



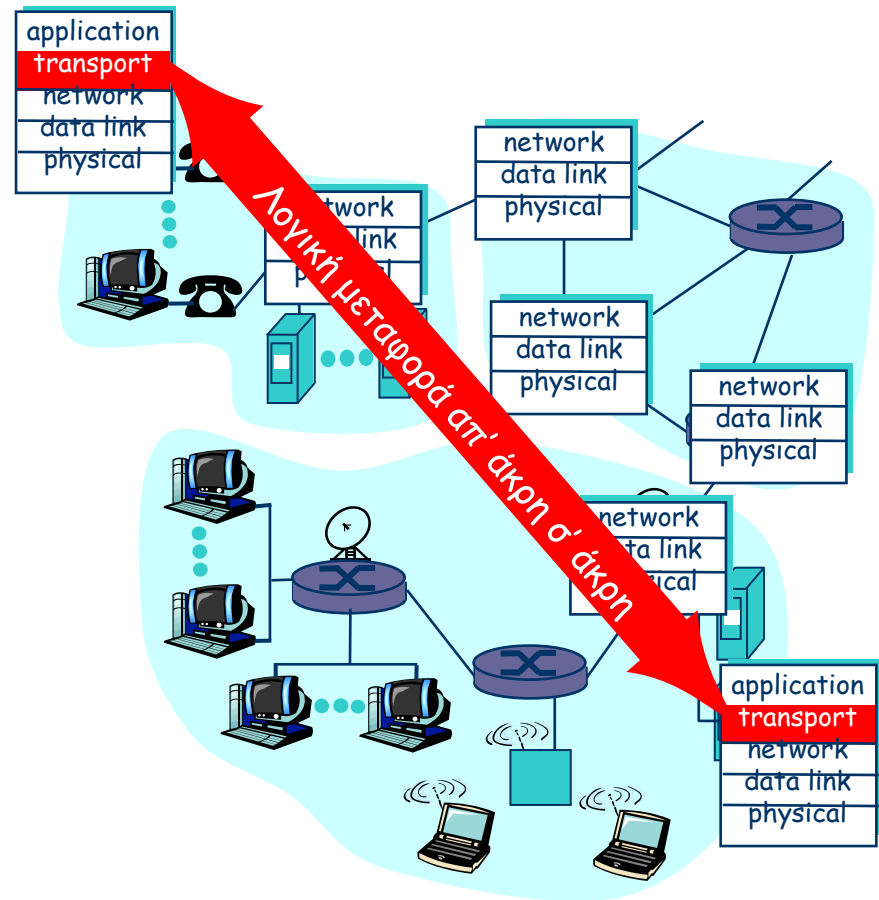
# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Έλεγχος και Αποφυγή  
Συμφόρησης στο TCP

# Υπηρεσίες μεταφοράς στο Internet



- αξιόπιστη, με τη σειρά παράδοση στον προορισμό: TCP
  - έλεγχος συμφόρησης
  - έλεγχος ροής
  - εγκατάσταση σύνδεσης
- αναξιόπιστη ("καλύτερης προσπάθειας"), χωρίς σειρά παράδοση δεδομενογραμμάτων στον προορισμό ή σε πολλαπλούς προορισμούς: UDP
- υπηρεσίες που δεν προσφέρονται:
  - πραγματικού χρόνου
  - εξασφάλιση εύρους ζώνης
  - αξιόπιστη διανομή σε πολλούς προορισμούς



# Τι είναι έλεγχος ροής/συμφόρησης/λαθών



- **Έλεγχος Ροής:** Αλγόριθμοι για την πρόληψη αποστολής πληροφορίας με μεγαλύτερο ρυθμό από αυτόν που μπορεί να παραληφθεί
  - **Έλεγχος Συμφόρησης:** Αλγόριθμοι για την πρόληψη υπερφόρτωσης του δικτύου από τον αποστολέα
  - **Έλεγχος Λαθών:** Αλγόριθμοι ανάκαμψης από την απώλεια πακέτων
- Οι σκοποί των μηχανισμών είναι διαφορετικοί
- Η υλοποίηση όμως γίνεται συνδυασμένα

# Συμφόρηση



- συμβαίνει σε διαφορετικές κλίμακες - από δύο πακέτα μέχρι πολλούς χρήστες
- είναι αναπόφευκτη
- **και μάλλον είναι καλό!**
  - Χρησιμοποιούμε μεταγωγή πακέτου για να έχουμε αποδοτικότερη χρήση των ζεύξεων, άρα οι χώροι αποθήκευσης στους δρομολογητές είναι συχνά γεμάτοι
  - Εάν οι χώροι ήταν άδειοι, η καθυστέρηση θα ήταν χαμηλή, αλλά η χρήση του δικτύου θα ήταν μικρή
  - Εάν οι χώροι είναι σχεδόν γεμάτοι, η καθυστέρηση είναι υψηλή, αλλά χρησιμοποιούμε το δίκτυο πιο αποδοτικά
- Πόσο μεγάλη πρέπει να είναι η συμφόρηση για να είναι υπερβολική?



# Φορτίο προς καθυστέρηση

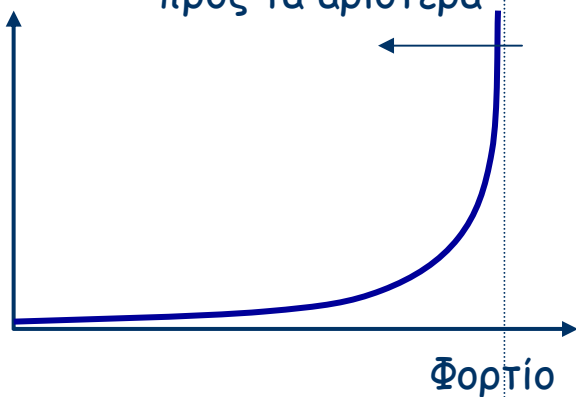
Συνήθης συμπεριφορά συστήματος αναμονής με τυχαίες αφίξεις:

Ένα απλό μέτρο της επίδοσης του δικτύου:

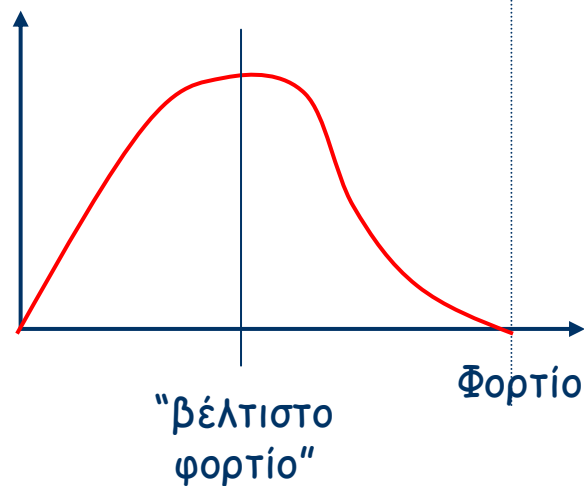
φορτίο/καθυστέρηση

Εκρηκτικότητα κίνησης μετακινεί την ασύμπτωτο προς τα αριστερά

Μέση καθυστέρηση πακέτων



Επίδοση



# Έλεγχος συμφόρησης



- Γιατί πρέπει να γίνει στο στρώμα μεταφοράς;
  - Η πραγματική λύση είναι η επιβράδυνση του αποστολέα
- Χρήση του νόμου "διατήρησης των πακέτων"
  - Κράτα τον αριθμό των πακέτων στο δίκτυο σταθερό
  - Μην εισάγεις νέα πακέτα εάν τα παλαιά δεν φύγουν
- Οι περισσότερες εκπνοές χρόνου στο Internet οφείλονται σε συμφόρηση!

# Επιλογές για έλεγχο συμφόρησης



1. Υλοποίηση στους υπολογιστές (αποστολέα, αποδέκτη) ή στο δίκτυο
2. Βασισμένες σε κρατήσεις ή σε ανάδραση
3. Βασισμένες σε παράθυρα ή σε ρυθμό

# Έλεγχος συμμόρφωσης στο TCP



- Το TCP διαθέτει μηχανισμό ελέγχου συμμόρφωσης:
  - υλοποιείται στον αποστολέα και βασίζεται στην ανάδραση και στο παράθυρο
- Οι πηγές TCP προσπαθούν να προσδιορίσουν τη διαθέσιμη χωρητικότητα του δικτύου
- Το TCP στέλνει πακέτα και αντιδρά σε παρατηρήσιμα γεγονότα



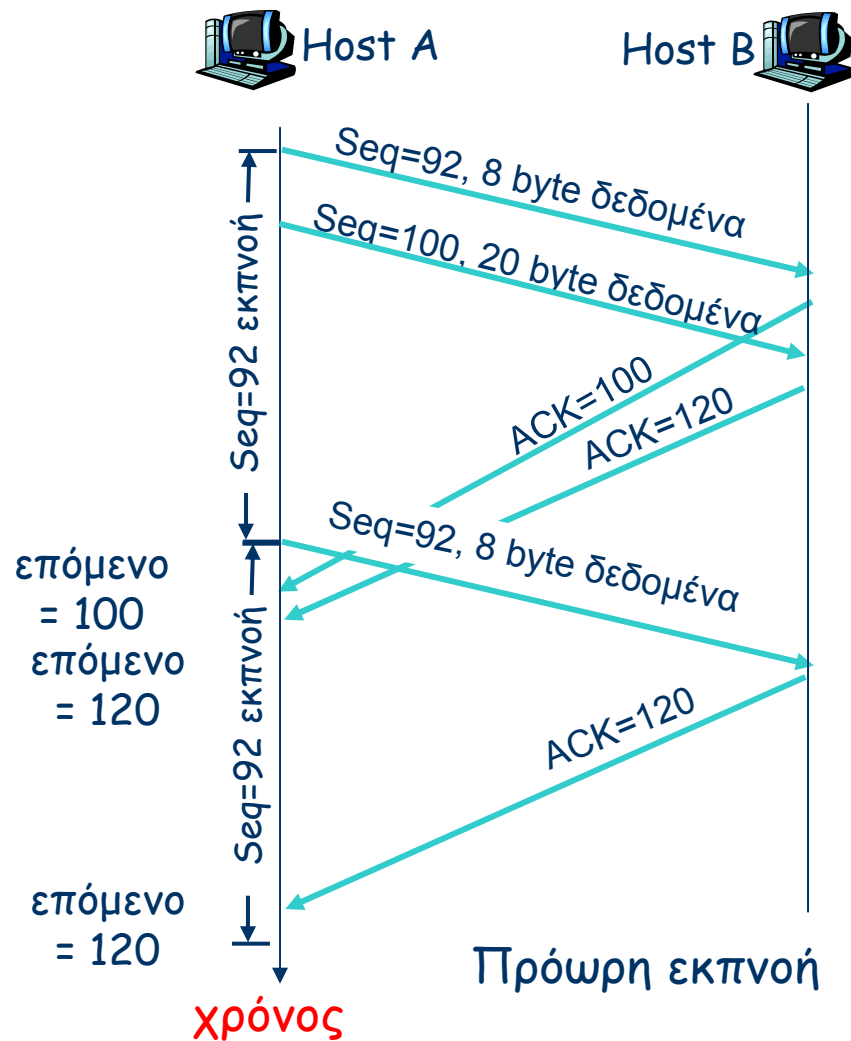
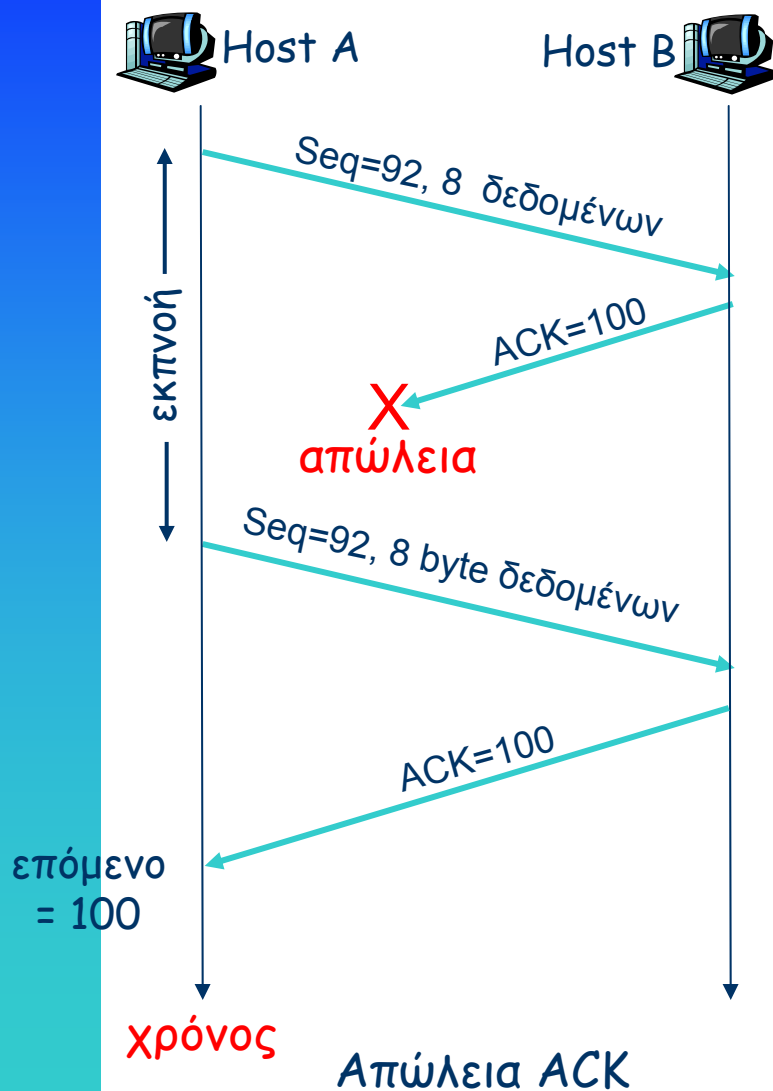
# Βασικά σημεία



- Οι αποστολείς TCP μπορούν να ανιχνεύσουν τη συμφόρηση και να μειώσουν τον ρυθμό αποστολής
- Το TCP τροποποιεί τον ρυθμό αποστολής σύμφωνα με τον κανόνα Προσθετικής Αύξησης, Πολλαπλασιαστικής Μείωσης AIMD (Additive Increase, Multiplicative Decrease)
- Για το ξεκίνημα της ροής, το TCP χρησιμοποιεί ένα **γρήγορο** μηχανισμό εκκίνησης που αποκαλείται "**αργή αρχή**"!
- Οι δρομολογητές επιβραδύνουν τους αποστολείς TCP αποθηκεύοντας τα πακέτα και αυξάνοντας έτσι την καθυστέρηση

