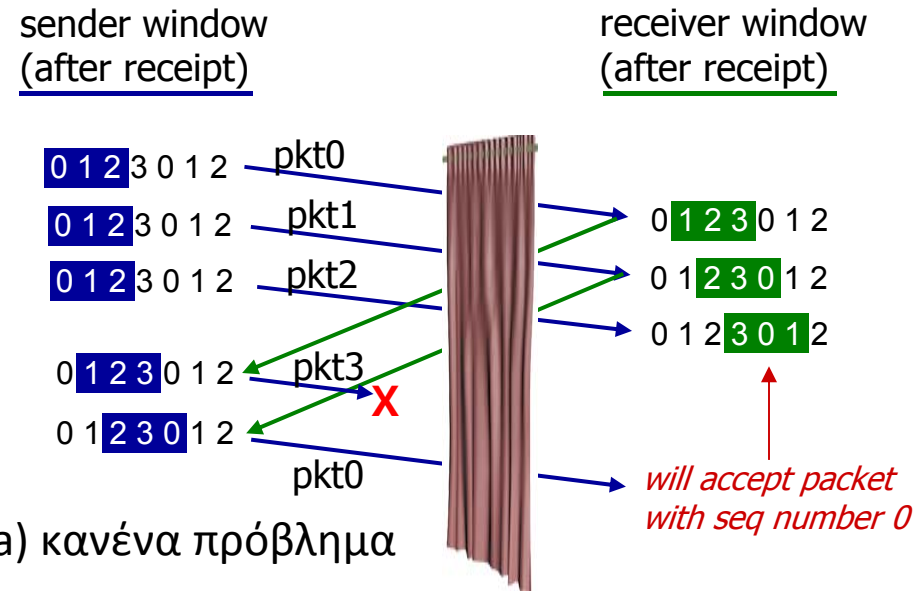


# Επιλεκτική επανάληψη: δίλημμα

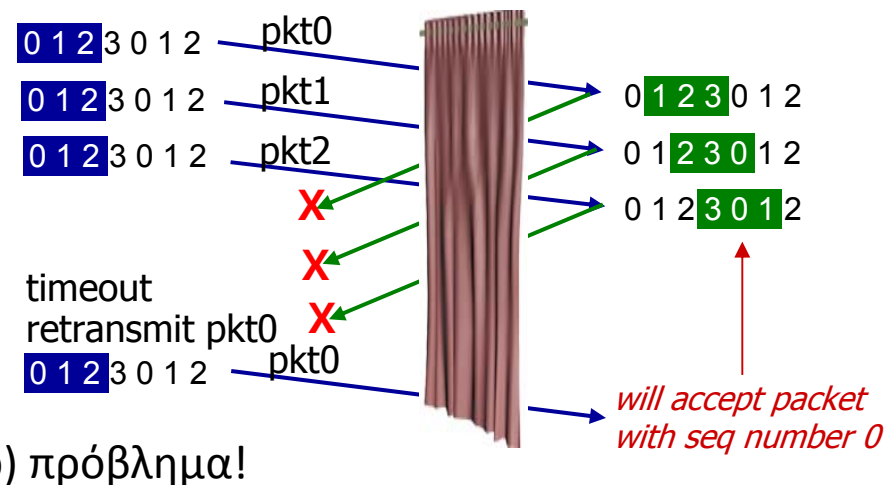
## Παράδειγμα:

- ❖ # ακολουθίας: 0, 1, 2, 3
- ❖ Μέγεθος παραθύρου=3
- ❖ Ο δέκτης δε βλέπει διαφορά στα δύο σενάρια!
- ❖ Τα διπλά (duplicate) δεδομένα γίνονται δεκτά σαν νέα στο (b)

**Ε:** Ποιά η σχέση μεταξύ μεγέθους #ακολουθίας και μεγέθους παραθύρου;



Ο δέκτης δεν μπορεί να δει την πλευρά του αποστολέα. Η συμπεριφορά του δέκτη ίδια και στις 2 περιπτώσεις! κάτι πάει (πολύ) στραβά!



# 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

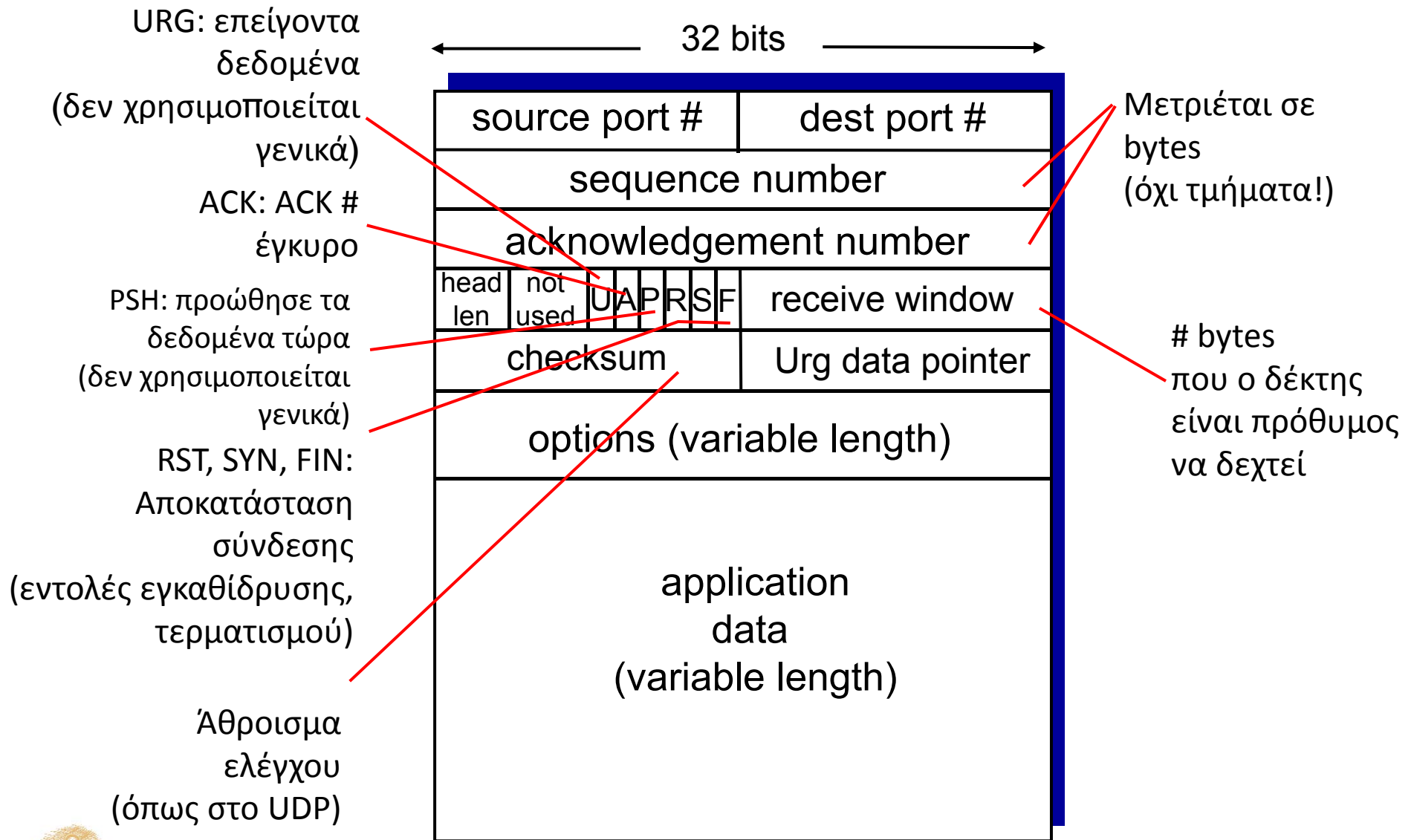


# TCP: Επισκόπηση RFCs: 793, 1122, 1323, 2018, 2581

- ❖ **Από σημείο προς σημείο:**
  - Ένας αποστολέας, ένας δέκτης
- ❖ **Αξιόπιστη, σε σειρά ροή από bytes:**
  - Χωρίς “όρια μηνυμάτων”
- ❖ **Με διοχέτευση:**
  - Οι έλεγχοι συμφόρησης και ροής του TCP καθορίζουν το μέγεθος του παραθύρου
- ❖ **Ενταμιευτές αποστολής και λήψης**
- ❖ **Πλήρως αμφίδρομα δεδομένα:**
  - Δικατευθυντική ροή δεδομένων στην ίδια σύνδεση
  - MSS: maximum segment size (μέγιστο μέγεθος τμήματος)
- ❖ **Συνδεδεσμένη:**
  - 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



