Transmission Control Protocol

Δρ. Κωνσταντίνος Σ. Χειλάς

Transmission Control Protocol

- TCP δουλεύει στο επίπεδο μεταφοράς (transport layer) της σουίτας πρωτοκόλλων του TCP/IP
- Παρέχει **αξιόπιστη** (reliable) μεταφορά δεδομένων στις εφαρμογές που το χρησιμοποιούν.
- Η σύνδεση που προσφέρει είναι **connection oriented.**
- Είναι ένα πρωτόκολλο που χειρίζεται ροές bytes (byte stream oriented protocol)

Το TCP ασχολείται με …

- Διευθυνσιοδότηση
- Υλοποίηση σύνδεσης (Connection Establishment)
- Απόλυση σύνδεσης (Connection release)
- Χειρισμό πολιτικών μετάδοσης
- Έλεγχο ροής και ενταμίευση (buffering)
- Διάφορες άλλες λειτουργίες

Παροχές από το TCP

- **Stream Data Transfer**
- **Reliability**
- **Flow Control**
- **Multiplexing**
- **Logical Connections**
- **Full Duplex**

Διευθυνσιοδότηση

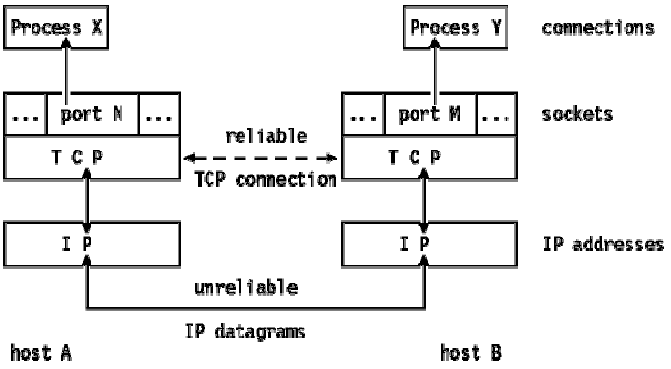
- Το TCP χρησιμοποιεί την έννοια του αριθμού της θύρας (port number) που λειτουργεί ως σημείο πρόσβασης της υπηρεσίας (SAP) και μέσω αυτού αναγνωρίζεται με μοναδικό τρόπο μια διαδικασία επικοινωνίας σε έναν υπολογιστή.
- Συνήθως οι εφαρμογές των εξυπηρετητών χρησιμοποιούν αυτές που είναι γνωστές ως *"Well known port numbers"* και έχουν οριστεί από οργανισμούς τυποποίησης.
- Οι εφαρμογές στην πλευρά του πελάτη χρησιμοποιούν τις εφήμερες θύρες "Ephemeral Ports" ορίζοντάς τες με τυχαίο τρόπο για κάθε σύνδεση.
- Τα παραπάνω σύνολα θυρών δεν έχουν τομή. Είναι αμοιβαία αποκλειόμενα.

Το μοντέλο υπηρεσιών του TCP

Port	Protocol	Use
21	FTP	File transfer
23	Telnet	Remote login
25	SMTP	E-mail
69	TFTP	Trivial File Transfer Protocol
79	Finger	Lookup info about a user
80	HTTP	World Wide Web
110	POP-3	Remote e-mail access
119	NNTP	USENET news

Μερικές ορισμένες θύρες.

Δύο διαδικασίες που επικοινωνούν μέσω TCP Sockets



Η επικεφαλίδα του TCP Segment

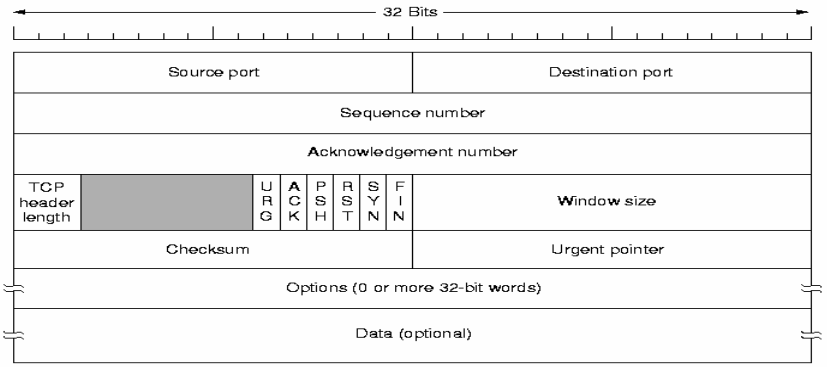


Fig. 6-24. The TCP header.