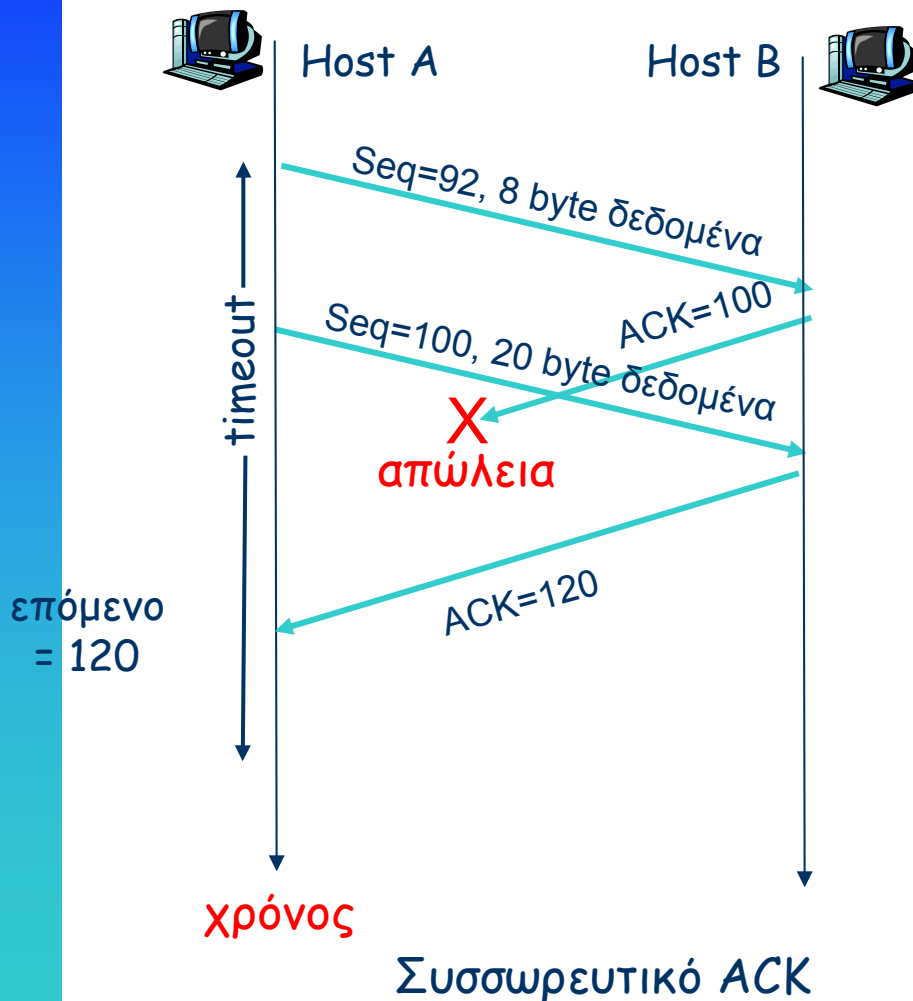


# Παρατηρήσιμα γεγονότα





# Παρατηρήσιμα γεγονότα



# Παραγωγή των ACK στο TCP



Γεγονός στον δέκτη	Ενέργεια TCP
Άφιξη στη σειρά τεμαχίου με τον αναμενόμενο seq #. Έχει επιβεβαιωθεί η λήψη των προηγούμενων	Καθυστέρηση ACK. Αναμονή μέχρι 500ms για το επόμενο τεμάχιο. Εάν δεν έρθει, αποστολή ACK
Άφιξη στη σειρά τεμαχίου με τον αναμενόμενο seq #. Εκκρεμεί ένα ACK.	Άμεση αποστολή ενός συσσωρευτικού ACK, για αμφότερα τα τεμάχια
Άφιξη τεμαχίου εκτός σειράς με seq # μεγαλύτερο του αναμενόμενου. Διαπίστωση κενού.	Άμεση αποστολή ταυτόσημου ACK, με seq # το αναμενόμενο byte
Άφιξη τεμαχίου που καλύπτει πλήρως ή μερικώς το κενό	Άμεση αποστολή ACK, εάν κλείνει το κάτω μέρος του κενού

# Μηχανισμός ελέγχου συμφόρησης TCP



- Υπάρχουν δύο ενδεχόμενα προβλήματα
  - Η χωρητικότητα του δικτύου
  - Η χωρητικότητα του αποδέκτη
- Απαιτείται χωριστή αντιμετώπιση του κάθε προβλήματος
- Κάθε αποστολέας διαθέτει δύο παράθυρα
  - Το **διαφημιζόμενο παράθυρο (advertised window)** από τον αποδέκτη
  - Το **παράθυρο συμφόρησης (congestion window)**

# Μηχανισμός ελέγχου συμφόρησης TCP



- Ο αποστολέας τηρεί τις μεταβλητές:
  - Congestion Window (*cwnd*)
  - Slow-start threshold Value (*ssthresh*)  
αρχική τιμή το διαφημιζόμενο μέγεθος παραθύρου
  - Advertised Window (*flow\_win*)
- Ο αποστολέας TCP αλλάζει τον ρυθμό αποστολής τροποποιώντας το παράθυρο αποστολής:

$$Window = \min \{ cwnd, flow\_win \}$$

Πομπός

Δέκτης

# Μηχανισμός ελέγχου συμφόρησης TCP



- Το πλήθος byte που μπορούν να αποσταλούν είναι το μικρότερο από τα δύο παράθυρα
- Δηλαδή, στείλε με τον ρυθμό του αργότερου:
  - δίκτυο ή αποδέκτης

# Το παράθυρο συμφόρησης



- Το εκτιμά ο αποστολέας από την ανάδραση που λαμβάνει από το δίκτυο
- **ιδανικά:** μετάδωσε όσο γρήγορα μπορείς χωρίς απώλειες (παράθυρο συμφόρησης όσο το δυνατό μεγαλύτερο)
- **"διερεύνηση"** για εύρος ζώνης που μπορεί να χρησιμοποιηθεί:
  - **αύξησε** το παράθυρο συμφόρησης μέχρι να υπάρξουν απώλειες (συμφόρηση)
  - **μείωσε** το παράθυρο συμφόρησης και συνέχισε τη διερεύνηση (αυξάνοντάς το) πάλι



# Φάσεις ελέγχου συμφόρησης TCP



- Ο έλεγχος συμφόρησης στο TCP λειτουργεί με δύο τρόπους:
  - **αργή αρχή (slow start)**
    - $(cwnd < ssthresh)$
  - **αποφυγή συμφόρησης (congestion avoidance)**
    - $(cwnd \geq ssthresh)$

# Αργή Αρχή



- Αρχική τιμή:  $cwnd = 1$ 
  - Σημείωση: Η μονάδα μέτρησης είναι το μέγεθος τεμαχίου
  - Το TCP στην πραγματικότητα μετρά byte και επομένως αυξάνει κατά 1 MSS (maximum segment size)
- Ο δέκτης στέλνει επαλήθευση ACK
  - Εν γένει, ο δέκτης TCP στέλνει ένα ACK για κάθε δεύτερο τεμάχιο
- Κάθε φορά που λαμβάνεται ACK στον αποστολέα, το παράθυρο αυξάνει κατά 1 τεμάχιο:  
 $cwnd = cwnd + 1$ 
  - Εάν το ACK επαληθεύει δύο τεμάχια, το  $cwnd$  πάλι αυξάνει κατά 1 τεμάχιο
  - Ακόμη και εάν το ACK επαληθεύει τεμάχιο μικρότερο από MSS byte, το  $cwnd$  πάλι αυξάνει κατά 1 τεμάχιο

# Αργή Αρχή

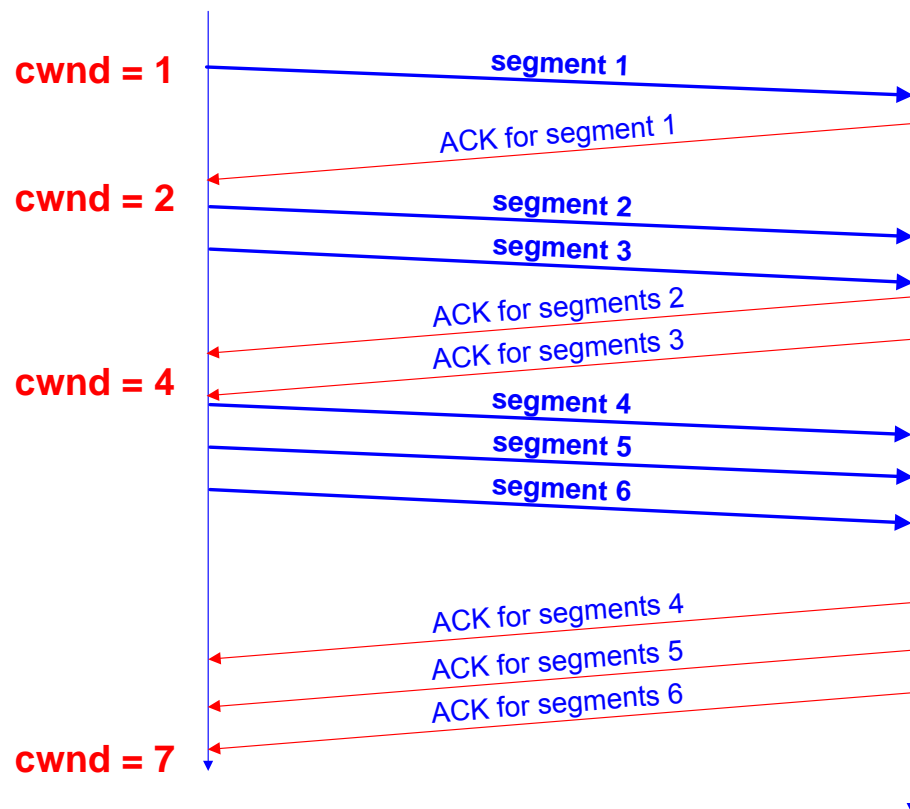


- Είναι η αργή αρχή πραγματικά **αργή**;
- Όχι, στην πραγματικότητα το *cwnd* αυξάνει **εκθετικά** ανά RTT
- Γιατί αποκαλείται αργή αρχή;
  - Επειδή το TCP αρχικά δεν διαθέτει μηχανισμό ελέγχου της συμφόρησης
  - Ο αποστολέας μπορούσε να στείλει ένα πλήρες παράθυρο δεδομένων
- Η "αργή αρχή" είναι αργή μόνο σε σχέση με το να αποσταλούν δεδομένα όσο και το μέγεθος του διαφημιζόμενου παράθυρου



# Παράδειγμα Αργής Αρχής

- Το παράθυρο συμφόρησης μεγαλώνει πολύ γρήγορα
  - Για κάθε ACK το **cwnd** αυξάνει κατά 1 άσχετα από τον αριθμό των τεμαχίων που έχουν λάβει ACK
- Το TCP μειώνει την αύξηση του **cwnd** όταν **cwnd  $\geq$  ssthresh**



# Αποφυγή Συμφόρησης



- Η φάση της αποφυγής συμφόρησης αρχίζει όταν το *cwnd* φτάσει το κατώφλι αργής αρχής
- Εάν *cwnd*  $\geq$  *ssthresh* τότε κάθε φορά που λαμβάνεται ένα ACK, η αύξηση του *cwnd* γίνεται ως εξής:

$$cwnd = cwnd + 1 / [cwnd]$$

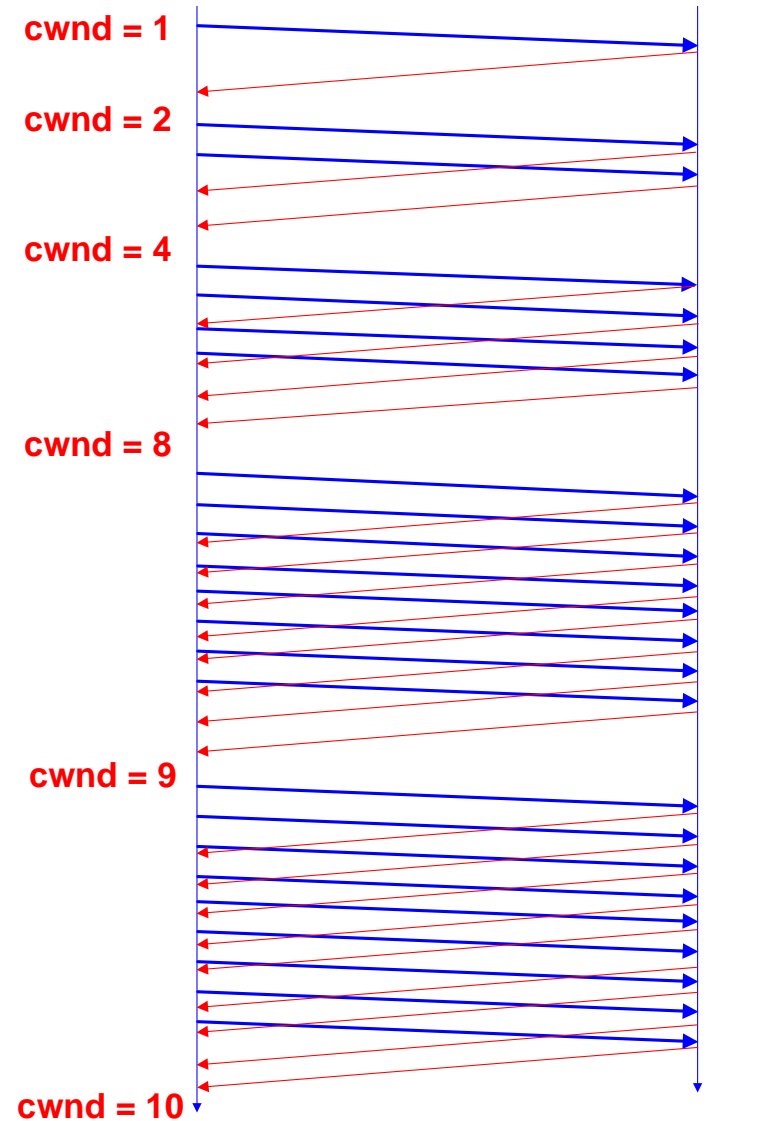
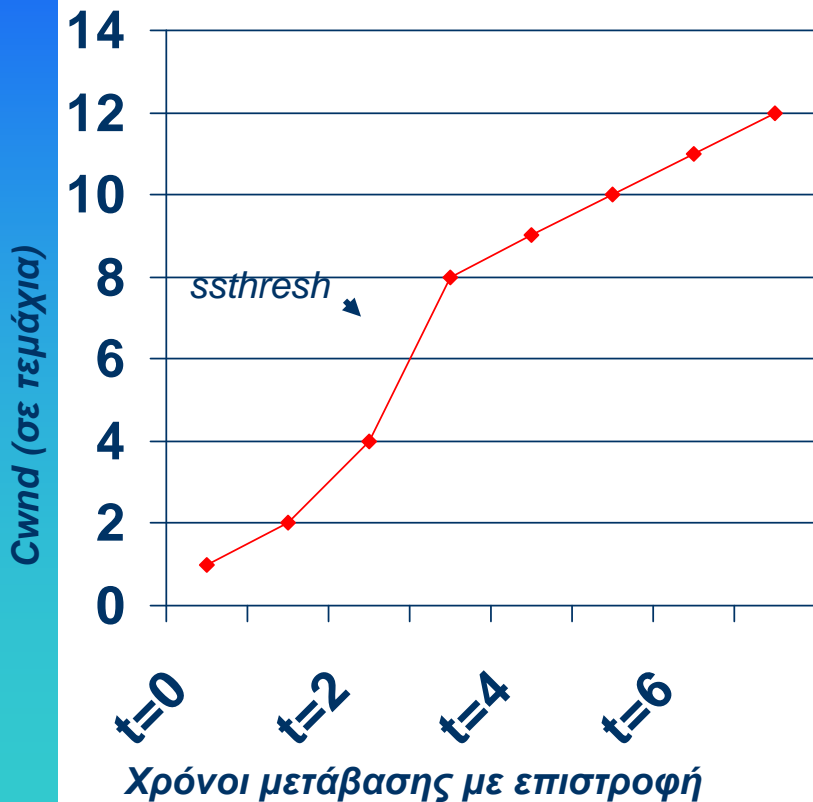
όπου  $[cwnd]$  είναι το ακέραιο μέρος του *cwnd*