# Δομή τμήματος TCP



# TCP: αριθμοί ακολουθίας και ACKs

## Αριθμοί ακολουθίας (Seq. #'s):

- Αριθμός του πρώτου byte των δεδομένων του τμήματος

## ACKs:

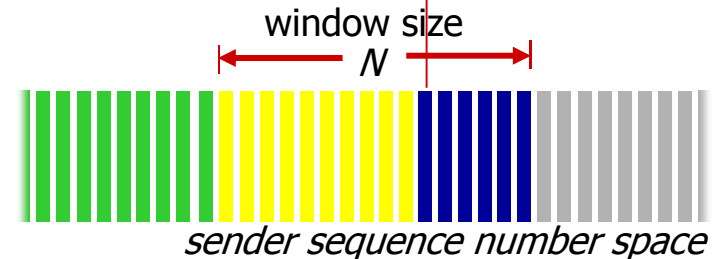
- seq # του επόμενου byte που αναμένεται από την άλλη πλευρά
- συσσωρευτικά ACK

## Ε: 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



sent  
ACKed

sent, not-  
yet ACKed  
("in-  
flight")

usable  
but not  
yet sent

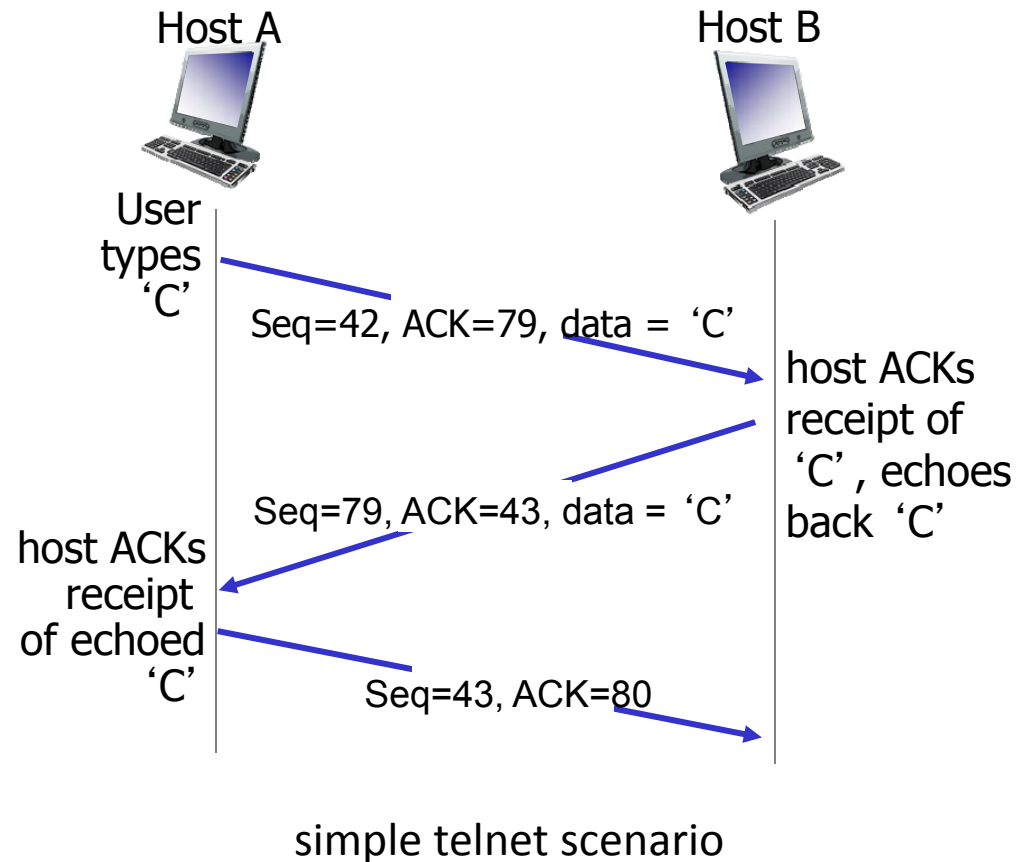
not  
usable

incoming segment to sender

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



# TCP αριθμοί ακολουθίας, ACKs



# 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



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

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

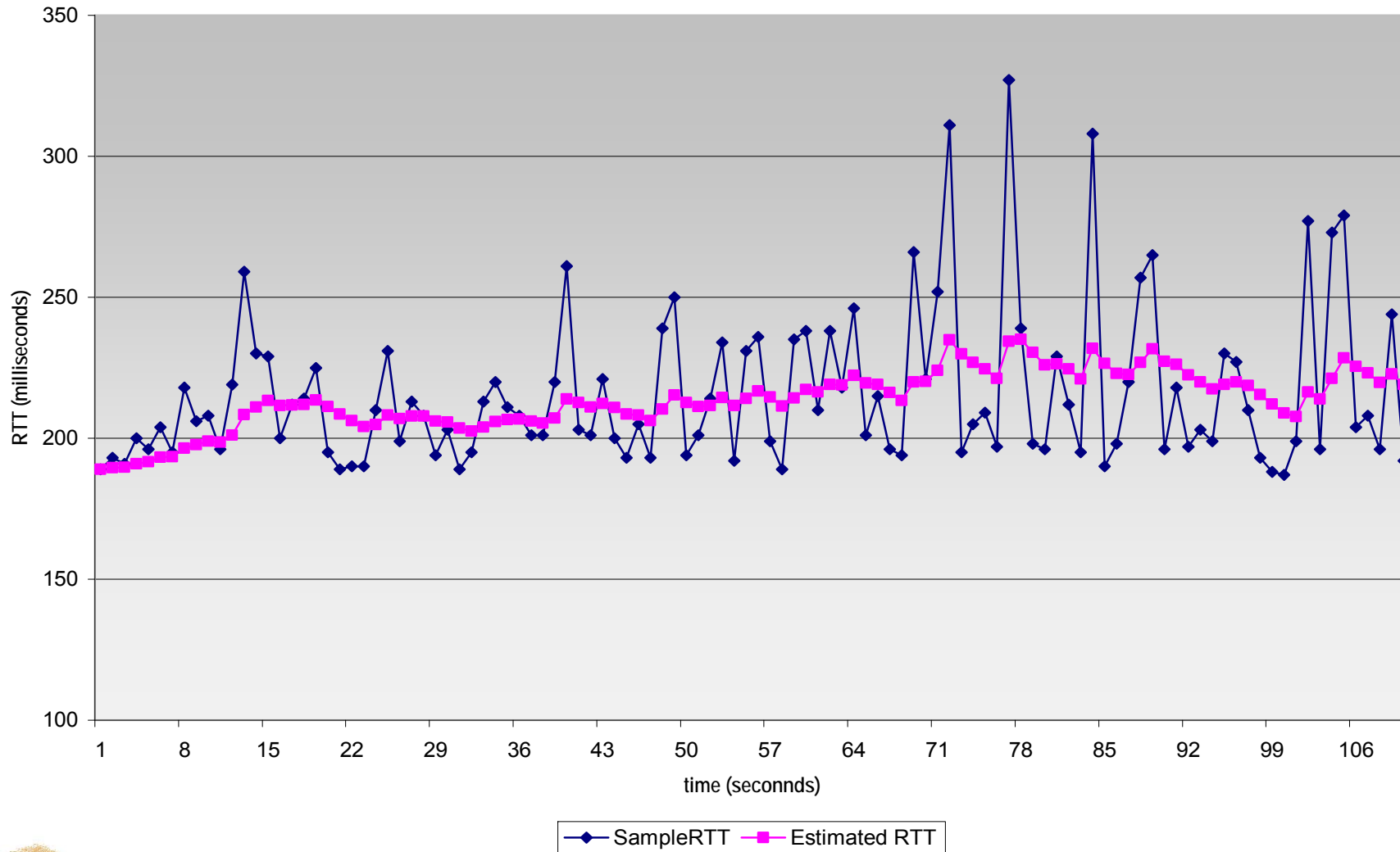
- Εκθετική σταθμισμένη κινητή μέση τιμή (Exponentially weighted moving average)
- η επίδραση των παλαιών δειγμάτων μειώνεται εκθετικά
- τυπική τιμή:  $\alpha = 0.125$





