

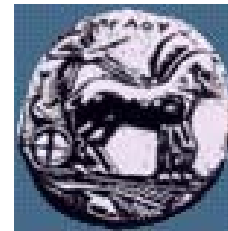
# Δίκτυα Επικοινωνιών II: Transmission Control Protocol

Δρ. Απόστολος Γκάμας

Διδάσκων 407/80

gkamas@uop.gr

# Μηχανισμοί πρωτοκόλλων προσανατολισμένων σε σύνδεση



- Λογική σύνδεση
- Εγκαθίδρυση
- Διατήρηση σύνδεση
- Αξιοπιστία
  
- Χαρακτηριστικό παράδειγμα: TCP

# Αξιόπιστες αιολουθιακές δικτυακές υπηρεσίες



- Υποθέτουν τυχαίο μέγεθος μηνυμάτων
- Υποθέτουν σχεδόν 100% αξιόπιστη μετάδοση πάνω από το δίκτυο
  - πχ αξιόπιστο δίκτυο μεταγωγής πακέτων πάνω από X.25
  - πχ frame relay με χρήση LAPF πρωτοκόλλου
  - πχ IEEE 802.3 με χρήση LLC υπηρεσίας προσανατολισμένη σε σύνδεση
- Μια υπηρεσία στο επίπεδο μεταφοράς είναι ένα πρωτόκολλο από άκρο σε άκρο ανάμεσα σε δύο συστήματα σε ίδιο δίκτυο

# Θέματα σε ένα απλό πρωτόκολλο μεταφοράς



- Διευθυνσιοδότηση
- Πολύπλεξη
- Έλεγχος ροής
- Εγκαθίδρυση σύνδεσης και τερματισμός σύνδεσης



# Διευθυνσιοδότηση

- Προσδιορισμός χρήστη User identification
  - Συνήθως σταθμός και θύρα (socket στην ορολογία του TCP)
  - Η θύρα αντιπροσωπεύει μια συγκεκριμένη υπηρεσίας μεταφοράς
- Οντότητα στο επίπεδο μεταφοράς
  - Ορισμός πρωτοκόλλου μεταφοράς (πχ TCP, UDP, ...)
- Διεύθυνσης σταθμού
  - Αποδίδεται σε κάθε δικτυακή συσκευή
  - Στο Διαδίκτυο μια μοναδική διεύθυνση



# Εύρεση διεύθυνσης

- Ήδη γνωστή διεύθυνση
- Well know διεύθυνση
- Χρήση Name Server



## Πολύπλεξη

- Πολλοί χρήστες χρησιμοποιούν το ίδιο πρωτόκολλο μεταφοράς
- Οι χρήστες προσδιορίζονται από τον αριθμό θύρας ή service access point (SAP)



# Έλεγχος ροής

- Μεγαλύτερη καθυστέρηση μετάδοσης σε σύγκριση με τον πραγματικό χρόνο μετάδοσης
  - Καθυστέρηση στην επικοινωνία των πληροφοριών ελέγχου ροής (επιπλέον της καθυστέρησης μετάδοσης δεδομένων)
- Μεταβλητή καθυστέρηση μετάδοσης
  - Δυσκολία στην χρήση των timeouts
- Ο έλεγχος ροής χρησιμοποιείται:
  - Όταν ο παραλήπτης δεν μπορεί να διατηρήσει το ρυθμό λήψης δεδομένων
  - Όταν ο αποστολέας δεν μπορεί να διατηρήσει το ρυθμό μετάδοσης δεδομένων
- Το αποτέλεσμα είναι η πλήρωση του buffer





# Πολιτικές ελέγχου ροής (1)

- Δεν χρησιμοποιείται έλεγχος ροής
  - Δεδομένα υπερχειλίζουν στο buffer και απορρίπτονται
  - Ο αποστολέας δεν θα λαμβάνει επιβεβαιώσεις (ACK) και θα ξανά-μεταδίδει τα χαμένα δεδομένων
    - Αυτό θα δημιουργήσει περισσότερα εισερχόμενα δεδομένα στον παραλήπτη
- Απορρίπτονται τα επιπλέον πακέτα δεδομένων
  - Δεν είναι επιθυμητό
  - Οι συνδέσεις που περιπλέκονται αντιμετωπίζονται ως μια ροή δεδομένων (ίσως άδικο για κάποιες ροές δεδομένων)



## Πολιτικές ελέγχου ροής (2)

- Χρησιμοποίηση πρωτοκόλλου κυλιόμενου παραθύρου (sliding window protocol)
  - Λειτουργεί αποτελεσματικά για αξιόπιστα δίκτυα
    - Αποτυχία στην λήψη επιβεβαίωσης ενεργοποιεί την διαδικασία ελέγχου ροής
  - Δεν λειτουργεί αποτελεσματικά σε μη αξιόπιστα δίκτυα
    - Δεν μπορεί να διακρίνει ξεκάθαρα την απώλεια πακέτων και την διαδικασία ελέγχου ροής
- Χρήση συστήματος βασισμένο σε credits



## Έλεγχος ροής με χρήση credits

- Μεγαλύτερος έλεγχος σε αξιόπιστα δίκτυα
- Περισσότερο αποτελεσματικός σε μη αξιόπιστα δίκτυα
- Ανεξαρτητοποίηση του ελέγχου ροής από τις επιβεβαιώσεις λήψης (ACK) των πακέτων
  - Η μετάδοση επιβεβαίωσης λήψης μπορεί να συνεχίζεται χωρίς να σχετίζονται με τα credits
- Κάθε οκτάδα (octet) έχει αριθμό ακολουθίας
- Κάθε πακέτο έχει αριθμό ακολουθίας, αριθμό επιβεβαίωσης και μέγεθος παραθύρου στην επικεφαλίδα