- Άρα το *cwnd* αυξάνει κατά 1 μόνο εάν επαληθευθούν όλα τα *cwnd* τεμάχια
- Οι επιτυχημένες μεταδόσεις προκαλούν γραμμική αύξηση του *cwnd* ανά RTT

# Παράδειγμα Αργής Αρχής - Αποφυγής Συμφόρησης



Υποθέστε ότι  $ssthresh = 8$



# Ανίχνευση της συμφόρησης



- Το TCP υποθέτει ότι υπάρχει συμφόρηση όταν ανιχνεύσει απώλεια κάποιου πακέτου
- Ο αποστολέας TCP μπορεί να ανιχνεύσει απώλεια πακέτων μέσω της:
  - Εκπνοής του χρονομέτρου επαναμετάδοσης
  - Λήψης ταυτόσημων ACK
- Το TCP υποθέτει ότι η απώλεια οφείλεται σε **υπερχείλιση** των χώρων αποθήκευσης στους δρομολογητές

# Αντίδραση στην συμφόρηση



- Το TCP ερμηνεύει την εκπνοή χρόνου με απόλυτο τρόπο (**συμφόρηση**)
- Όταν συμβεί εκπνοή χρόνου, ο αποστολέας:
  - μειώνει το *ssthresh* στο μισό της τρέχουσας τιμής  
 $ssthresh = cwnd / 2$
  - θέτει το *cwnd* στην αρχική τιμή  $cwnd = 1$
  - και επανέρχεται στην φάση της αργής αρχής



# Ψευδοκώδικας ελέγχου συμφόρησης



Αρχικά:

```
cwnd = 1;
```

```
ssthresh = advertised window  
size;
```

Όταν ληφθεί Ack:

```
if (cwnd < ssthresh)
```

```
    /* Αργή Αρχή */
```

```
    cwnd = cwnd + 1;
```

```
else
```

```
    /* Αποφυγή Συμφόρησης */
```

```
    cwnd = cwnd + 1/[cwnd];
```

Εκπνοή χρόνου:

```
    /* Πολλαπλασιαστική μείωση */
```

```
    ssthresh = cwnd/2;
```

```
    cwnd = 1;
```

```
while (next < unack + win)  
    μετάδοση επόμενου  
    πακέτου;
```

Όπου

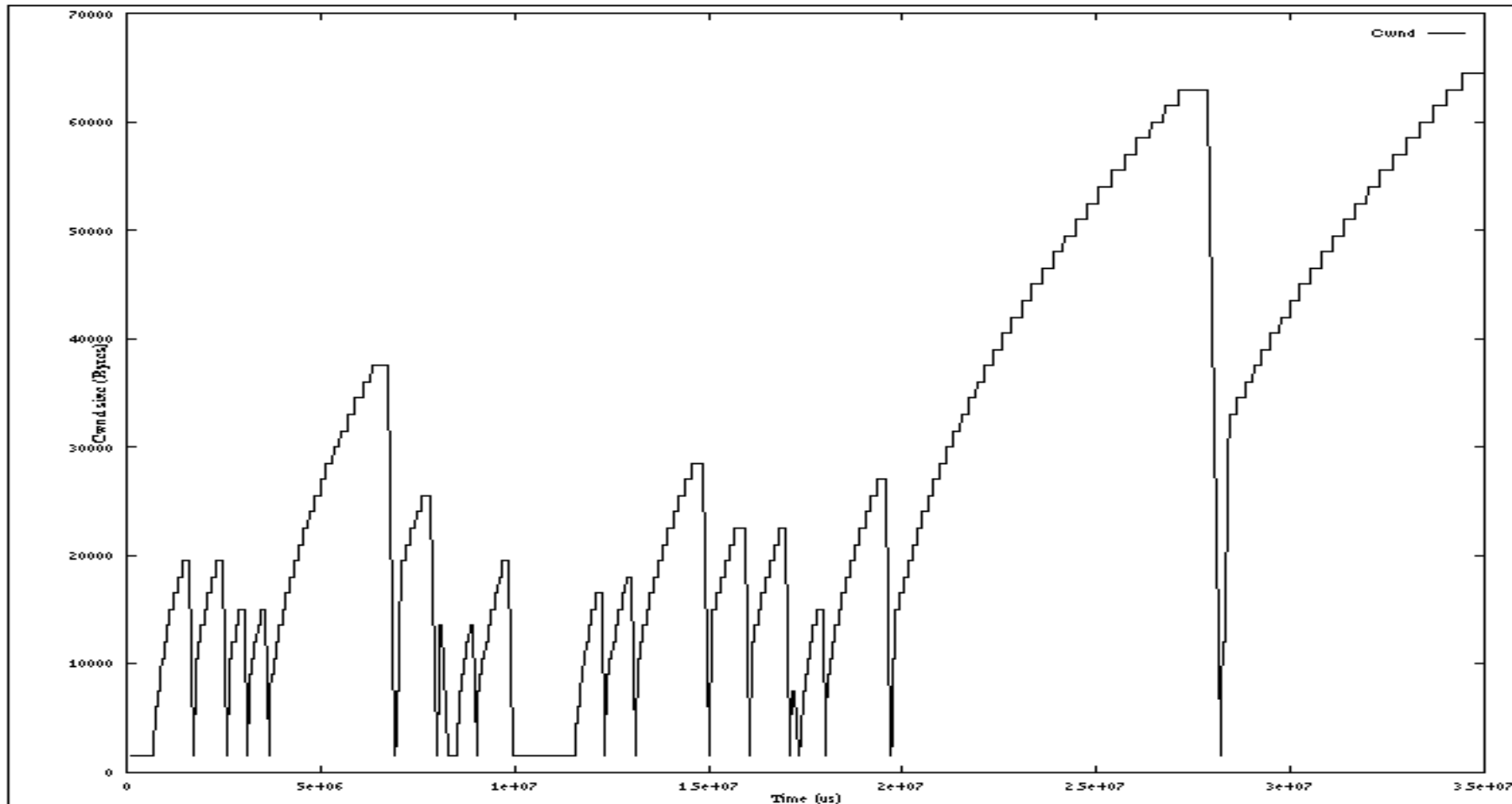
```
win = min(cwnd, adv_win);
```



# Το "πριόνι" TCP: Αργή Αρχή / Αποφυγή Συμφόρησης



- Τυπικό διάγραμμα του *cwnd* για σύνδεση TCP (MSS = 1500 byte) και TCP Tahoe



# Παραλλαγές του Ελέγχου Συμφόρησης TCP

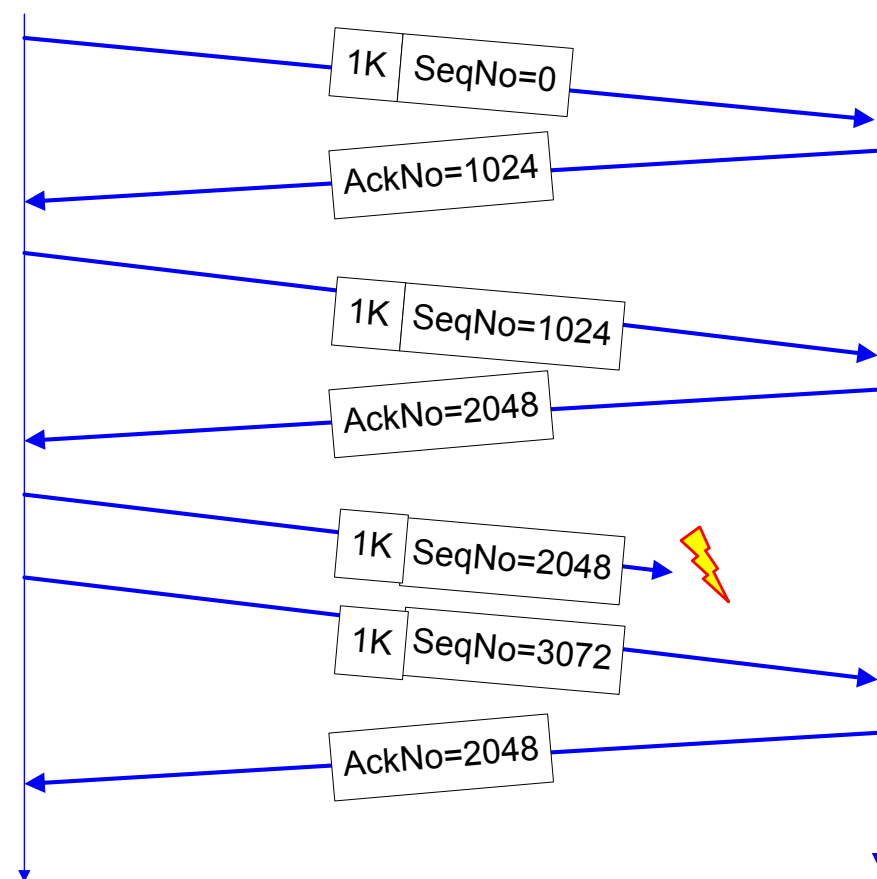


- **TCP Tahoe** (1988, FreeBSD 4.3 Tahoe)
  - Αργή Αρχή
  - Αποφυγή Συμφόρησης
  - Ταχεία Επαναμετάδοση
- **TCP Reno** (1990, FreeBSD 4.3 Reno)
  - Ταχεία Ανάκαμψη
- **RED** (Floyd and Jacobson 1993)
- **New Reno** (1996)
- **SACK** (1996)

# Επαληθεύσεις στο TCP



- Ο δέκτης στέλνει ACK στον αποστολέα
  - Το ACK χρησιμοποιείται για έλεγχο ροής, έλεγχο λαθών και έλεγχο συμφόρησης
- Ο αύξων αριθμός ACK που αποστέλλεται είναι ο επόμενος αναμενόμενος αύξων αριθμός
- Καθυστερημένο ACK: Ο δέκτης TCP συνήθως καθυστερεί την αποστολή του ACK
  - **Γιατί;**
- Το ACK δεν καθυστερεί εάν ληφθεί πακέτο εκτός σειράς
  - **Γιατί;**

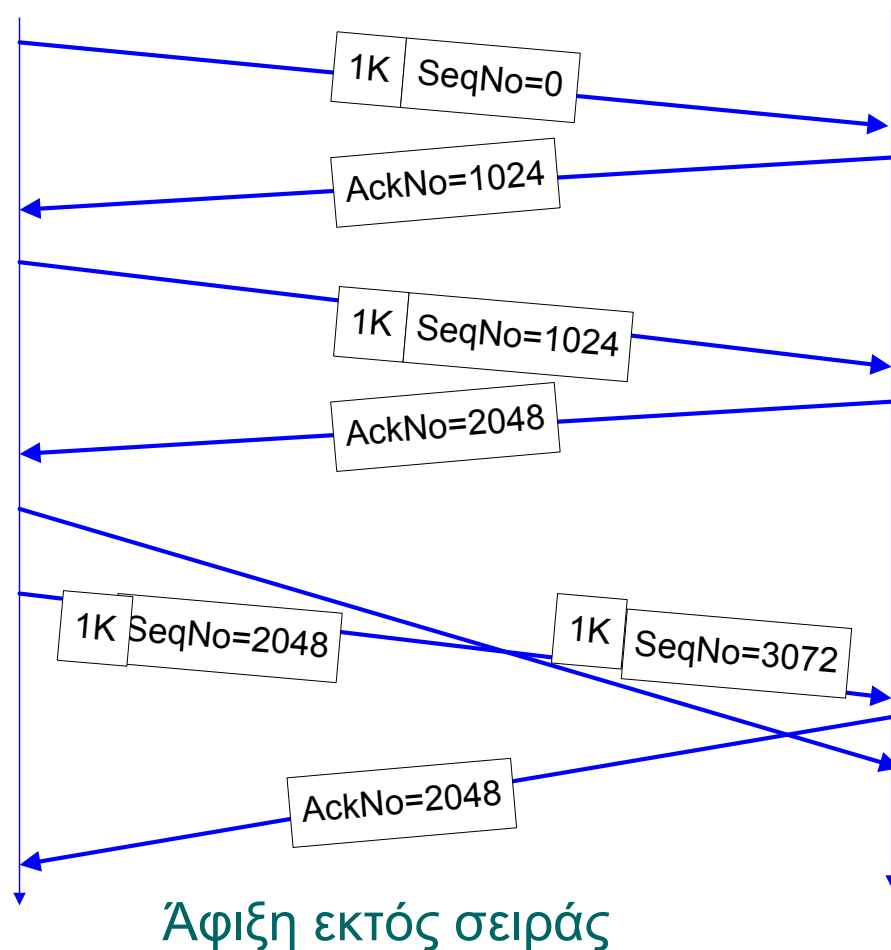


Απώλεια τεμαχίου



# Επαληθεύσεις στο TCP

- Ο δέκτης στέλνει ACK στον αποστολέα
  - Το ACK χρησιμοποιείται για έλεγχο ροής, έλεγχο λαθών και έλεγχο συμφόρησης
- Ο αύξων αριθμός ACK που αποστέλλεται είναι ο επόμενος αναμενόμενος αύξων αριθμός
- Καθυστερημένο ACK: Ο δέκτης TCP συνήθως καθυστερεί την αποστολή του ACK
  - **Γιατί;**
- Το ACK δεν καθυστερεί εάν ληφθεί πακέτο εκτός σειράς
  - **Γιατί;**



