



# ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ

Έλεγχος και Αποφυγή  
Συμφόρησης στο TCP

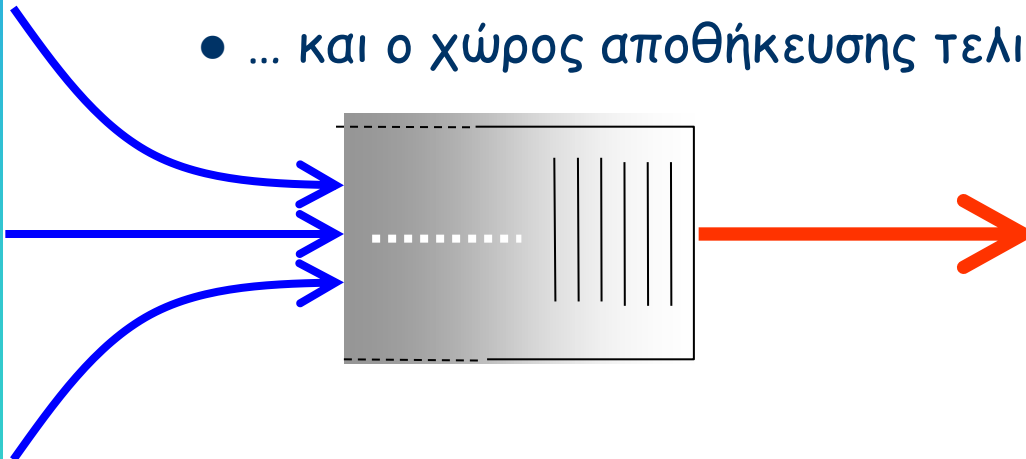


# Περί συμφόρησης



# Η συμφόρηση είναι αναπόφευκτη

- Συμβαίνει σε διαφορετικές κλίμακες - από δύο πακέτα μέχρι πολλούς χρήστες
  - Δύο πακέτα φτάνουν την ίδια στιγμή
    - Ο κόμβος μπορεί να μεταδώσει μόνο το ένα
    - ... και είτε αποθηκεύει είτε απορρίπτει το άλλο
  - Εάν πολλά πακέτα φτάσουν σε ένα μικρό χρονικό διάστημα
    - Ο κόμβος δεν προλαβαίνει να εξυπηρετήσει την κίνηση
    - ... και ο χώρος αποθήκευσης τελικά θα **υπερχειλίσσει**



# Συμφόρηση



- 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?



# Κατάρρευση λόγω συμφόρησης

- Ορισμός: Αύξηση του φορτίου οδηγεί σε μείωση της διέλευσης (των πακέτων που παραδίδονται)
- Που οφείλεται;
  - Στα πακέτα που δεν παραδίδονται
    - Τα πακέτα έχουν καταναλώσει πόρους του δικτύου, ενώ αργότερα απορρίπτονται από το δίκτυο
  - Σε αναμεταδόσεις πακέτων που δεν έχουν ακόμη παραδοθεί (καθυστερούν)
    - Αυξάνουν περισσότερο το φορτίο
    - Σαν να ρίχνεις λάδι στη φωτιά
- Στο μέσο της δεκαετίας του 1980 το διαδίκτυο ακινητοποιούταν συχνά λόγω συμφόρησης μέχρι που εφευρέθηκε ο έλεγχος συμφόρησης στο TCP



# Φορτίο προς καθυστέρηση

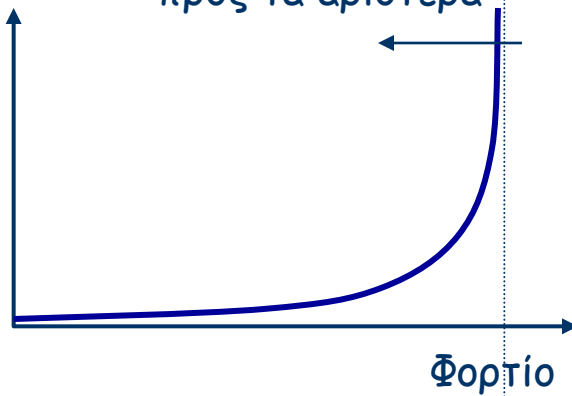
Συνήθης συμπεριφορά συστήματος αναμονής με τυχαίες αφίξεις:

Ένα απλό μέτρο της επίδοσης του δικτύου:

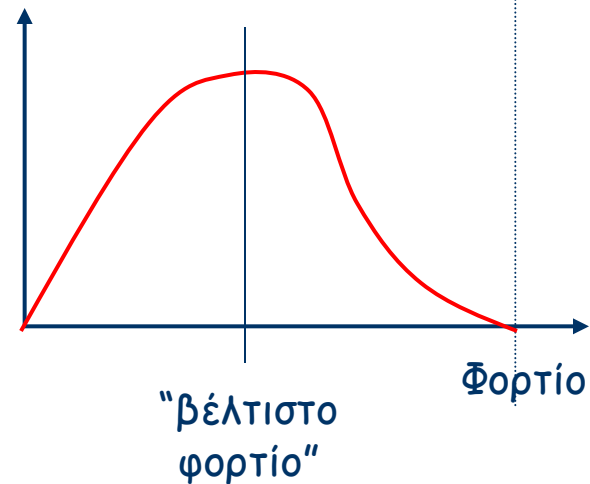
φορτίο/καθυστέρηση

Εκρηκτικότητα κίνησης μετακινεί την ασύμπτωτο προς τα αριστερά

Μέση καθυστέρηση πακέτων



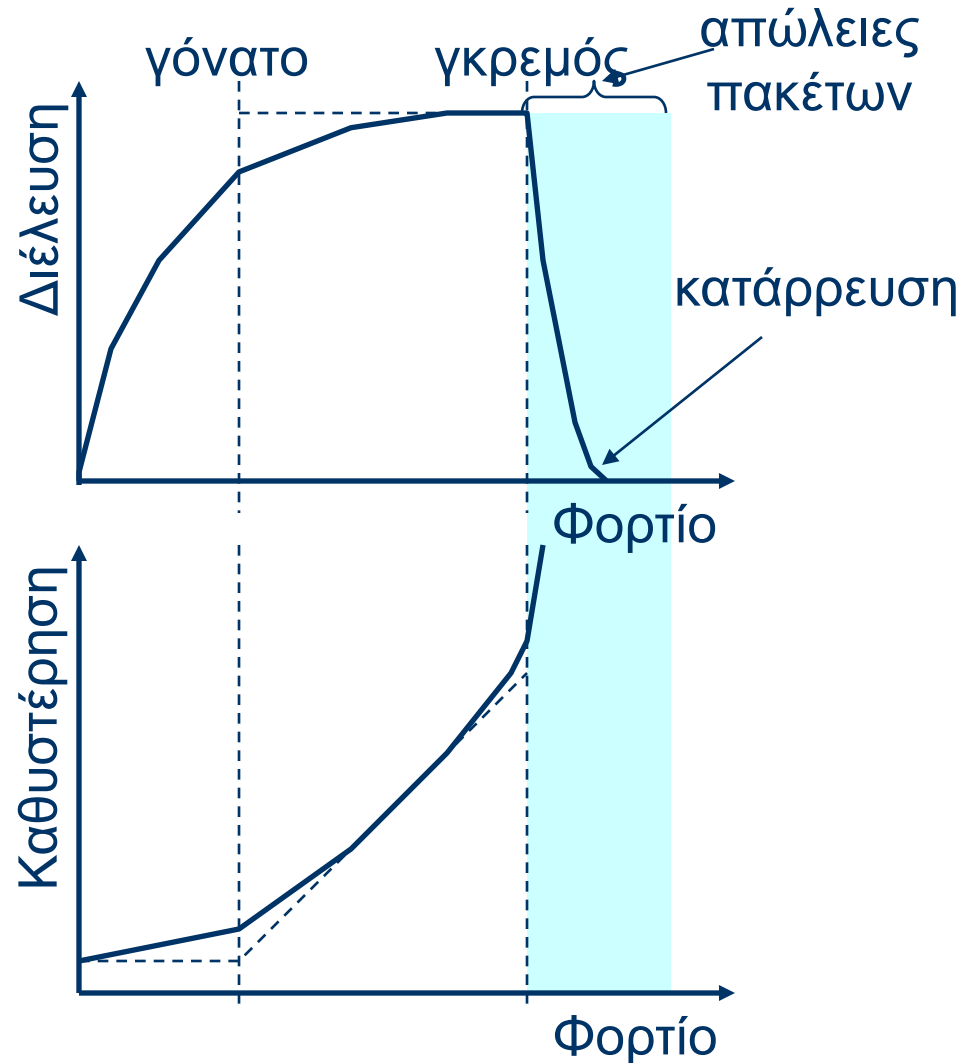
Επίδοση





# Κατάρρευση λόγω συμφόρησης

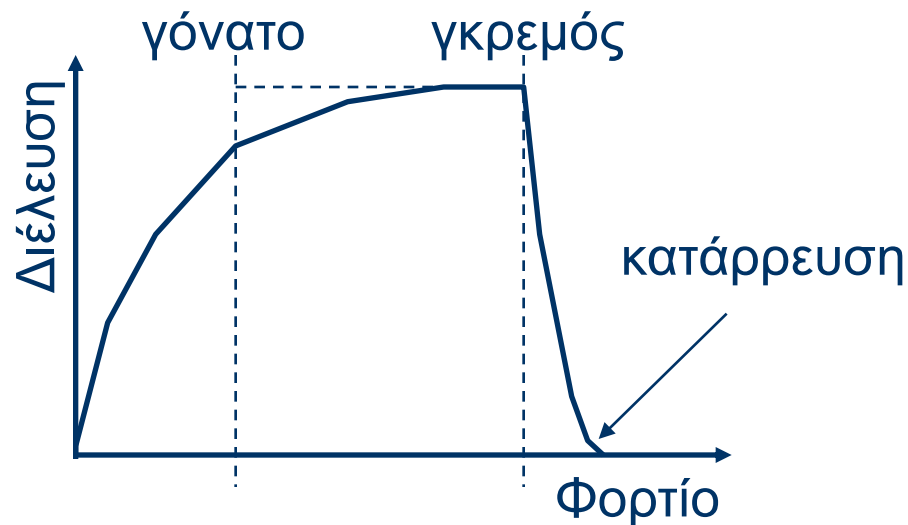
- Γόνατο (knee) - σημείο μετά από το οποίο
  - Η διέλευση (throughput) αυξάνει αργά
  - Η καθυστέρηση (delay) αυξάνει γρήγορα
- Γκρεμός (cliff) - σημείο μετά από το οποίο
  - Η διέλευση αρχίζει να μειώνεται πολύ γρήγορα προς το μηδέν
  - Η καθυστέρηση πλησιάζει το άπειρο



# Έλεγχος συμφόρησης και αποφυγή συμφόρησης



- Έλεγχος συμφόρησης είναι
  - Να μη πέσουμε στον γκρεμό
- Αποφυγή συμφόρησης
  - Να μη περάσουμε το γόνατο





# Τρόποι αντιμετώπισης



## (1) Κρατήσεις

- Δέσμευση εύρους ζώνης πριν τη χρήση
- Απαιτεί διαπραγμάτευση

## (2) Τιμολόγηση

- Μην απορρίπτεις πακέτα αυτών που πληρώνουν πιο πολύ
- Απαιτεί μοντέλο πληρωμών

## (3) Δυναμική προσαρμογή

- Δοκιμή για να βρεθεί το επίπεδο συμφόρησης
- Επιτάχυνε όταν δεν έχει συμφόρηση
- Επιβράδυνε όταν έχει συμφόρηση

# Τρόποι αντιμετώπισης



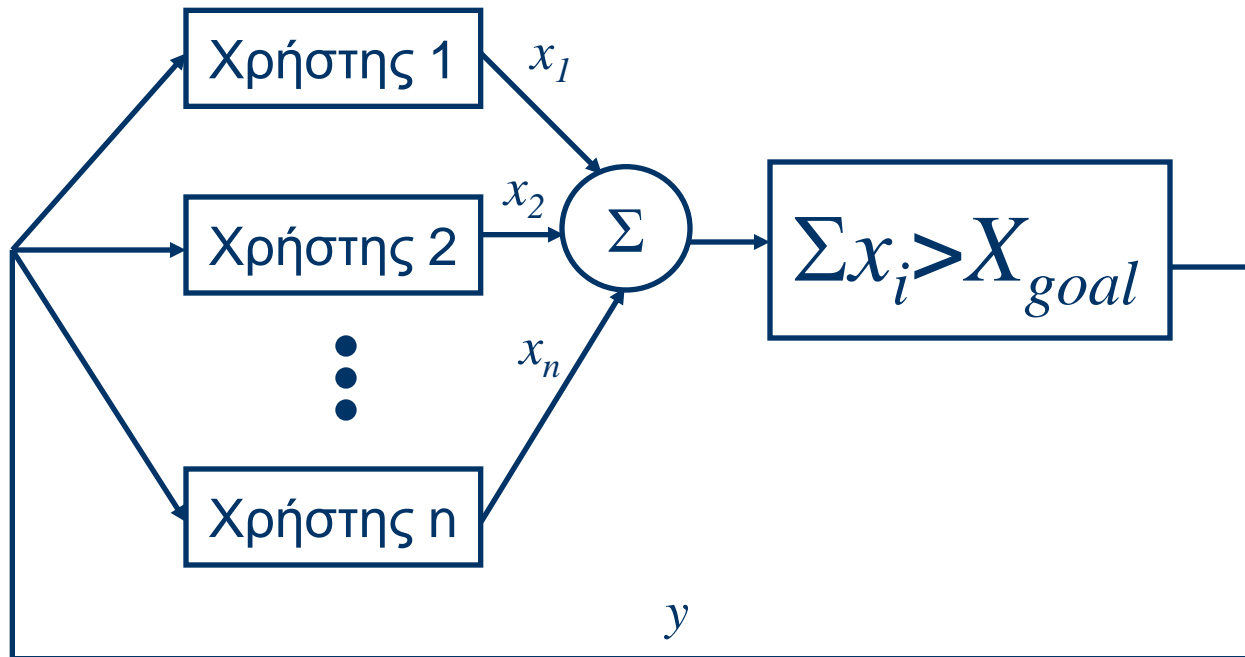
- Στο διαδίκτυο έχει επικρατήσει η τελευταία προσέγγιση
  - Υπο-βέλτιστη λύση
  - Ανώμαλη δυναμική συμπεριφορά
  - Εύκολη στην υλοποίηση
    - παρότι δύσκολη εκ πρώτης όψεως