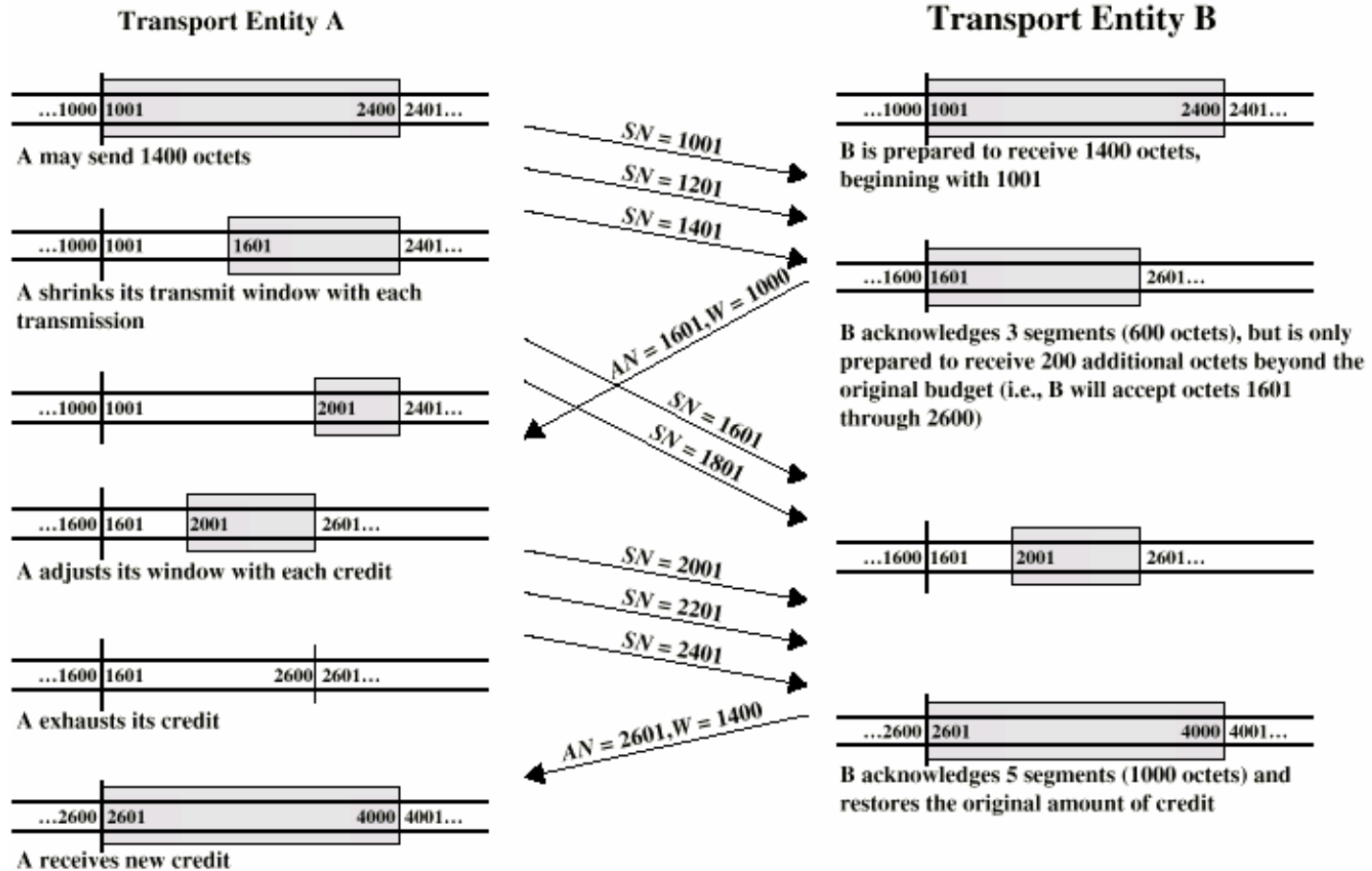


## Χρήση της επιεφαλίδας του πακέτου

- Όταν μεταδίδεται ένα πακέτο, αριθμός ακολουθίας προστίθεται ίσος με τον αριθμό ακολουθίας του πρώτου octet του πακέτου
- Η επιβεβαίωση (ACK) ενός πακέτου περιλαμβάνει  $AN=i$ ,  $W=j$
- Όλα τα δεδομένα μέχρι το  $SN=i-1$  έχουν ληφθεί σωστά
  - Αναμονή του octet  $i$
- Ο Αποστολέας έχει το δικαίωμα να μεταδώσει επιπλέον δεδομένα όσο αυτό ορίζεται από το  $W=j$ 
  - Για παράδειγμα τα octet ως το  $i+j-1$