# Ταχεία αναμετάδοση (Fast Retransmit)



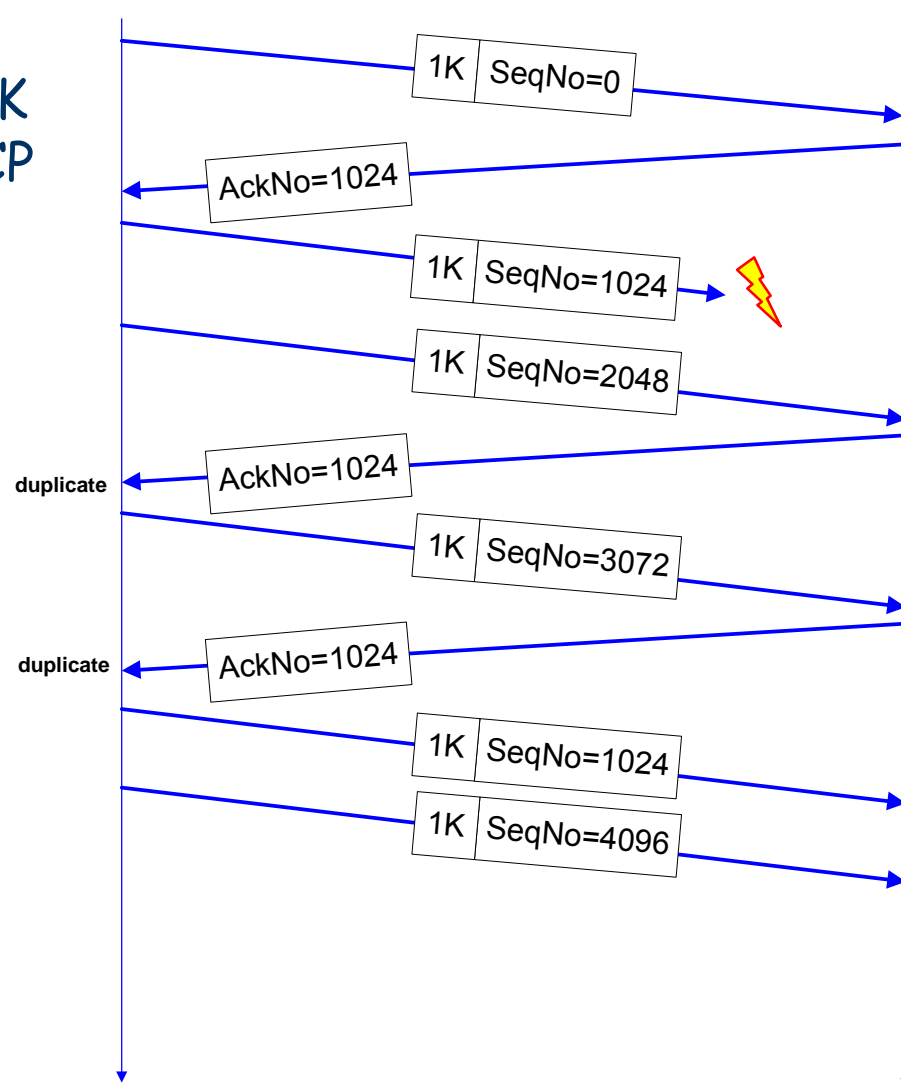
- Μερικές φορές η αναμονή μέχρι να λήξει το χρονόμετρο μπορεί να είναι σχετικά μεγάλη
- Η απώλεια μπορεί να ανιχνευθεί με τη λήψη ταυτόσημων ACK
- Ταχεία αναμετάδοση → πρόωρη αποστολή (πριν τη λήξη του χρονόμετρου)



# Ταχεία αναμετάδοση (Fast Retransmit)

- Εάν υπάρξουν τρία ή περισσότερα ταυτόσημα ACK στη σειρά, ο αποστολέας TCP πιστεύει ότι το τεμάχιο χάθηκε
- Τότε το TCP προχωρά σε αναμετάδοση του τεμαχίου που νομίζει ότι χάθηκε, **χωρίς** να περιμένει την **εκπνοή χρόνου**
- και επανέρχεται στην αργή αρχή

$$ssthresh = cwnd/2$$
$$cwnd = 1$$



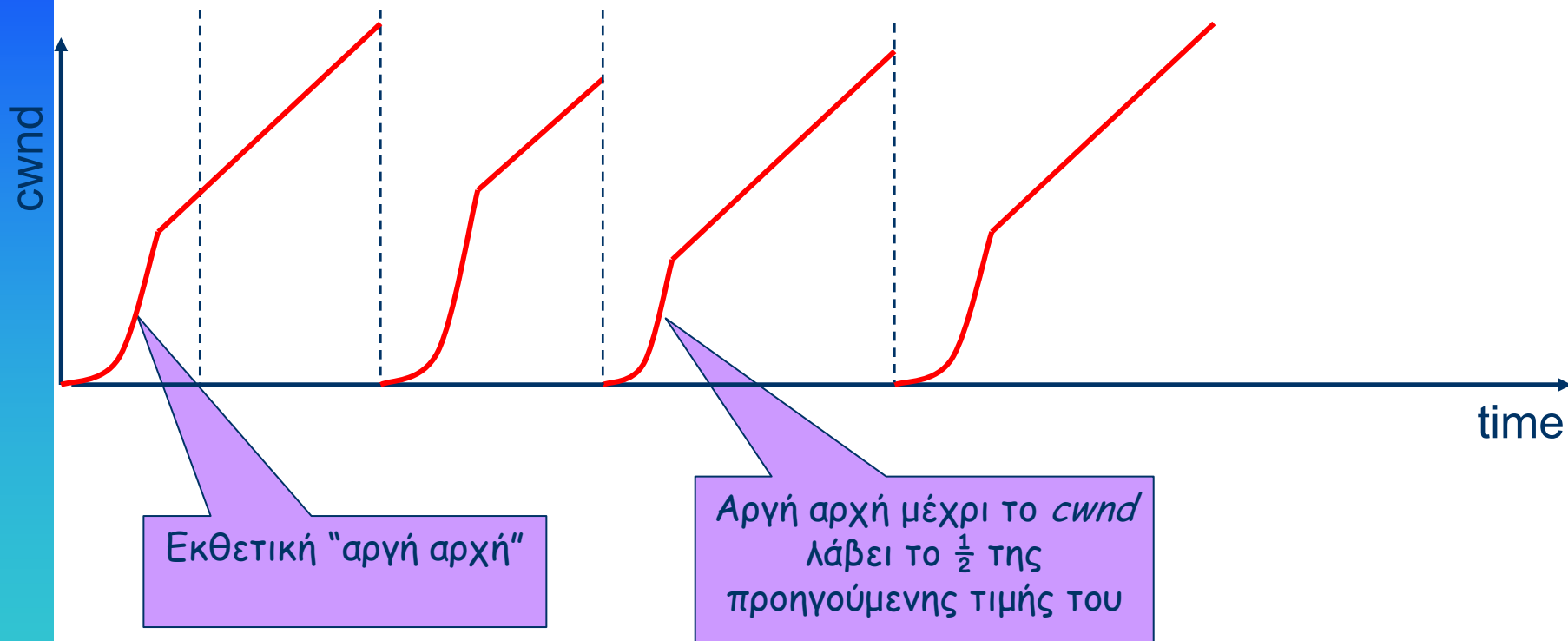
# TCP Tahoe



- Κάθε φορά που συμβαίνει "απώλεια"
  - $ssthresh = cwnd/2$
  - $cwnd = 1$
  - και επανεκκίνηση της αργής αρχής
- Πρόβλημα με το Tahoe: η ανάκαμψη καθυστερεί λόγω της αργής αρχής
  - Επίσης η ταχεία αναμετάδοση είναι απίθανο να ενεργοποιηθεί εάν δεν υπάρχουν πολλά προς μετάδοση πακέτα (δεν είναι χρήσιμη για μικρές μεταφορές, π.χ. σελίδες web)



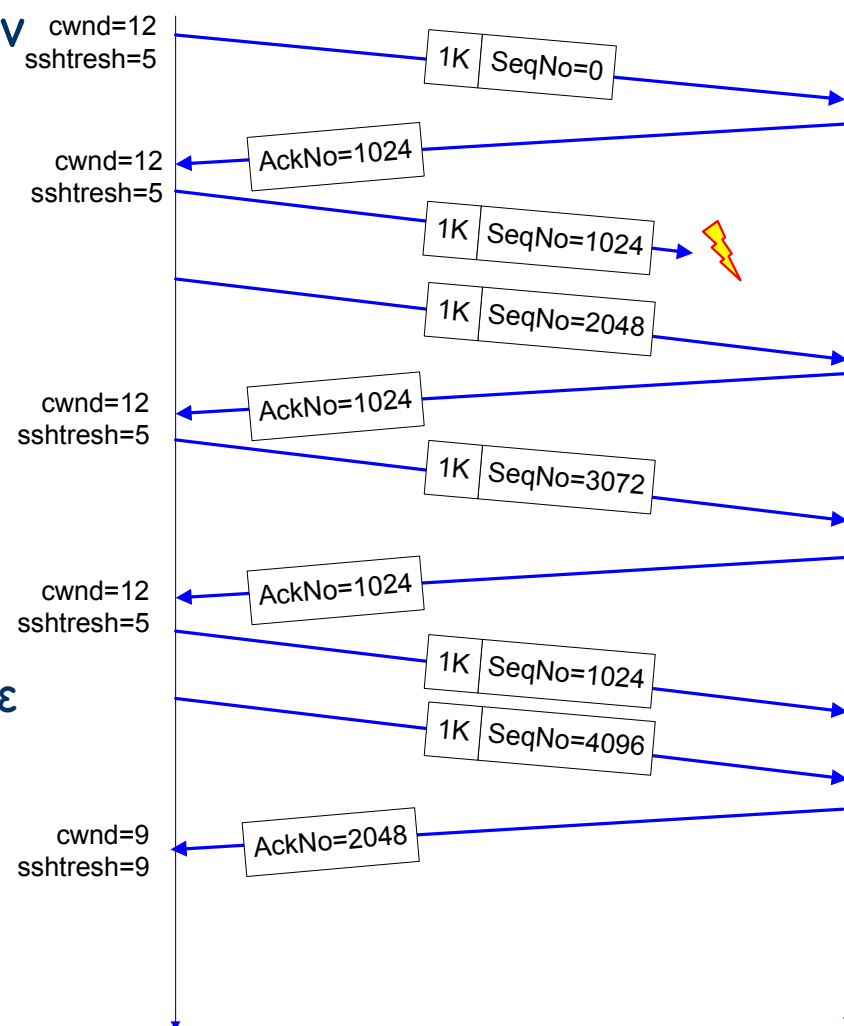
# Δυναμική συμπεριφορά TCP





# Ταχεία ανάκαμψη (Fast Recovery)

- Η ταχεία ανάκαμψη αποφεύγει την αργή αρχή μετά από την ταχεία αναμετάδοση
  - **Διαίσθηση:** Ταυτόσημα ACK δείχνουν ότι τα δεδομένα προωθούνται
- Μετά τρία ταυτόσημα ACK
  - Αναμετάδοση του "χαμένου πακέτου"
  - $ssthresh = cwnd/2$
  - $cwnd = ssthresh + 3$
  - Αποφυγή συμφόρησης
  - Αύξηση του  $cwnd$  κατά 1 για κάθε επιπλέον ταυτόσημο ACK
- Όταν φτάσει ACK για "νέα δεδομένα" (εδώ: AckNo=2028)
  - $cwnd = ssthresh$
  - Αποφυγή συμφόρησης



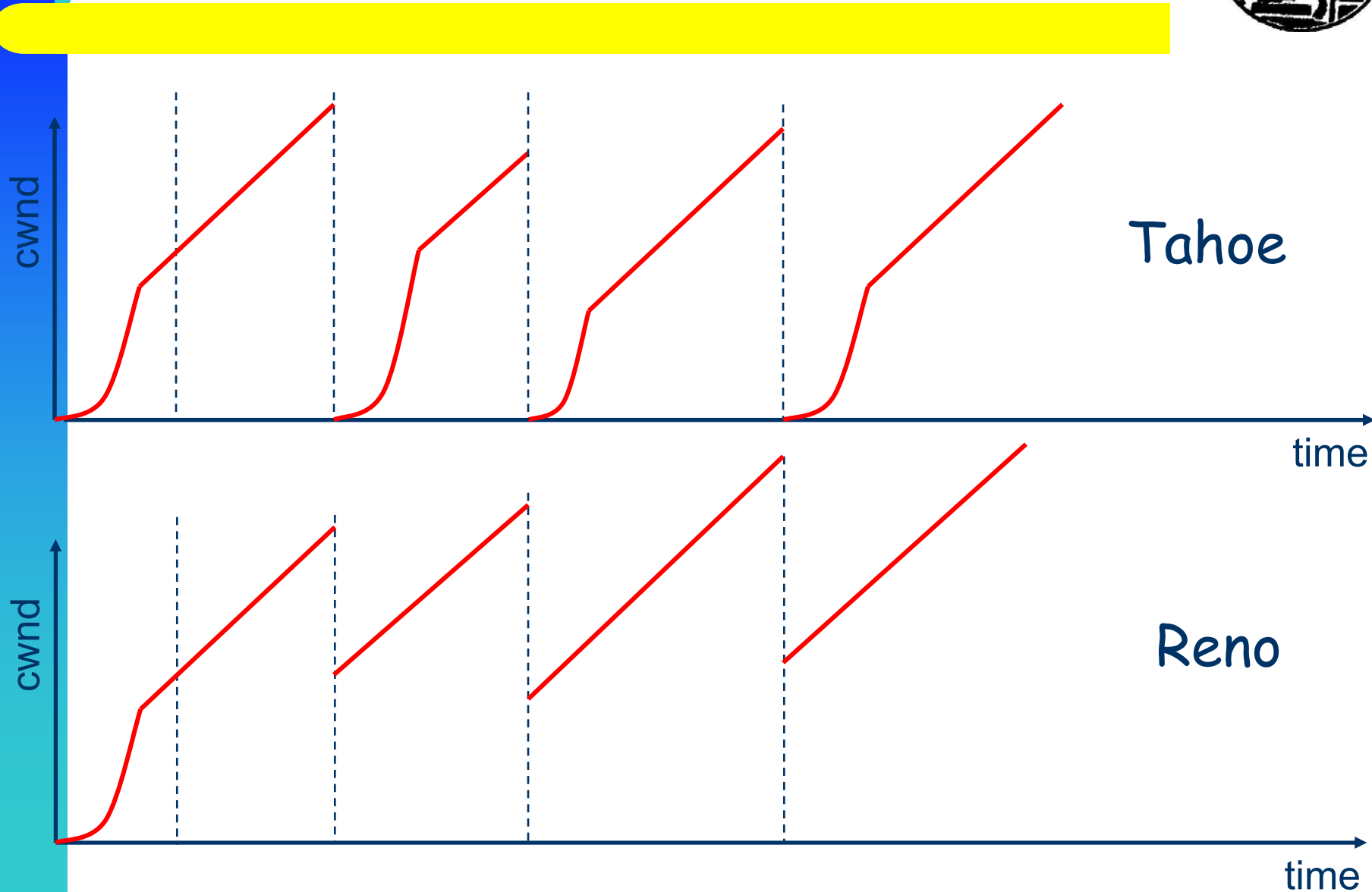
# TCP Reno



- Ταυτόσημα ACK:
  - Ταχεία αναμετάδοση (Fast retransmit)
  - Ταχεία ανάκαμψη (Fast recovery)
    - Η ταχεία ανάκαμψη αποφεύγει την αργή αρχή
- Εκπνοή χρόνου:
  - Αναμετάδοση
  - Αργή Αρχή
- Το TCP Reno **βελτιώνει** το TCP Tahoe όταν έχουμε **μία** απώλεια πακέτου μέσα στο διάστημα μετάδοσης μετ' επιστροφής (round-trip time)