# Δυναμική προσαρμογή



# Μοντέλο συστήματος ελέγχου



- Απλό μοντέλο
- Δυαδικό σήμα για συμφόρηση



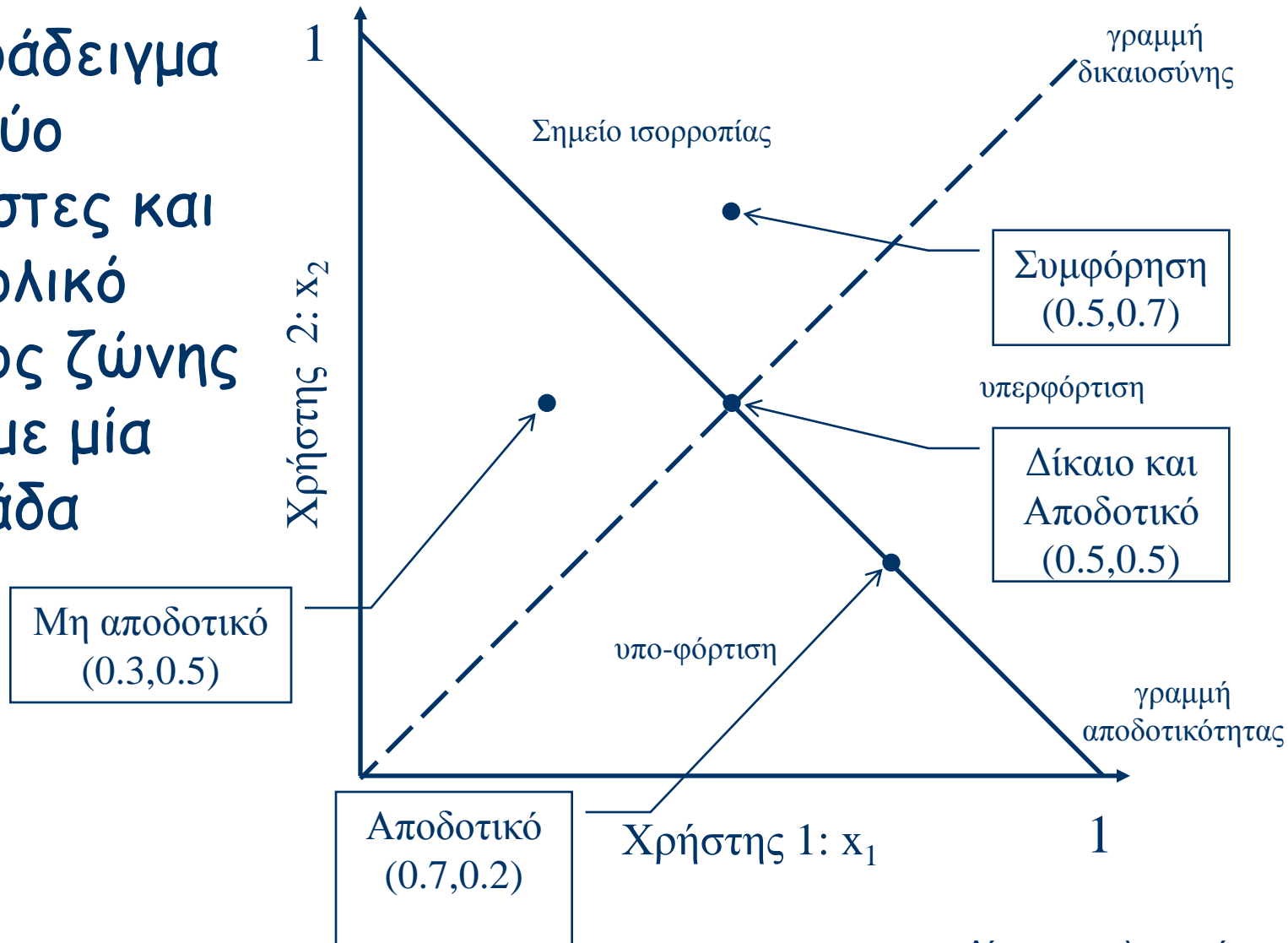
$$x_i(t+1) = \begin{cases} a_I + b_I x_i(t) & \text{αύξηση} \\ a_D + b_D x_i(t) & \text{μείωση} \end{cases}$$

- Πολλαπλασιαστική Αύξηση, Προσθετική Μείωση
  - $a_I=0, b_I>1, a_D<0, b_D=1$
- Προσθετική Αύξηση, Προσθετική Μείωση
  - $a_I>0, b_I=1, a_D<0, b_D=1$
- Πολλαπλασιαστική Αύξηση, Πολλαπλασιαστική Μείωση
  - $a_I=0, b_I>1, a_D=0, 0<b_D<1$
- Προσθετική Αύξηση, Πολλαπλασιαστική Μείωση
  - $a_I>0, b_I=1, a_D=0, 0<b_D<1$
- Τι να διαλέξουμε;



# Δικαιοσύνη και αποδοτικότητα

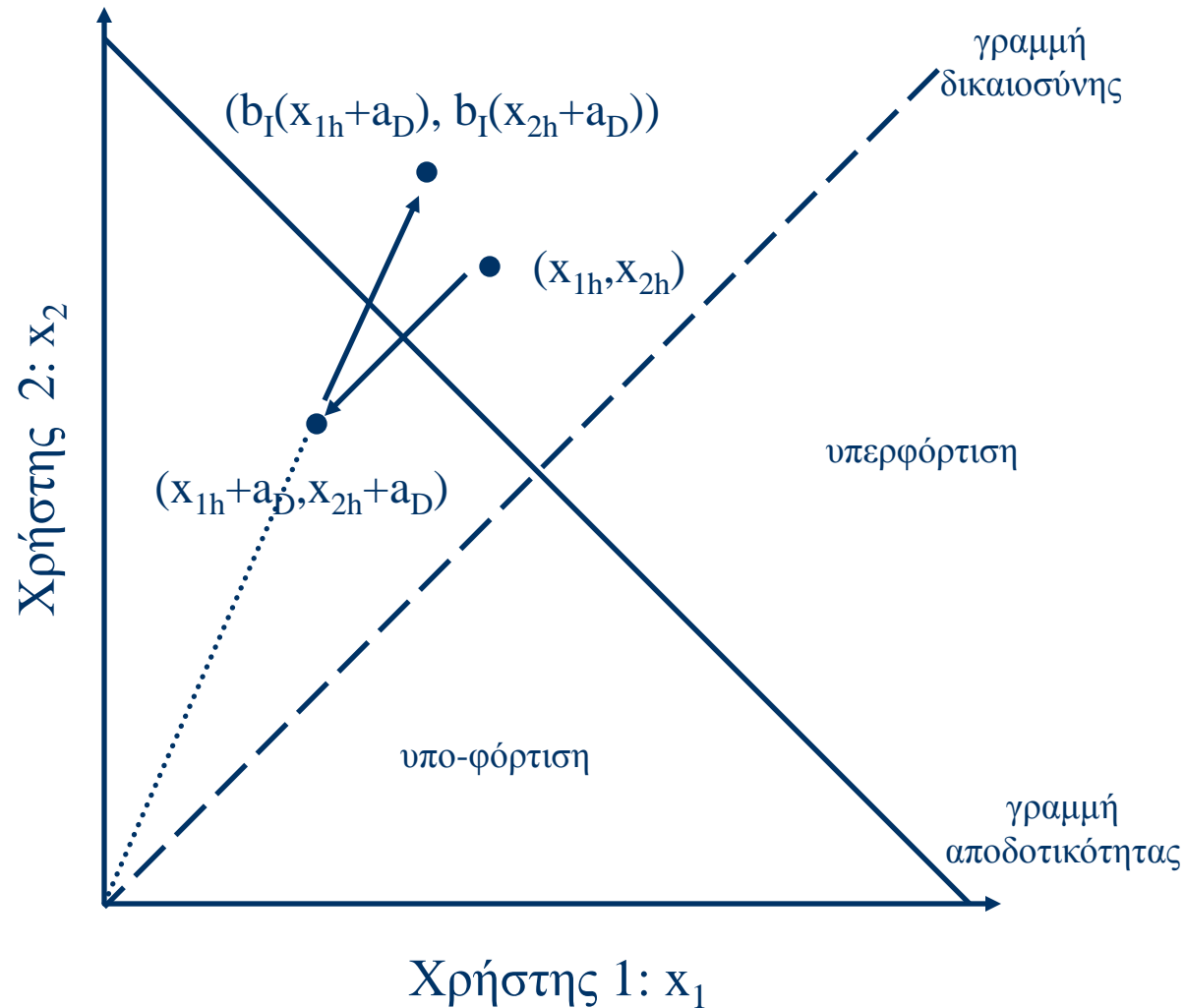
- Παράδειγμα με δύο χρήστες και συνολικό εύρος ζώνης ίσο με μία μονάδα