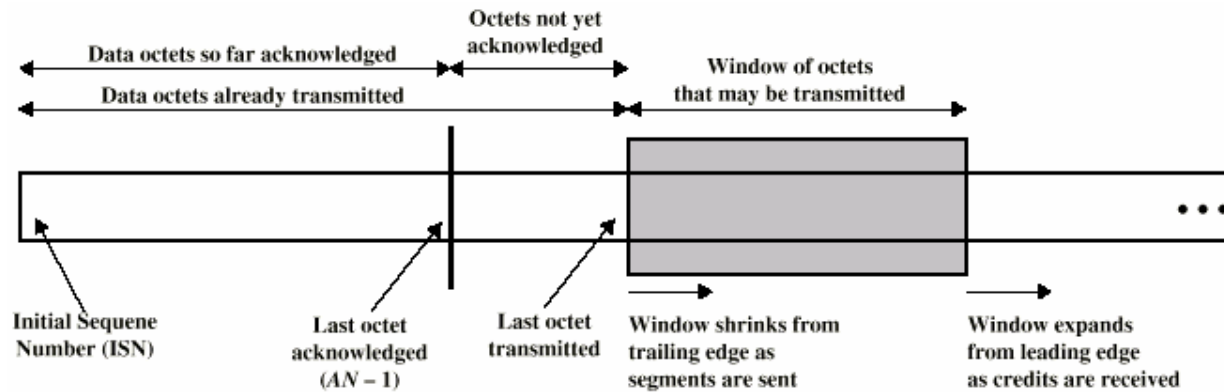


# Χρήση των credits

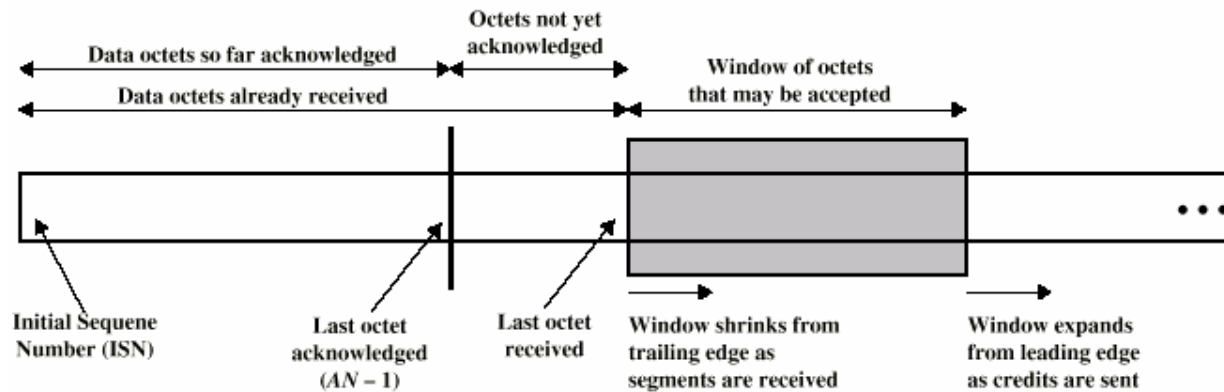




# Αποστολή και λήψη δεδομένων



(a) Send sequence space

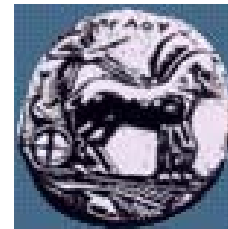


(b) Receive sequence space

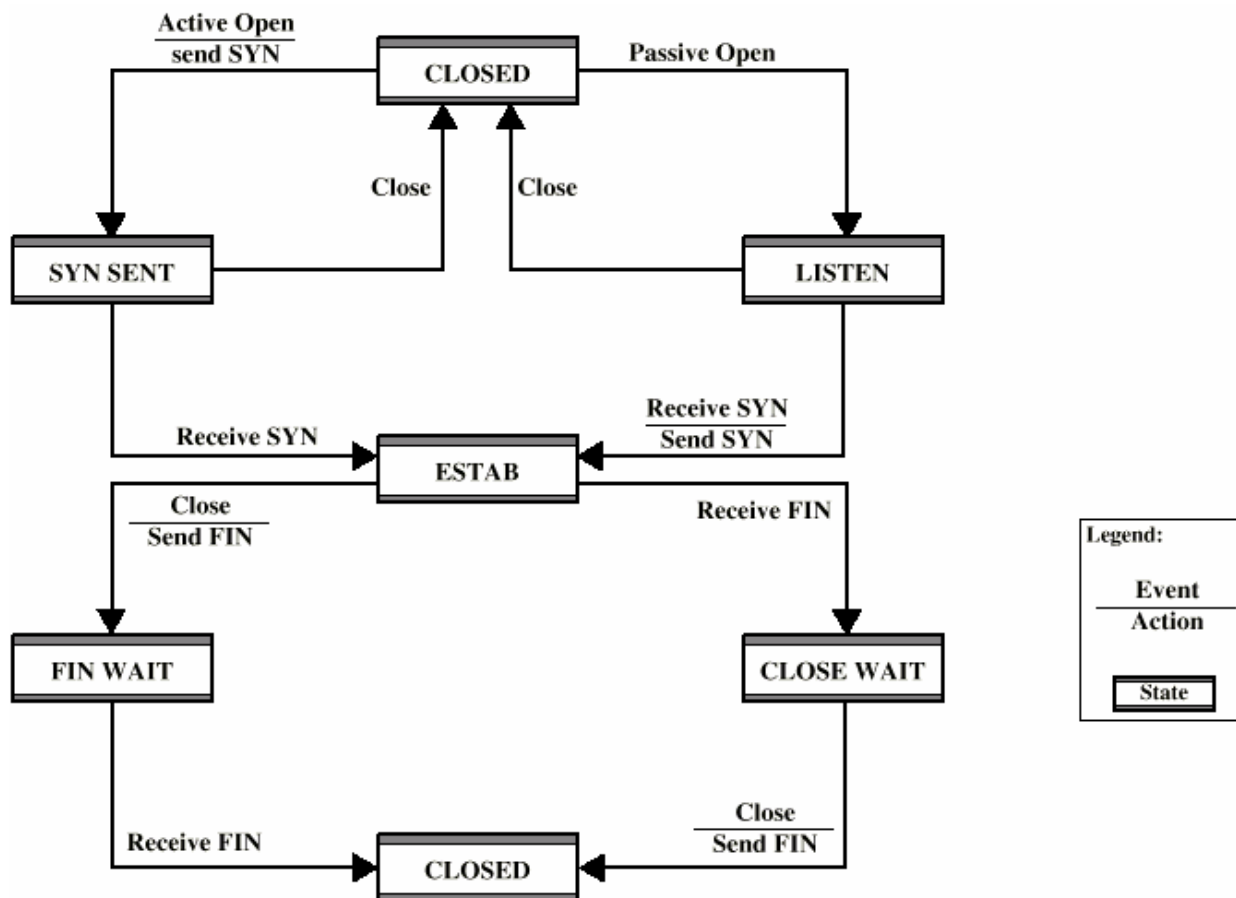


## Εδραίωση και τερματισμός σύνδεσης

- Επιτρέπει στα δύο άκρα που επιθυμούν να επικοινωνήσουν να γνωρίζουν την ύπαρξη τους
- Διαπραγμάτευση των παραμέτρων σύνδεσης
- Δέσμευση των πόρων στο επίπεδο μεταφοράς



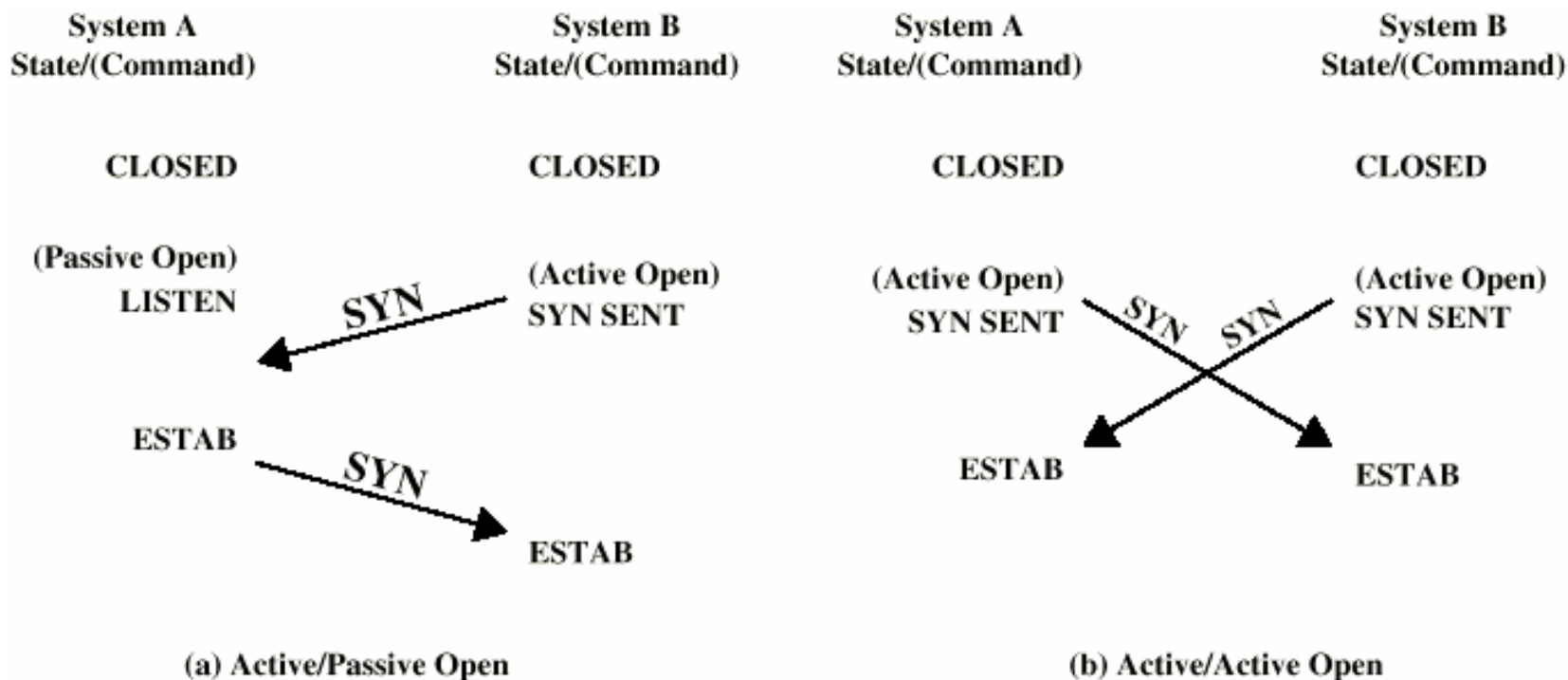
# Διάγραμμα κατάστασης σύνδεσης







# Εδραίωση σύνδεσης





## Τερματισμός σύνδεσης

- Μπορεί να κάνει τερματισμό σύνδεση οποιοδήποτε από τα δύο μέρη
- Μετά από συνεννόηση
- Αιφνίδιος τερματισμός
- Ομαλός τερματισμός
  - Αναμονή λήψης όλων των εισερχόμενων δεδομένων μέχρι την λήψη του μηνύματος FIN