# Σύγκριση TCP Tahoe και TCP Reno

(για απώλειες ενός τεμαχίου)



# Το νέο TCP Reno



- Όταν έχουμε πολλαπλές απώλειες πακέτων, το TCP Reno αντιμετωπίζει πρόβλημα
- Μερικό ACK:
  - Συμβαίνει όταν χάνονται πολλά πακέτα
  - Το μερικό ACK επιβεβαιώνει μερικά, αλλά όχι όλα, τα πακέτα που εκκρεμούν κατά την αρχή της ταχείας ανάκαμψης, οπότε ο αποστολέας θα βγει από την φάση της ταχείας ανάκαμψης  
→ Ο αποστολέας θα περιμένει να συμβεί εκπνοή χρόνου
- Το νέο Reno:
  - Το μερικό ACK δεν βγάζει τον αποστολέα από την ταχεία ανάκαμψη
  - Το μερικό ACK προκαλεί την αναμετάδοση του τεμαχίου που ακολουθεί το τεμάχιο που επιβεβαιώθηκε
- Το νέο Reno μπορεί να χειριστεί πολλαπλές απώλειες τεμαχίων χωρίς να εισέλθει στην αργή αρχή

# SACK (Selective acknowledgment)



- Το πρόβλημα: Το Reno και το νέο Reno αναμεταδίδουν το πολύ 1 χαμένο πακέτο ανά χρόνο μετάδοσης μετ' επιστροφής (RTT)
- **Επιλεκτική αναμετάδοση:** Ο δέκτης μπορεί να επιβεβαιώσει μη συνεχείς ομάδες δεδομένων (π.χ., SACK (0-1023, 2040-3071))
- **TCP SACK:**
  - Εισέρχεται στην ταχεία ανάκαμψη με τη λήψη 3 ταυτόσημων ACK
  - Ο αποστολέας παρακολουθεί τα SACK και συμπεραίνει το κατά πόσο κάποιο τεμάχιο χάθηκε
  - Ο αποστολέας αναμεταδίδει το επόμενο τεμάχιο από τη λίστα των τεμαχίων που θεωρεί ότι χάθηκαν

# Αποφυγή συμφόρησης στους δρομολογητές



- Rate adaptation:
  - Οι δρομολογητές ειδοποιούν άμεσα τις πηγές για τη συμφόρηση
  - π.χ. DECbit
- Active queue management (AQM) :
  - Οι δρομολογητές ειδοποιούν εμμέσως τις πηγές απορρίπτοντας πακέτα
  - π.χ., Random Early Detection (RED) απορρίπτει πακέτα τυχαία ως συνάρτηση του βαθμού συμφόρησης

# DECbit

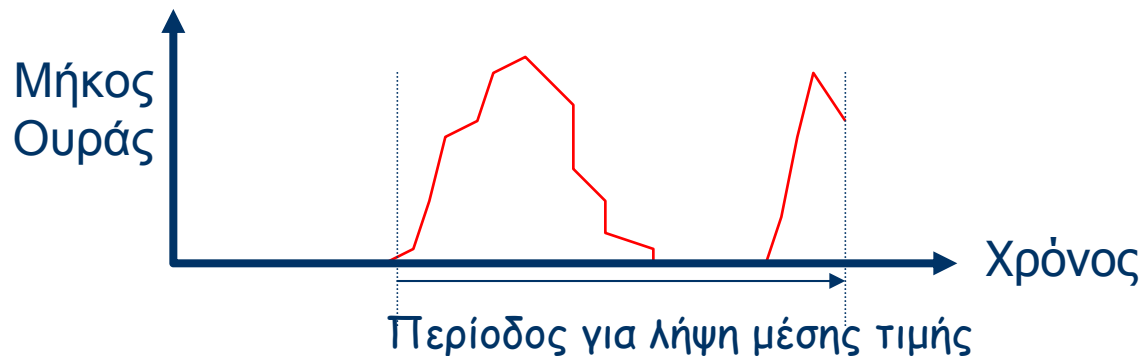


- Κάθε πακέτο έχει ένα bit στην επικεφαλίδα του που αποκαλείται DECbit και χρησιμοποιείται ως δείκτης συμφόρησης
- Εάν κάποιος δρομολογητής της διαδρομής εμφανίζει συμφόρηση, θέτει το DECbit
  - π.χ. εάν η μέση τιμή της ουράς  $\geq 1$  πακέτου
- Για να ειδοποιηθεί η πηγή, ο παραλήπτης αντιγράφει το DECbit στα ACK





- Η πηγή προσαρμόζει τον ρυθμό αποστολής για να αποφύγει τη συμφόρηση
  - μετρά το ποσοστό των DECbit σε κάθε παράθυρο
  - Εάν  $< 50\%$ , αυξάνει τον ρυθμό προσθετικά
  - Εάν  $\geq 50\%$ , μειώνει τον ρυθμό πολλαπλασιαστικά επί  $7/8$



# Πώς χάνονται τα πακέτα;



- Υπερχείλιση των ουρών στους δρομολογητές
- Το πακέτο που βρίσκει την ουρά γεμάτη χάνεται (**Droptail**)
- Προβλήματα με την υπερχείλιση:
  - Μπορεί να απορριφθεί μια ριπή πακέτων της ίδιας ροής
    - Εμποδίζει τη λειτουργία της γρήγορης αναμετάδοσης/ανάκαμψης
  - Μπορεί να δημιουργήσει εκπνοές χρόνου για πολλές ροές ταυτόχρονα
    - Οδηγεί σε συγχρονισμό των αφίξεων: όλες οι ροές TCP θα ξεκινήσουν μαζί και σύντομα θα έχουμε νέα απόρριψη

# Ενεργή διαχείριση ουρών



- Διαχείριση της συμπεριφοράς των δρομολογητών σε σχέση με την απόρριψη πακέτων ώστε να βελτιωθεί η επίδοση του TCP
  - Τυχαία απόρριψη από την ουρά
  - Απόρριψη προτού γεμίσει η ουρά

# Τυχαία απόρριψη



- Έμμεση ειδοποίηση του αποστολέα να μειώσει τον ρυθμό του απορρίπτοντας πακέτα στην τύχη σε περίπτωση συμφόρησης
- Όταν γεμίσει η ουρά, απορρίπτεται στην τύχη ένα πακέτο από την ουρά
  - Λιγότερο πιθανό να απορριφθούν περισσότερα από ένα πακέτα της ίδιας ροής (καλό)
  - Πιο πιθανό να απορριφθεί ένα πακέτο που ανήκει σε μια μεγάλη ροή (καλό)
  - Δύσκολο στην υλοποίηση
  - Μεγάλες απαιτήσεις σε χώρους προσωρινής αποθήκευσης
- Η τυχαία απόρριψη δουλεύει καλά με το TCP

# Πρόωρη τυχαία ανίχνευση (RED)



- Τα πακέτα απορρίπτονται τυχαία προτού γεμίσει η ουρά
  - Η πιθανότητα απόρριψης βασίζεται στο μέγεθος της ουράς
  - Όσο πιο μεγάλη είναι, τόσο υψηλότερη η πιθανότητα απόρριψης
- Ευκολότερη στην υλοποίηση σε σχέση με την τυχαία απόρριψη
  - Δεν απορρίπτονται πακέτα που βρίσκονται ήδη στην ουρά
- Δύσκολη ρύθμιση των παραμέτρων λειτουργίας

# Πρόωρη τυχαία ανίχνευση (RED)



- Λαμβάνεται (γεωμετρικός) κινητός μέσος όρος του μήκους ουράς
  - έτσι ώστε να ανιχνεύεται η μακροχρόνια συμφόρηση
  - αλλά να επιτρέπονται και οι σύντομες ριπές κίνησης

$$AvgLen_{n+1} = (1 - a) \times AvgLen_n + a \times Length_n$$

$$\text{δηλαδή, } AvgLen_{n+1} = \sum_{i=1}^n Length_i(a)(1 - a)^{n-1}$$

# Πρόωρη τυχαία ανίχνευση (RED)



- Η πιθανότητα απόρριψης αυξάνει καθώς αυξάνει το μέσο μήκος της ουράς
  - Εάν το *AvgLen* είναι μικρό, το πακέτο μπαίνει στην ουρά
  - Αλλιώς, εάν το *AvgLen* είναι μεγαλύτερο από κάποιο ελάχιστο κατώφλι, το πακέτο απορρίπτεται με πιθανότητα που αυξάνει γραμμικά
  - Τέλος, εάν το *AvgLen* ξεπεράσει το μέγιστο κατώφλι, τότε το πακέτο απορρίπτεται



# Παράμετροι λειτουργίας

- Ορίζεται ένα ελάχιστο μήκος ουράς  $Th_{min}$  κάτω από το οποίο δεν γίνονται απορρίψεις πακέτων  
 $AvgLen < Th_{min}$
- Ορίζεται ένα μέγιστο μήκος ουράς  $Th_{max}$  πάνω από το οποίο απορρίπτεται κάθε εισερχόμενο πακέτο  
 $AvgLen > Th_{max}$
- Όταν  $Th_{max} > AvgLen > Th_{min}$  ένα εισερχόμενο πακέτο απορρίπτεται με κάποια πιθανότητα  $P$
- Προσοχή: κατά το δυνατόν δεν πρέπει να απορρίπτονται διαδοχικά πακέτα μιας ριπής  
- γιατί;





# Πιθανότητα απόρριψης

- Η πιθανότητα  $P$  αυξάνει γραμμικά ως προς το  $AvgLen$  μέχρι τη μέγιστη τιμή  $P_{max}$

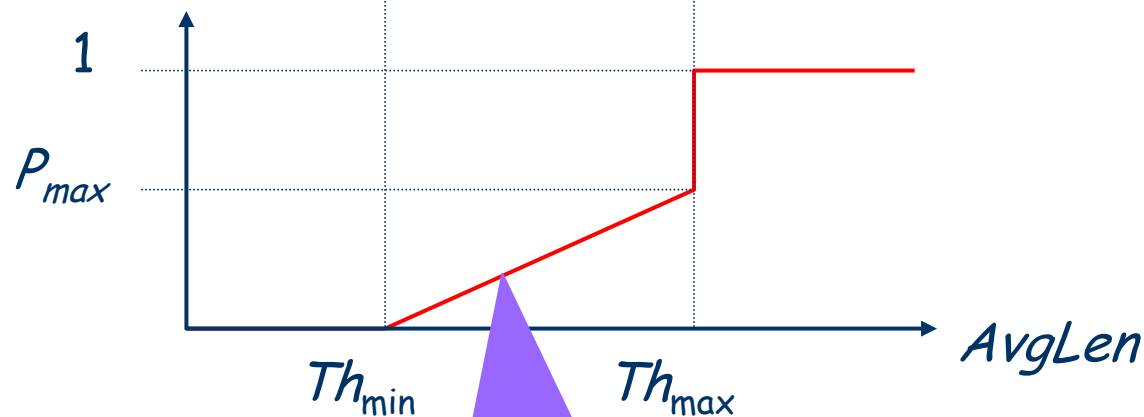
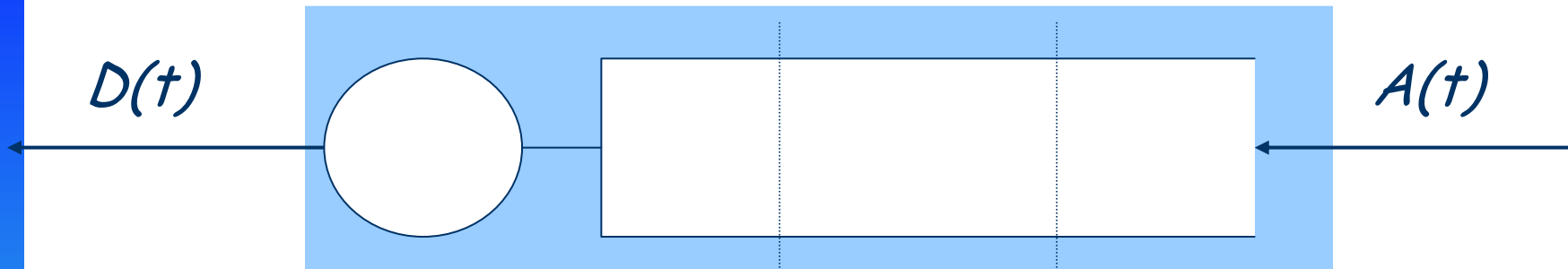
$$\hat{p} = P_{max} \left\{ \frac{AvgLen - Th_{min}}{Th_{max} - Th_{min}} \right\}$$

- Η μεταβλητή  $count$  μετρά πόσες φορές το  $AvgLen$  βρέθηκε στο διάστημα  $Th_{min}$  έως  $Th_{max}$  μετά την τελευταία απόρριψη πακέτου

$$Pr(Drop) = \frac{\hat{p}}{1 - count \times \hat{p}}$$

- Δηλαδή, οι απορρίψεις απέχουν ώστε να μειωθεί η πιθανότητα επανεισόδου στην αργή αρχή