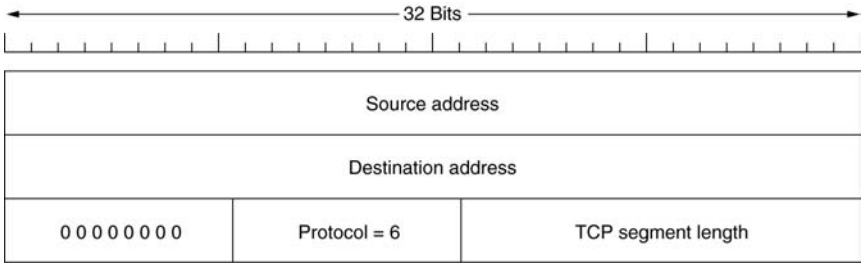
- **SrcPort** και **DstPort**. Μαζί με τις IP διευθύνσεις ορίζουν μια΄TCP σύνδεση με μοναδικό τρόπο.
- Ο αριθμός ακολουθίας (**sequence number)** προσδιορίζει τον αριθμό του byte, μέσα στη ροή δεδομένων που στέλνει ο αποστολέας, που έχει το πρώτο byte του segment.
- Ο αριθμός επιβεβαίωσης (**Acknowledgement number)** περιέχει τον επόμενο αριθμό ακολουθίας που περιμένει να λάβει ο αποστολέας του ACK. Αυτός είναι ο αριθμός ακολουθίας του τελευταίου byte που έλαβε με επιτυχία συν 1. Το πεδίο είναι έγκυρο μόνο εφόσον η σημαία ACK είναι ενεργοποιημένη. Από τη στιγμή που υλοποιείται μια σύνδεση η σημαία είναι ενεργοποιημένη συνεχώς.

- Τα πεδία **Acknowledgement**, **SequenceNum**, και **AdvertisedWindow** εμπλέκονται στη λειτουργία του αλγορίθμου του κινούμενου παραθύρου «sliding window algorithm».
- Τα πεδία Acknowledgement και AdvertisedWindow περιέχουν πληροφορία σχετική με τη ροή των δεδομένων προς την αντίθετη κατεύθυνση.
- Ο δέκτης διαφημίζει προς τον αποστολέα το μέγεθος ενός παραθύρου χρησιμοποιώντας το πεδίο AdvertisedWindow. Με τον τρόπο αυτό περιορίζεται ο αποστολέας, σε κάθε χρονική στιγμή, να μην έχει στείλει περισσότερα δεδομένα από το μέγεθος του παραθύρου, χωρίς να λάβει επιβεβαίωση.
  - Ο δέκτης καθορίζει την τιμή αυτή λαμβάνοντας υπόψη του το μέγεθος της μνήμης που έχει δεσμεύσει για την εξυπηρέτηση της σύνδεσης (buffering).

- Το **header length** δίνει το μέγεθος της επικεφαλίδας σε λέξεις των 32-bit. Αυτό είναι απαραίτητο από τη στιγμή που το μέγεθος του πεδίου options είναι μεταβλητό.
- Το μεγέθους 6-bit πεδίο **Flags** χρησιμοποιείται για την ανταλλαγή πληροφοριών ελέγχου μεταξύ των άκρων της επικοινωνίας. Οι πιθανές σημαίες είναι: SYN, FIN, RESET, PUSH, URG, και ACK.
  - Οι SYN και FIN χρησιμοποιούνται κατά την ενεργοποίηση ή τη λύση μιας TCP σύνδεσης αντίστοιχα.
  - Η σημαία ACK είναι ενεργή κάθε φορά που υπάρχει έγκυρο πεδίο επιβεβαίωσης (Acknowledgement). Υπονοείται ότι ο δέκτης πρέπει να λάβει το πεδίο υπόψη του.
  - Η σημαία URG δηλώνει ότι το παρόν segment περιέχει επείγοντα δεδομένα. Όταν η σημαία είναι ενεργοποιημένη, τότε το πεδίο UrgPtr δείχνει το σημείο του πακέτου από το οποίο ξεκινούν τα μη επείγοντα δεδομένα. (το τελευταίο byte επειγόντων δεδομένων)
  - Η σημαία PUSH σημαίνει ότι ο αποστολέας ενεργοποίησε τη διαδικασία Push (άμεσης προώθησης δεδομένων) και η διαδικασία του TCP στην πλευρά του παραλήπτη πρέπει να ενημερώσει άμεσα την εφαρμογή που λαμβάνει τα δεδομένα.
  - Τέλος, η σημαία RESET δηλώνει ότι ο δέκτης βρίσκεται σε σύγχυση και θέλει να κλείσει τη σύνδεση.