# Παράδειγμα εκτίμησης του RTT:

RTT: gaia.cs.umass.edu to fantasia.eurecom.fr



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

❖ Καθορισμός timeout: **EstimatedRTT** συν «περιθώριο ασφάλειας»

- Μεγάλη μεταβολή στο **EstimatedRTT** -> μεγαλύτερο περιθώριο ασφάλειας
- Πρώτα εκτιμάται πόσο αποκλίνει το **SampleRTT** από το **EstimatedRTT**:

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

(τυπικά,  $\beta = 0.25$ )

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 * \text{DevRTT}$$



↑  
estimated RTT

↑  
“safety margin”



# 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



# Αξιόπιστη μεταφορά δεδομένων (αμδ) του TCP

- ❖ Το TCP δημιουργεί υπηρεσία «αμδ» πάνω από την αναξιόπιστη υπηρεσία του IP
  - Τμήματα σε διοχέτευση
  - Σωρευτικά acks
  - Το TCP χρησιμοποιεί ένα μόνο χρονομετρητή αναμεταδόσεων
- ❖ Αναμεταδόσεις προκαλούνται από
  - συμβάντα λήξης χρόνου (timeouts)
  - διπλές επιβεβαιώσεις (duplicate ACKs)
- ❖ Αρχικά θεωρούμε απλοποιημένο αποστολέα TCP:
  - αγνοούνται διπλά ACKS
  - αγνοείται έλεγχος ροής, έλεγχος συμφόρησης



# Γεγονότα του αποστολέα TCP:

## 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**

## Λήξη χρόνου (timeout):

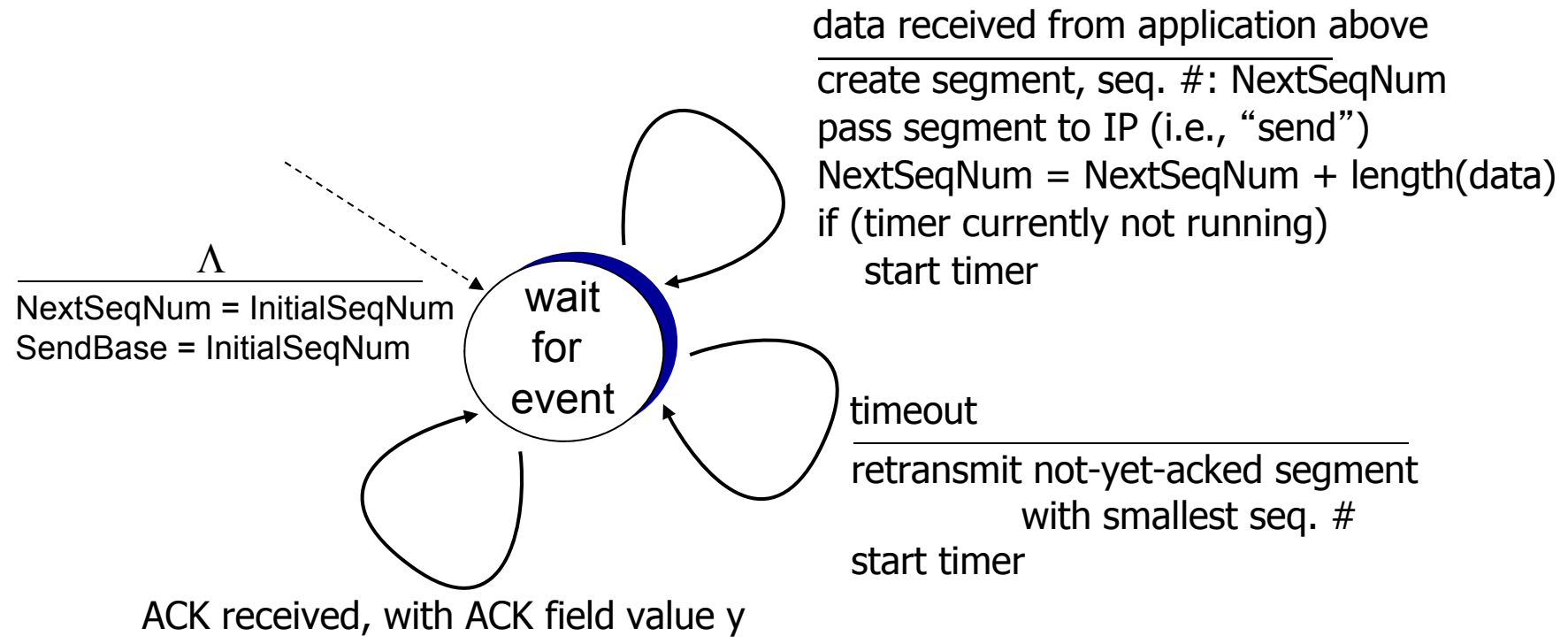
- ❖ επαναμετάδοση του τμήματος που προκάλεσε το timeout
- ❖ επανεκκίνηση χρονομετρητή

## Λήψη ACK:

- ❖ αν επιβεβαιώνει τμήματα που δεν έχουν ήδη επιβεβαιωθεί
  - ανανέωση του τι είναι γνωστό ότι έχει επιβεβαιωθεί
  - εκκίνηση χρονομετρητή αν εξακολουθούν να υπάρχουν τμήματα



