



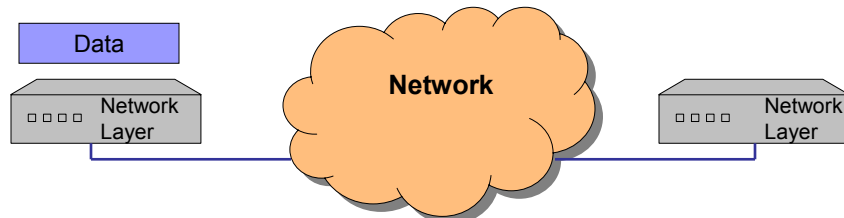
Επίπεδο Μεταφοράς (Transport Layer)

Περίληψη



- Ο ρόλος του επιπέδου μεταφοράς
- Τα στοιχεία του πρωτοκόλλου μεταφοράς
- Τα πρωτόκολλα μεταφοράς του Διαδικτύου
 - UDP (User Datagram Protocol)
 - RPC (Remote Procedure Call)
 - RTP (Real-time Transport Protocol)
 - TCP (Transport Control Protocol)
 - Άλλες μορφές του TCP
- Θέματα απόδοσης

Υπηρεσίες IP: “Best-Effort”

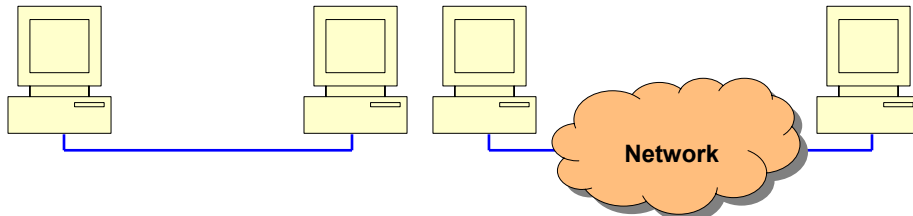


- Η υπηρεσία που τελικά προσφέρει το επίπεδο δικτύου έχει τα ακόλουθα χαρακτηριστικά
 - ☐
 - ☐ Πακέτα τελικά παραδίνονται εκτός σειράς (out-of-order)
 - ☐ Μπορεί να παρουσιαστούν πακέτα περισσότερες από μια φορές (duplicates).
 - ☐
 - ☐ Τα πακέτα παραδίδονται μετά από **αυθαίρετη καθυστέρηση**.

Υπηρεσίες του Επιπέδου Μεταφοράς

- Προσφέρει ότι και το επίπεδο Ζεύξης Δεδομένων!
 - ☐ **Εγγυάται** την παράδοση των μηνυμάτων
 - ☐ Παραδίδει τα μηνύματα με τη **σωστή σειρά**
 - ☐
 - ☐ Υποστηρίζει αυθαίρετο μέγεθος μηνυμάτων
 - ☐ Υποστηρίζει το συγχρονισμό μεταξύ των επικοινωνούντων
 - ☐ Επιτρέπει το έλεγχο της ροής προς το παραλήπτη
 - ☐ Υποστηρίζει την πολυπλεξία πολλαπλών χρηστών
- Γιατί τότε χρειάζεται το επίπεδο μεταφοράς;

Γιατί Χρειάζεται το Επίπεδο Μεταφοράς;



"If all real networks were flawless and all had the same service primitives and were guaranteed never, ever to change, the transport layer might not be needed"

A. Tanenbaum

■ Επίπεδο ζεύξης δεδομένων:

- ☐
- ☐ Η σύνδεση είναι γενικά «εύκολη»
- ☐ Δεν υπάρχει περίπτωση παράδοσης δεδομένων εκτός σειράς
- ☐ Περιορισμένος αριθμός συνδέσεων