- Ο έλεγχος σφαλμάτων (**Checksum**) καλύπτει ολόκληρο το TCP segment, δηλαδή τα δεδομένα και την επικεφαλίδα. Η ύπαρξη και η χρήση του είναι υποχρεωτικές. Η τιμή του πρέπει να υπολογιστεί από τον αποστολέα και να επιβεβαιωθεί από τον παραλήπτη.
- Το πεδίο **Option** σχετίζεται με την επιλογή του μέγιστου μεγέθους του segment που είναι γνωστή ως MSS. Ορίζεται από τα άκρα της επικοινωνίας μέσα στο πρώτο segment που ανταλλάσσεται και ορίζει το μέγιστο μέγεθος του segment που επιθυμεί να δέχεται ο αποστολέας.
- Η ύπαρξη δεδομένων στο TCP segment είναι προαιρετική.

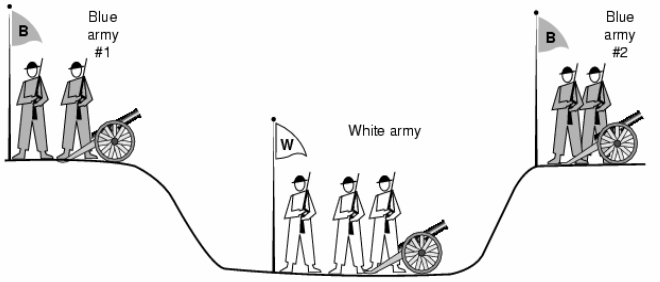
pseudoheader



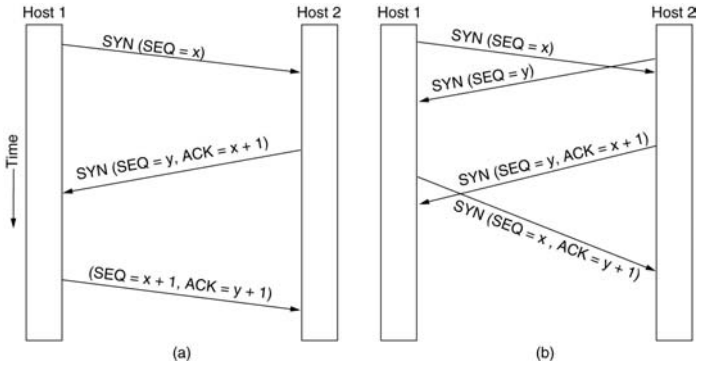
Options

- Maximum segment size (MSS) option: Ορίζει το μέγιστο αποδεκτό μέγεθος segment μεταξύ αποστολέα και παραλήπτη. Η προεπιλεγμένη τιμή είναι 536 bytes (το τυπικό μέγεθος ενός IP datagram, 576, μείον τα μεγέθη των επικεφαλίδων) και συνίσταται να χρησιμοποιείται όταν οι σταθμοί που επικοινωνούν βρίσκονται σε διαφορετικά υποδίκτυα. Η τιμή του MSS συμφωνείται κατά τη ανταλλαγή των πακέτων SYN και SYN/ACK.
- Timestamp option: Συνδυάζεται με την επιλογή timestamp reply option. Χρησιμοποιείται από τον αποστολέα για τον υπολογισμό των χρόνων RTT (round trip time) και RTO (retransmission timeout timer). Είναι πεδίο ενεργοποιημένο σε όλα τα TCP segments. Στα πακέτα ACK επιστρέφεται η τιμή που είχε το πακέτο του αποστολέα το οποίο επιβεβαιώνεται.
- Window scale option: Το μέγεθος παραθύρου που ορίζεται με τα 16 bit του πεδίου Window Size είναι πολύ μικρό για κανάλια με μεγάλο εύρος ζώνης και μεγάλες καθυστερήσεις μετάδοσης. Η τιμή του πεδίου Window Scale μπορεί να είναι από 0 έως 14, δίνει την κλίμακα μεγέθυνσης του παραθύρου και συμφωνείται κατά τη διάρκεια της τριπλής χειραψίας στα πακέτα SYN. Παραμένει σταθερό σε όλη τη διάρκεια της επικοινωνίας.