# Multiplicative Increase, Additive Decrease



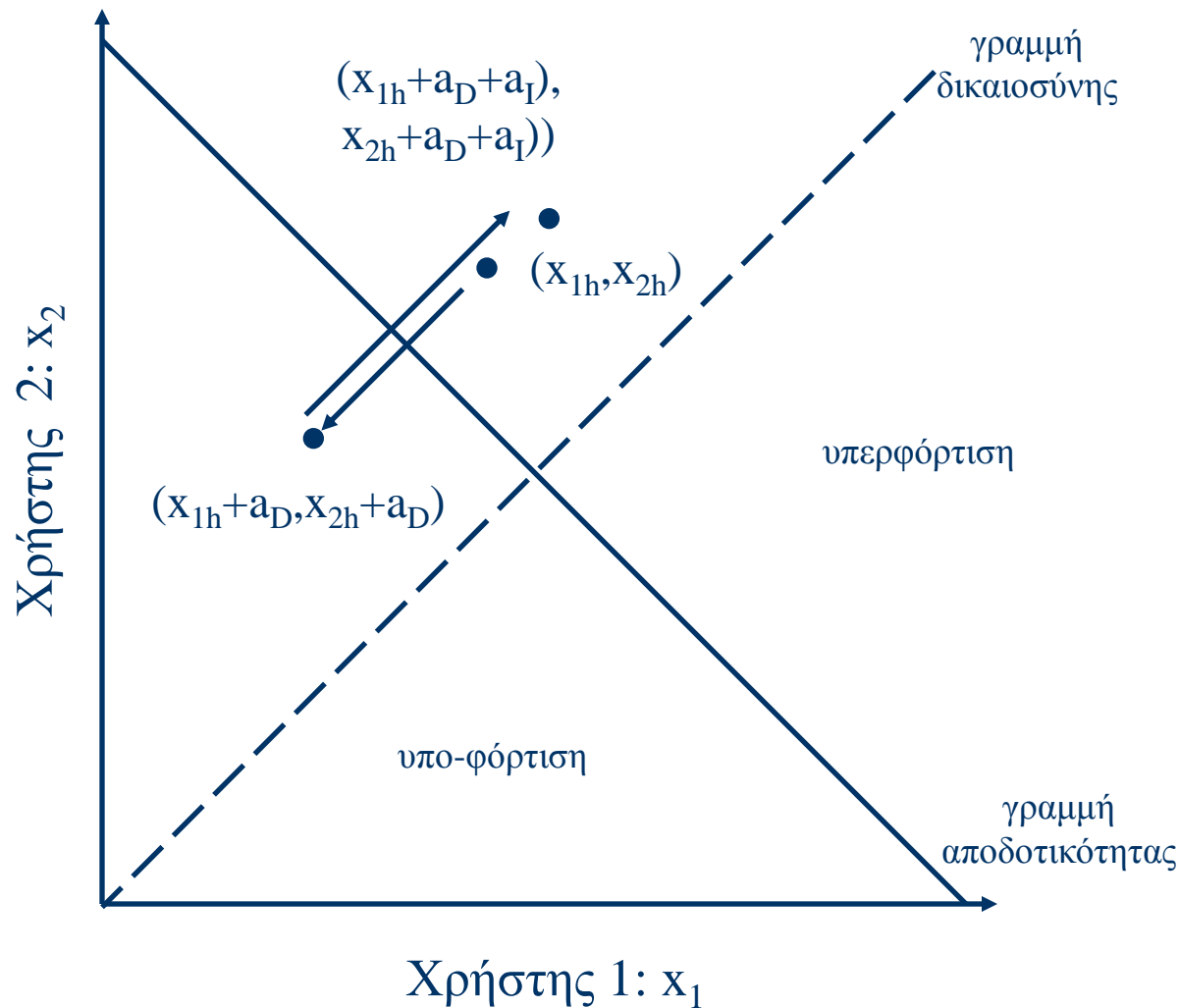
- Δε συγκλίνει, ασταθές



# Additive Increase, Additive Decrease



- Δε συγκλίνει, ταλαντώνεται

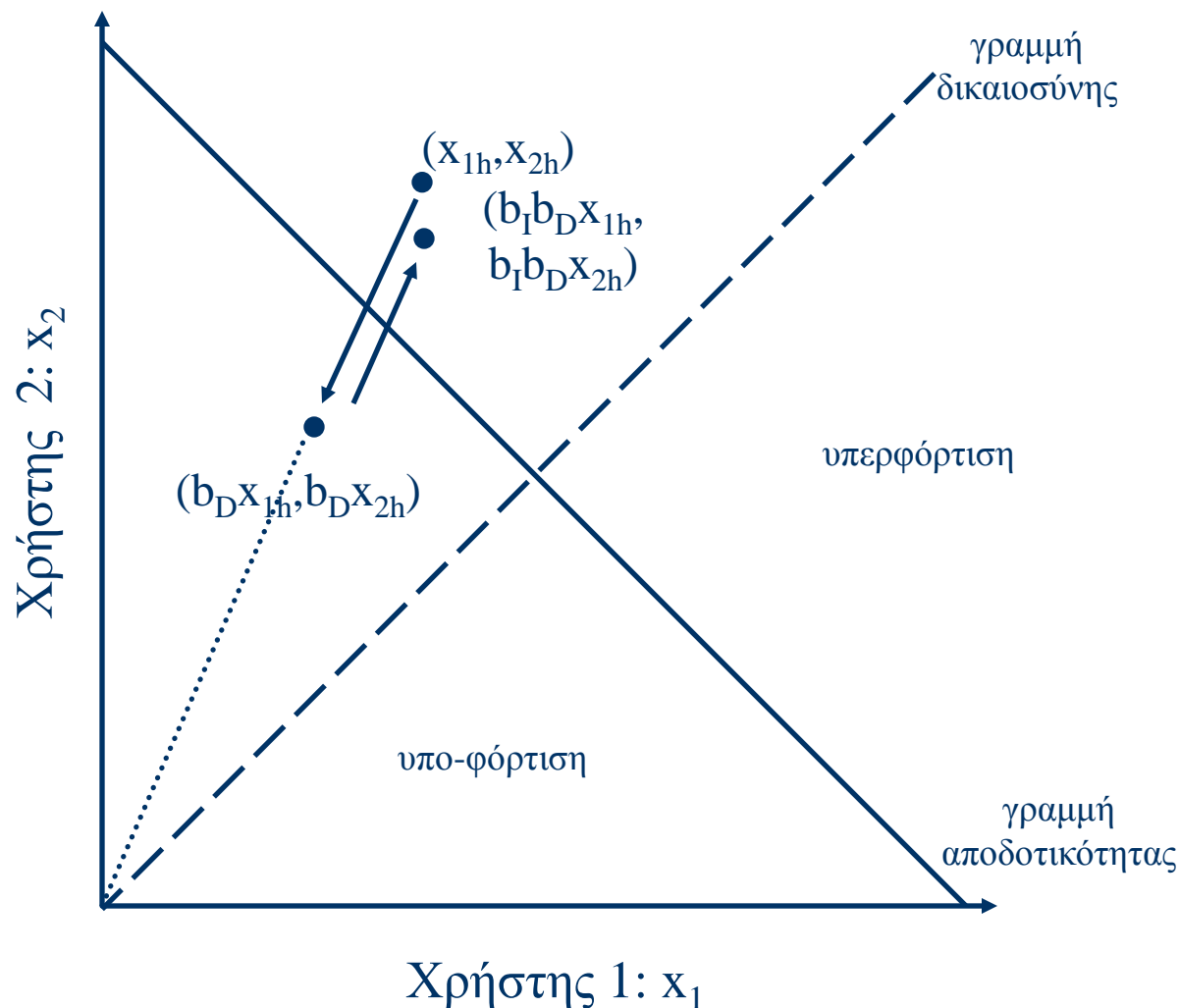




# Multiplicative Increase, Multiplicative Decrease



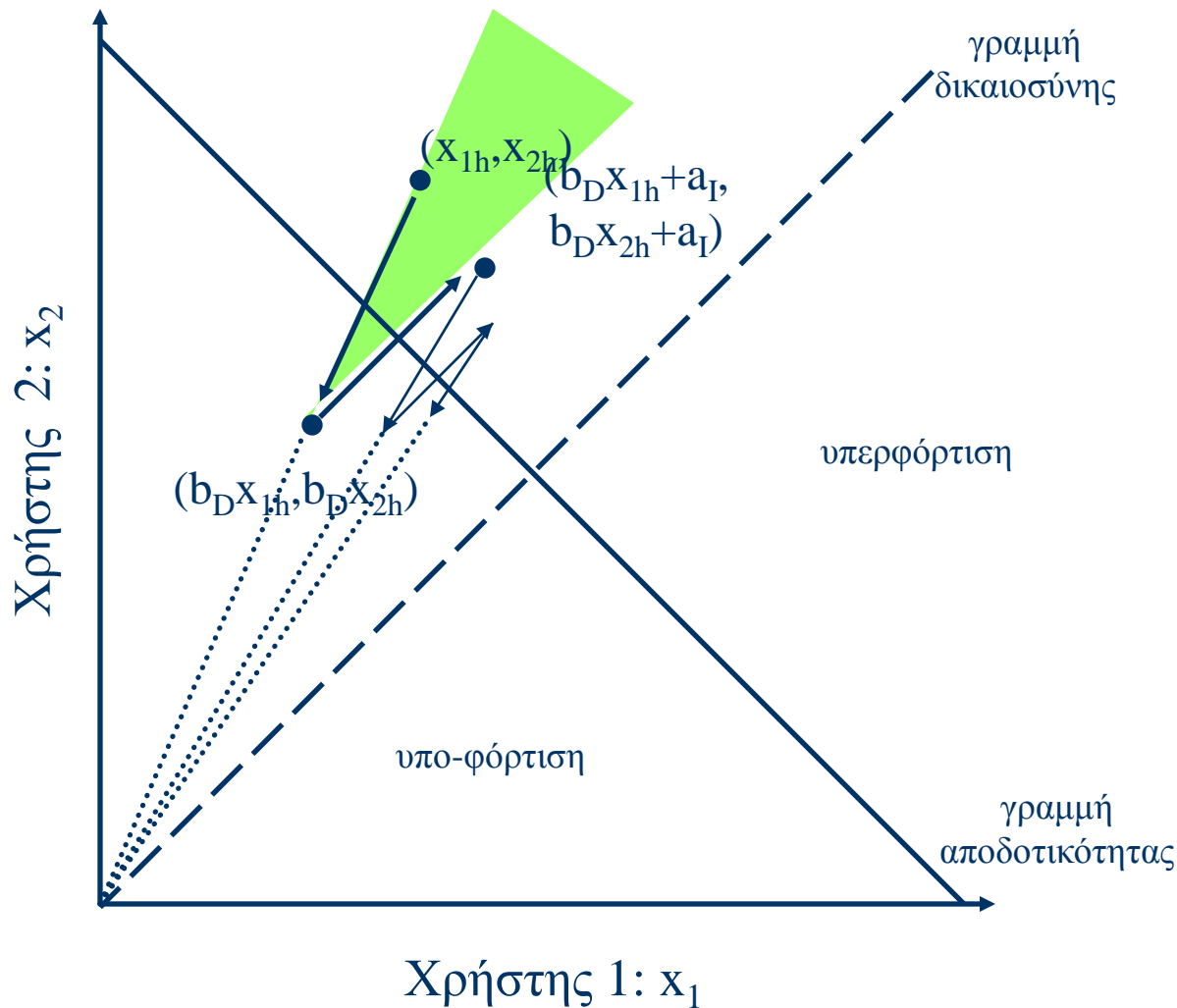
- Δε συγκλίνει σε σχέση με τη δικαιοσύνη
- Συγκλίνει σε σχέση με την αποδοτικότητα



# Additive Increase, Multiplicative Decrease



- Συγκλίνει σε σχέση με τη δικαιοσύνη
- Συγκλίνει σε σχέση με την αποδοτικότητα
- Οι αυξήσεις μικραίνουν καθώς αυξάνει η δικαιοσύνη



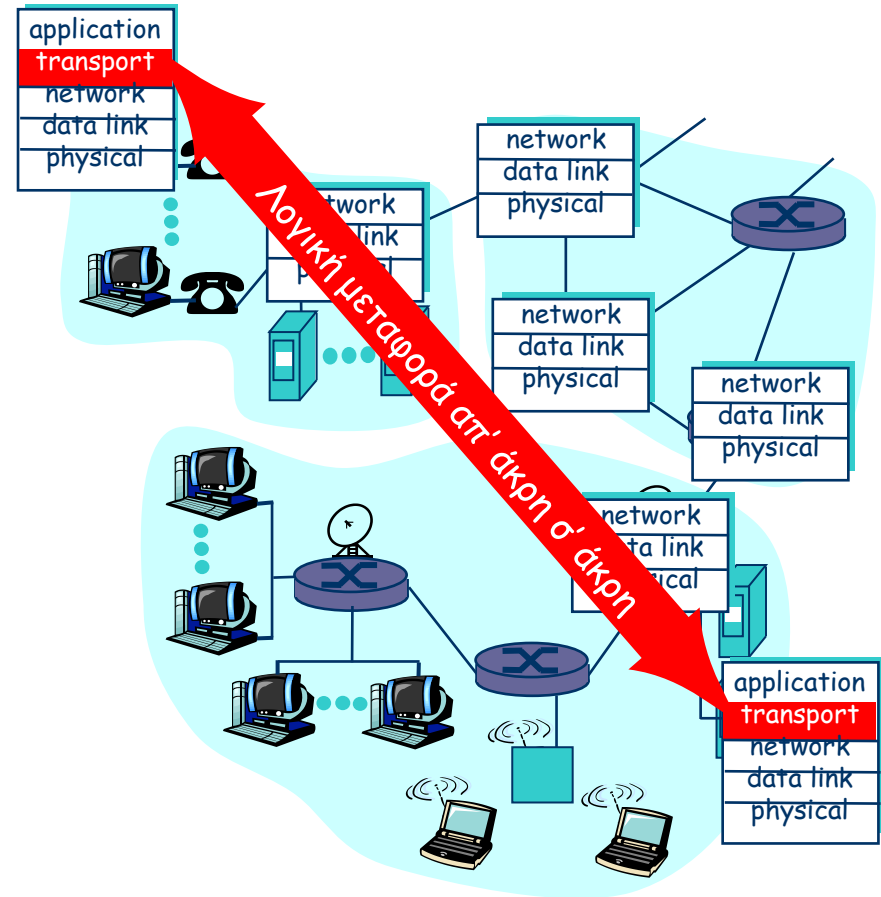


# Έλεγχος συμμόρφωσης στο Internet



# Υπηρεσίες μεταφοράς στο Internet

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





# Έλεγχος συμφόρησης

- 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

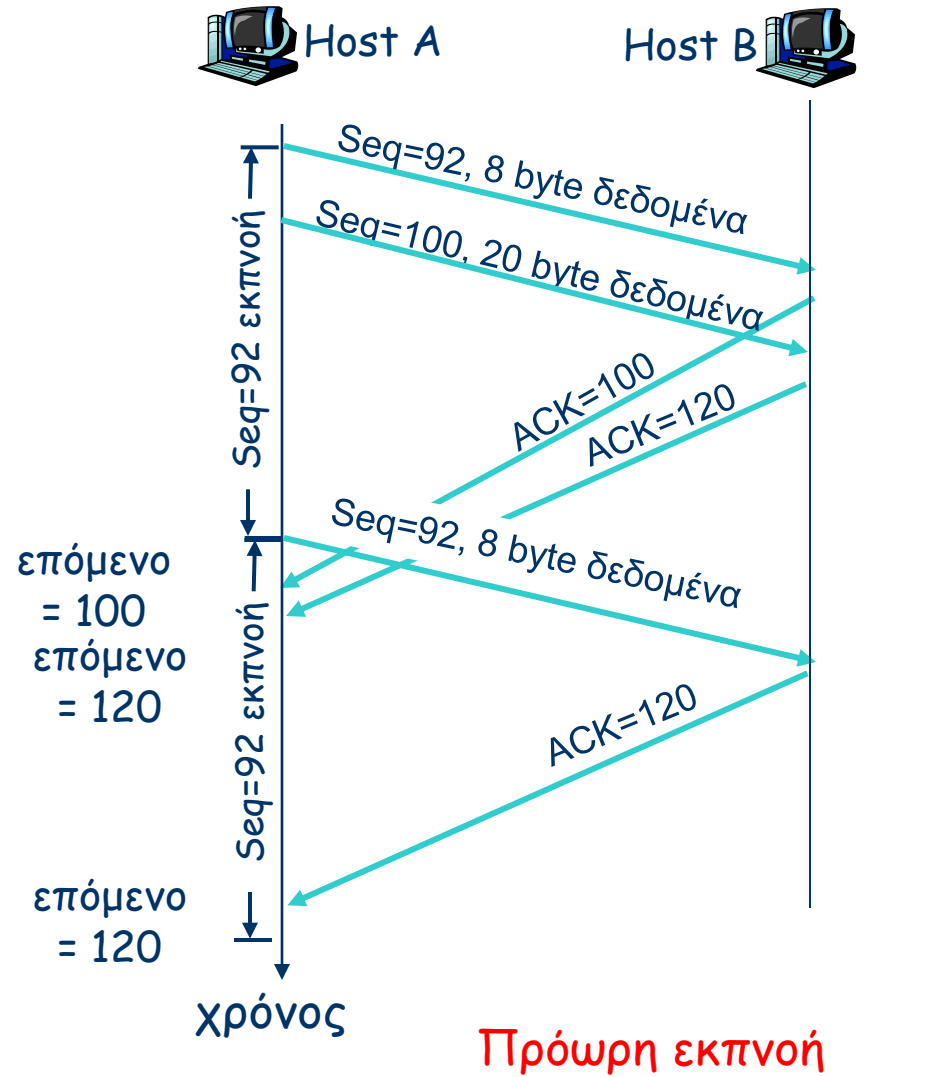
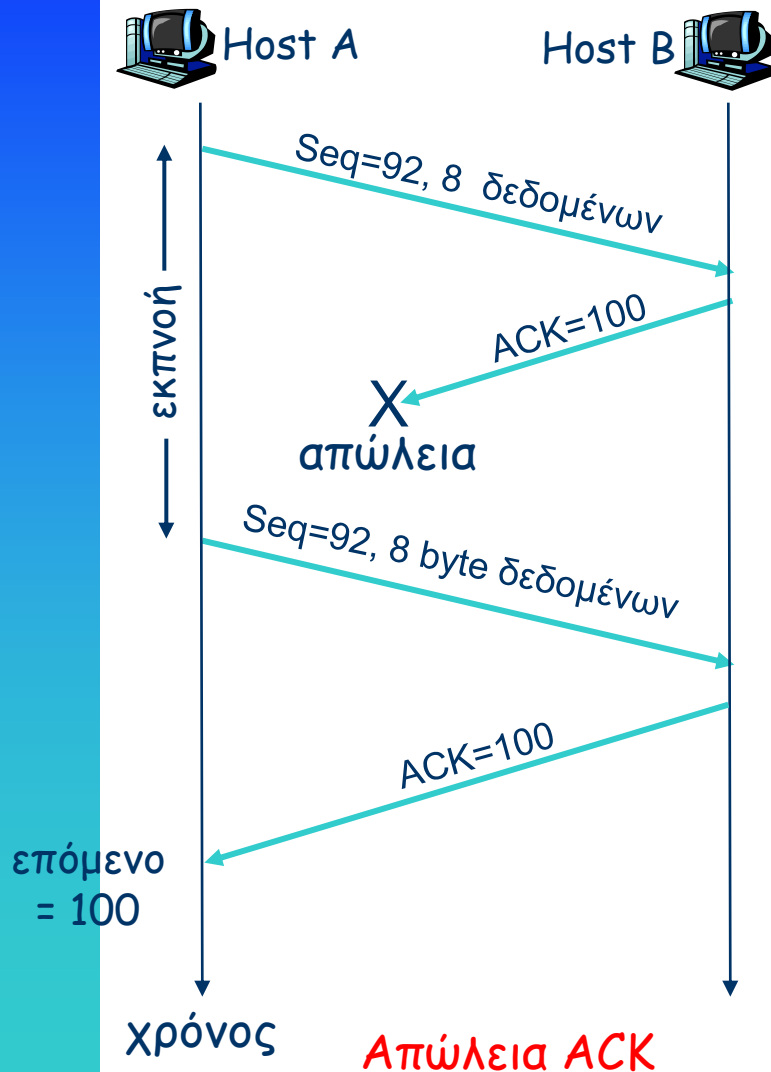


# Έλεγχος συμφόρησης στο TCP

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

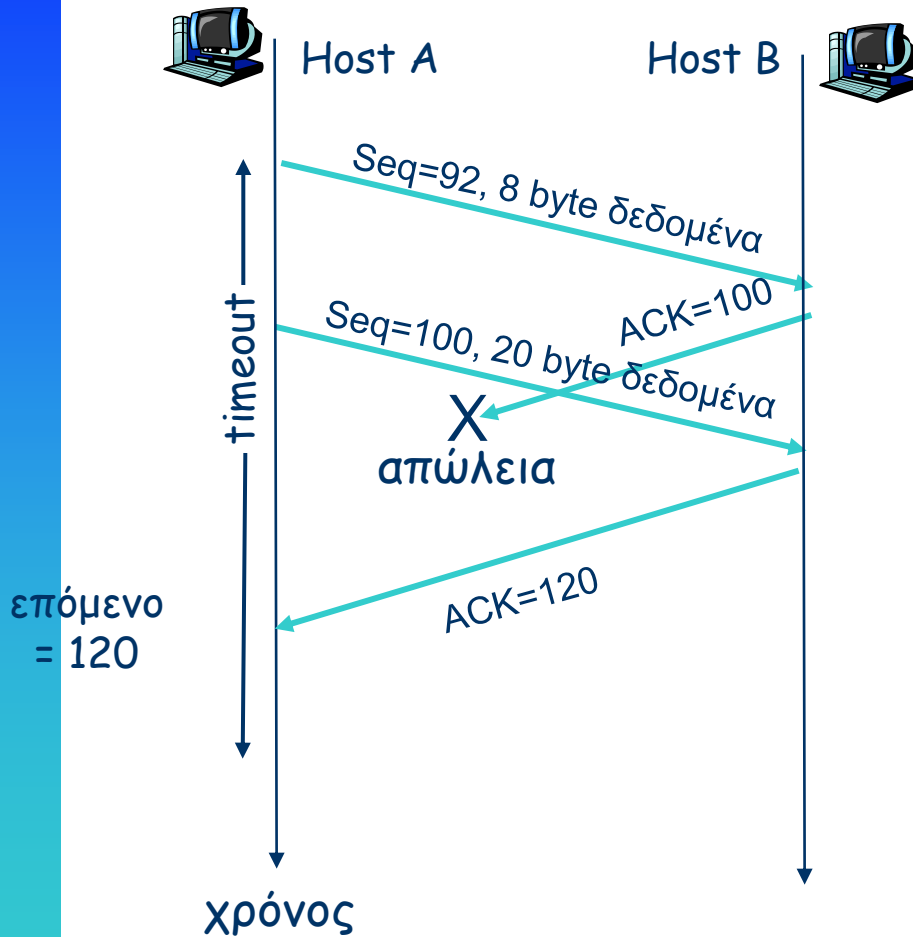


# Παρατηρήσιμα γεγονότα





# Παρατηρήσιμα γεγονότα



**Συσσωρευτικό ACK**





# Βασικά σημεία

- Οι περισσότερες εκπνοές χρόνου στο Internet οφείλονται σε συμφόρηση!
  - Οι αποστολείς TCP ανιχνεύσουν τη συμφόρηση και μειώσουν τον ρυθμό αποστολής
  - Οι δρομολογητές επιβραδύνουν τους αποστολείς TCP απορρίπτοντας πακέτα
- 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**"!