# Αποστολέας TCP (απλοποιημένος)

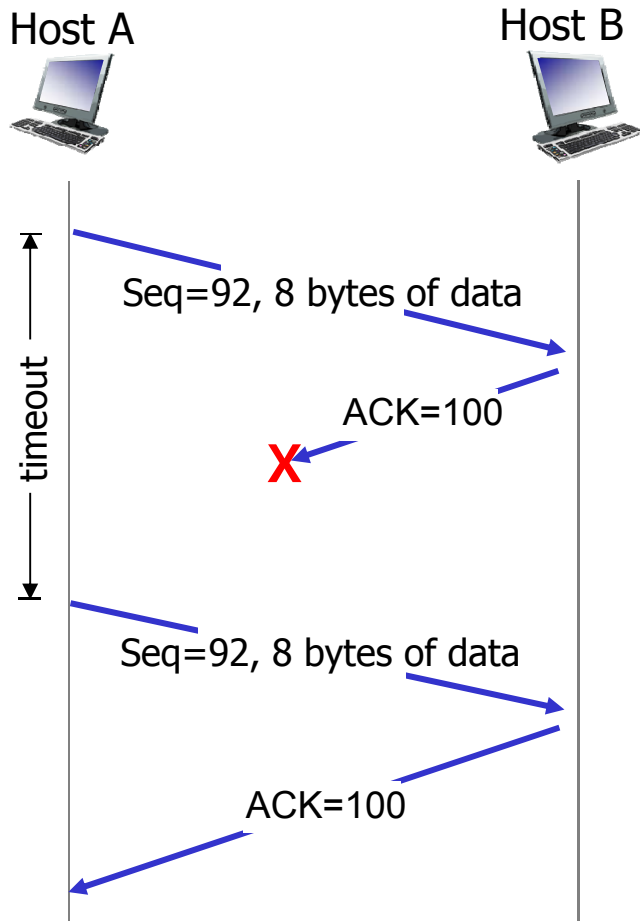


ACK received, with ACK field value  $y$

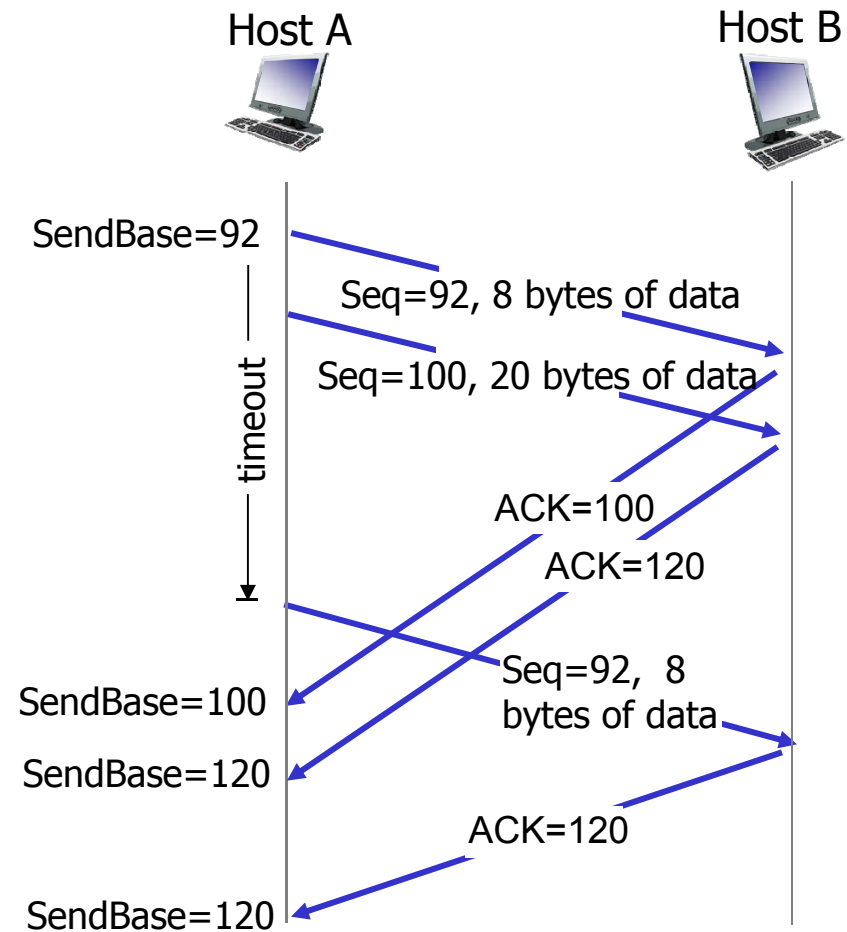
```
if (y > SendBase) {  
    SendBase = y  
    /* SendBase-1: last cumulatively ACKed byte */  
    if (there are currently not-yet-acked segments)  
        start timer  
    else stop timer  
}
```



# Σενάρια αναμεταδόσεων TCP



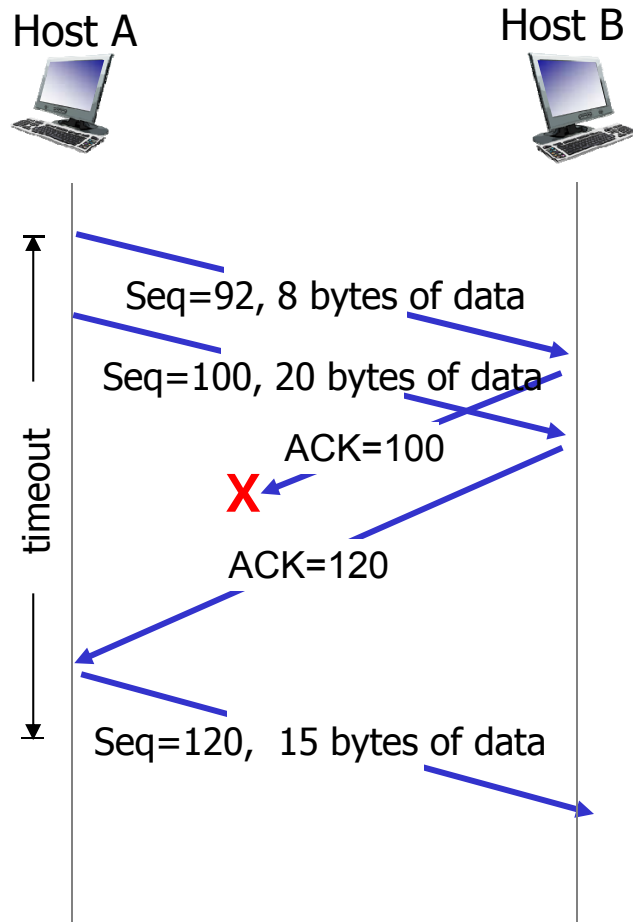
Σενάριο χαμένου ACK



Πρώιμη λήξη χρόνου  
(premature timeout)



# Σενάρια αναμεταδόσεων TCP (συν.)



Σενάριο συσσωρευτικού ACK





# Παραγωγή TCP ACK [RFC 1122, RFC 2581]

<b>Συμβάν στο δέκτη</b>	<b>Ενέργεια δέκτη TCP</b>
Άφιξη τμήματος σε σειρά με αναμενόμενο # ακολουθίας. Όλα τα δεδομένα μέχρι τον αναμενόμενο # ακολουθίας έχουν επιβεβαιωθεί	Καθυστερημένο ACK. Αναμονή 500ms για το επόμενο τμήμα. Αν όχι επόμενο τμήμα στείλε ACK
Άφιξη τμήματος σε σειρά με αναμενόμενο # ακολουθίας. Ένα άλλο τμήμα περιμένει για μετάδοση ACK	Άμεση αποστολή ενός συσσωρευτικού ACK που κάνει επιβεβαίωση και για τα δύο τμήματα που έφτασαν σε σειρά
Άφιξη τμήματος εκτός σειράς με μεγαλύτερο του αναμενόμενου # ακολουθίας. Ανίχνευση κενού	Άμεση αποστολή <i>duplicate</i> ACK που δηλώνει # ακολουθίας επόμενου αναμενόμενου byte
Άφιξη τμήματος που μερικώς ή πλήρως συμπληρώνει κενό στα ληφθέντα δεδομένα	Άμεση αποστολή ACK, αρκεί το τμήμα αυτό να αρχίζει στο κάτω άκρο του κενού