# Τερματισμός: Άκρο το οποίο αρχικοποιεί τον τερματισμό



- Ο χρήστης επιθυμεί να τερματίσει την σύνδεση
- Ο μηχανισμός στο επίπεδο μεταφοράς στέλνει το μήνυμα FIN και ζητά τερματισμό της σύνδεσης
- Η σύνδεση τίθεται σε κατάσταση αναμονής FIN WAIT
  - Συνεχίζει να λαμβάνει δεδομένα
  - Σταματά να στέλνει δεδομένα
- Όταν το μήνυμα FIN ληφθεί ενημερώνει τον χρήστη και τερματίζει την σύνδεση

# Τερματισμός: Άκρο το οποίο δεν αρχικοποιεί τον τερματισμό



- Λήψη του μηνύματος FIN
- Ενημερώνει τον χρήστη και θέτει την σύνδεση σε κατάσταση CLOSE WAIT
  - Συνεχίζει να μεταδίδει δεδομένα
- Ο χρήστης επιβεβαιώνει τον τερματισμό της σύνδεσης
- Ο μηχανισμός στο επίπεδο μεταφοράς στέλνει το μήνυμα FIN
- Η σύνδεση τερματίζεται
- Όλα τα ειδικά δεδομένα αποστέλλονται και από τις 2 πλευρές



## Μη αξιόπιστες δικτυακές υπηρεσίες

- Π.χ.
  - Διαδίκτυο με χρήση IP,
  - frame relay με χρήση LARF
  - IEEE 802.3 με χρήση LLC χωρίς επιβεβαιώσεις
- Πακέτα δεδομένων μπορεί να χαθούν
- Πακέτα δεδομένων μπορεί να φτάσουν σε λάθος σειρά



# Προβλήματα

- Μετάδοση στην σωστή σειρά
- Στρατηγικές επανα-μετάδοσης
- Ανίχνευση διπλών μεταδόσεων
- Έλεγχος ροής
- Εδραίωση σύνδεσης
- Τερματισμός σύνδεσης
- Ανάκαμψη από πρόβλημα



## Μετάδοση στην σωστή σειρά

- Πακέτα μπορεί να φτάσουν σε λάθος σειρά
- Τα πακέτα αριθμούνται σειριακά
- Το TCP αριθμοί κάθε octet ακολουθιακά
- Τα πακέτα αριθμούνται με τον αριθμό του πρώτου octet στο πακέτο



## Στρατηγικές επανα-μετάδοσης

- Πακέτα μπορεί να καταστραφούν στην μετάδοση
- Πακέτα μπορεί να αποτύχουν να φτάσουν στον προορισμό
- Ο αποστολέας δεν γνωρίζει ότι ένα πακέτο απέτυχε να μεταδοθεί
- Ο παραλήπτης θα πρέπει να επιβεβαιώνει την επιτυχημένη λήψη πακέτων
- Χρησιμοποιείται αθροιστική επιβεβαίωση
- Υπάρχει ένα περιθώριο αναμονής (timeout) μέχρι να ληφθεί η επιβεβαίωση (ACK) διαφορετικά γίνεται επανα-μετάδοση





# Η χρήση timeout

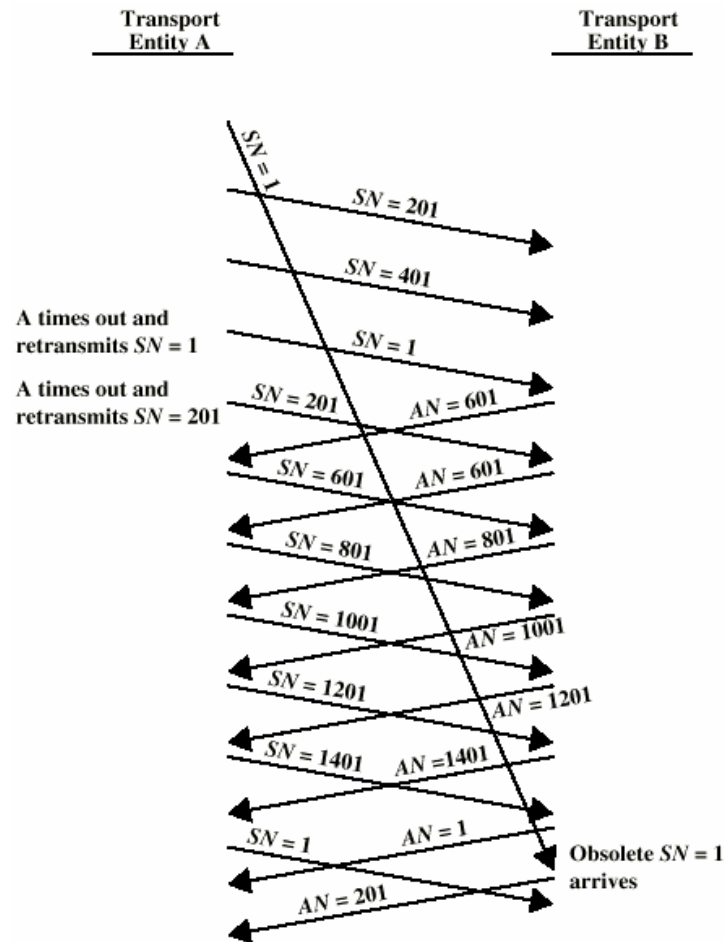
- Σταθερό timeout
  - Βασίζεται στην κατανόηση της συμπεριφοράς του δικτύου
  - Δεν μπορεί να προσαρμοστεί στις αλλαγές του δικτύου
  - Εάν είναι πολύ μικρό οδηγεί σε μη απαραίτητες επανα-μεταδόσεις
  - Εάν είναι πολύ μεγάλο οδηγεί καθυστέρηση ανίχνευσης χαμένων πακέτων
  - Πρέπει να είναι λίγο μεγαλύτερο από το round trip time
- Προσαρμογή του timeout
  - Μπορεί να μην στέλνεται επιβεβαίωση ACK αμέσως
  - Δεν μπορεί να διαχωρίσει επιβεβαιώσεις ACK ανάμεσα στα αρχικά πακέτα και πακέτα τα οποία επανα-μεταδόθηκαν
  - Οι συνθήκες μπορεί να αλλάξουν δραματικά