# Σχέση με έλεγχο ροής/λαθών



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



# Έλεγχος συμμόρφωσης στο TCP



# Μηχανισμός ελέγχου

- Υπάρχουν δύο ενδεχόμενα προβλήματα
  - Η χωρητικότητα του δικτύου
  - Η χωρητικότητα του αποδέκτη
- Απαιτείται χωριστή αντιμετώπιση του κάθε προβλήματος
- Ο αποστολέας τηρεί τις μεταβλητές:
  - **Advertised Window (*flow\_win*)**  
το διαφημίζει ο αποδέκτης
  - **Congestion Window (*cwnd*)**  
τροποποιείται με βάση την ανάδραση από το δίκτυο
  - **Slow-start threshold Value (*ssthresh*)**  
αρχική τιμή το διαφημιζόμενο μέγεθος παραθύρου



# Μηχανισμός ελέγχου

- Ο αποστολέας TCP αλλάζει τον ρυθμό αποστολής τροποποιώντας το παράθυρο αποστολής:

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

⏟      ⏟  
Πομπός      Δέκτης

- Το πλήθος byte που μπορούν να αποσταλούν είναι το μικρότερο από τα δύο παράθυρα
  - Το διαφημιζόμενο παράθυρο (advertised window)
  - Το παράθυρο συμφόρησης (congestion window)
- Δηλαδή, στείλε με τον ρυθμό του αργότερου:
  - δίκτυο ή αποδέκτης



# Το παράθυρο συμφόρησης

- **ιδανικά:** μετάδωσε όσο γρήγορα μπορείς χωρίς απώλειες (παράθυρο συμφόρησης όσο το δυνατό μεγαλύτερο)
- Εκτιμάται από τον αποστολέα βάσει της ανάδρασης που λαμβάνει από το δίκτυο
- **"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

- 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:  $cwnd = 1$ 
  - Η μονάδα μέτρησης είναι το μέγεθος τεμαχίου
  - Το TCP στην πραγματικότητα μετρά byte και επομένως αυξάνει κατά 1 MSS (maximum segment size)
- 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
  - Εν γένει, ο δέκτης TCP στέλνει ένα ACK για κάθε δεύτερο τεμάχιο
- Ακόμη και εάν το ACK επαληθεύει τεμάχιο μικρότερο από MSS byte, το  $cwnd$  πάλι αυξάνει κατά 1 τεμάχιο





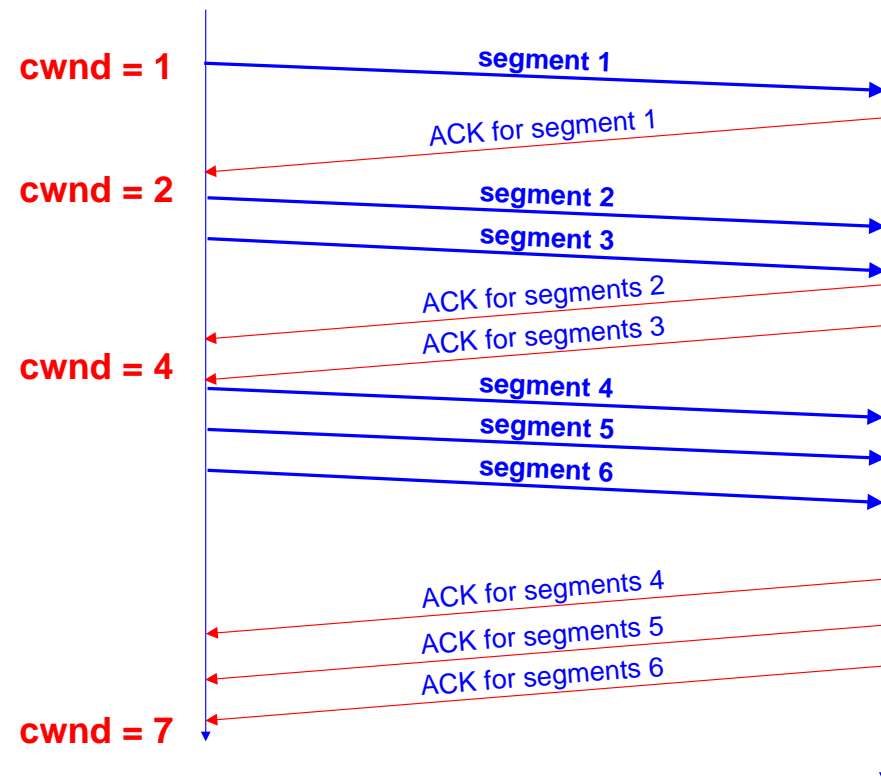
# Αργή Αρχή

- Είναι η αργή αρχή πραγματικά **αργή**;
- Όχι, στην πραγματικότητα το **cwnd** αυξάνει **εκθετικά** ανά RTT
- Γιατί αποκαλείται αργή αρχή;
  - Επειδή το TCP αρχικά δεν διαθέτε 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* ≥ *ssthresh***





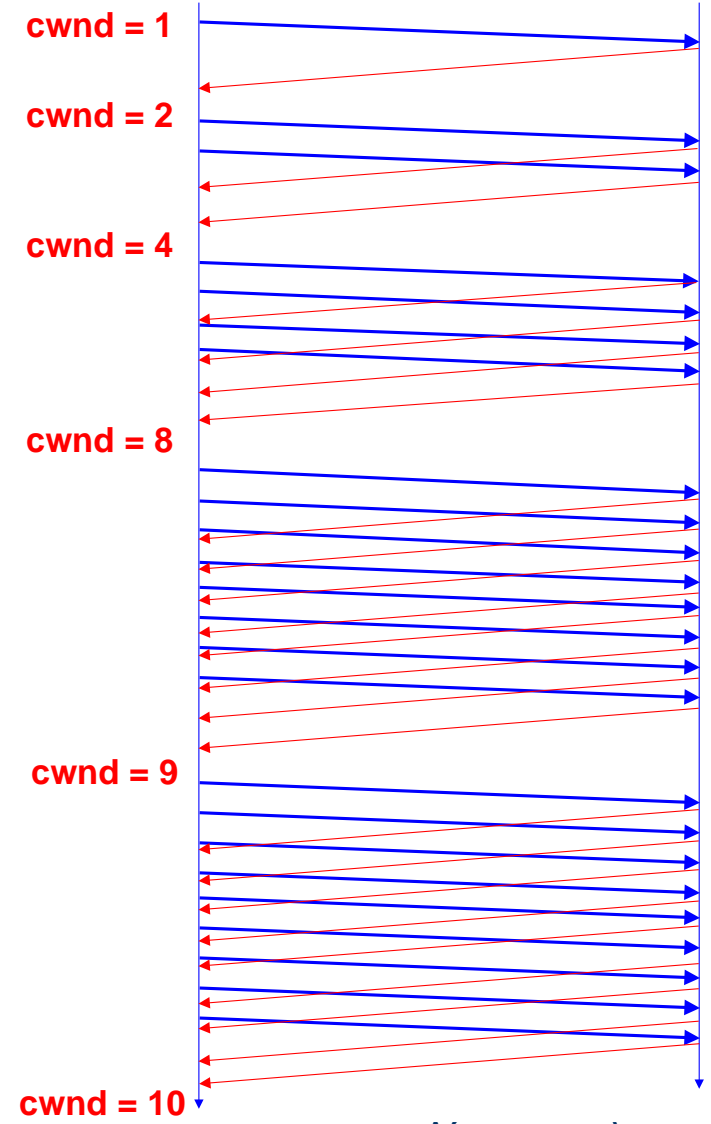
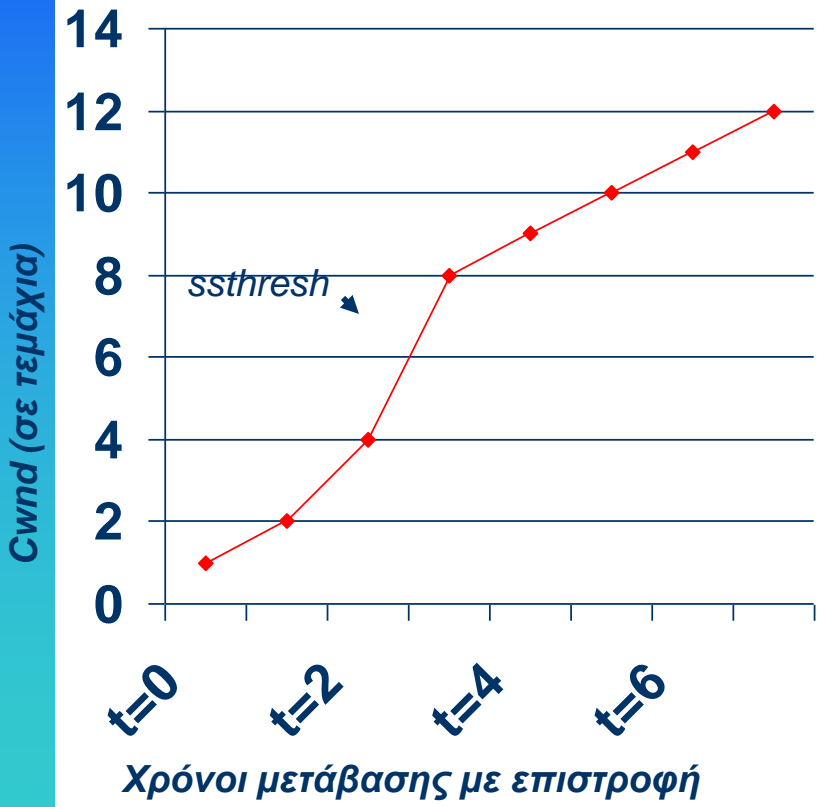
# Αποφυγή Συμφόρησης

- 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  $ssthresh = 8$



Δίκτυα υπολογιστών



# 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



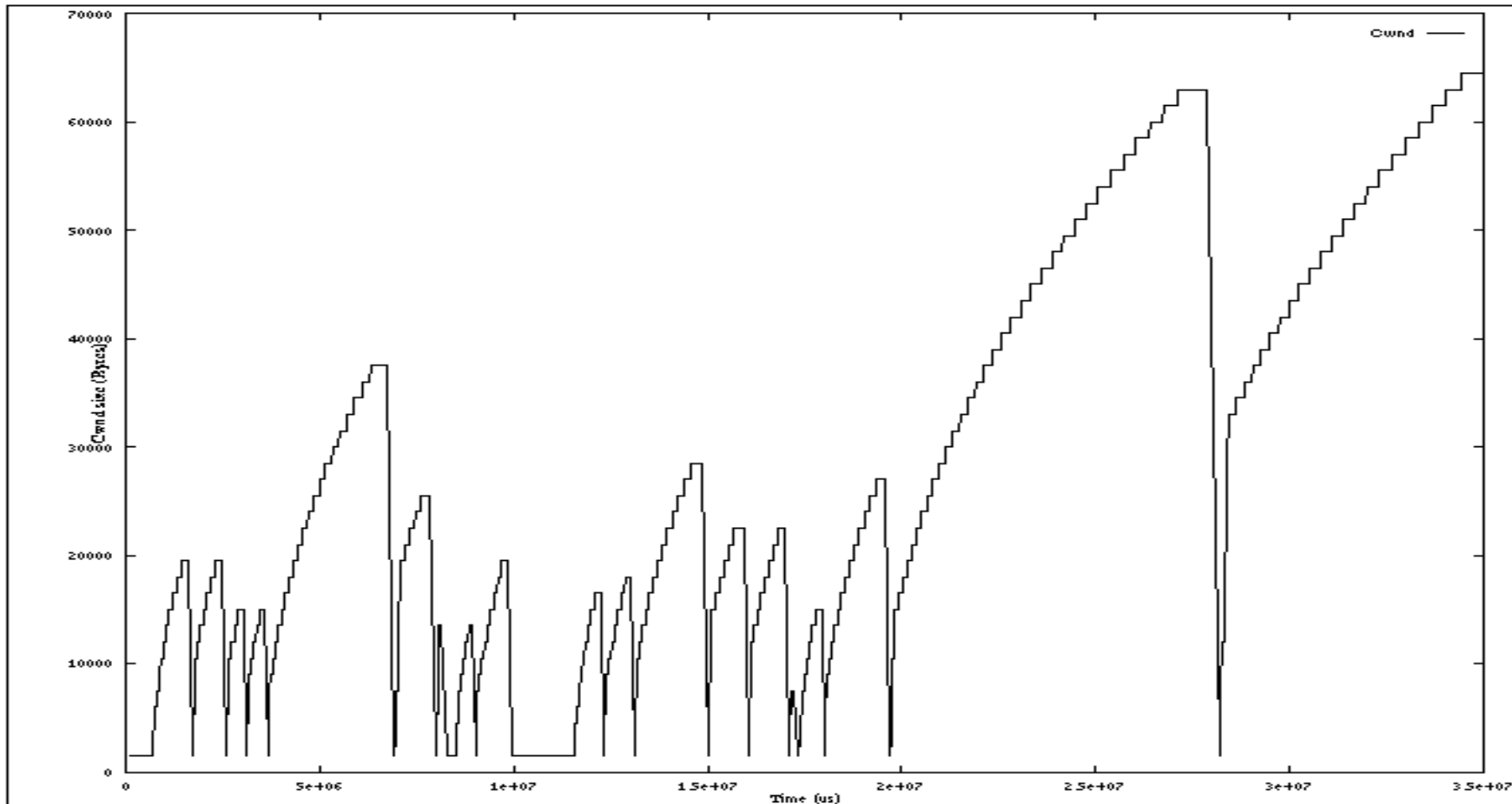
# Αντίδραση στην συμφόρηση

- 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
- Φαινόμενο πριονιού TCP



# Το "πριόνι" TCP

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



# Ψευδοκώδικας ελέγχου συμφόρησης TCP



Αρχικά:

$cwnd = 1;$

$ssthresh = \text{advertised window size};$

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

$win = \min(cwnd, adv\_win);$







# Πρόβλημα με την αργή αρχή

- Η αργή αρχή υποθέτει ότι οι απώλειες οφείλονται σε συμφόρηση του δικτύου
  - υπάρχουν περιπτώσεις, π.χ. ασύρματα δίκτυα, όπου οι απώλειες οφείλονται στην κακή ποιότητα της ζεύξης δεδομένων
  - σε τέτοιες περιπτώσεις η επίδοση του TCP είναι κακή
- Η αργή αρχή έχει κακή επίδοση όταν οι συνδέσεις έχουν μικρή διάρκεια
  - σε μικρές μεταφορές, π.χ. σελίδες web

# Βελτιώσεις



- **Ταχεία αναμετάδοση**
  - πρόωρη αποστολή (πριν τη λήξη του χρονόμετρου)
- **Ταχεία ανάκαμψη**
  - επιτάχυνση της διαδικασίας ανάκαμψης