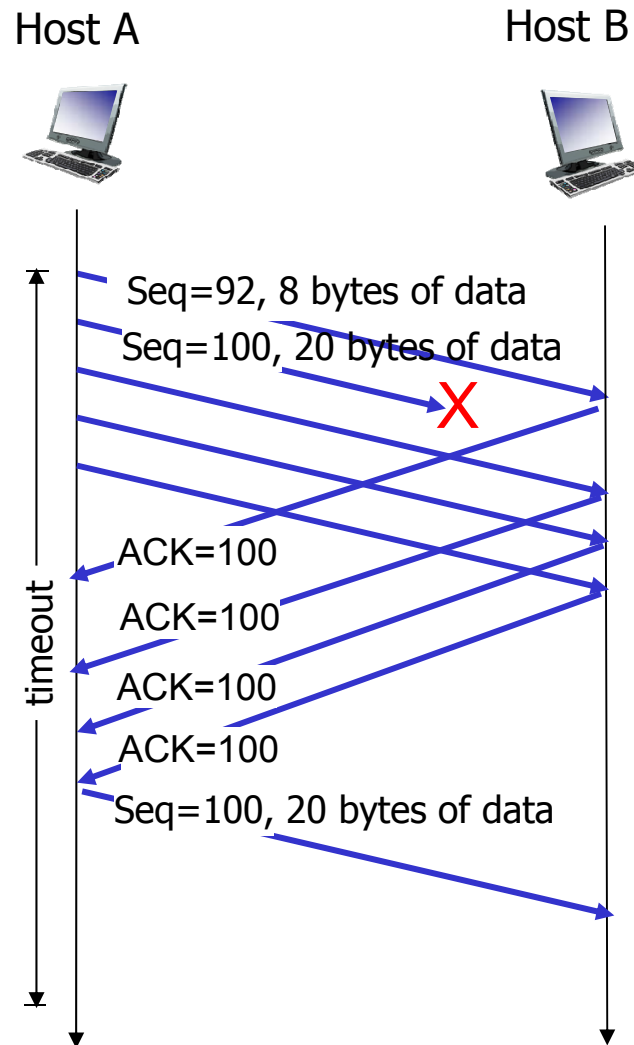


# Ταχεία αναμετάδοση (Fast retransmit)

- ❖ Το διάστημα λήξης χρόνου είναι συχνά σχετικά μεγάλο:
  - μεγάλη καθυστέρηση πριν ξανασταλεί το χαμένο πακέτο
- ❖ Ανίχνευση χαμένων τμημάτων με duplicate ACKs
  - Ο αποστολέας συχνά στέλνει πολλά τμήματα το ένα πίσω από το άλλο
  - Αν ένα τμήμα χαθεί, θα υπάρξουν πιθανώς πολλά διπλά ACKs.
- ❖ **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



# Ταχεία Αναμετάδοση TCP



Ταχεία αναμετάδοση μετά τη λήψη από τον αποστολέα του τριπλού διπλότυπου ACK



# 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

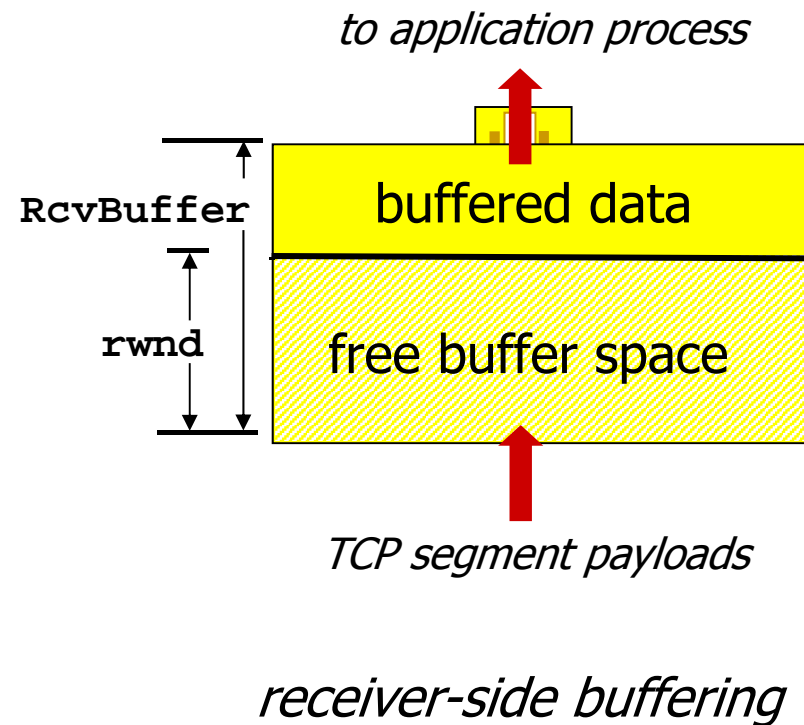
3.7 Έλεγχος συμφόρησης του TCP





# Έλεγχος ροής του TCP

- ❖ Ο δέκτης κοινοποιεί τον ελεύθερο χώρο του ενταμιευτή περιλαμβάνοντας την τιμή *rwnd* στην TCP κεφαλίδα των τμημάτων από το δέκτη προς στον αποστολέα
  - 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 δεδομένα στην τιμή **rwnd** του δέκτη
- ❖ εγγυάται ότι ο ενταμιευτής στον δέκτη δεν υπερχειλίζει



# 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

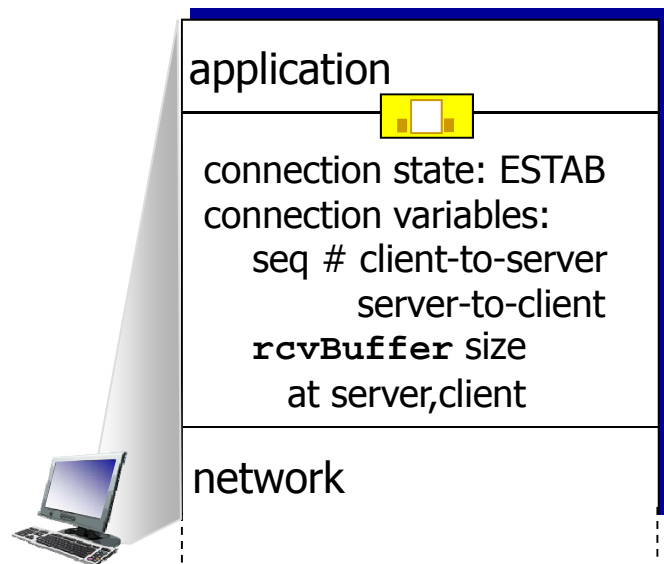
3.7 Έλεγχος συμφόρησης του TCP



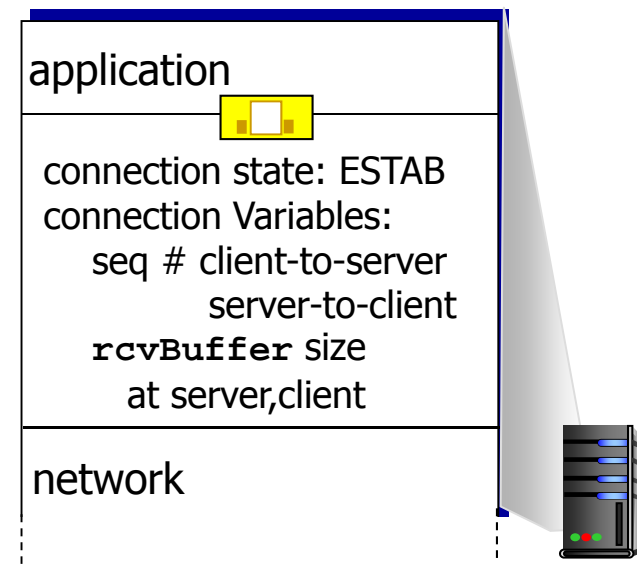
# Διαχείριση σύνδεσης TCP

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



```
Socket clientSocket =  
newSocket("hostname", "port  
number");
```