# Ιδιότητες της πιθανότητας απόρριψης



$$\hat{p} = P_{\max} \left\{ \frac{AvgLen - Th_{\min}}{Th_{\max} - Th_{\min}} \right\}$$

$$Pr(\text{Drop}) = \frac{\hat{p}}{1 - count \times \hat{p}}$$

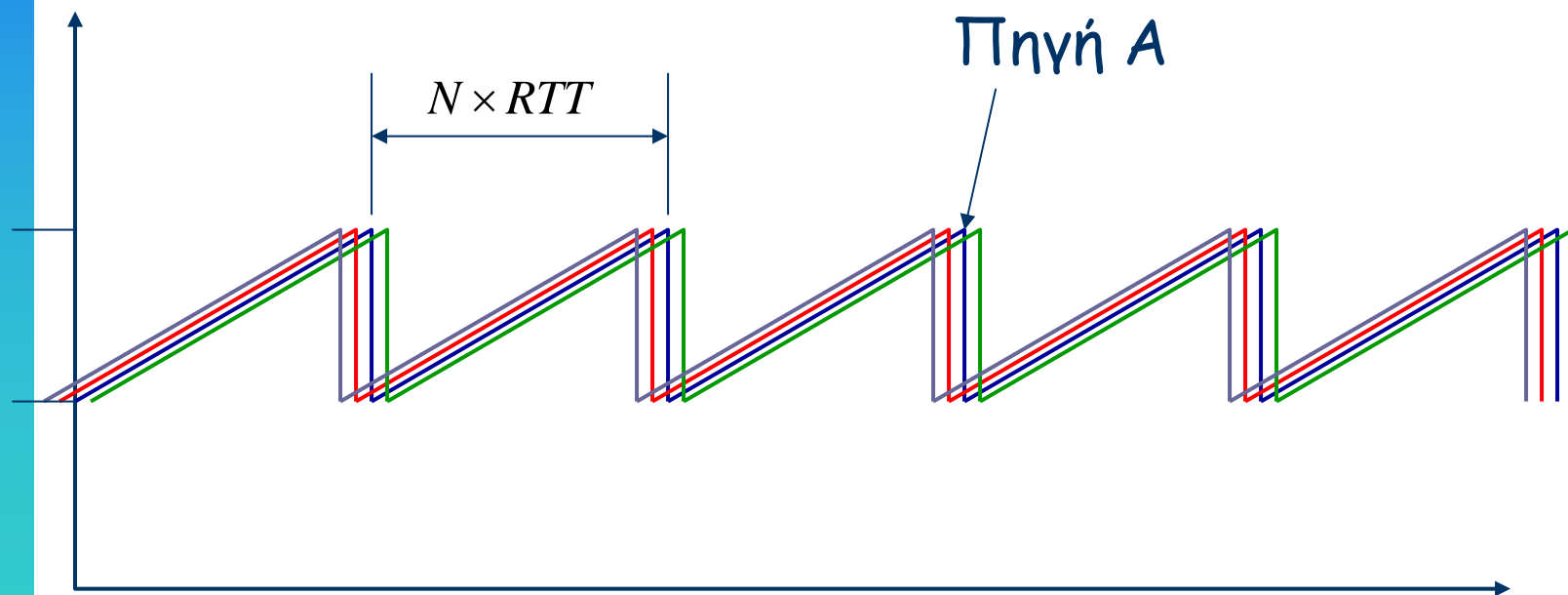
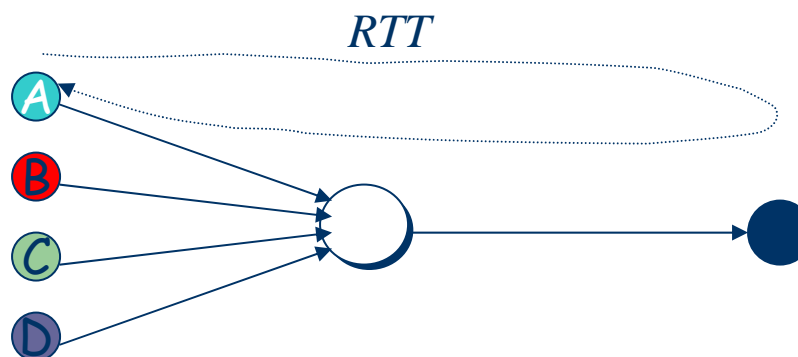
# Ιδιότητες RED



- Απορρίπτει πακέτα προτού γεμίσει η ουρά, ελπίζοντας ότι θα μειωθεί ο ρυθμός μερικών ροών
- Οι απορρίψεις πακέτων για κάθε ροή είναι περίπου ανάλογες με τον ρυθμό της
- Οι απορρίψεις απέχουν στον χρόνο
- Επειδή χρησιμοποιεί μέση τιμή του μήκους ουράς, η RED είναι ανεκτική στις ριπές
- Οι τυχαίες απορρίψεις (ελπίζεται ότι) **αποσυγχρονίζουν** τις πηγές TCP

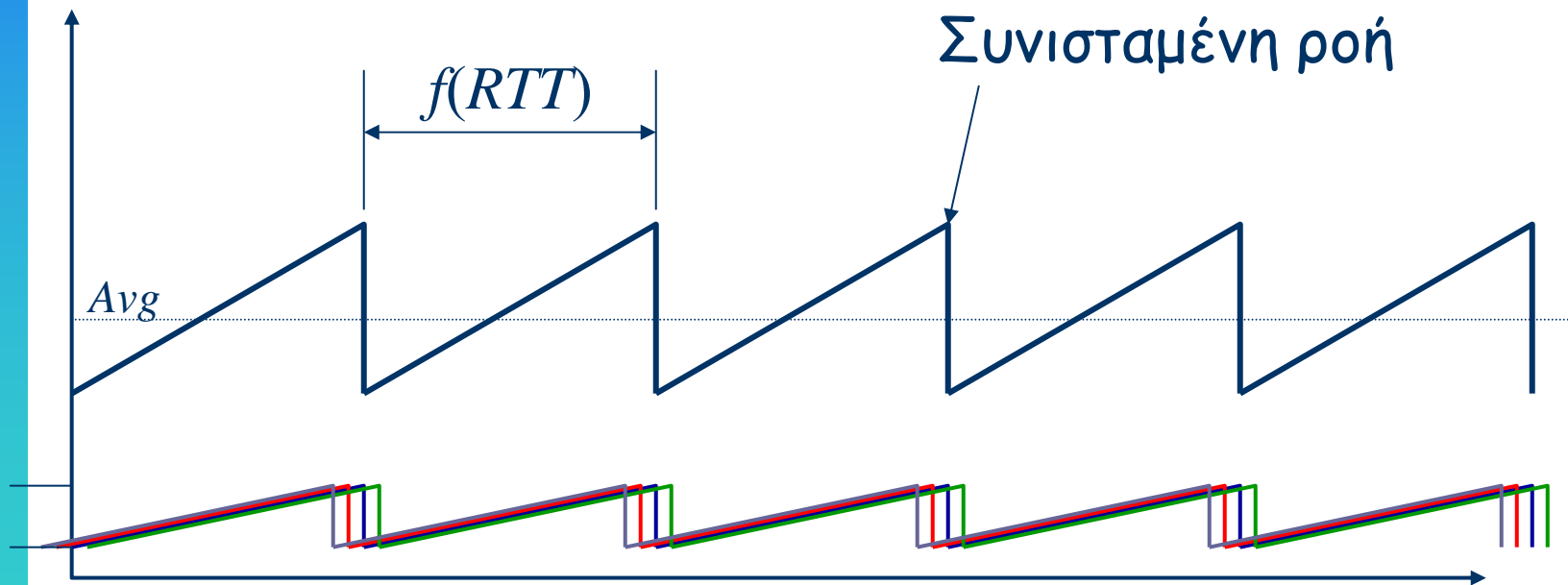
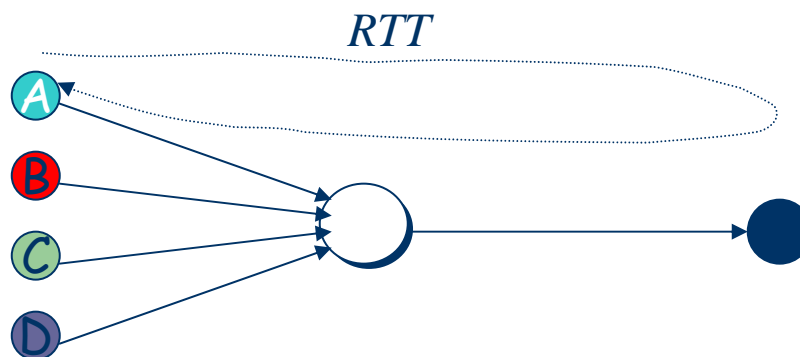


# Συγχρονισμός πηγών



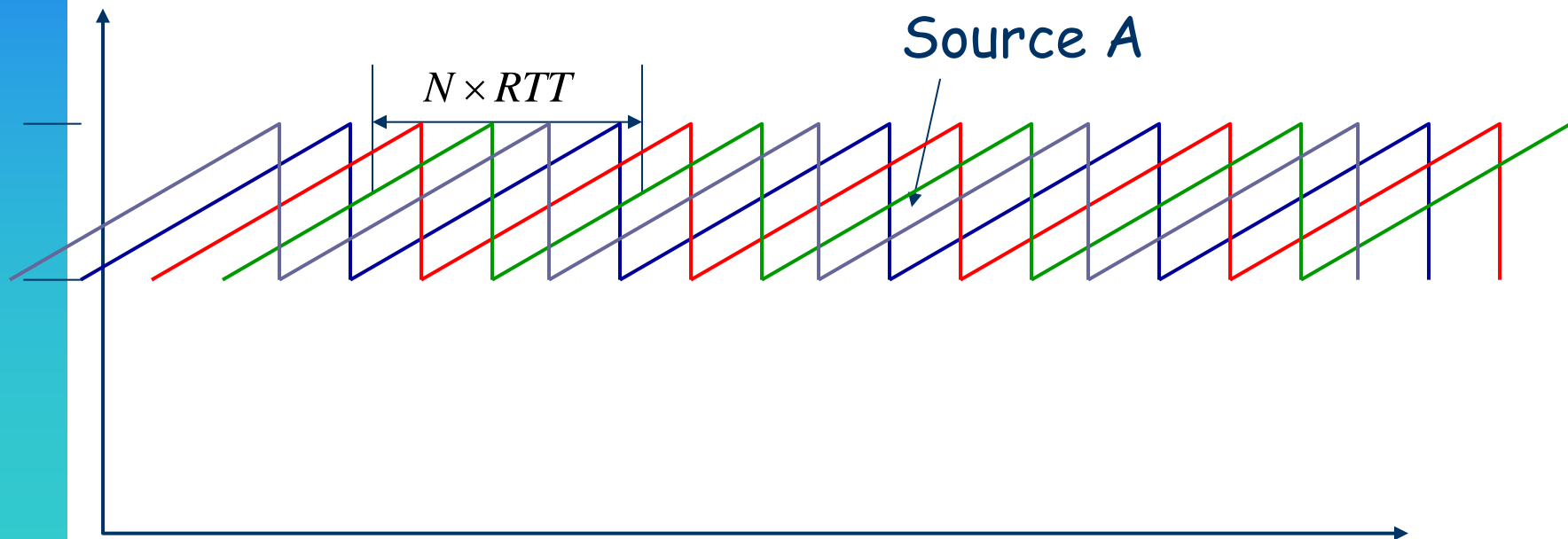
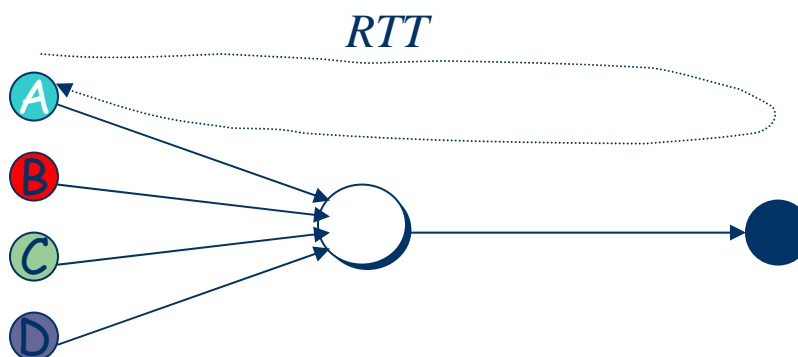


# Συγχρονισμός πηγών



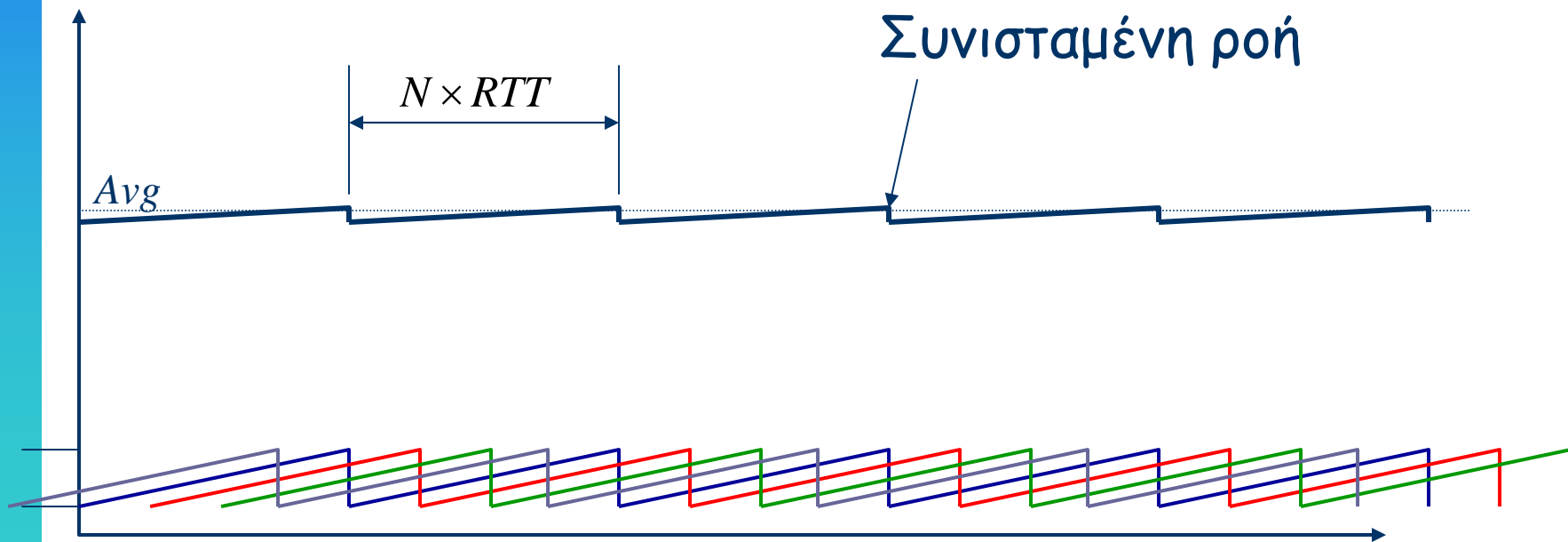
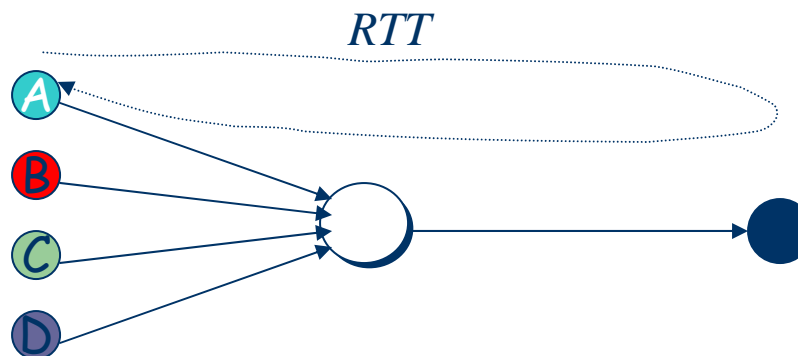