Το πρόβλημα των δύο στρατών



Διαχείριση TCP συνδέσεων
Three-way Handshake



- (α) Τυπική υλοποίηση σύνδεσης TCP.
- (β) Σύγκρουση κλήσεων.

Υλοποίηση σύνδεσης

- Το TCP χρησιμοποιεί για την εγκαθίδρυσης μιας σύνδεσης, έναν μηχανισμό τριπλής «χειραψίας».
- Κάθε πλευρά επιλέγει έναν αρχικό αριθμό ακολουθίας που πρέπει να επιβεβαιωθεί από τον απέναντι.
- Οι ακολουθιακοί αριθμοί των πακέτων που αποτελούν τη σύνδεση υπολογίζονται με βάση τη σχέση:

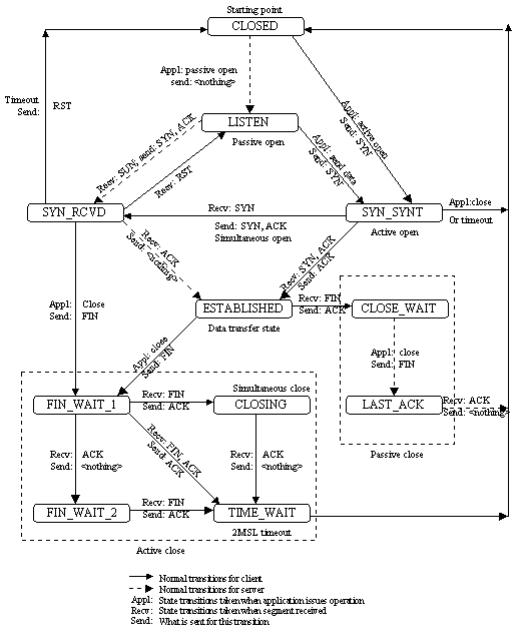
  $Seq(P_n) = Seq(P_n\text{-}1) + Length(P_n\text{-}1)$

  $Ack(LP_n) = Seq(RP_n\text{-}1) + Length(RP_n\text{-}1) + 1$
- Λέμε ότι η πλευρά που στέλνει το πρώτο μήνυμα SYN εκτελεί ενεργό άνοιγμα της γραμμής (active open) ενώ ο απέναντι κάνει ένα παθητικό άνοιγμα (passive open).
- Είναι δυνατό και οι δύο πλευρές να εκτελέσουν active open

Τερματισμός της σύνδεσης (4 μηνύματα)

- Οι συνδέσεις TCP λειτουργούν ως full duplex (μπορεί κανείς να τις θεωρήσει και ως δύο simplex).
- Κάθε μία μπορεί να απελευθερωθεί (τερματιστεί) ανεξάρτητα.
- Το πρόβλημα των δύο στρατών μπορεί να αποφευχθεί με τη χρήση χρονομετρητών (2MSL_Timer).
- Η πλευρά που στέλνει πρώτη το μήνυμα FIN λέμε ότι κάνει active close. Παρόλα αυτά και οι δυο πλευρές μπορούν να εκτελέσουν active close
- Το κλείσιμο της σύνδεσης σημαίνει ότι δεν πρέπει να αποσταλούν άλλα δεδομένα προς την κατεύθυνση που έκανε active close

TCP Finite State Machine



Οι καταστάσεις που εμφανίζονται στο finite state machine του TCP.