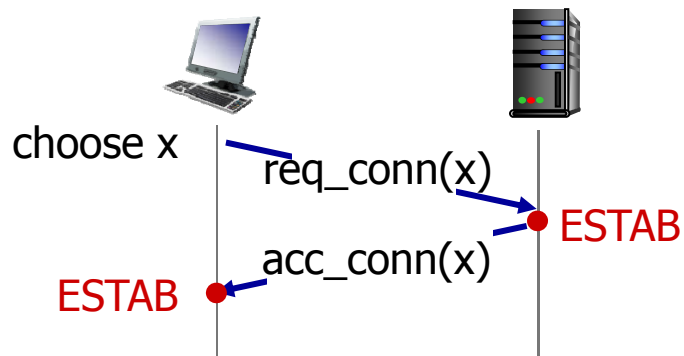
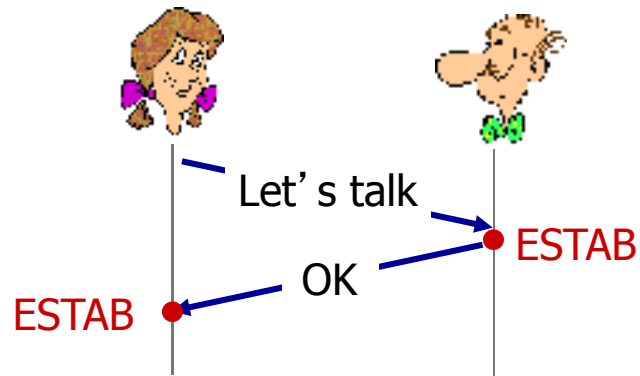
```
Socket connectionSocket =  
welcomeSocket.accept();
```





# Συμφωνία για δημιουργία σύνδεσης

Διμερής χειραψία (2-way handshake):



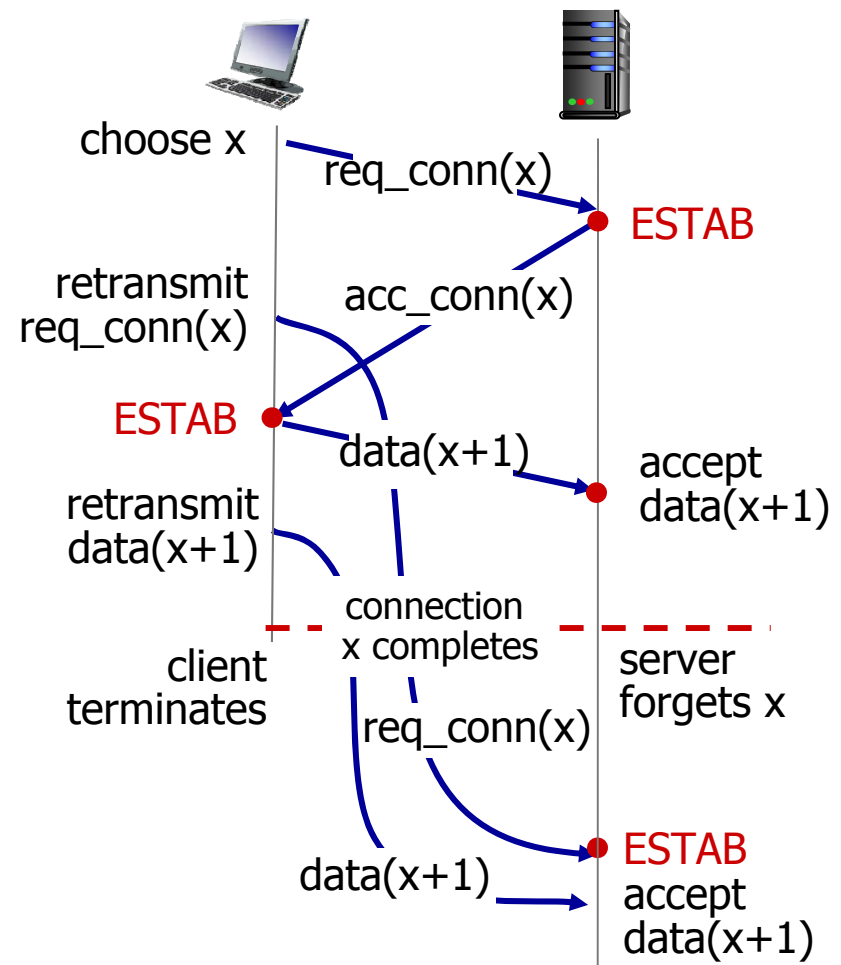
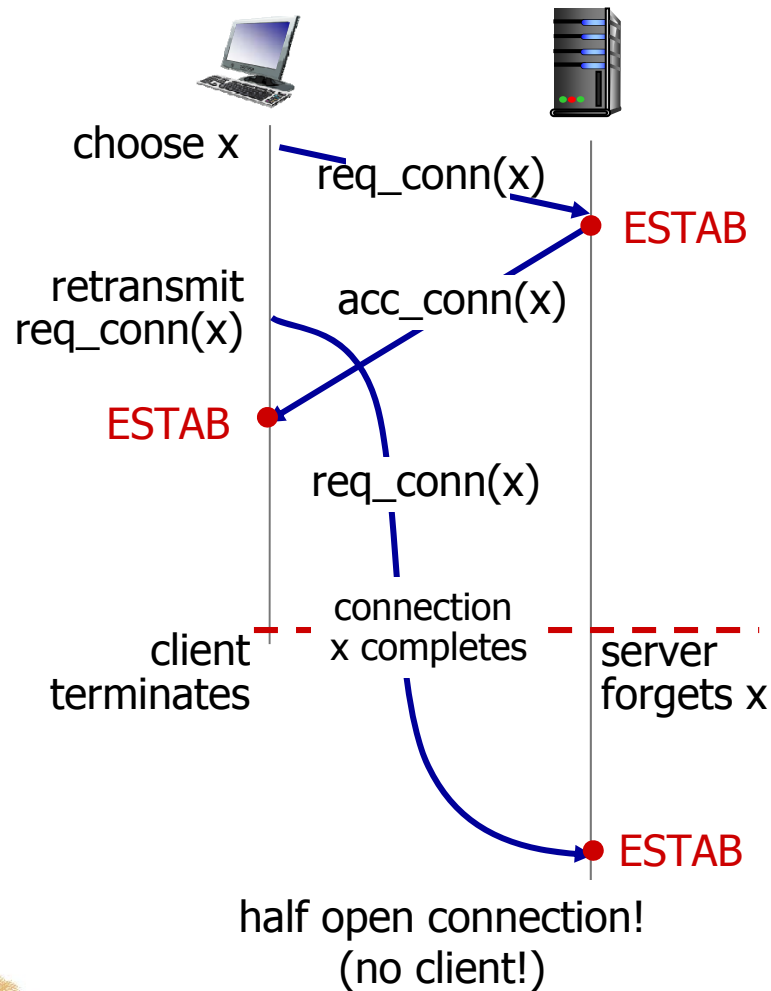
**E:** η διμερής χειραψία δουλεύει πάντα στο δίκτυο;

- ❖ μεταβλητές καθυστερήσεις
- ❖ επαναμεταδιδόμενα μηνύματα (π.χ.  $req\_conn(x)$ ) λόγω απωλειών μηνυμάτων
- ❖ αναδιάταξη μηνύματος
- ❖ δεν μπορεί να δει την "άλλη" πλευρά



# Συμφωνία για δημιουργία σύνδεσης

Σενάρια αποτυχίας στην διμερή χειραψία:



# TCP 3-μερής χειραψία

*client state*

LISTEN

SYNSENT

ESTAB

Διαλέγει αρχικό # ακολουθίας,  $x$  στέλνει TCP SYN msg

Το ληφθέν SYNACK( $x$ ) υποδεικνύει ότι ο server είναι ζωντανός. Στέλνει ACK για το ληφθέν SYNACK. Αυτό το τμήμα μπορεί να περιέχει δεδομένα από πελάτη σε εξυπηρέτη