# Αποσυγχρονισμός πηγών





# Αποσυγχρονισμός πηγών





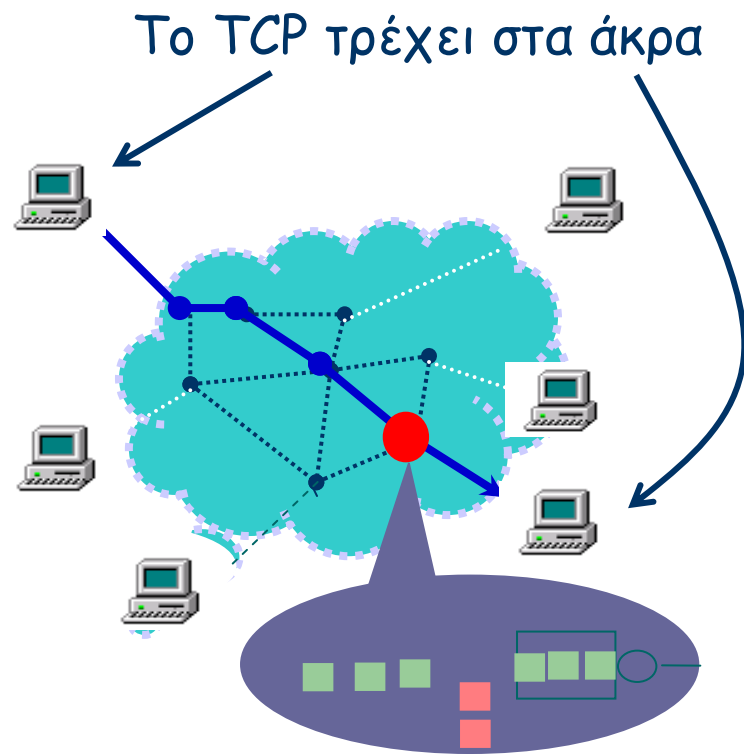
## Επίδοση TCP





# Συμπεριφορά του TCP

- Αποφυγή συμφόρησης:
  - μείωση του ρυθμού αποστολής μόλις ανιχνευθεί απώλεια, αύξηση όσο δεν υπάρχουν απώλειες
- δρομολογητές
  - απορρίπτουν ή σηματοδοτούν τα πακέτα όταν υπάρχει συμφόρηση
- πώς μοντελοποιούμε αυτή τη συμπεριφορά μεταξύ ακραίων συστημάτων (TCP) και δρομολογητών;
  - ποσοτικοποίηση



Ο δρομολογητής απορρίπτει πακέτα όταν υπάρχει συμφόρηση

# Διέλευση (throughput) στο TCP

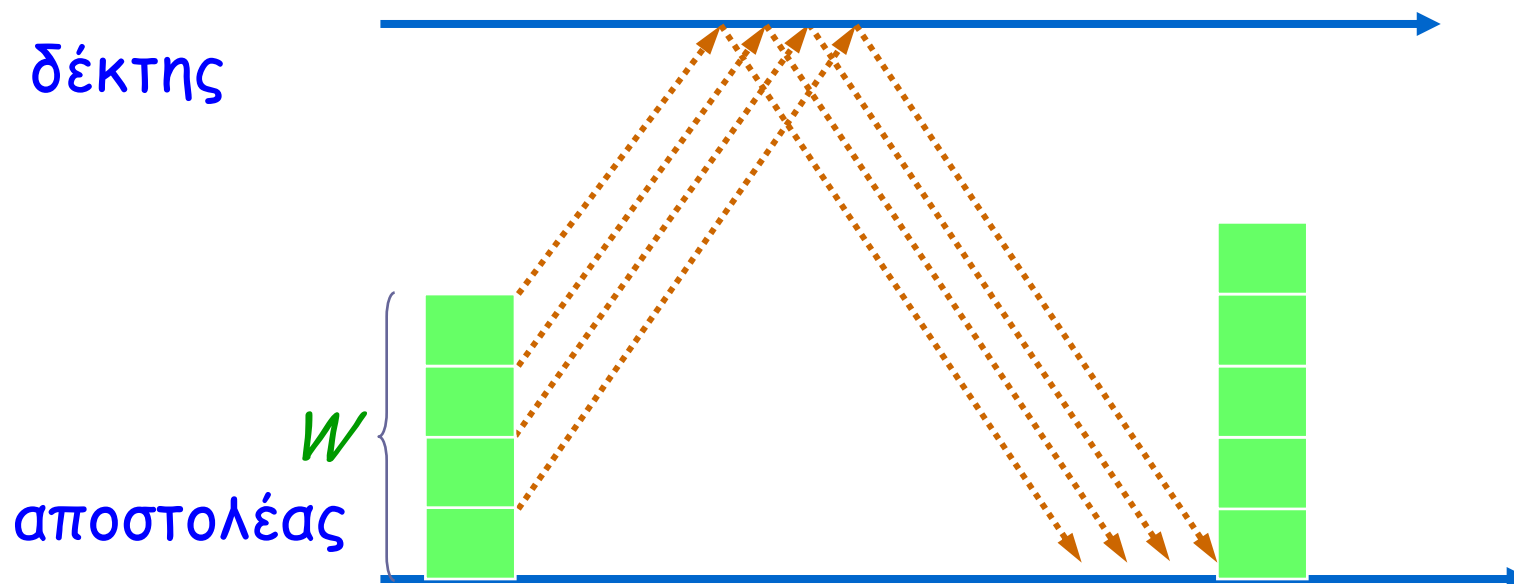


- Ποια είναι η διέλευση (μέσος ρυθμός μετάδοσης) ως συνάρτηση του μέγεθος παραθύρου και του RTT?
- Με απλοποιητικές παραδοχές
  - Μια απώλεια ανά παράθυρο
  - Αγνοώντας την αργή αρχή (μακροχρόνια ροή)
  - Χωρίς εκπνοές
  - κλπ
- Διέλευση = 
$$\frac{1.22 \cdot MSS}{RTT \sqrt{p_{loss}}}$$
- όπου  $p_{loss}$  = η πιθανότητα απώλειας



## Συμπεριφορά του TCP

- Παράθυρο συμφόρησης ( $W$ )
  - μέχρι  $W$  τεμάχια στο δίκτυο
  - κάθε ACK επιτρέπει την αποστολή άλλου ένα τεμαχίου
  - συσσωρευτικά ACK
- Αύξηση του παραθύρου κατά ένα για κάθε RTT
$$W \leftarrow W + 1 / W \text{ ανά ACK}$$
$$\Rightarrow W \leftarrow W + 1 \text{ ανά RTT}$$
- Διερεύνηση του διαθέσιμου εύρους ζώνης δικτύου

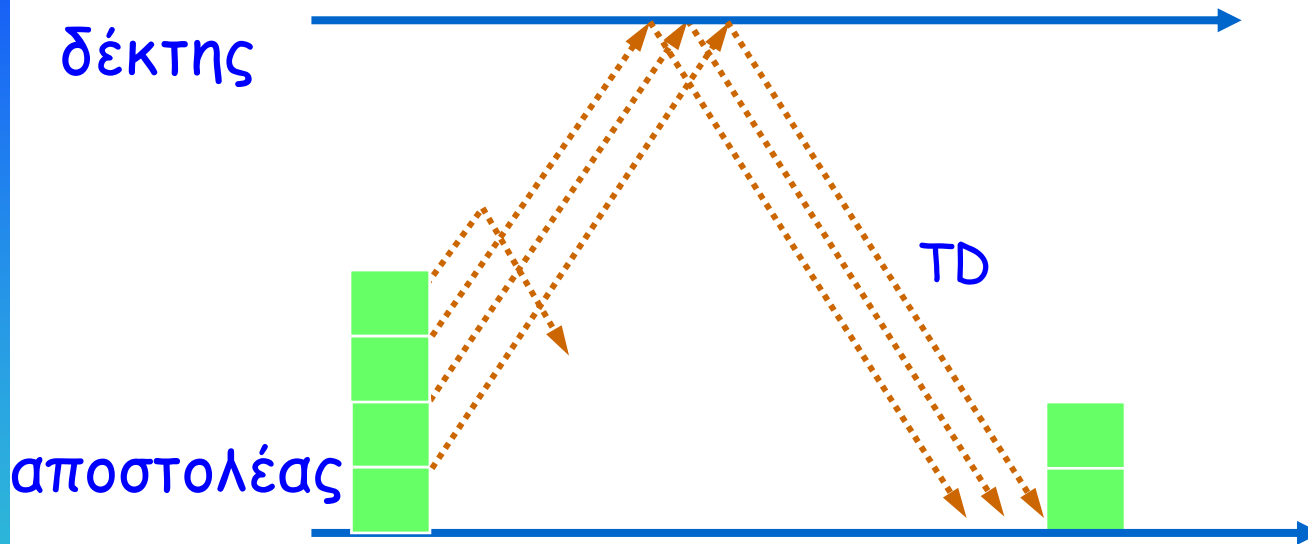




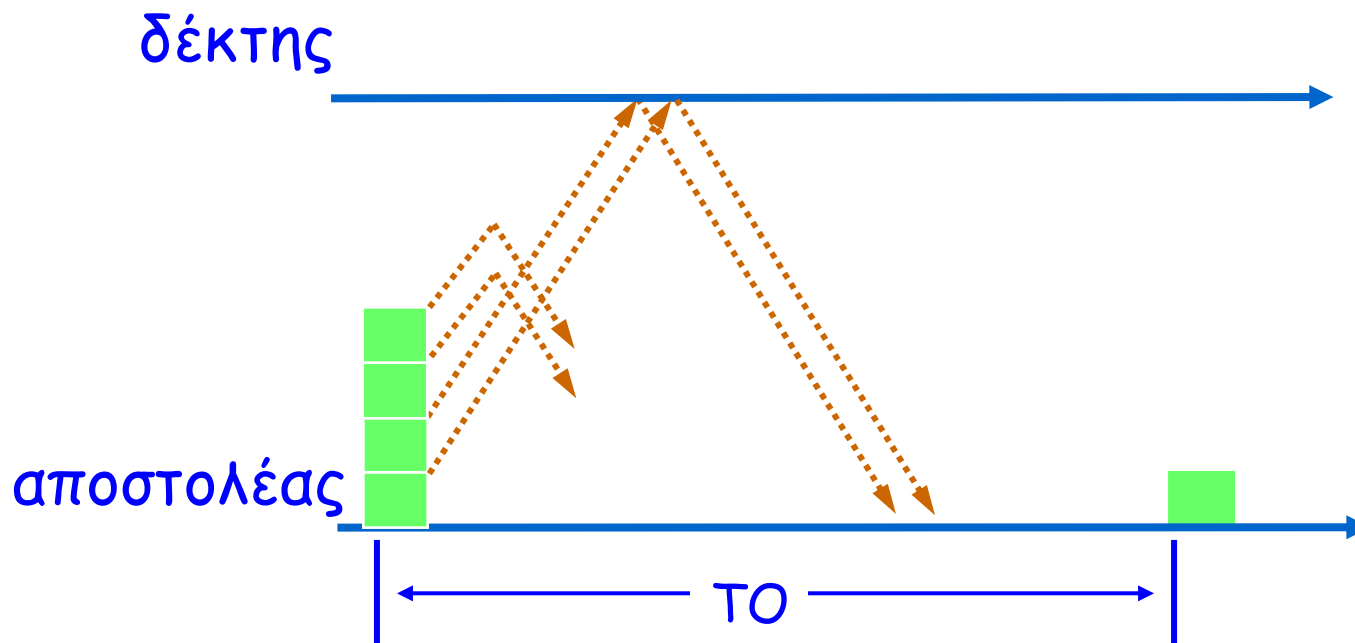
## Συμπεριφορά του TCP

- Η απώλεια δείχνει συμφόρηση
- Μείωση του παραθύρου στο μισό με την ανίχνευση απώλειας (τριπλό ACK)
- $W \leftarrow W/2$
- Εκπνοή χρόνου λόγω έλλειψης ACK, μείωση του παραθύρου στο ένα
- $W \leftarrow 1$
- Τα διαδοχικά διαστήματα χρόνων εκπνοής μεγαλώνουν εκθετικά μέχρι έξι φορές

# Απώλεια

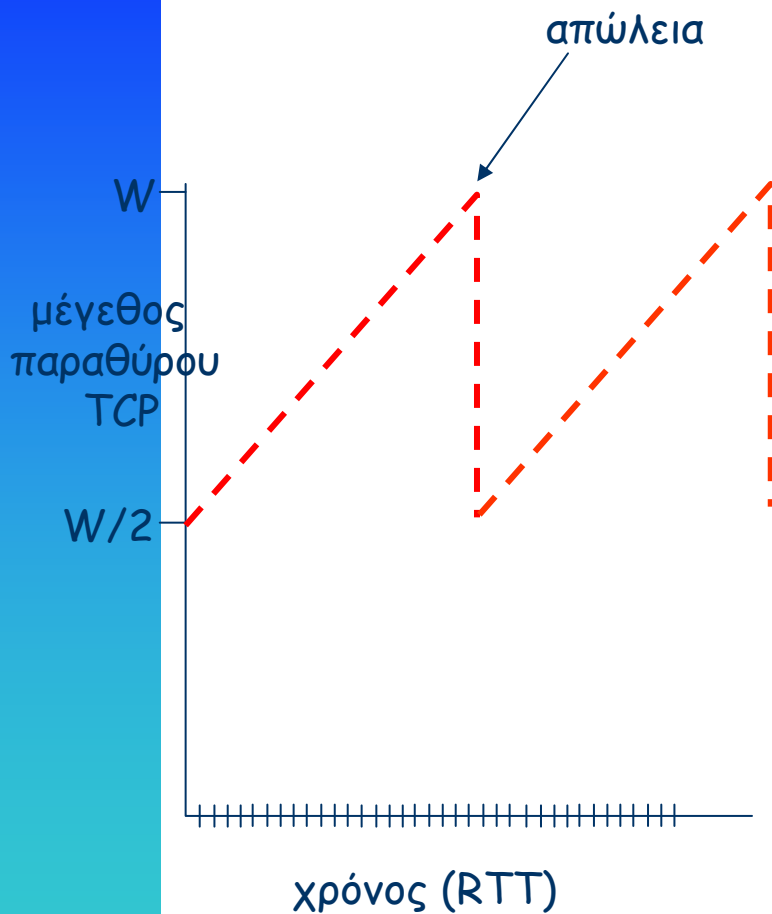


# Εκπνοή





# Σχέση διέλευσης/απωλειών στο TCP



## Εξιδανικευμένο μοντέλο:

- Το  $W$  είναι το μέγιστο ανεκτό παράθυρο (μετά έχουμε απώλεια)
- Το παράθυρο TCP αρχίζει από το  $W/2$ , μετά αυξάνει στο  $W$ , μετά μειώνεται στο μισό, μετά αυξάνει στο  $W$ , μετά μειώνεται ...
- Με κάθε RTT αποστέλλεται ένα παράθυρο τεμαχίων
- *ζητείται:* η διέλευση ως συνάρτηση της πιθανότητας απώλειας και του RTT