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

Πολιτική εκπομπής στο TCP

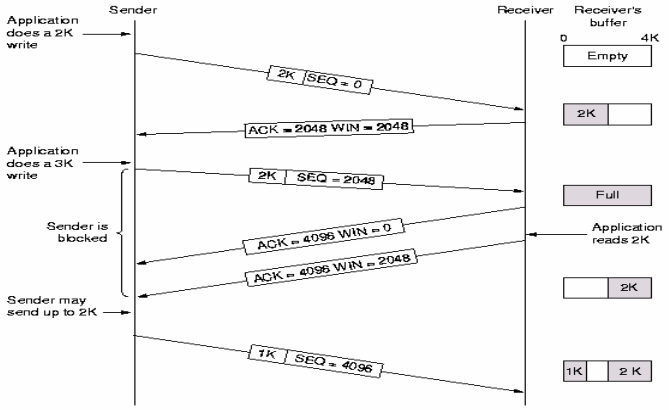
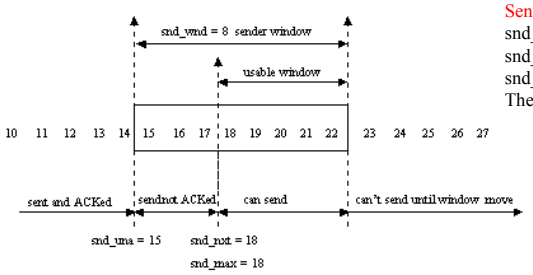
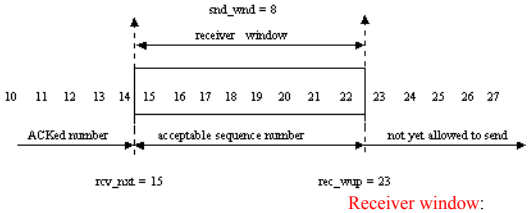


Fig. 6-29. Window management in TCP.



TCP Sender Window[8]



TCP Receiver Window

Sender window:
snd_una - oldest unacknowledged sequence number.
snd_nxt - next send sequence number.
snd_wnd - offered window (advertised by receiver)
The acceptable ACK is: snd_una < ACK <= snd_next

Receiver window:
rec_next - next receive sequence number.
rec_wnd - receiver window (advertised to sender).
The acceptable segment is:
rcv_next <= beginning sequence number of segment < rcv_nxt + rcv_wnd
rcv_nxt <= ending sequence number < rec_nxt + rec_wnd.

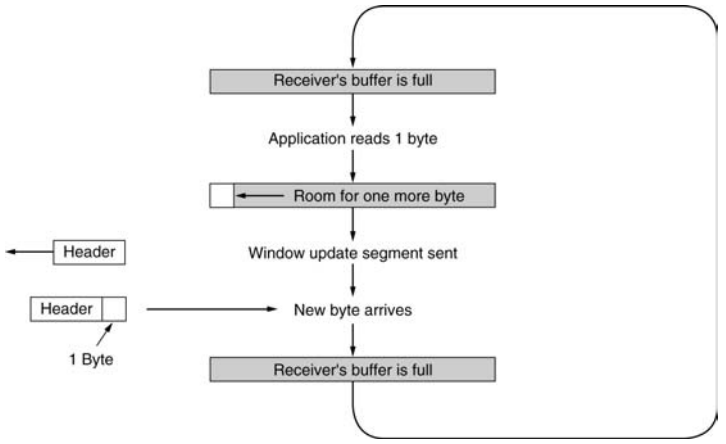
Πολιτική εκπομπής στο TCP με διαδραστικά δεδομένα

- π.χ. telnet σε έναν επεξεργαστή κειμένου που πρέπει να αντιδρά σε κάθε πλήκτρο. Μπορεί να χρειάζεται μέχρι και 162bytes ($41_{send}+40_{ACK}+40_{window\text{-}update}+41_{echo}$) για ένα γράμμα.

- **Ο αλγόριθμος του Nagle**
  - Τα δεδομένα φτάνουν στην TCP οντότητα ως μια ροή από bytes με ρυθμό 1 byte τη φορά.
  - Στείλε το πρώτο byte
  - Ενταμίευσε τα υπόλοιπα μέχρι να έρθει το ACK για το πρώτο byte
  - Στείλε όλα τα αποθηκευμένα δεδομένα με ένα TCP segment
  - Ξεκίνα να ενταμιεύεις (buffer) μέχρι να λάβεις ACK.

Πρόβλημα με περιβάλλοντα όπως τα X Windows όπου πρέπει να στέλνονται οι κινήσεις του ποντικιού.