*server state*

LISTEN

SYN RCVD

ESTAB

SYNbit=1, Seq= $x$

SYNbit=1, Seq= $y$   
ACKbit=1; ACKnum= $x+1$

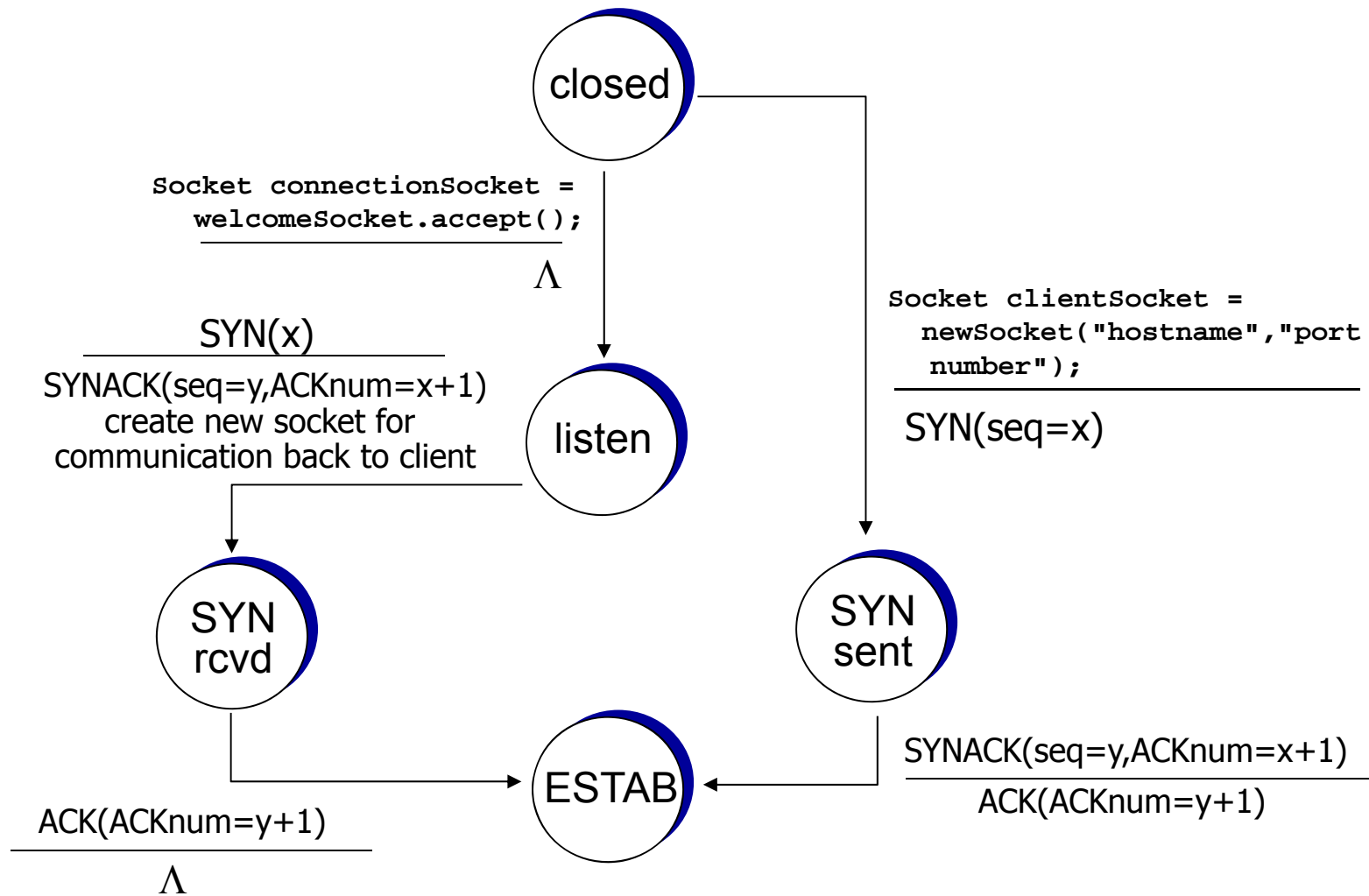
ACKbit=1, ACKnum= $y+1$

Διαλέγει αρχικό #ακολουθίας,  $y$  στέλνει TCP SYNACK msg, βεβαιώνοντας το ληφθέν SYN

Το ληφθέν ACK( $y$ ) υποδεικνύει ότι ο client είναι ζωντανός



# TCP 3-μερής χειραψία FSM



# TCP: Κλείσιμο σύνδεσης

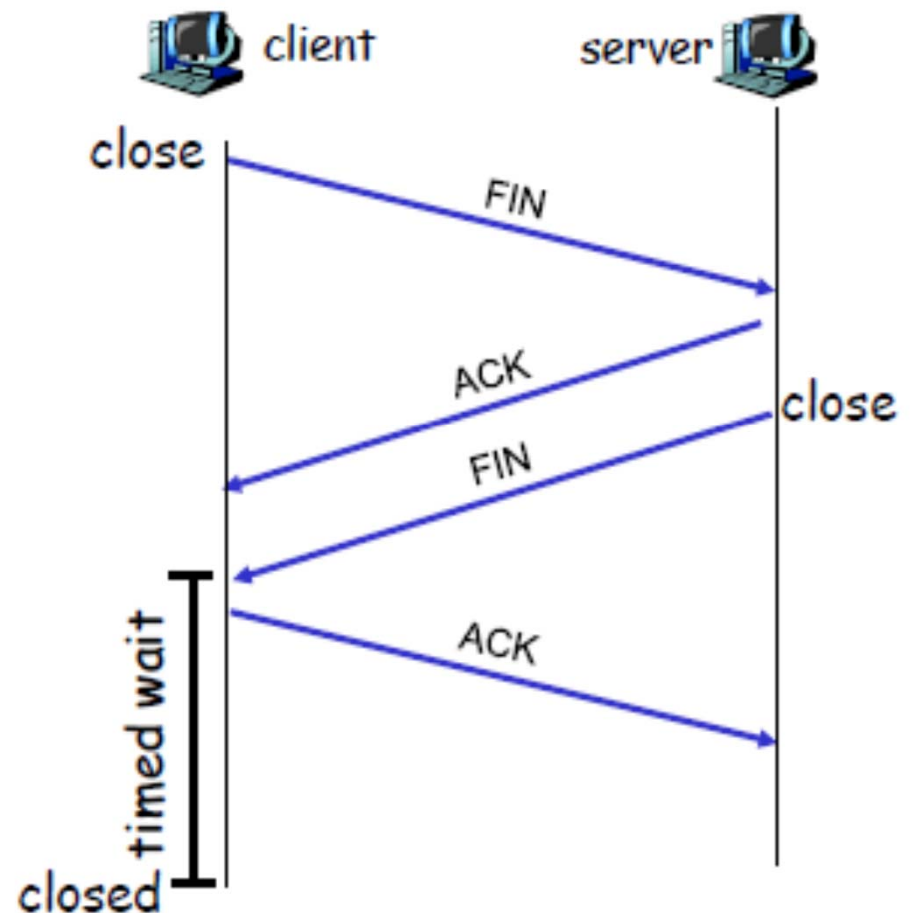
## Κλείσιμο μιας σύνδεσης:

Ο πελάτης κλείνει τη socket:

`clientSocket.close()`

**Βήμα 1:** Το τερματικό σύστημα πελάτης στέλνει τμήμα ελέγχου του TCP FIN στον εξυπηρέτη.

**Βήμα 2:** Ο εξυπηρέτης λαμβάνει FIN, απαντά με ACK. Κλείνει τη σύνδεση, στέλνει FIN.



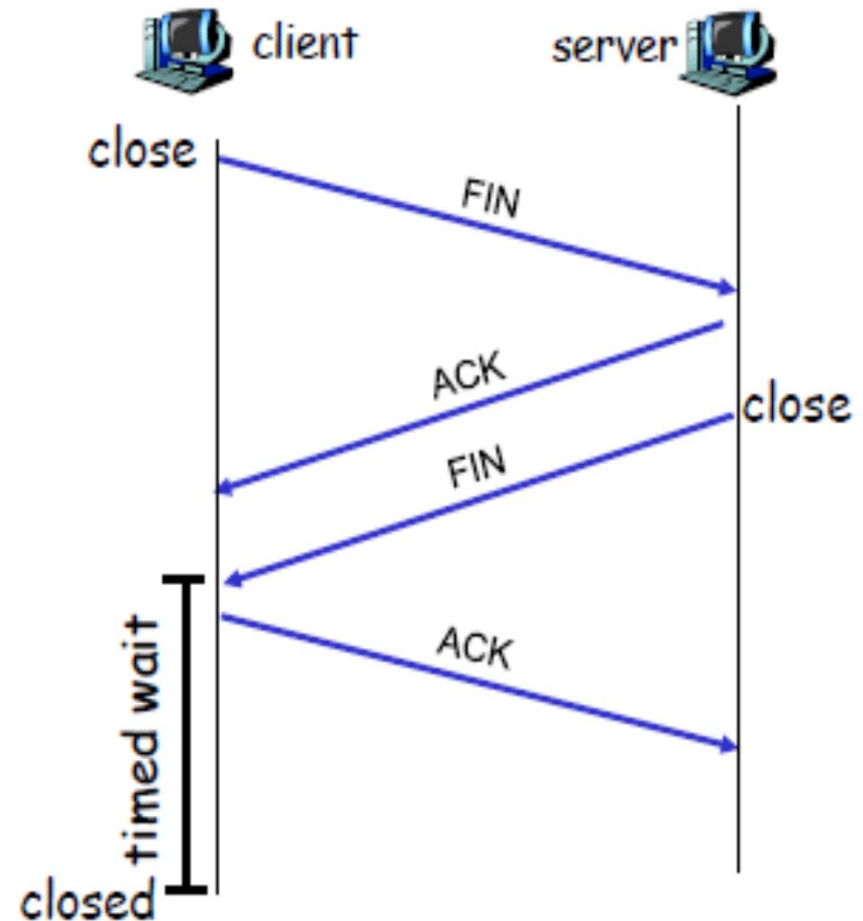
# TCP: Κλείσιμο σύνδεσης (συν.)

**Βήμα 3:** ο πελάτης λαμβάνει FIN, απαντά με ACK.

- Εισέρχεται σε πεπερασμένη αναμονή (“timed wait”) – θα απαντήσει με ACK σε λαμβανόμενα FINs

**Βήμα 4:** Ο εξυπηρέτης, λαμβάνει ACK. Κλειστή σύνδεση.

**Σημείωση:** με μικρή τροποποίηση, μπορεί να διαχειριστεί ταυτόχρονα FINs.



# TCP: Κλείσιμο σύνδεσης (ανάλυση)

*client state*

ESTAB

`clientSocket.close()`

FIN\_WAIT\_1

can no longer send but can receive data

FIN\_WAIT\_2

wait for server close

TIMED\_WAIT

timed wait for  $2 * \text{max segment lifetime}$

CLOSED



FINbit=1, seq=x

ACKbit=1; ACKnum=x+1

FINbit=1, seq=y

ACKbit=1; ACKnum=y+1

can still send data

can no longer send data

*server state*

ESTAB

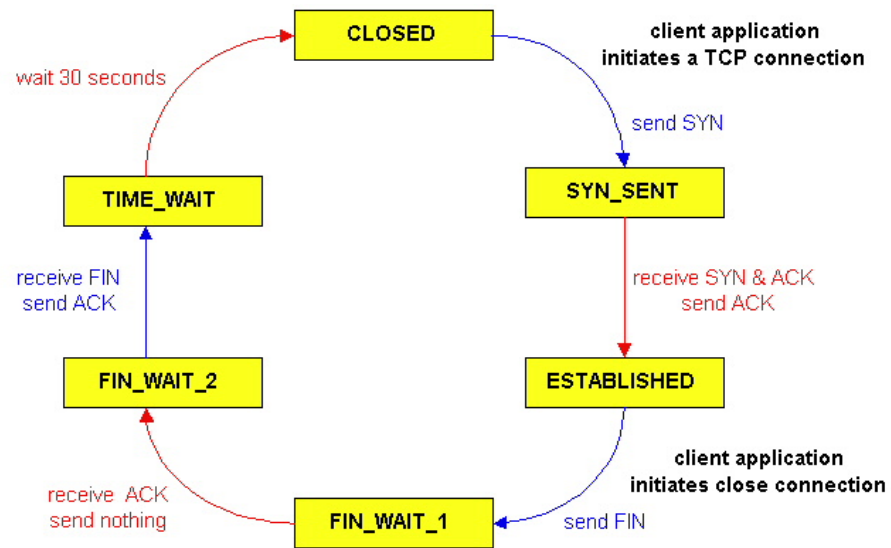
CLOSE\_WAIT

LAST\_ACK

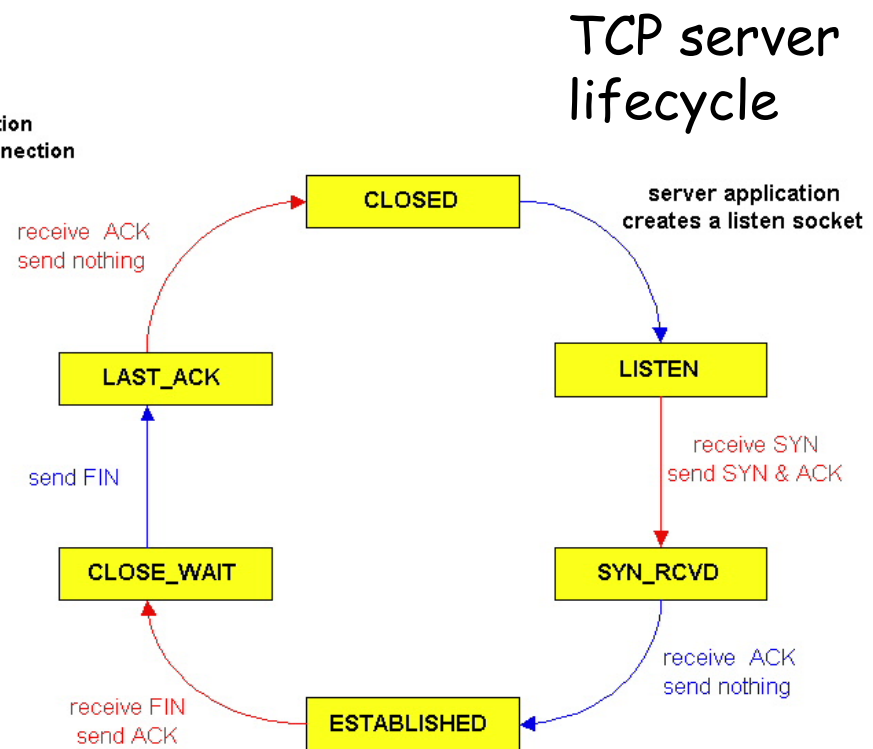
CLOSED



# Ο κύκλος ζωής μιας TCP σύνδεσης



TCP client lifecycle



TCP server lifecycle





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