## Ανίχνευση διπλών μεταδόσεων

- Ένα η επιβεβαίωση ACK ενός πακέτου χαθεί, τότε το πακέτο επανα-μεταδίδεται
- Ο παραλήπτης θα πρέπει να αναγνωρίζει τις επανα-μεταδόσεις
- Διπλές μεταδόσεις πριν το κλείσιμο της σύνδεσης
  - Ο παραλήπτης θεωρεί πως η επιβεβαίωση ACK χάθηκε και επανα-μεταδίδει την επιβεβαίωση ACKs
  - Ο αποστολέας δεν θα πρέπει να μπερδεύεται με πολλές επιβεβαιώσεις
  - Ο αριθμός ακολουθίας θα πρέπει είναι αρκετά μεγάλος ώστε να μην εξαντλείται στην «μέγιστη διάρκεια ζωής» ενός πακέτου
- Διπλές μεταδόσεις μετά το κλείσιμο της σύνδεσης

# Λανθασμένη ανίχνευση διπλών μεταδόσεων





## Έλεγχος ροής

- Διανομή Credits
- Πρόβλημα αν  $AN=i$ ,  $W=0$  στο closing window
- Μετάδοση  $AN=i$ ,  $W=j$  για επανεκκίνηση, αλλά αυτό χάνεται
- Ο αποστολέας πιστεύει ότι το window είναι κλειστό, ενώ ο παραλήπτης πιστεύει ότι το window είναι ανοικτό
- Χρήση του window timeout
- Εάν περάσει το timeout τότε μετέδωσε
  - Πχ επανα-μετάδοσης του προηγούμενου πακέτου

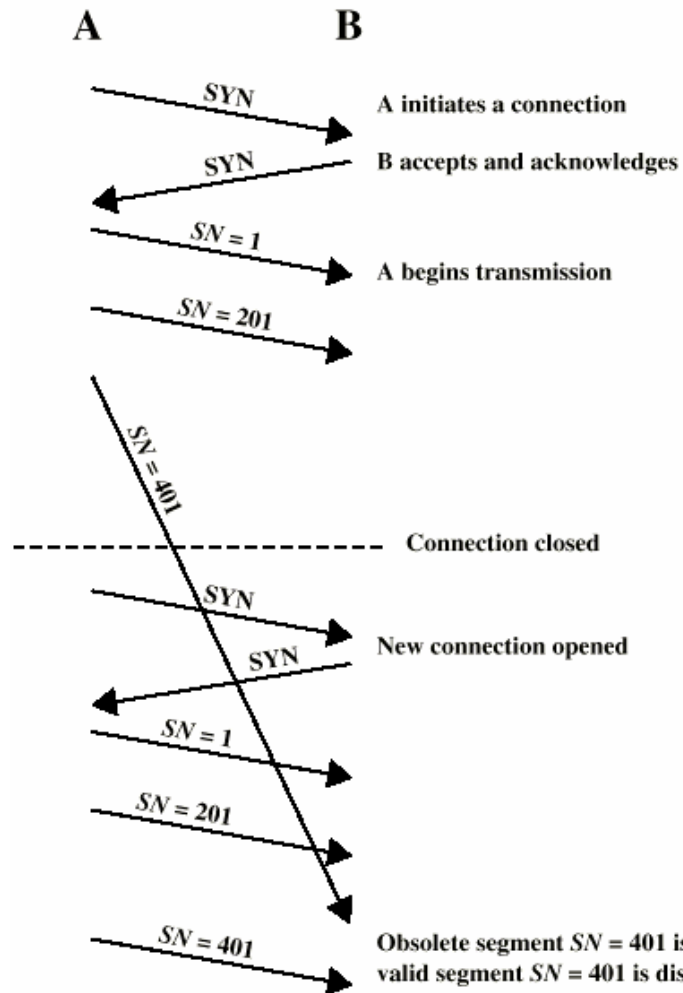


## Εγκαθίδρυση σύνδεσης

- Επικοινωνία και προς τις δύο κατευθύνσεις
  - Το ένα άκρο στέλνει το μήνυμα SYN, και το άλλο άκρο απαντά με το μήνυμα SYN
  - Σε περίπτωση απώλειας του μηνύματος SYN υπάρχει επαναμετάδοση
    - Μπορεί να οδηγήσει σε διπλή μετάδοση μηνυμάτων SYNs
  - Τα επιπλέον μηνύματα SYN αγνοούνται μετά την εγκαθίδρυση σύνδεσης
- Χαμένα ή καθυστερημένα πακέτα δημιουργούν προβλήματα σύνδεσης
  - Πακέτα από παλιές συνδέσεις
  - Αρχικοποίηση αριθμών ακολουθίας σε σχέση με προηγούμενες συνδέσεις (χρήση SYNi, και three way Εγκαθίδρυση σύνδεσης)