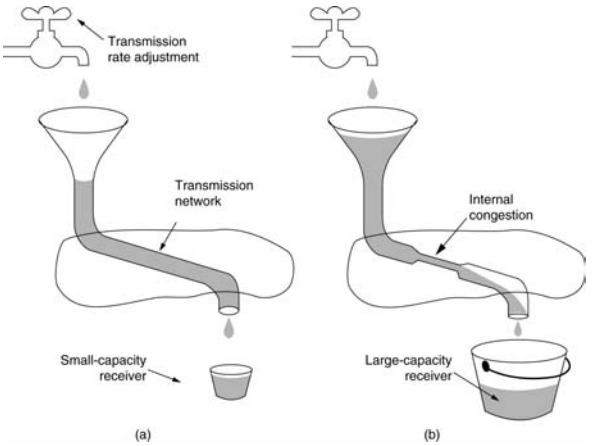
Silly window syndrome (Clark, 1982)



Έστω μια διαδραστική εφαρμογή που διαβάζει 1 byte τη φορά αλλά ο αποστολέας στέλνει πολλά bytes μαζί

Απαγορεύεται να διαπραγματευτεί παράμετρο 1 byte

Έλεγχος συμφόρησης στο TCP



(α) Ένα γρήγορο δίκτυο που τροφοδοτεί ένα δέκτη με μικρή μνήμη
(β) Ένα αργό δίκτυο που τροφοδοτεί έναν δέκτη με μεγάλη χωρητικότητα

Έλεγχος συμφόρησης (Congestion Control)

- Slow start
- Congestion avoidance
- Fast retransmit
- Fast recovery

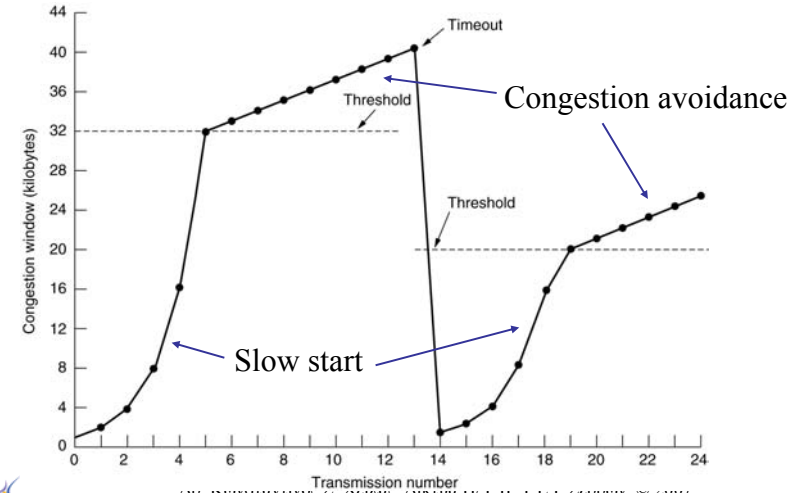
TCP Congestion Control

- Η μέθοδος Slow Start του Jacobson χρησιμοποιείται για να επιλύσει προβλήματα συμφόρησης στην πλευρά του παραλήπτη
  - Receiver Window
  - Congestion Window (cwnd)
- Μια μικρή παραλλαγή της μεθόδου χρησιμοποιείται για να αντεπεξέλθει στην εσωτερική συμφόρηση.
  - Receiver Window
  - Congestion Window
  - Congestion threshold (αρχικά 64KB, αν συμβεί συμφόρηση γίνεται το μισό του τρέχοντος παραθύρου)

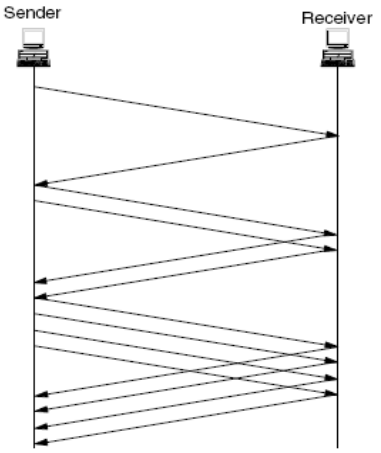
Network Congestion Algorithm

Το κατώφλι είχε αρχικά τεθεί στα 64K

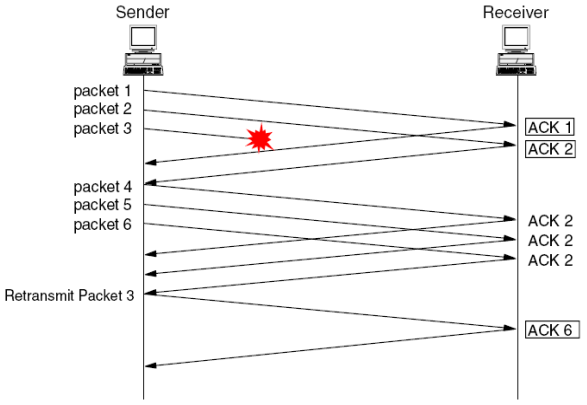
Έστω ότι παρατηρήθηκε συμφόρηση και έγινε 32K