# Ταχεία αναμετάδοση (Fast Retransmit)



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

# Υπενθύμιση για τα ACK

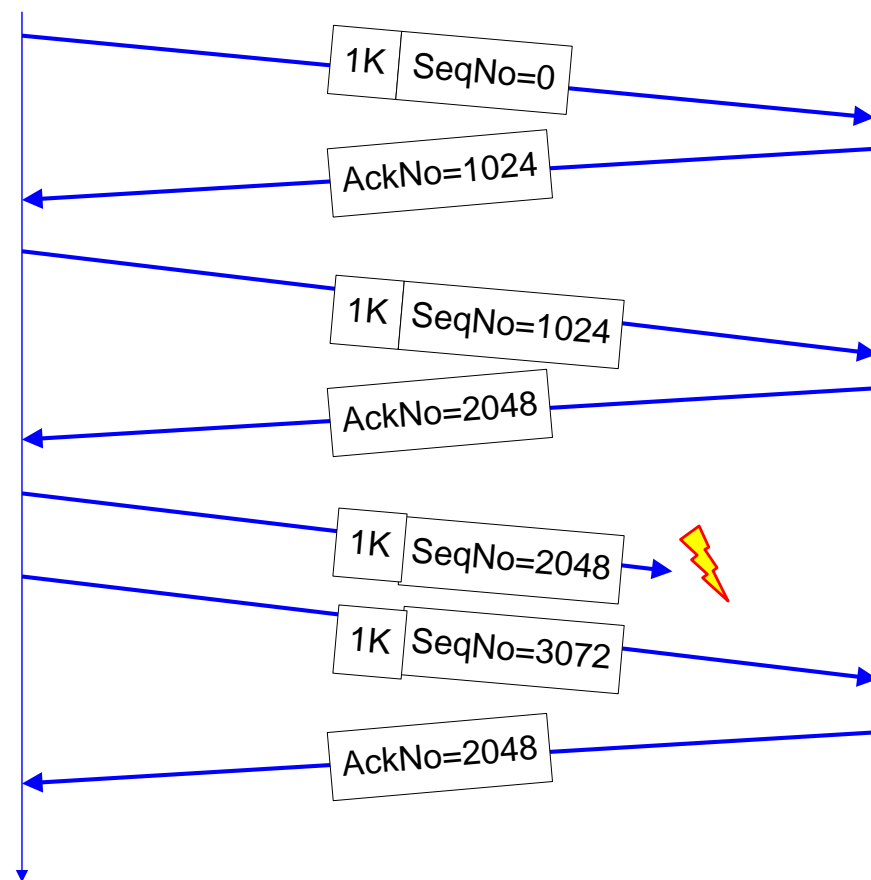


- Το TCP παράγει ένα ταυτόσημο (duplicate) ACK όταν λαμβάνεται τεμάχιο εκτός σειράς
- Αυτό το ταυτόσημο ACK δεν πρέπει να καθυστερήσει
  - Ο σκοπός του είναι να πει στον αποστολέα ότι έφτασε ένα τεμάχιο εκτός σειράς και
  - να δηλώσει το αύξοντα αριθμό που αναμένεται



# Επαληθεύσεις - Απώλεια τεμαχίου

- 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

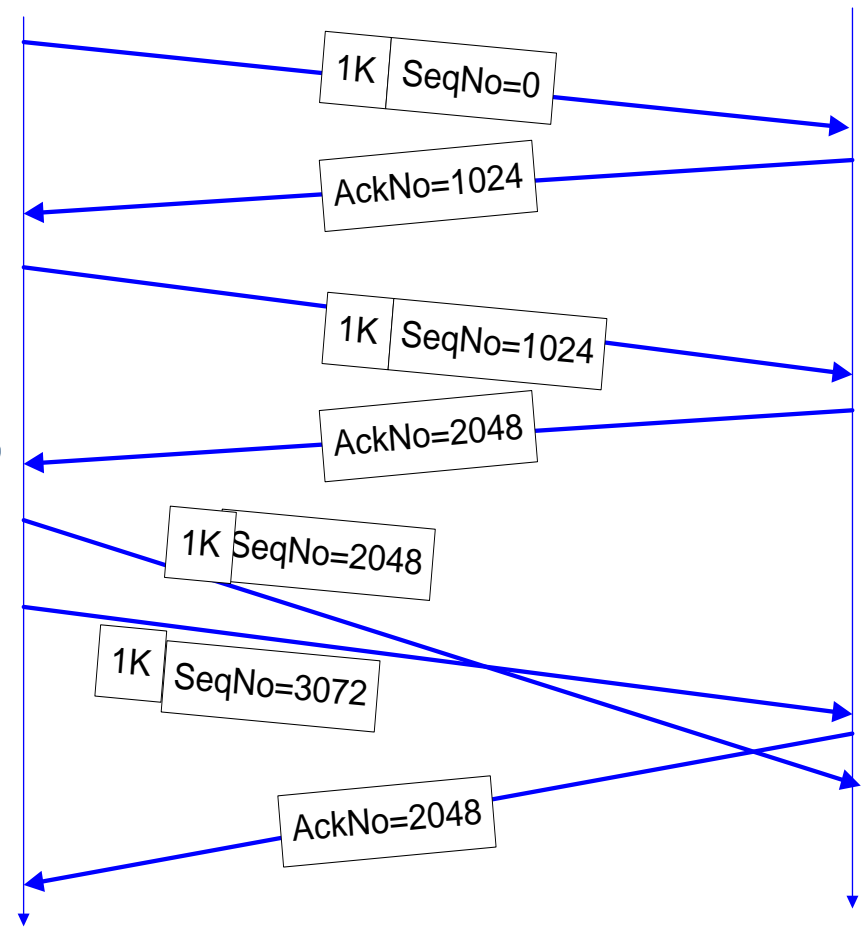


**Απώλεια τεμαχίου**



# Επαληθεύσεις - Άφιξη εκτός σειράς

- 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



Άφιξη εκτός σειράς

# Ταυτόσημα ACK



- Όμως ο αποστολέας δε ξέρει εάν το ταυτόσημο ACK προέρχεται από απώλεια τεμαχίου ή από άφιξη εκτός σειράς
  - Για αυτό περιμένει για λίγα ακόμη ταυτόσημα ACK
  - Εάν πρόκειται για άφιξη εκτός σειράς, θα υπάρξουν ένα ή δύο ταυτόσημα ACK πριν την παραγωγή νέου ACK
  - Όμως, τρία ή περισσότερα ταυτόσημα ACK στη σειρά, αποτελούν ισχυρή ένδειξη ότι έχει χαθεί ένα τεμάχιο

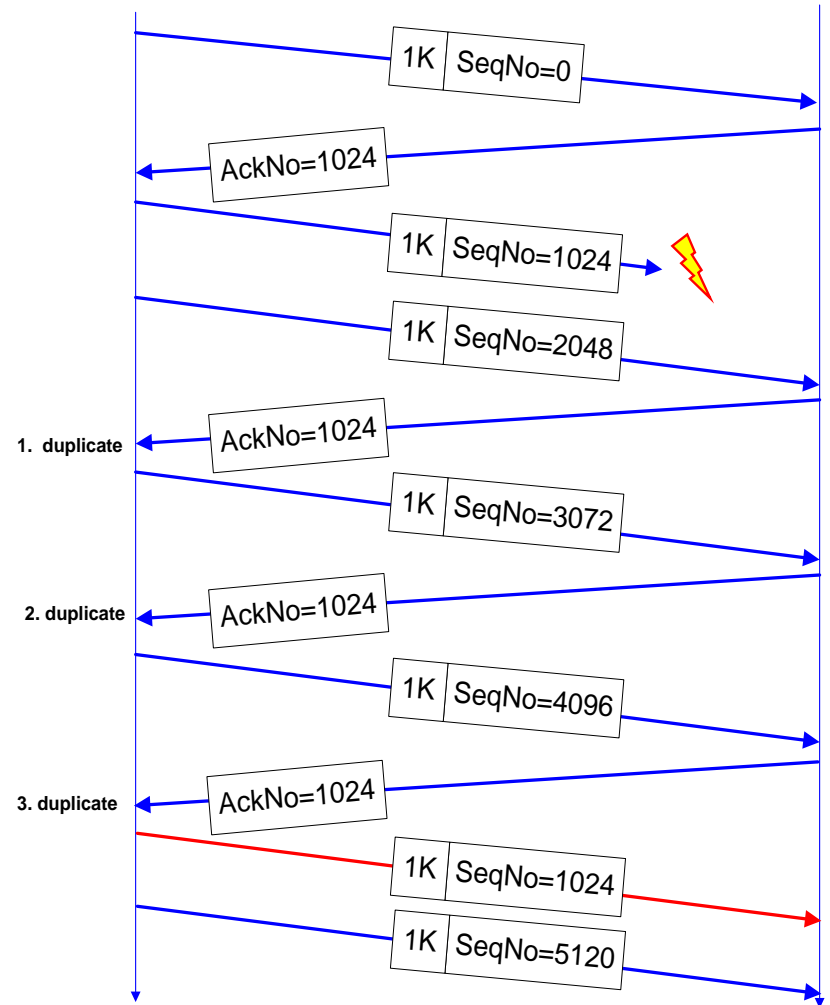


# Ταχεία αναμετάδοση (Fast Retransmit)

- 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

$$ssthresh = cwnd/2$$

$$cwnd = 1$$





# Πρόβλημα με την ταχεία αναμετάδοση



- Η ταχεία αναμετάδοση είναι απίθανο να ενεργοποιηθεί εάν δεν υπάρχουν πολλά προς μετάδοση πακέτα
  - δεν είναι χρήσιμη για μικρές μεταφορές, π.χ. σελίδες web

# Ταχεία ανάκαμψη (Fast Recovery)

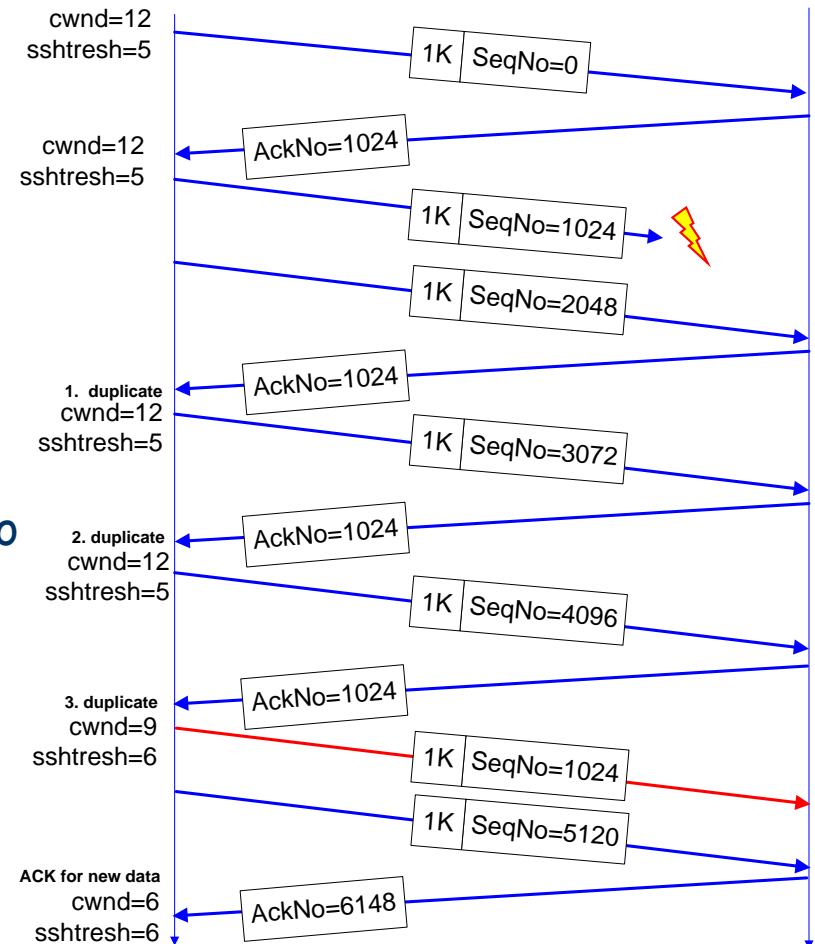


- Αποφυγή της αργής αρχής μετά από την ταχεία αναμετάδοση
  - Η λήψη ταυτόσημων ACK, εκτός από πιθανή απώλεια, δείχνει επιπλέον ότι τα δεδομένα προωθούνται στον προορισμό
  - Δεν υπάρχει λόγος να αρχίσει η διαδικασία διερεύνησης από την αρχή



# Ταχεία ανάκαμψη (Fast Recovery)

- Μετά τρία ταυτόσημα ACK
  - Αναμετάδοση του "χαμένου τεμαχίου"
  - $ssthresh = cwnd/2$
  - $cwnd = ssthresh + 3$
  - Αύξηση του  $cwnd$  κατά 1 για κάθε επιπλέον ταυτόσημο ACK
    - αποστολή νέων τεμαχίων εάν το επιτρέπει η νέα τιμή του  $cwnd$
- Όταν φτάσει ACK για "νέα δεδομένα" (εδώ: AckNo=6148)
  - $cwnd = ssthresh$
  - Αποφυγή συμφόρησης



# Ταχεία ανάκαμψη (Fast Recovery)



- Η ταχεία ανάκαμψη αποτελεί βελτίωση μόνο όταν έχουμε **μία** απώλεια πακέτου μέσα στο διάστημα μετάδοσης μετ' επιστροφής (round-trip time)
- Εάν υπάρξει εκπνοή χρόνο, τότε ακολουθεί αναμετάδοση και αργή αρχή

# Παραλλαγές Ελέγχου Συμφόρησης

## TCP

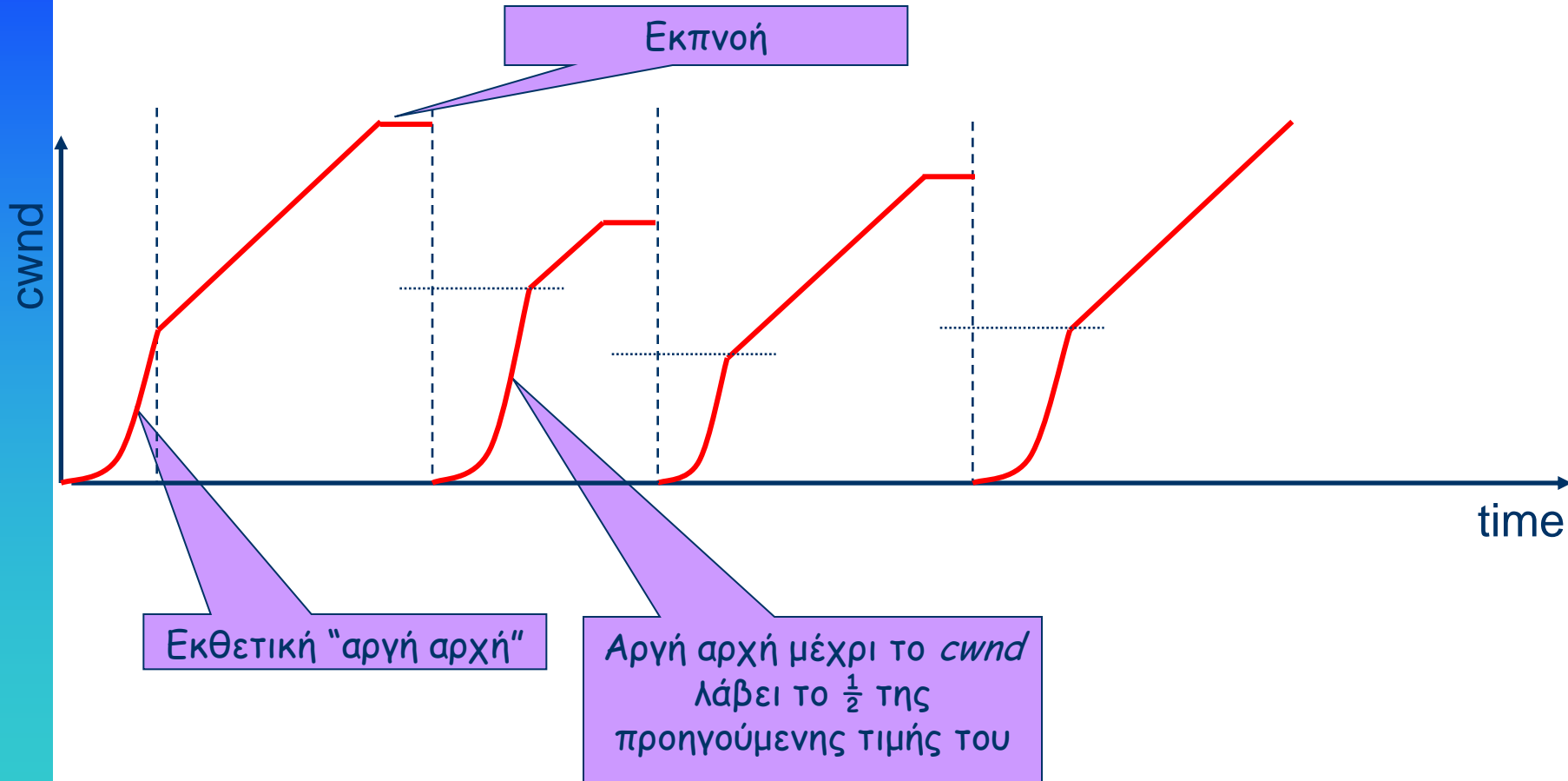


- **TCP Tahoe** (1988, FreeBSD 4.3 Tahoe)
  - Αργή Αρχή
  - Αποφυγή Συμφόρησης
  - Ταχεία Αναμετάδοση
- **TCP Reno** (1990, FreeBSD 4.3 Reno)
  - Ταχεία Ανάκαμψη
- **New Reno** (1996)
- **SACK** (1996)