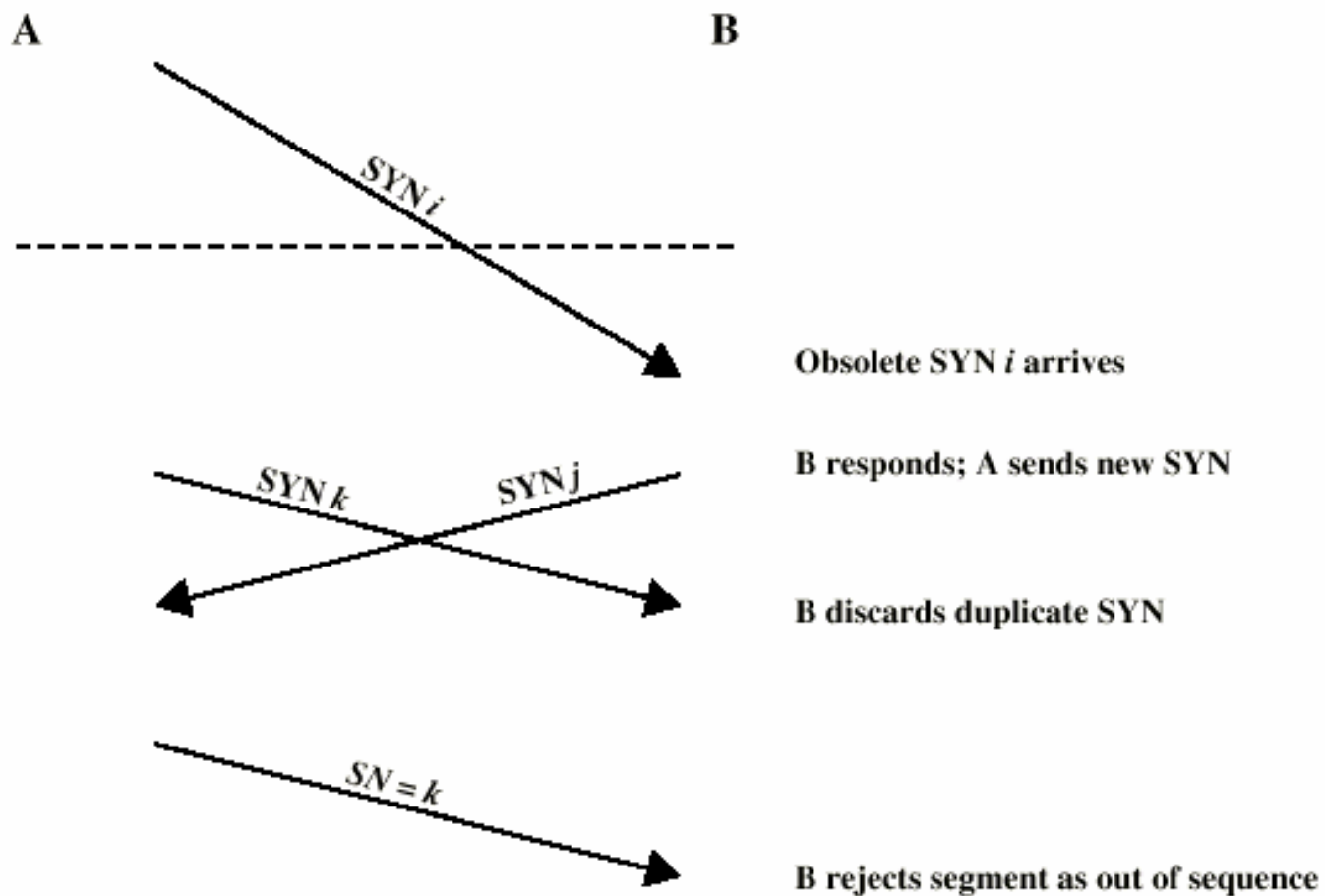


# Απώλεια πακέτου



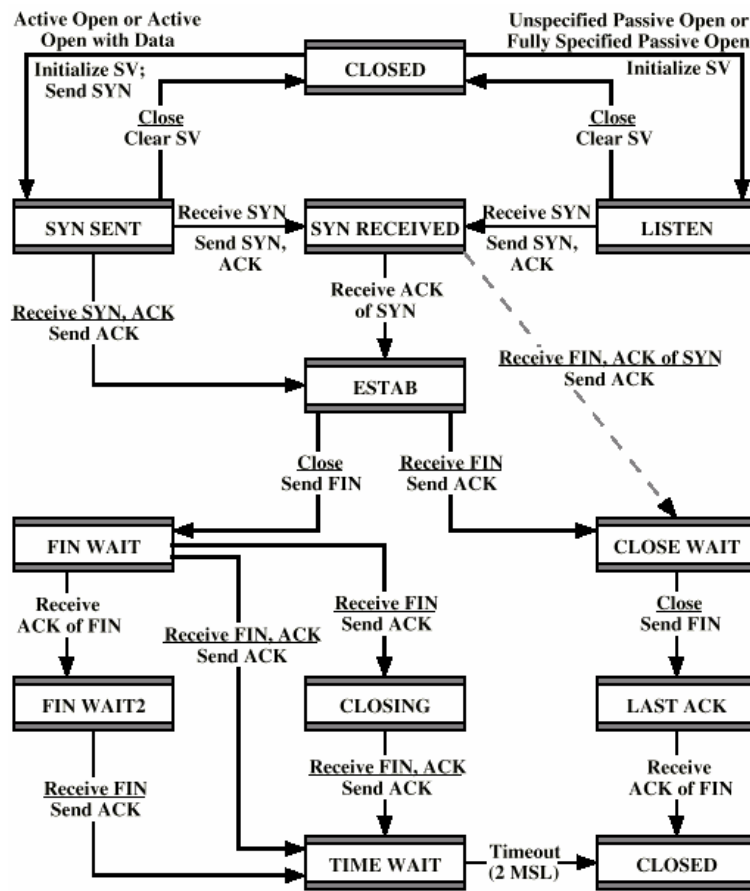


# Απώλεια SYN πακέτου





# Three way Εγκαθίδρυση σύνδεσης

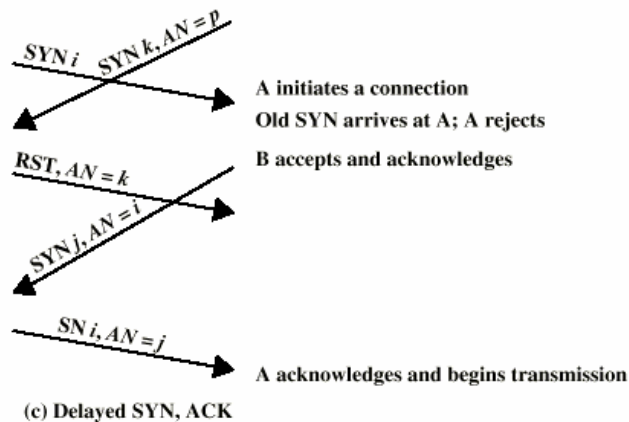
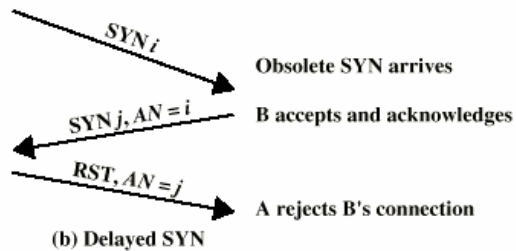
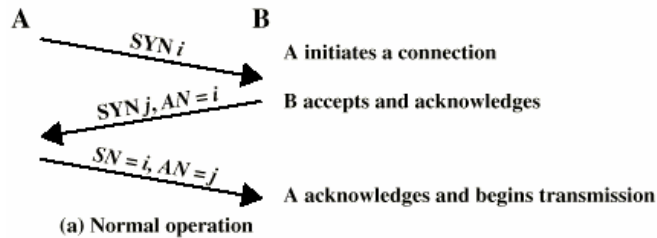


SV = state vector  
MSL = maximum segment lifetime





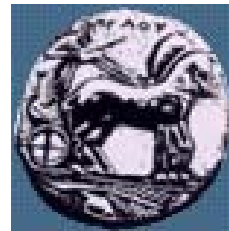
# Three way Εγκαθίδρυση σύνδεσης





## Τερματισμός σύνδεσης

- Η οντότητα σε κατάσταση CLOSE WAIT στέλνει το τελευταίο πακέτο ακολουθημένο από το μήνυμα FIN
- Το μήνυμα FIN φτάνει πριν το τελευταίο πακέτο δεδομένων
- Ο παραλήπτης λαμβάνει το μήνυμα FIN
  - Κλείνει την σύνδεση
  - Χάνει το τελευταίο πακέτο δεδομένων
- Συσχέτιση αριθμών ακολουθίας με μηνύματα FIN
- Ο παραλήπτης αναμένει όλα τα πακέτα δεδομένων πριν το αριθμό ακολουθίας του μηνύματος FIN
- Απώλεια πακέτων και παλαιών πακέτων.....
  - Θα πρέπει να επιβεβαιώνονται τα μηνύματα FIN



## Ομαλός τερματισμός

- Αποστολή μηνύματος FIN  $i$  και λήψη μηνύματος AN  $i$
- Λήψη μηνύματος FIN  $j$  και αποστολή μηνύματος AN  $j$
- Αναμονή διπλάσιου χρόνου από τον αναμενόμενο «χρόνο ζωής» του πακέτου