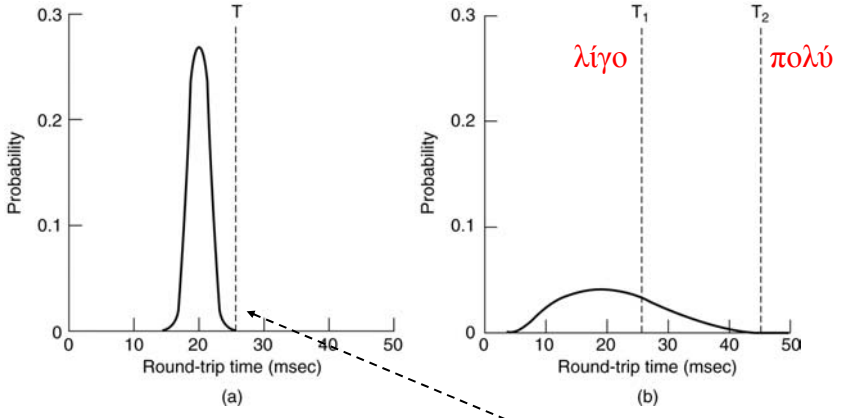
Slow start



Fast retransmit



Η διαχείριση του χρόνου στο TCP



(α) Η πυκνότητα πιθανότητας για τις αφίξεις των ACK στο επίπεδο σύνδεσης (data link layer).

(β) Η πυκνότητα πιθανότητας για τις αφίξεις των ACK στο TCP.

TCP Timers

- Connection establishment timer:
  - εκκινεί μόλις λάβει SYN. Αν δεν λάβει ACK εντός 75sec κλείνει τη σύνδεση.
- Retransmission timer:
  - Ξεκινάει μόλις το TCP στείλει δεδομένα. Αν δεν έρθει ACK πριν τη λήξη του χρονομέτρου, τα δεδομένα ξαναστέλνονται. Ο χρόνος ρυθμίζεται δυναμικά, βάσει του *RTT*:
  - $RTT = a*RTT+(1 - a)*M$, όπου M η τελευταία μετρημένη τιμή του *RTT*, $\alpha=7/8$.
  - $D = a*D+(1 - a)*|RTT - M|$, διόρθωση με βάση τη μέση διακύμανση (Jacobson, 1988).
  - $RTO=Retransmit\text{-}TimeOut = RTT + 4 \times D$

Αλγόριθμος του Karn

- Αν ένα TCP segment έχει επανεκπεμφθεί τότε το νεοαφιχθέν ACK μπορεί να είναι:
  - Για το πρώτο αντίγραφο του segment που εκπέμφθηκε
    - Άρα το RTT είναι μεγαλύτερο από το αναμενόμενο
  - Για το δεύτερο αντίγραφο
- Δεν υπάρχει τρόπος να τα διακρίνουμε
- Αλγόριθμος Karn
  - Δεν υπολογίζεται το RTT για τις επανεκπομπές
  - Μόλις διαπιστώσεις επανεκπομπή ενεργοποίησε τη λειτουργία οπισθοδρόμησης (backoff)
  - Χρησιμοποίησε το ήδη υπολογισμένο backoff RTO μέχρις ότου λάβεις ACK για ένα segment που δεν επανεκπέμφθηκε.

TCP Timers

- Delayed ACK timer:
  - Ενεργοποιείται μόλις το TCP στείλει δεδομένα των οποίων η λήψη δε χρειάζεται να επιβεβαιωθεί άμεσα. Στο Linux ο χρόνος είναι 300msec.
- Persistence timer
  - Αν το απέναντι άκρο διαφημίσει μηδενικό παράθυρο αλλά ο αποστολέας έχει δεδομένα να στείλει, τότε παρατηρεί το παράθυρο στη διάρκεια ενός retransmission interval και αν δεν πάρει ACK με νέο παράθυρο στέλνει ένα probe segment για να δημιουργήσει ACK. (για να εξασφαλίσει ότι το ACK/WIN δεν έχει χαθεί)
- Keepalive timer:
  - Αν η σύνδεση είναι ανενεργή (idle) για δυο ώρες, στέλνει ένα ειδικό segment (keepalive probe). Αν το άλλο άκρο είναι κλειστό, ο αποστολέας θα λάβει RST και θα κλείσει. Αν λάβει ACK ρυθμίζει πάλι τον timer στις 2 ώρες.