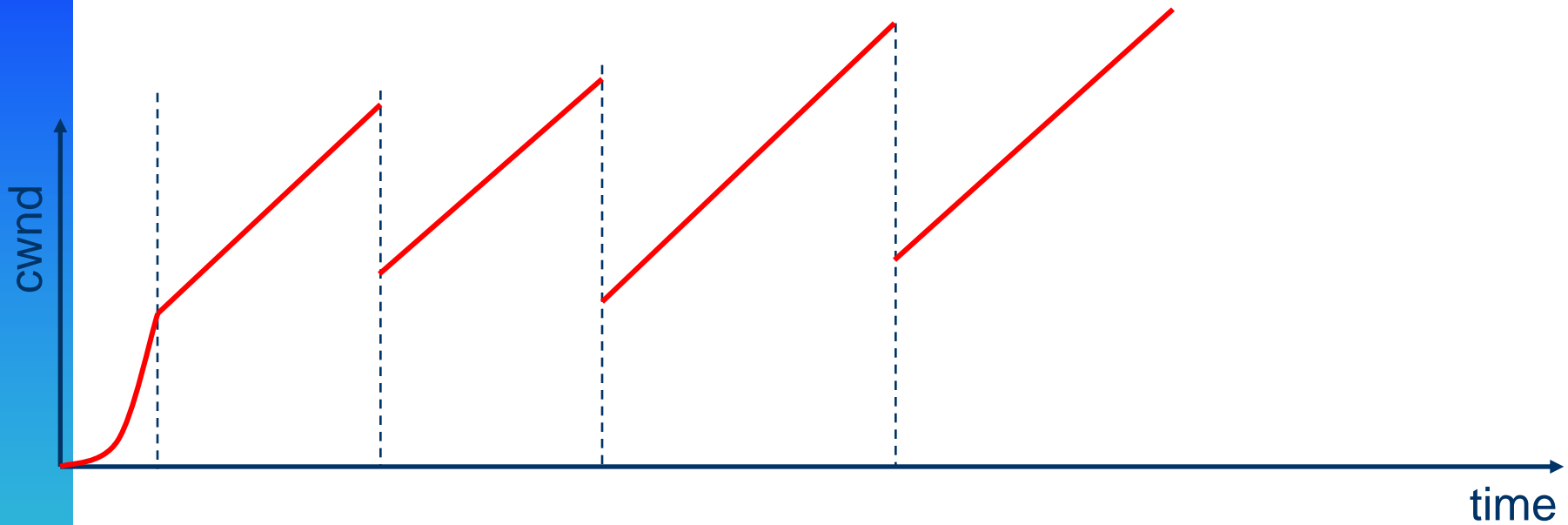
# Δυναμική συμπεριφορά TCP Tahoe

- Η ανάκαμψη καθυστερεί λόγω της αργής αρχής





# Δυναμική συμπεριφορά TCP Reno

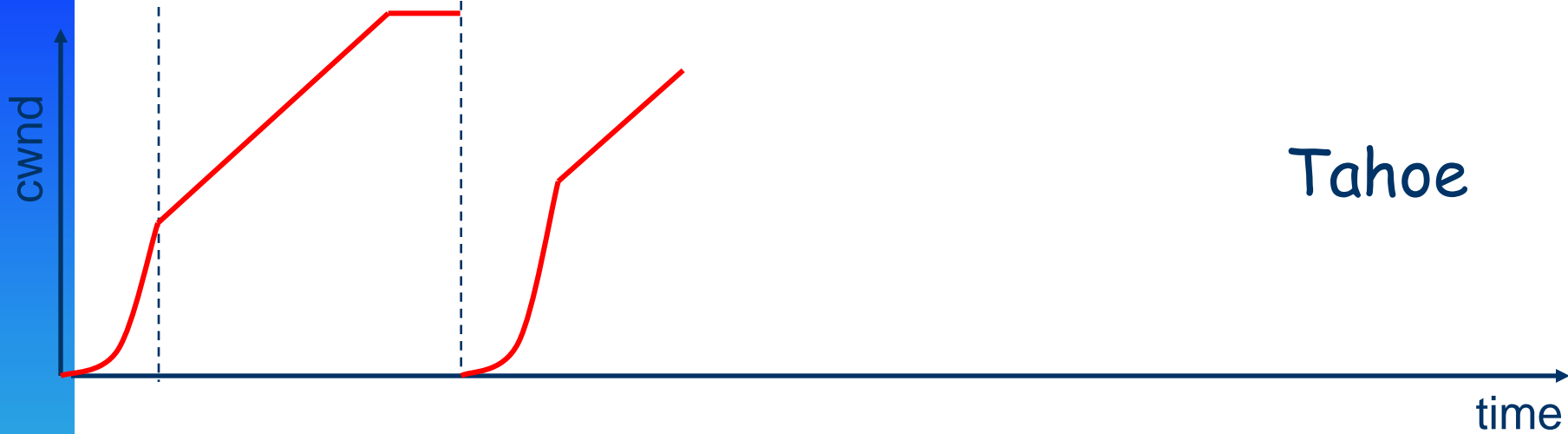


- Στη μόνιμη κατάσταση το *cwnd* ταλαντώνεται γύρω από τη βέλτιστη τιμή του παραθύρου



# Σύγκριση TCP Tahoe και TCP Reno

(για απώλειες ενός τεμαχίου)





# To new Reno



- Όταν έχουμε πολλαπλές απώλειες πακέτων, το TCP Reno αντιμετωπίζει πρόβλημα
- Μερικό (partial) ACK:
  - Συμβαίνει όταν χάνονται πολλά τεμάχια
  - Το μερικό ACK επιβεβαιώνει μερικά, αλλά όχι όλα τα τεμάχια που εκκρεμούν κατά την αρχή της ταχείας ανάκαμψης
  - Ο αποστολέας θα βγει από την φάση της ταχείας ανάκαμψης όταν επέλθει η εκπνοή χρόνου
- Το new 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 new 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 SACK:** 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

# Τι μπορεί να κάνουν 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., RED (Floyd and Jacobson 1993)
    - Random Early Detection (RED) απορρίπτει πακέτα στην τύχη ως συνάρτηση του βαθμού συμφόρησης



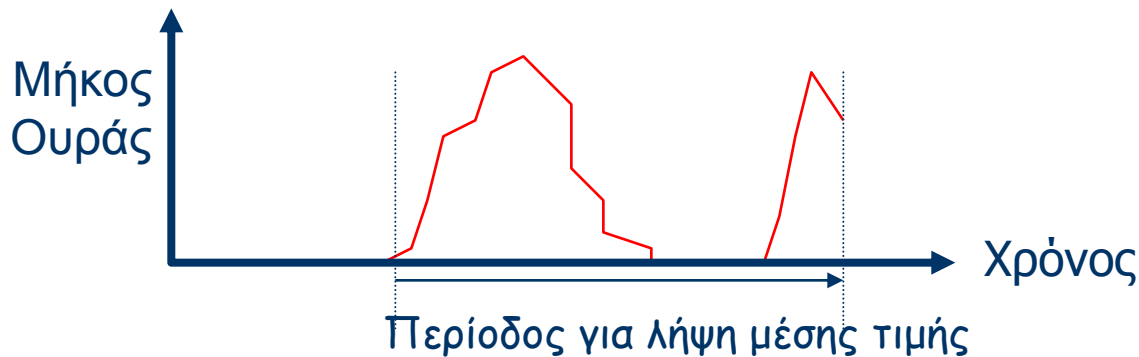
# DECbit

- 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



# DECbit

- 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, όταν γεμίσει
  - Απόρριψη προτού γεμίσει η ουρά
- Έμμεση ειδοποίηση του αποστολέα να μειώσει τον ρυθμό του απορρίπτοντας πακέτα στην τύχη σε περίπτωση συμφόρησης





# Τυχαία απόρριψη

- 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 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 απόρριψης υπολογίζεται ως

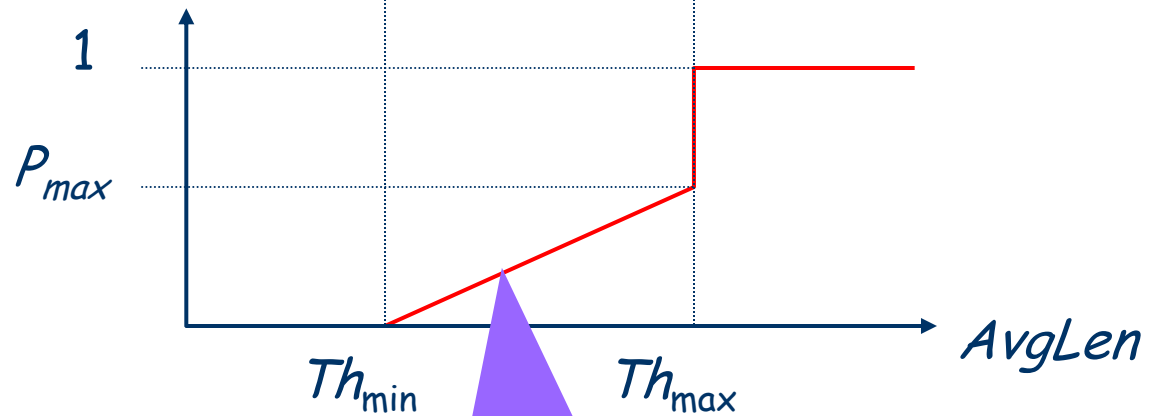
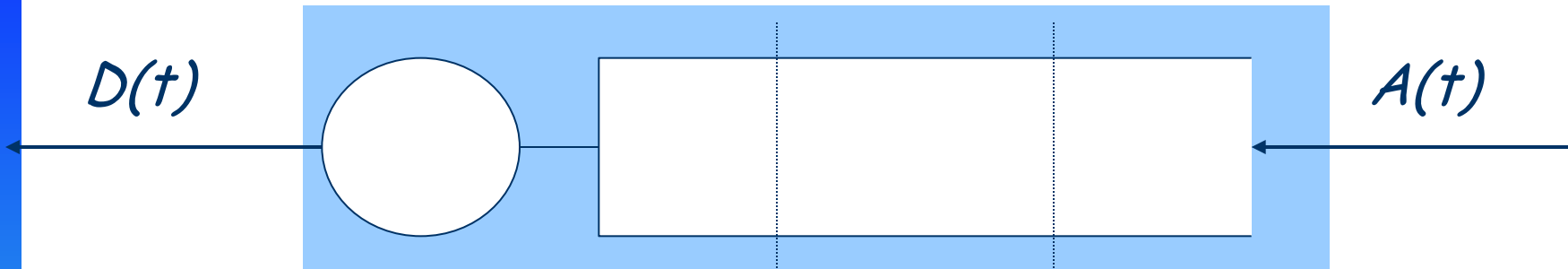
$$\rho = \frac{\hat{\rho}}{1 - \text{count} \times \hat{\rho}} \quad \hat{\rho} = \rho_{\max} \left\{ \frac{\text{AvgLen} - Th_{\min}}{Th_{\max} - Th_{\min}} \right\}$$

όπου η μεταβλητή *count* μετρά πόσες φορές το *AvgLen* βρέθηκε στο διάστημα  $Th_{\min}$  έως  $Th_{\max}$  μετά την τελευταία απόρριψη πακέτου

- Η πιθανότητα  $\hat{\rho}$  αυξάνει γραμμικά ως προς το *AvgLen* μέχρι τη μέγιστη τιμή  $\rho_{\max}$  και αντιστοιχεί στην πιθανότητα απόρριψης του πρώτου πακέτου
- Όσο διαρκεί η συμφόρηση η πιθανότητα απόρριψης αυξάνει, με αποτέλεσμα οι απορρίψεις περίπου να ισαπέχουν και έτσι μειώνεται η πιθανότητα επανεισόδου στην αργή αρχή



# Πιθανότητα απόρριψης



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

$$P = \frac{\hat{p}}{1 - count \times \hat{p}}$$

# Ιδιότητες RED

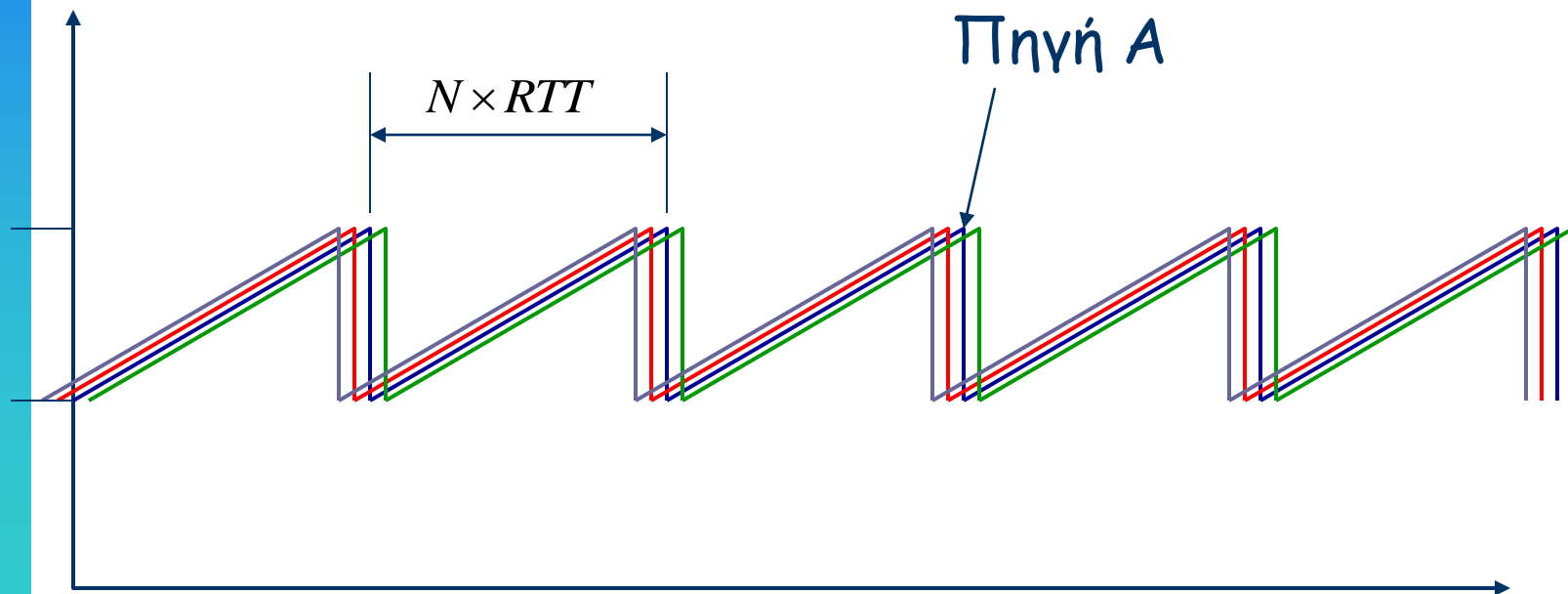
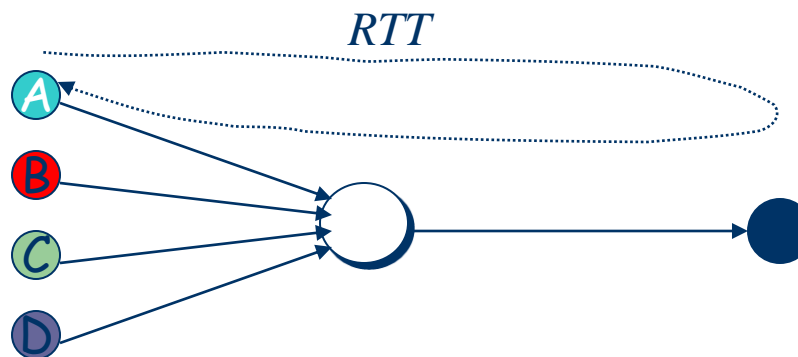