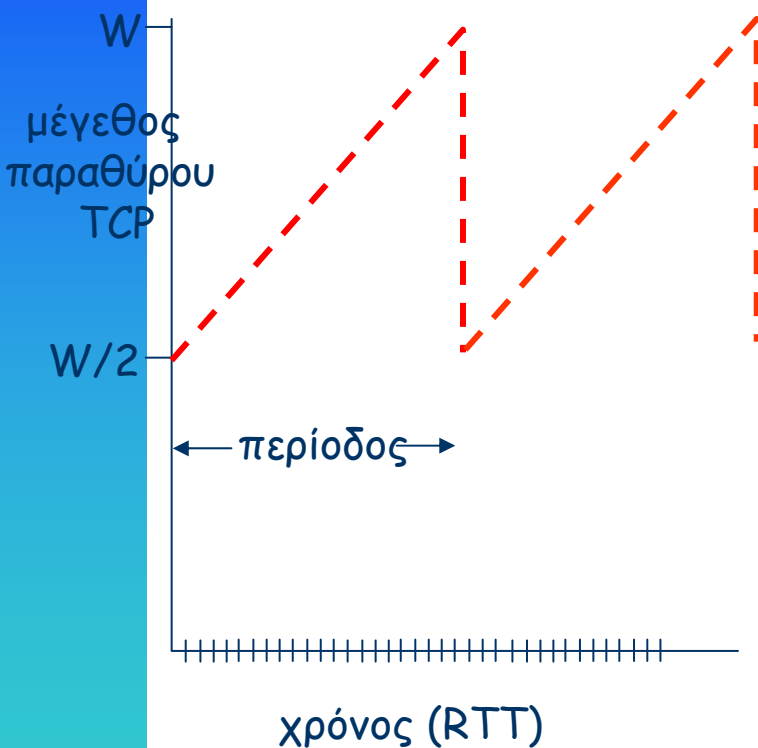




# Σχέση διέλευσης/απωλειών στο TCP

αριθμός τεμαχίων ανά “περίοδο” =

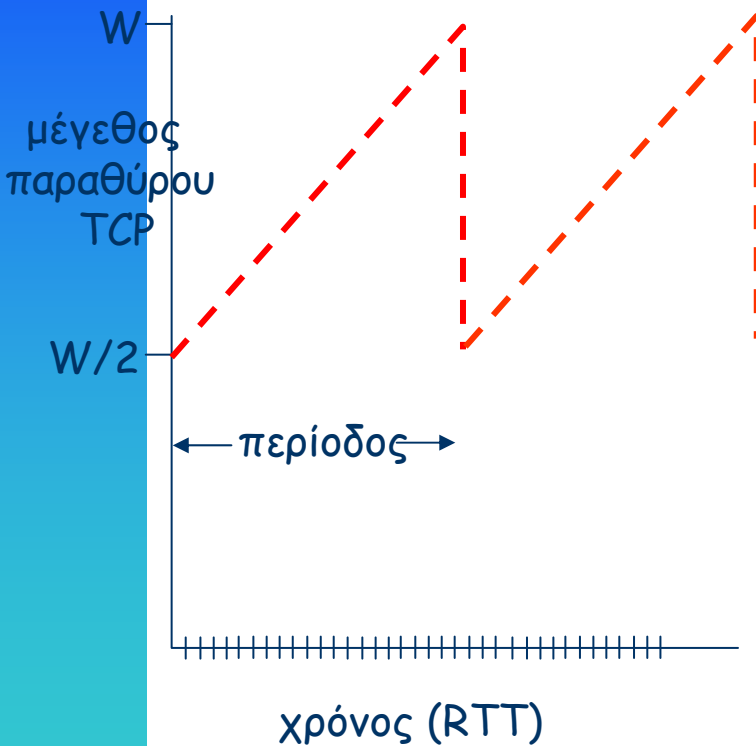
$$\begin{aligned} \frac{W}{2} + \left(\frac{W}{2} + 1\right) + \dots + W &= \sum_{n=0}^{W/2} \left(\frac{W}{2} + n\right) \\ &= \left(\frac{W}{2} + 1\right) \frac{W}{2} + \sum_{n=0}^{W/2} n \\ &= \left(\frac{W}{2} + 1\right) \frac{W}{2} + \frac{W/2(W/2 + 1)}{2} \\ &= \frac{3}{8} W^2 + \frac{3}{4} W \\ &\approx \frac{3}{8} W^2 \end{aligned}$$





# Σχέση διέλευσης/απωλειών στο TCP

αριθμός τεμαχίων ανά “περίοδο”  $\approx \frac{3}{8} W^2$



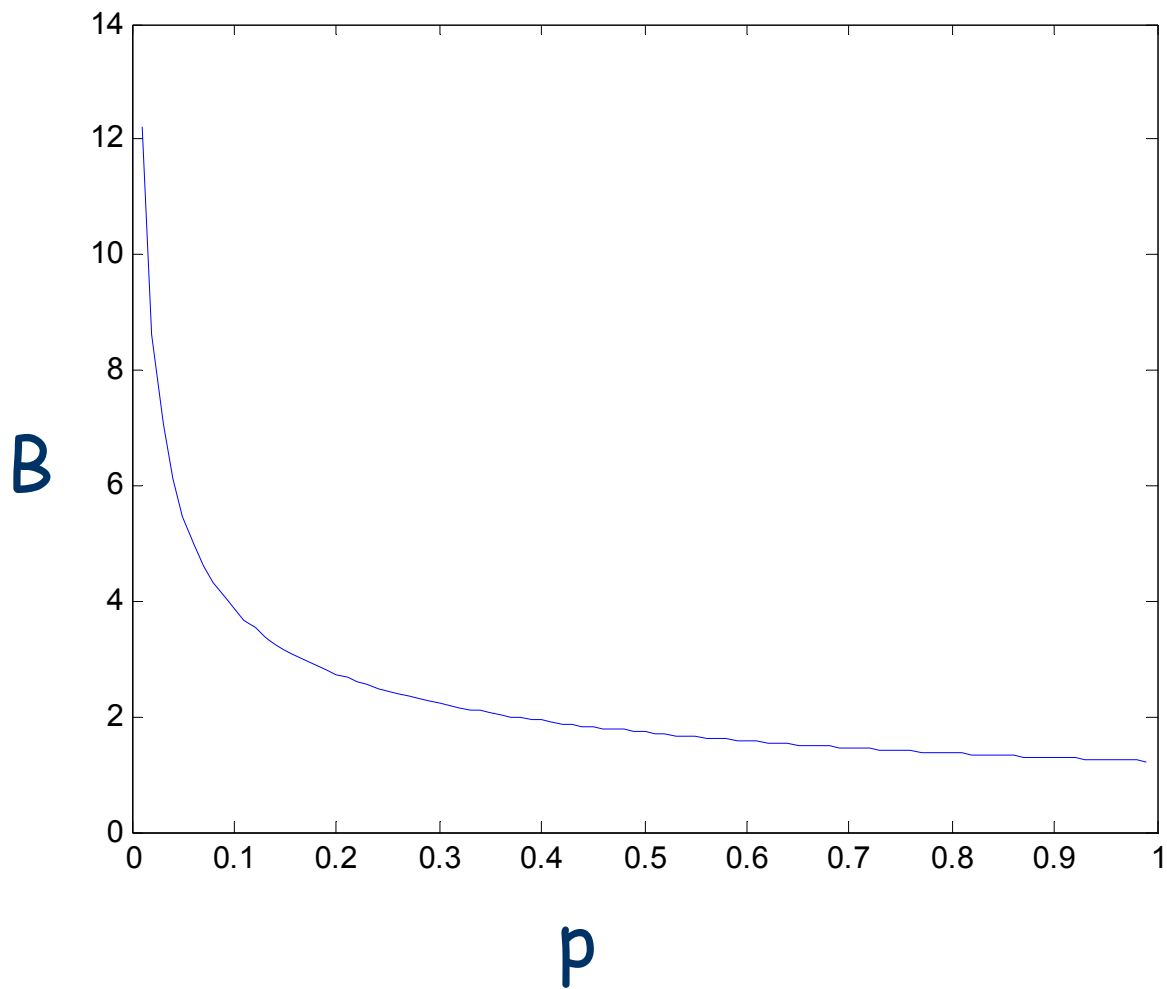
1 απώλεια ανά “περίοδο” σημαίνει:

$$p_{\text{loss}} \approx \frac{8}{3W^2} \quad \text{ή} \quad W = \sqrt{\frac{8}{3p_{\text{loss}}}}$$

$$B = \text{διέλευση} = \frac{3}{4} W \frac{\text{τεμάχια}}{\text{RTT}}$$

$$B = \text{διέλευση} = \frac{1.22}{\sqrt{p_{\text{loss}}}} \frac{\text{τεμάχια}}{\text{RTT}}$$

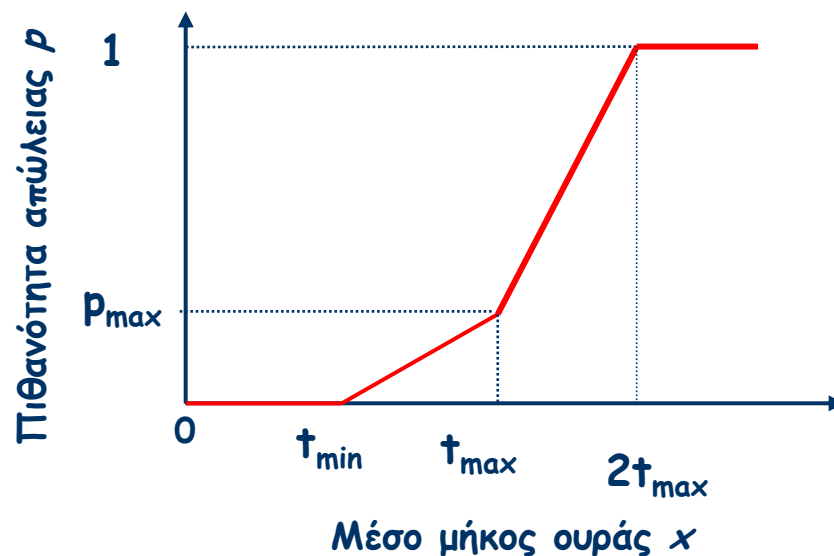
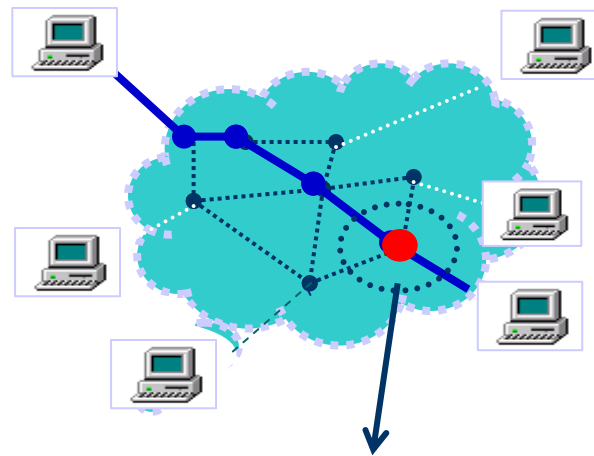
Σημ.: Η σχέση για το B δυνατό να επεκταθεί ώστε να περιλάβει εκπνοές και την αργή αρχή





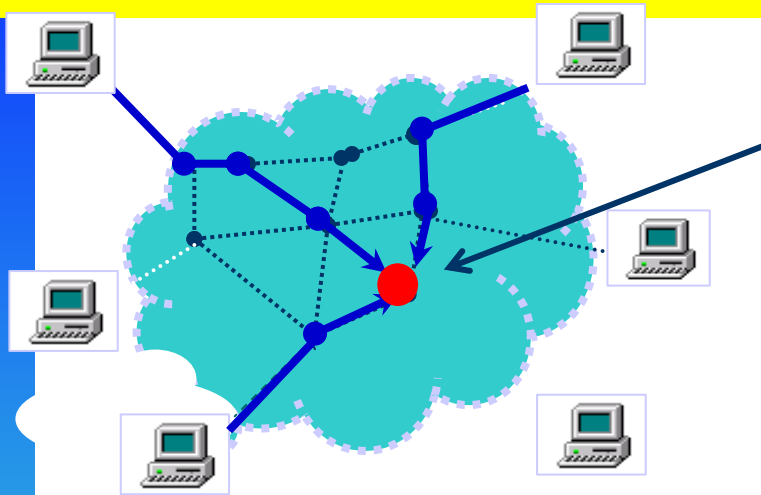
# Διαχείριση ουρών RED

- Η απώλεια είναι συνάρτηση του μέσου μήκους ουράς  $\rightarrow p = p(x)$





# Συμπεριφορά στο σημείο συμφόρησης



Ο δρομολογητής που παρουσιάζει συμφόρηση:

- Χρησιμοποιεί πλήρως τη διαθέσιμη χωρητικότητα
- Όλες οι ροές βλέπουν την ίδια πιθανότητα απώλειας
- Έχουν την ίδια διέλευση?

$$\sum_i B_i(p, RTT_i) = C$$

$C$  = χωρητικότητα δρομολογητή

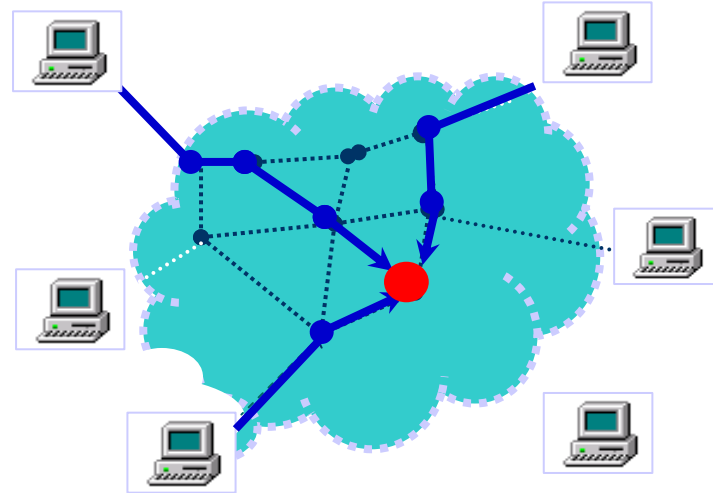
$B_i$  = διέλευση ροής  $i$

# Μοναδικό σημείο συμφόρησης, άπειρης διάρκειας ροές



- $N$  ροές TCP άπειρης διάρκειας

- Καθυστέρηση διάδοσης  $A_i$ ,  $i = 1, \dots, N$
- διέλευση  $B_i(p, RTT_i)$



- Ένας δρομολογητής με συμφόρηση

- Ουρές RED
  - μέσο μήκος ουράς  $x$ ; πιθανότητα απώλειας  $p(x)$

- ζητείται

- $B_i$ : διέλευση ανά ροή TCP,
- παράμετροι δρομολογητή: μέσο μήκος ουράς  $x$ ; πιθανότητα απώλειας  $p(x)$



## Μοντέλο και λύση

- Μοντέλο

$$p = p(x)$$

$$RTT_i = A_i + x / C$$

$$\sum_j B_j(p, RTT_j) = C \quad \text{για } j=1, \dots, N$$

- λύση ως προς  $x$        $\sum_j B_j(x) = C$  για  $j=1, \dots, N$
- Μοναδική λύση αφού το  $B$  είναι μονοτονική και συνεχής συνάρτηση του  $x$
- Από το  $x$  λαμβάνουμε  $RTT_j$  και  $p$