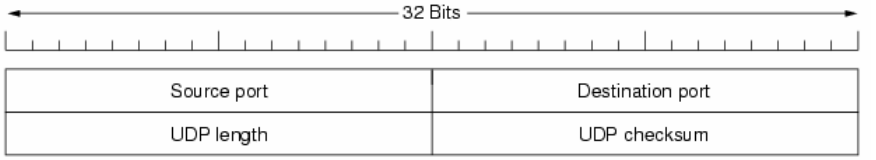
TCP Timers

- FIN_WAIT_2 timer:
  - Τίθεται στα 10 λεπτά όταν η σύνδεση μεταβεί από την κατάσταση FIN_WAIT_1 στην FIN_WAIT_2 και δεν μπορεί να λάβει άλλα δεδομένα. Όταν λήξει τίθεται στα 75sec κι αν ξαναλήξει, η σύνδεση κλείνει.
- 2MSL timer:
  - Ενεργοποιείται όταν η σύνδεση γίνει ενεργώς κλειστή (active closed). Η σύνδεση μένει σε κατάσταση TIME_WAIT για χρόνο ίσο με το διπλάσιο του Maximum Segment Lifetime (ο χρόνος ζωής ενός πακέτου στο δίκτυο πριν διαγραφεί) για την περίπτωση που θα χρειαστεί να ξαναστείλει το τελευταίο ACK που έστειλε.

User Datagram Protocol

User Datagram Protocol

- Πρωτόκολλο μεταφοράς – χωρίς σύνδεση (connectionless)
- Παρέχει μη-αξιόπιστη υπηρεσία
- Η παράδοση των πακέτων δεν είναι εγγυημένη
- Επίσης δεν μπορεί να εγγυηθεί τη λήψη διπλοτύπων



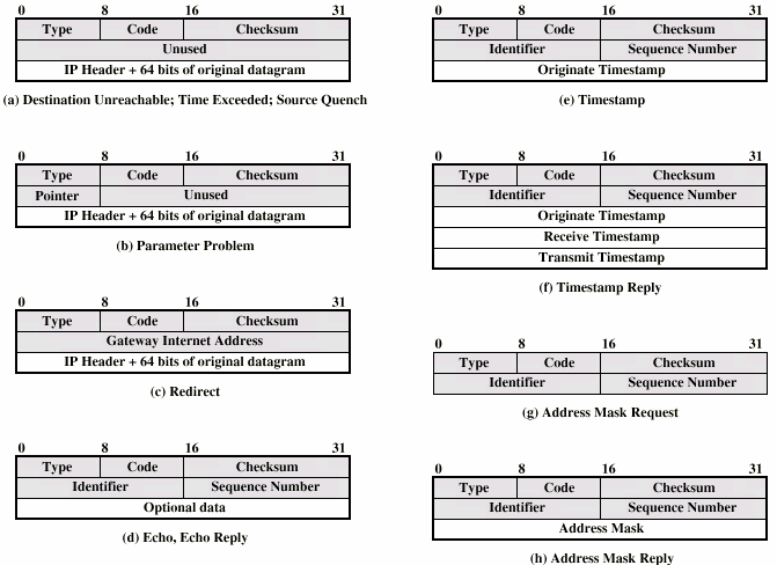
ICMP

Δρ. Κωνσταντίνος Σ. Χειλάς, Δίκτυα Η/Υ ΙΙ, Τ.Ε.Ι. Σερρών, © 2007

ICMP

- Internet Control Message Protocol
- RFC 792
- Μεταφορά μηνυμάτων ελέγχου από δρομολογητές και hosts σε hosts.
- Παρέχει ενημέρωση για προβλήματα
  - π.χ. time to live expired
- Ενθυλακώνεται σε ένα IP datagram
  - Δεν είναι αξιόπιστο

ICMP Message Formats



ΤΕΙ ΣΕΡΡΩΝ