- Απορρίπτει πακέτα προτού γεμίσει η ουρά, ελπίζοντας ότι θα μειωθεί ο ρυθμός μερικών ροών
- Οι απορρίψεις πακέτων για κάθε ροή είναι περίπου ανάλογες με τον ρυθμό της
- Οι απορρίψεις απέχουν στον χρόνο
- Επειδή χρησιμοποιεί γεωμετρικό μέσο του μήκους ουράς, η RED είναι ανεκτική στις ριπές
- Οι τυχαίες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



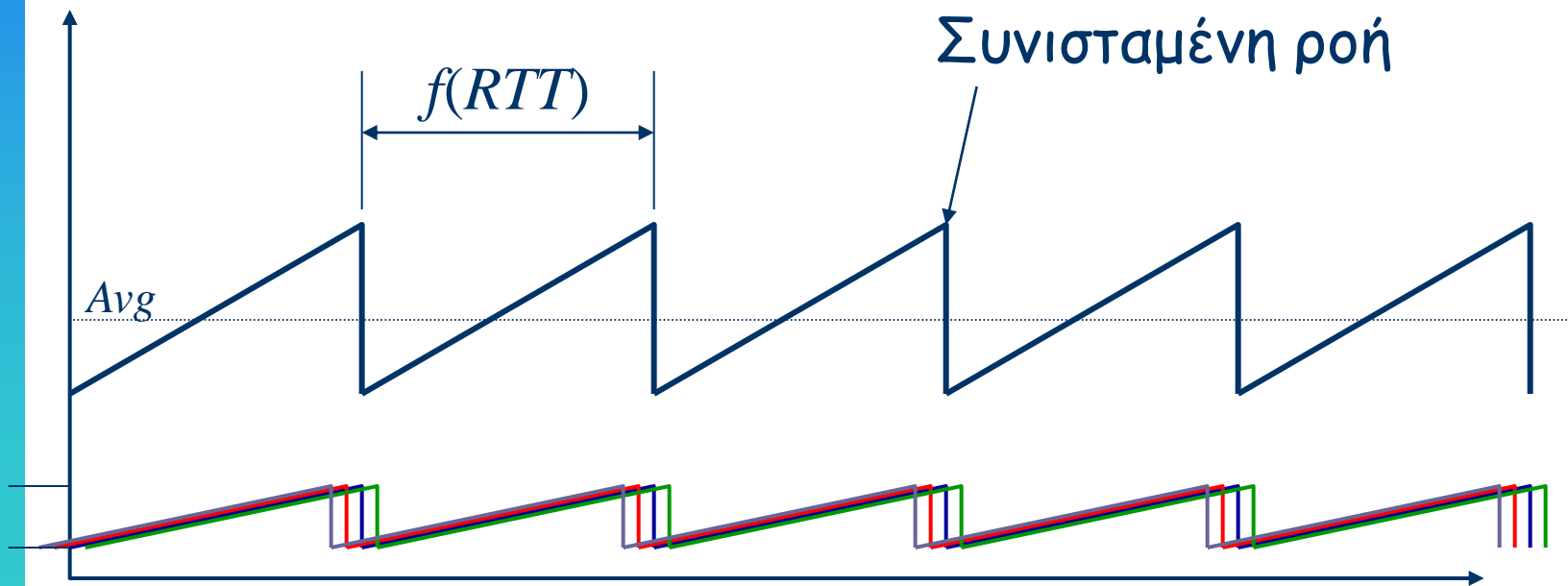
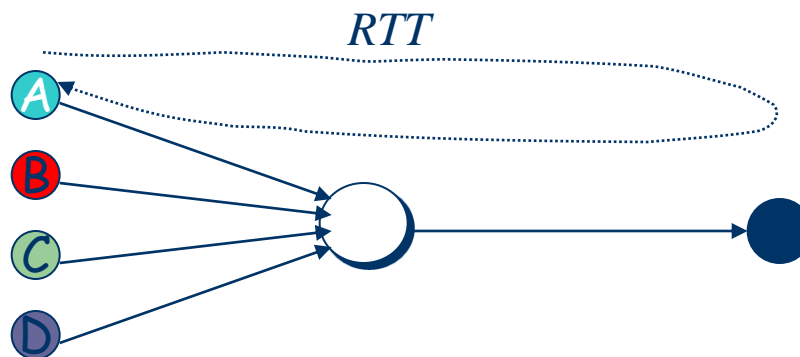


# Συγχρονισμός πηγών

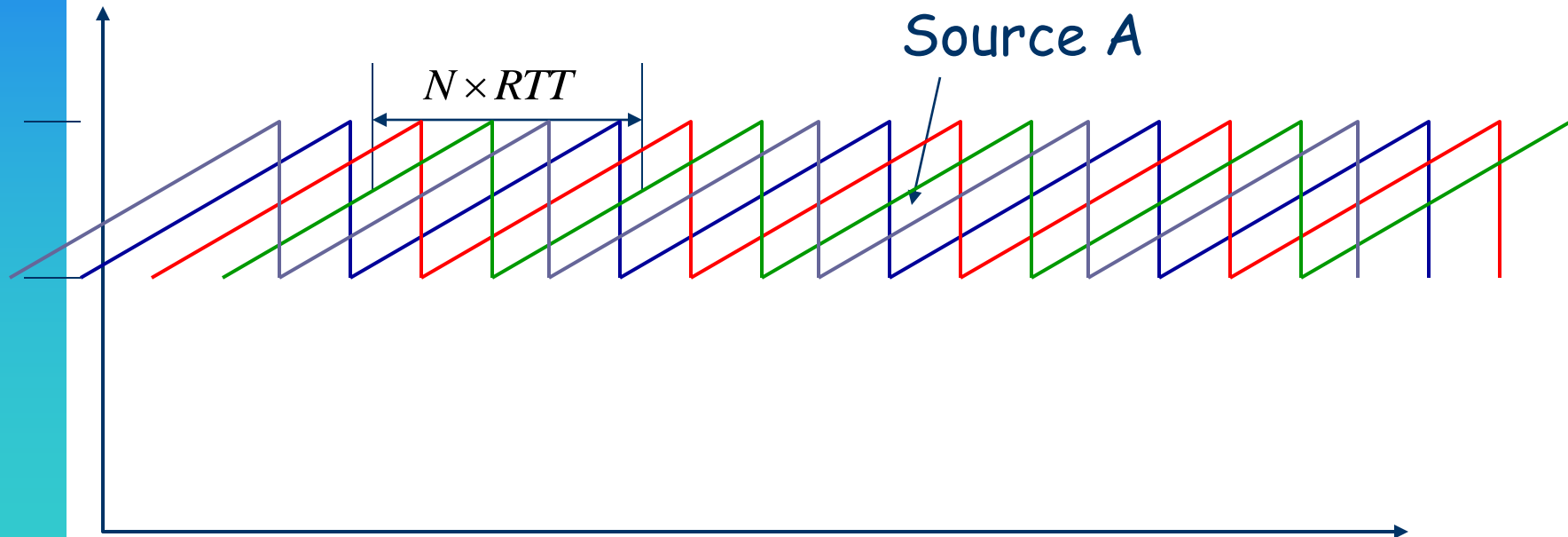
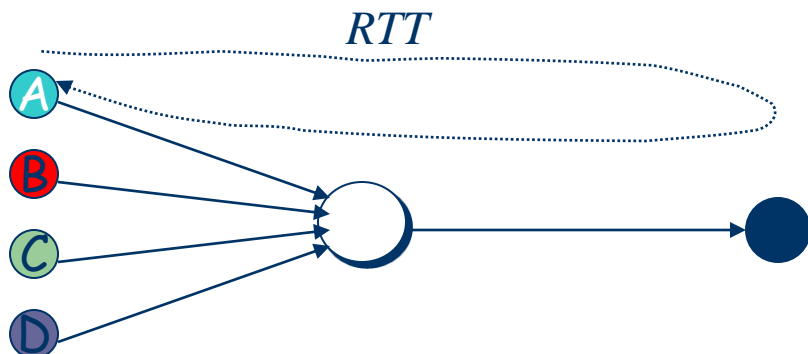




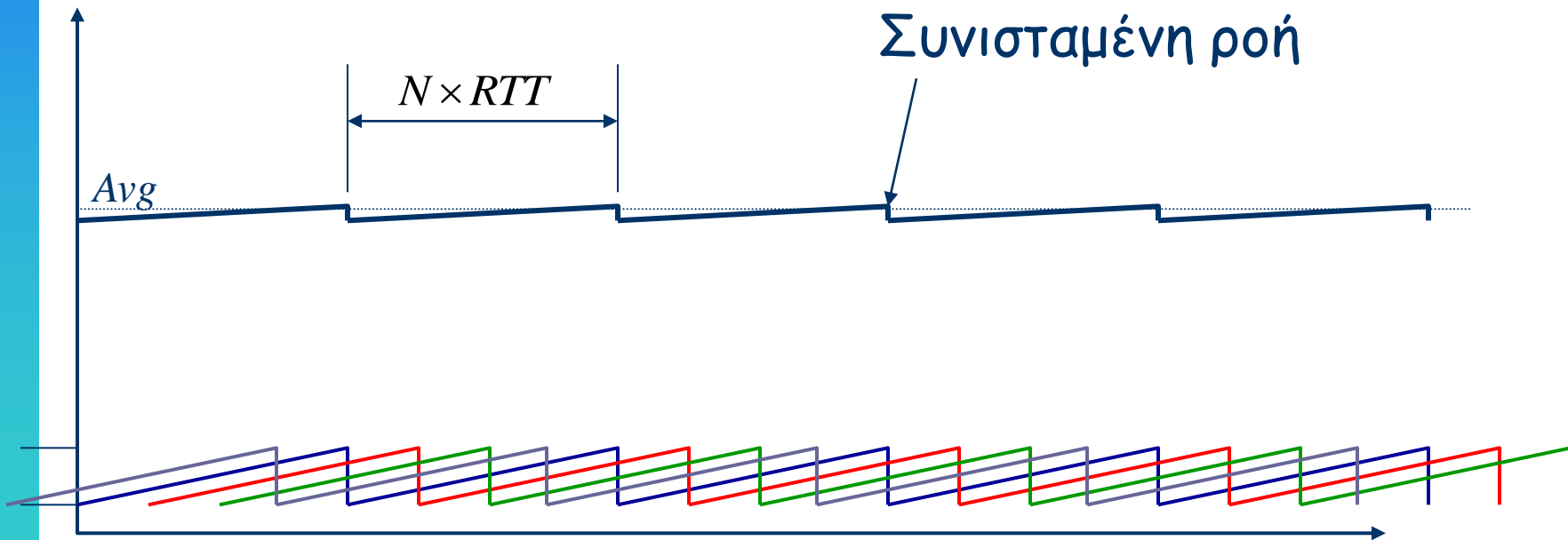
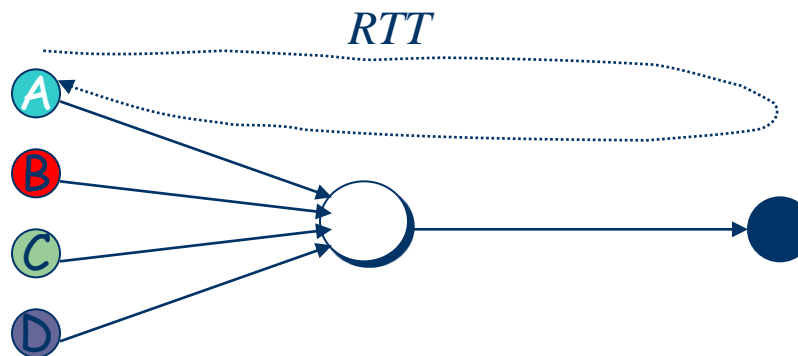
# Συγχρονισμός πηγών



# Αποσυγχρονισμός πηγών



# Αποσυγχρονισμός πηγών



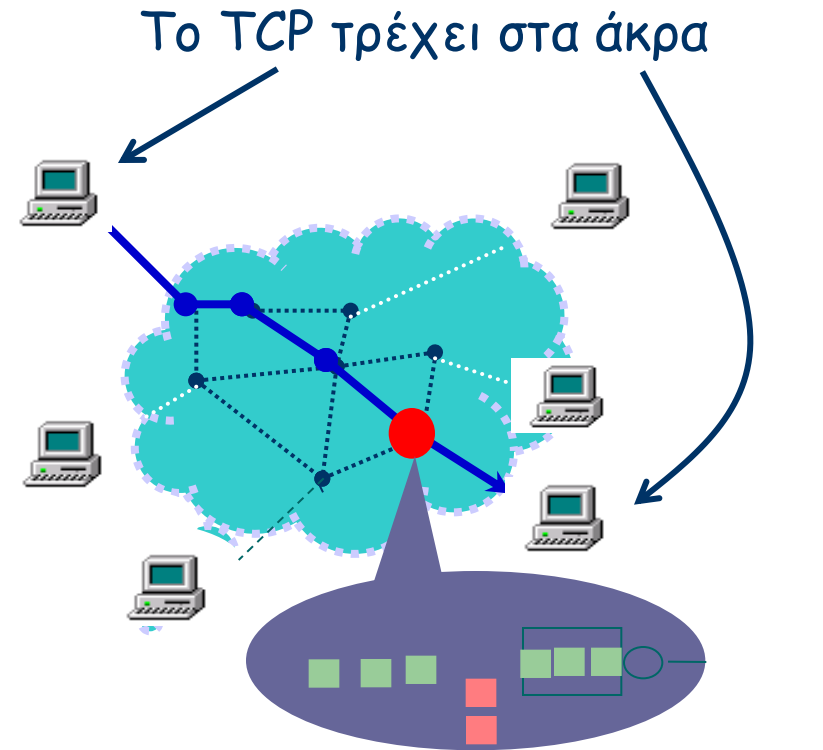


## Επίδοση TCP

# Συμπεριφορά του TCP



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





# 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 = 
$$\frac{1.22 \cdot MSS}{RTT \sqrt{p}}$$
- όπου  $p$  = η πιθανότητα απώλειας