## Ανάκαμψη από πρόβλημα

- Μετά από επανεκκίνηση όλες οι πληροφορίες κατάστασης χάνονται
- Οι συνδέσεις είναι ανοικτές μόνο από το ένα άκρο
  - Οι πλευρά η οποία δεν έχει πρόβλημα ακόμη νομίζει ότι είναι συνδεδεμένη
- Η σύνδεση κλείνει μετά από κάποιο timeout
  - Αναμονή για μήνυμα ACK για χρόνο timeout\*αριθμό επανα-μεταδόσεων
  - Μετά την παραπάνω αναμονή κλείσει την σύνδεση και ενημέρωσε τον χρήστη
- Αποστολή μηνύματος RST  $i$  ως ανταπόκριση για κάθε  $i$  πακέτο το οποίο λαμβάνεται
- Ο χρήστης θα πρέπει να αποφασίσει εάν θα ξανα-συνδεθεί
  - Πρόβλημα με χαμένα πακέτα ή διπλές μεταδόσεις πακέτων



# TCP & UDP

- Transmission Control Protocol
  - Προσανατολισμένο σε σύνδεση
  - RFC 793
- User Datagram Protocol (UDP)
  - Όχι προσανατολισμένο σε σύνδεση
  - RFC 768

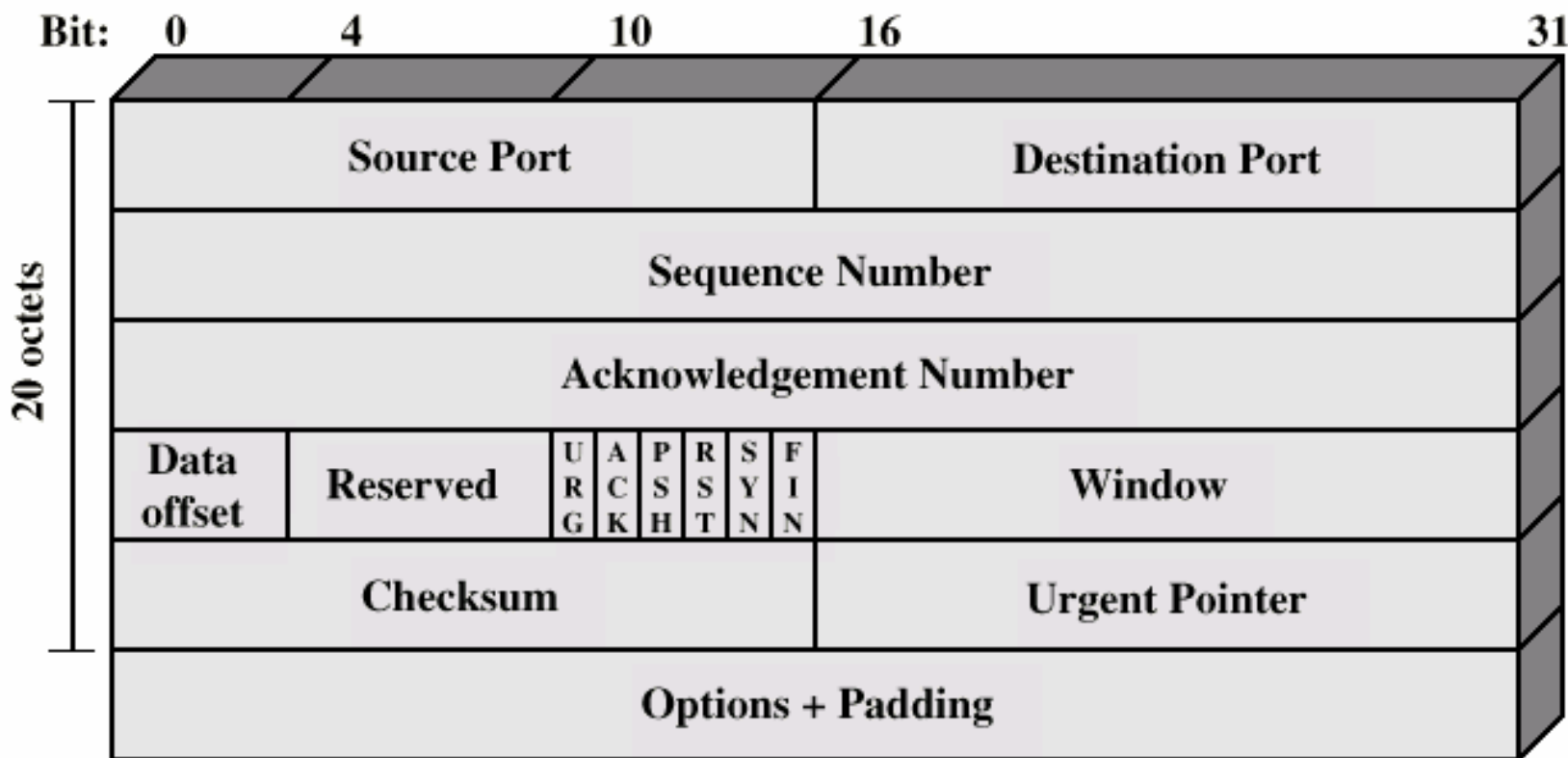


# TCP Υπηρεσίες

- Αξιόπιστη επικοινωνία ανάμεσα σε δύο διεργασίες
- Λειτουργεί σε μια ποικιλία αξιόπιστων και μη αξιόπιστων δικτύων
- Δυνατότητα push δεδομένων
  - Μετάδοση όλων των δεδομένων μέχρι το push flag
  - Ο παραλήπτης θα λαμβάνει με τον ίδιο ρυθμό
- Χρήση urgent data signal
  - Σηματοδοτεί την μετάδοση επείγων δεδομένων
  - Ο χρήστης αποφασίζει πως θα τα χειριστεί



# TCP Επιεφαλίδα





# Παράμετροι οι οποίοι δίνονται στο IP

- Το TCP δίνει κάποιες παραμέτρους στο IP
  - Precedence
  - Κανονικό delay / χαμηλό delay
  - Κανονικό throughput/υψηλό throughput
  - Κανονικό reliability/υψηλό reliability
  - Security





# TCP Μηχανισμοί (1)

- Εγκαθίδρυση σύνδεσης
  - Three way εγκαθίδρυση σύνδεσης
  - Ανάμεσα σε δύο θύρες
  - Μία θύρα μπορεί να συνδεθεί σε πολλαπλούς προορισμούς



## TCP Μηχανισμοί (2)

- Μεταφορά δεδομένων
  - Λογικές ροές από octets
  - Τα octets αριθμούνται με modulo  $2^{23}$
  - Έλεγχος ροής με χρήση credit στον αριθμό των octets
  - Τα δεδομένα αποθηκεύονται προσωρινά τόσο στον αποστολέα όσο και στον παραλήπτη



## TCP Μηχανισμοί (3)

- Τερματισμός σύνδεσης
  - Ομαλός τερματισμός
  - Ο χρήστης ανακοινώνει την επιθυμία να τερματίσει την σύνδεση
  - Η οντότητα στο επίπεδο μεταφοράς θέτει το FIN flag στο τελευταίο πακέτο το οποίο στέλνει
  - Αιφνίδιος τερματισμός
    - Η οντότητα στο επίπεδο μεταφοράς δεν στέλνει ούτε λαμβάνει δεδομένα
    - Το RST πακέτο μεταδίδεται