# Συμπεριφορά του TCP

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



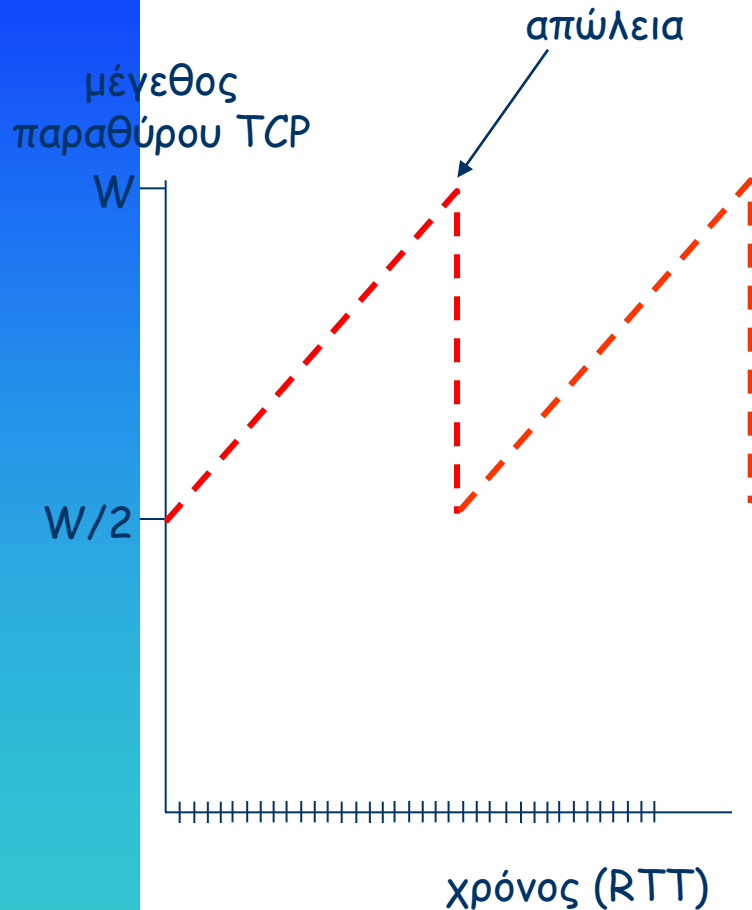


# Συμπεριφορά του TCP

- Μείωση του παραθύρου στο μισό με την ανίχνευση απώλειας (τριπλό ταυτόσημο 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



# Σχέση διέλευσης/απωλειών στο TCP

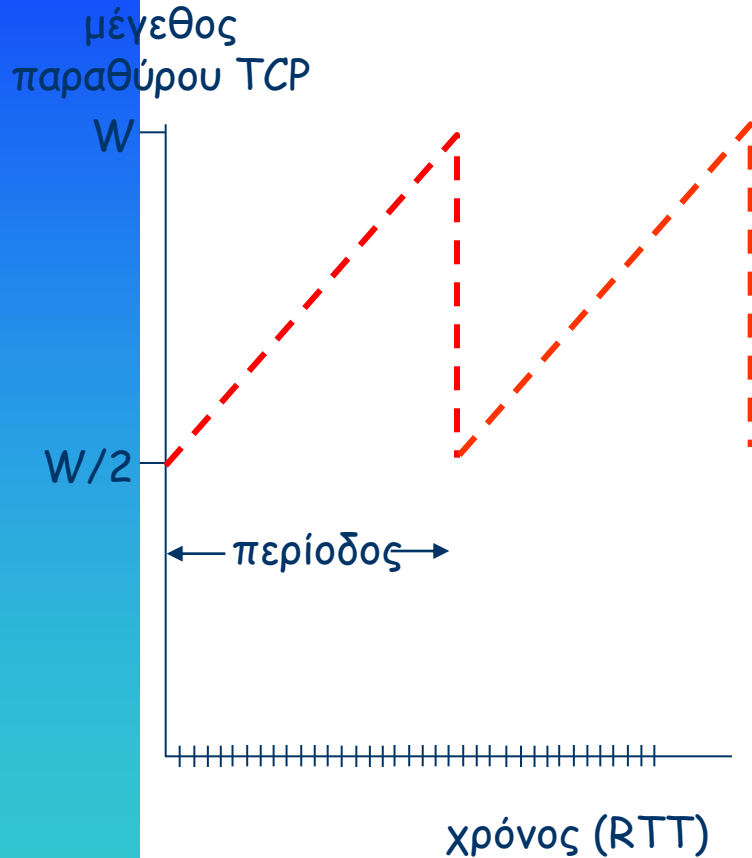


## Εξιδανικευμένο μοντέλο:

- 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



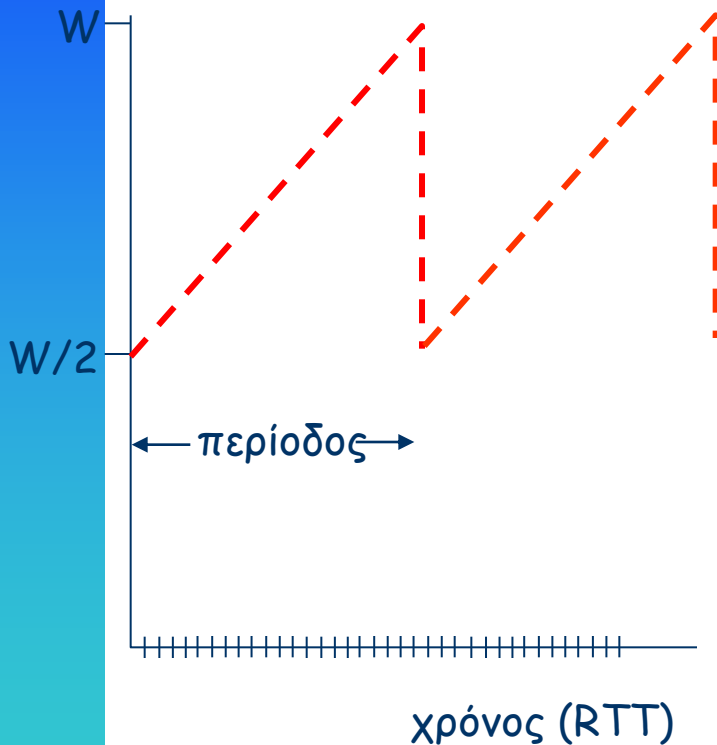
αριθμός τεμαχίων ανά “περίοδο” =

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



# Σχέση διέλευσης/απωλειών στο TCP

μέγεθος  
παραθύρου TCP



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

1 απώλεια ανά “περίοδο” σημαίνει:

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

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

$$T = \text{διέλευση} = \frac{1.22}{\sqrt{p}} \frac{\text{τεμάχια}}{\text{RTT}}$$

Η σχέση για τη διέλευση  $T$  είναι δυνατό να επεκταθεί ώστε να περιλάβει εκπνοές και την αργή αρχή



# Ένα καλύτερο μοντέλο

Εάν ληφθεί υπόψη και η περίπτωση απώλειας ACK (εκπνοή χρόνου)

$$T = \frac{s}{RTT \sqrt{\frac{2p}{3}} + RTO \left( 3 \sqrt{\frac{2p}{3}} \right) p (1 + 32p^2)}$$

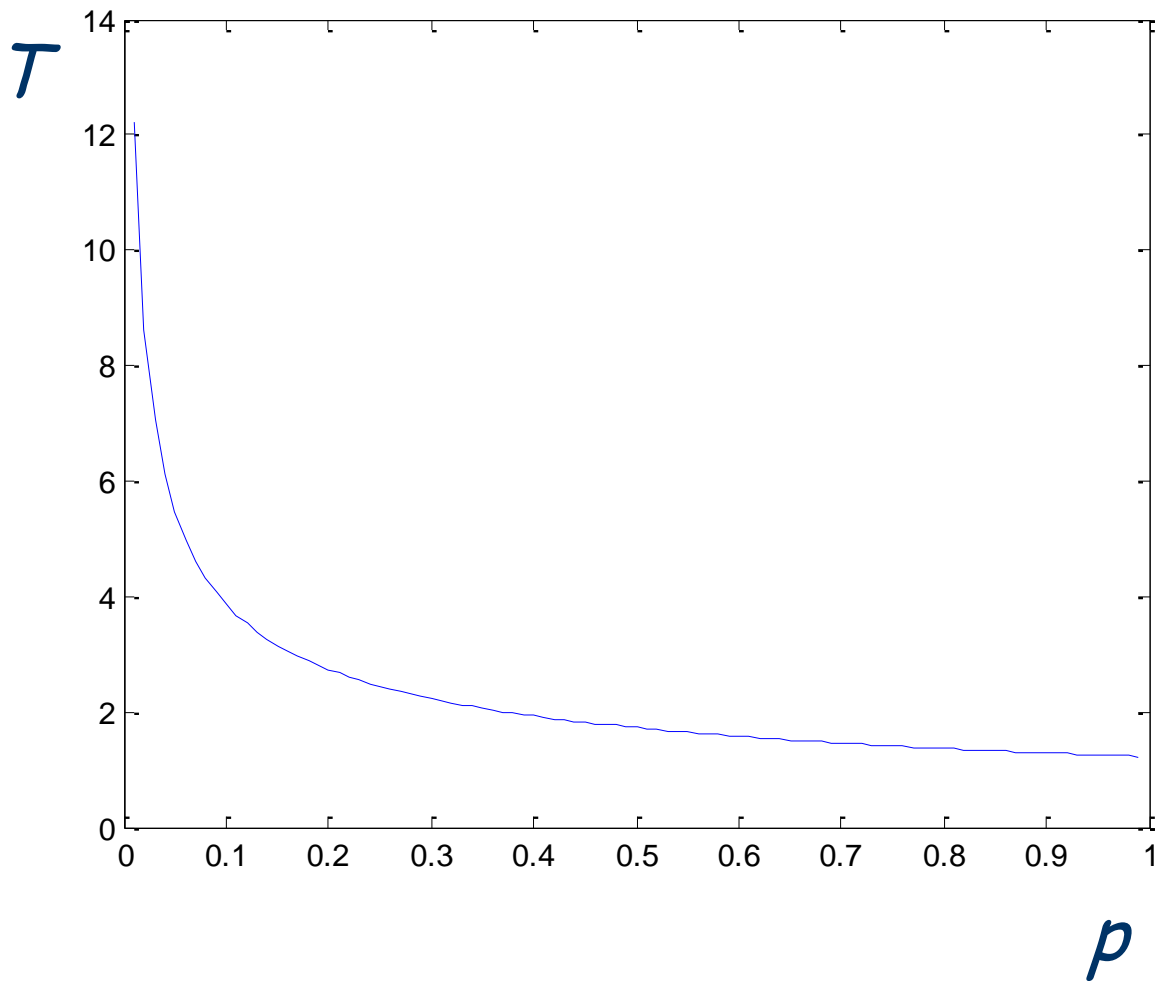
$T$  = η διέλευση σε byte/δευτερόλεπτο

$RTT$  = Round Trip Time

$p$  = πιθανότητα απώλειας

$RTO$  = Retransmission Timeout

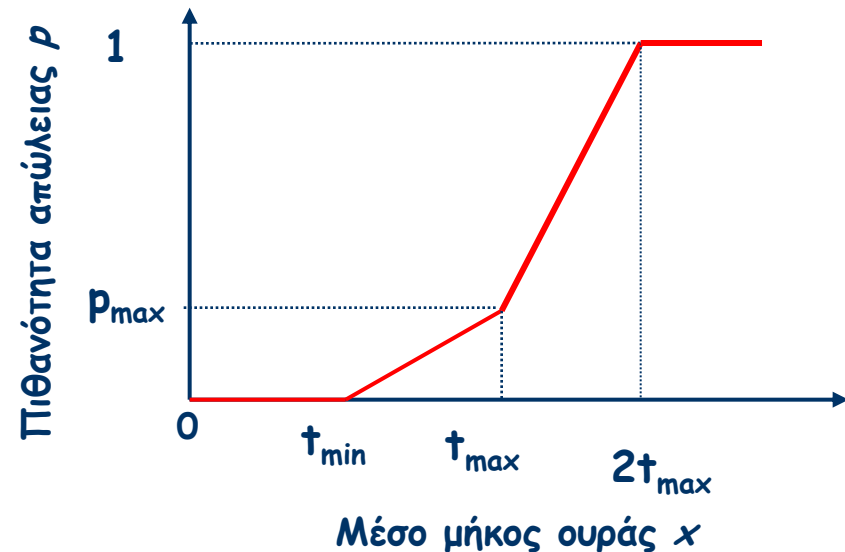
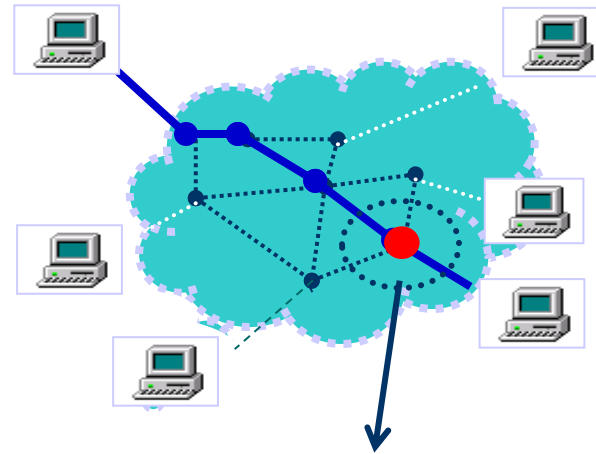
$s$  = μέγεθος τεμαχίου σε byte



# Διαχείριση ουρών RED

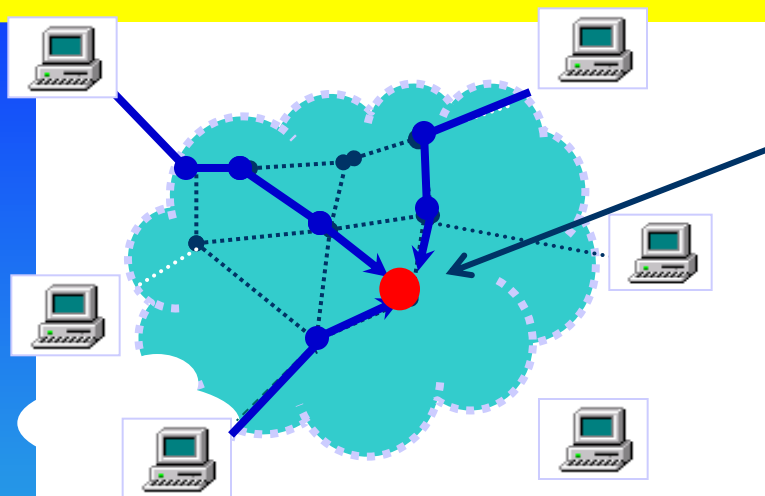


- Η απώλεια είναι συνάρτηση του μέσου μήκους ουράς  $\rightarrow p = p(x)$





# 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;

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

$C$  = χωρητικότητα δρομολογητή

$T_i$  = διέλευση ροής  $i$

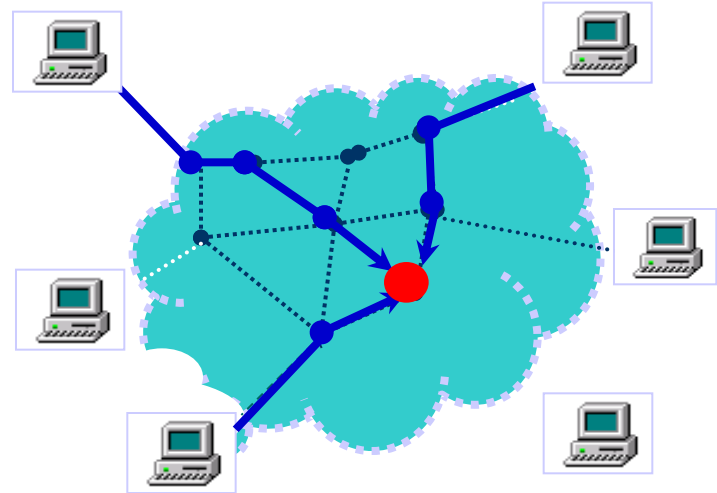


# Μοναδικό σημείο συμφόρησης, άπειρης διάρκειας ροές



- $N$  ροές TCP άπειρης διάρκειας

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



- Ένας δρομολογητής με συμφόρηση

- Ουρές RED

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

- ζητείται

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



# Μοντέλο και λύση

- Μοντέλο

$$p = p(x)$$

$$RTT_j = A_j + x/C$$

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

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