# Υλοποίηση επιλογών πολιτικής

- Μετάδοση
- Λήψη
- Αποδοχή
- Επανα-μετάδοση
- Επιβεβαίωση λήψης

# Μετάδοση



- Στην απουσία push, ή τερματισμού το TCP μεταδίδει δεδομένα με τον δικό του ρυθμό
- Τα δεδομένα αποθηκεύονται προσωρινά πριν την μετάδοση
- Μπορεί να δημιουργηθούν πακέτα ανάλογα με την ροή εισόδου δεδομένων
- Μπορεί να αναμένει μέχρι να λάβει μια συγκεκριμένη ποσότητα δεδομένων

# Λήψη



- Στην απουσία push παραδίδει δεδομένα με τον δικό του ρυθμό
- Παραδίδει δεδομένα όταν λαμβάνει πακέτα δεδομένων
- Μπορεί να αποθηκεύει προσωρινά δεδομένα από περισσότερα του ενός πακέτα



# Αποδοχή

- Τα πακέτα μπορεί να φτάσουν με διαφορετική σειρά από ότι μεταδόθηκαν
- In order
  - Αποδοχή μόνο πακέτων στην σωστή σειρά
  - Απόρριψη πακέτων σε λάθος σειρά
- In windows
  - Αποδοχή όλων των πακέτων μέσα στο receive window



## Επανα-μετάδοση

- Το TCP διατηρεί μια ουρά με τα πακέτα τα οποία έχει μεταδώσει αλλά δεν έχει λάβει επιβεβαίωση για την σωστή λήψη τους
- Το TCP θα επανα-μεταδώσει εάν δεν λάβει επιβεβαίωση σε ένα ορισμένο χρονικό διάστημα
  - First only
  - Batch
  - Individual





# Επιβεβαίωση λήψης

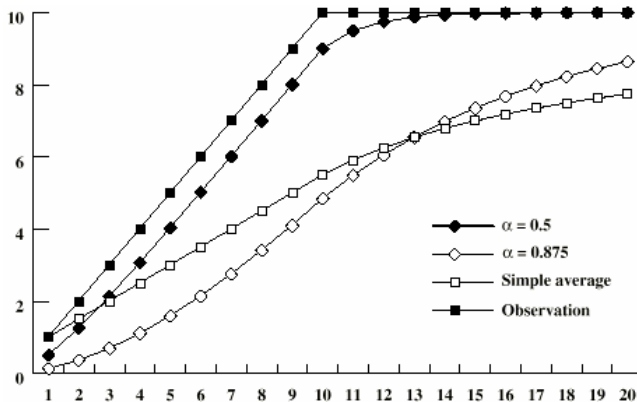
- Άμεση
- Αθροιστική



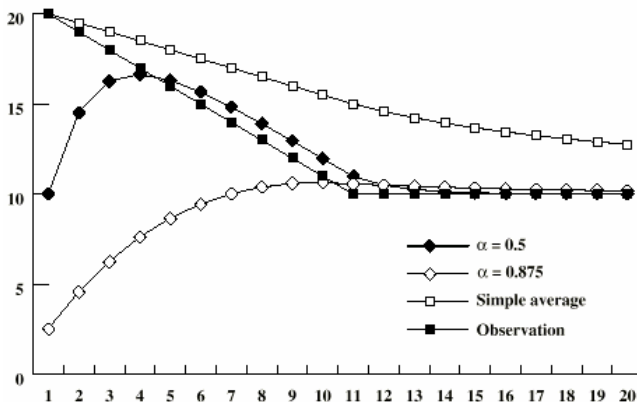
# Έλεγχος συμφόρησης

- RFC 1122, απαιτήσεις για κόμβους του Διαδικτύου
- Διαχείριση timeout επανα-μετάδοσης
  - Εκτίμηση του χρόνου round trip delay μελετώντας τα μοτίβα της καθυστέρησης
  - Ορισμός του timeout σε κάτι μεγαλύτερο από την παραπάνω εκτίμηση
  - Απλός μέσος όρος
  - Εκθετικός (exponential) μέσος όρος
  - Εκτίμηση διακύμανσης RTT (αλγόριθμος Jacobson)

# Χρήση εκθετικού (exponential) μέσου όρου



(a) Increasing function



(b) Decreasing function



# Backoff

- Εκτιμώντας ότι η λήξη του timeout οφείλεται σε συμφόρηση, η διατήρηση του RTO δεν είναι καλή ιδέα
- Αύξηση του RTO κάθε φορά που ένα πακέτο επαναμεταδίδεται
- $RTO = q * RTO$
- Συνήθως  $q=2$ 
  - Binary exponential backoff



# Αλγόριθμος Karn

- Εάν ένα πακέτο επανα-μεταδοθεί, η επιβεβαίωση θα αφορά:
  - Το πρώτο αντίγραφο του πακέτου
    - RTT περισσότερο χρόνο από ότι αναμενόταν
  - Το δεύτερο αντίγραφο
- Δεν υπάρχει τρόπος να το πεις (No way to tell)
- Μην μετράς τον χρόνο RTT για πακέτα τα οποία επανα-μεταδίδονται
- Υπολόγισε το backoff όταν γίνεται επανα-μετάδοση ενός πακέτου
- Χρησιμοποίησε backoff RTO μέχρι να λάβεις επιβεβαίωση για πακέτο το οποίο δεν έχει επανα-μεταδώσει



## Διαχείριση παραθύρου (window)

- Slow start
  - $awnd = \text{MIN}[\text{credit}, \text{cwnd}]$
  - Όταν ξεκινά η σύνδεση  $\text{cwnd}=1$
  - Αύξησε το  $\text{cwnd}$  σε κάθε επιβεβαίωση μέχρι ένα άνω όριο
- Δυναμικό μέγεθος παραθύρου σε περιόδους συμφόρησης
  - Όταν υπάρξει timeout
  - Θέσε το slow start threshold στο μισό του τρέχοντος μεγέθους του παραθύρου
    - $\text{ssthresh} = \text{cwnd}/2$
  - Θέσε  $\text{cwnd} = 1$  και slow start μέχρι  $\text{cwnd} = \text{ssthresh}$ 
    - Αύξησε το  $\text{cwnd}$  κατά 1 σε κάθε επιβεβαίωση την οποία λαμβάνεις
  - Για  $\text{cwnd} \geq \text{ssthresh}$ , αύξησε το  $\text{cwnd}$  κατά 1 για κάθε χρόνο RTT

# UDP



- User datagram protocol
- RFC 768
- Υπηρεσία μη προσανατολισμένη σε σύνδεση
  - Παράδοση και έλεγχος διπλής μετάδοσης δεν εγγυάται
- Μειωμένη πολυπλοκότητα
- Διάφορες χρήσεις (πχ μετάδοση πολυμέσων)



# Χρήσεις UDP

- Συλλογή δεδομένων (Inward data collection)
- Διασπορά δεδομένων (Outward data dissemination)
- Αίτηση – απόκριση
- Εφαρμογές πραγματικού χρόνου





# Επιεφαλίδα UDP