Έλεγχος Ροής και Έλεγχος Συμφόρησης

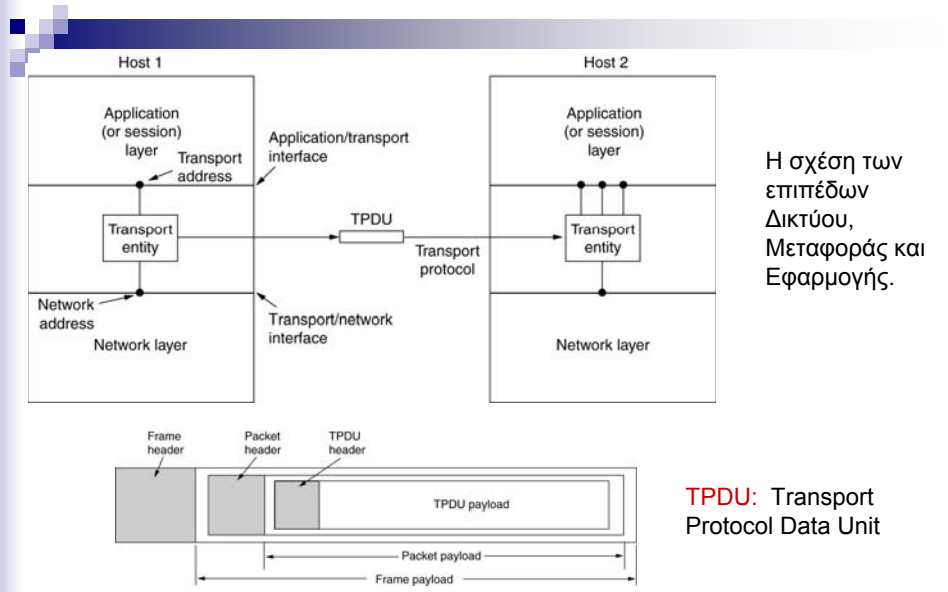
■ Έλεγχος Ροής (Flow Control)

- ☐

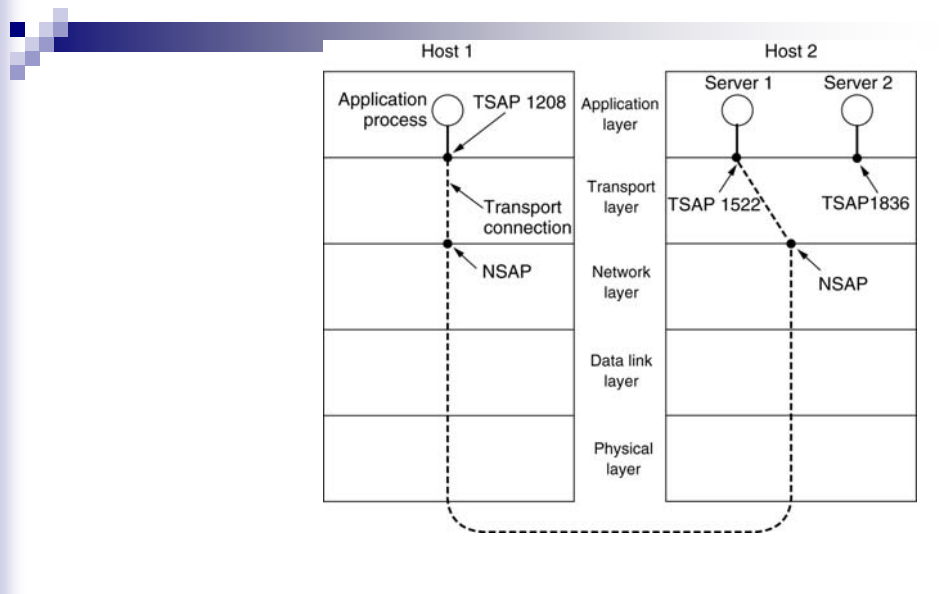
■ Έλεγχος Συμφόρησης (Congestion Control)

- ☐ Ελέγχει τη ροή του αποστολέα έτσι ώστε να μη στέλνει περισσότερα δεδομένα από όσα μπορεί να μεταφέρει το **δίκτυο**

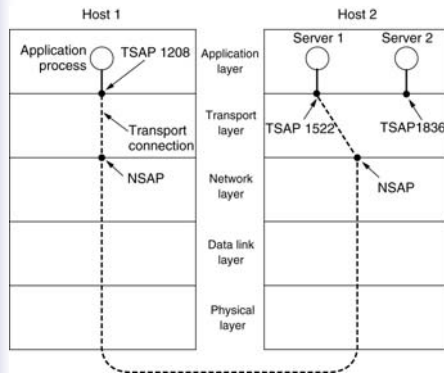
Στοιχεία του Πρωτοκόλλου Μεταφοράς



Διευθύνσεις και Πολυπλεξία



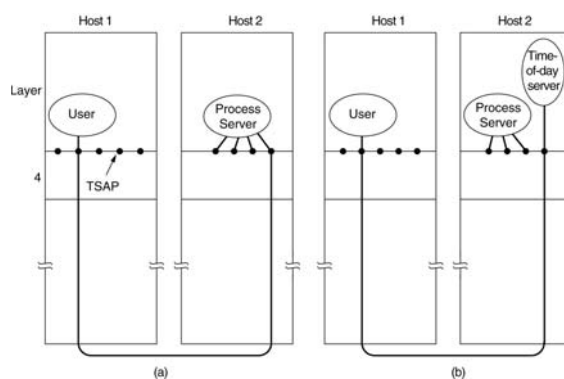
Διευθύνσεις και Πολυπλεξία



- Μια διαδικασία «δημιουργείται» στον υπολογιστή 2 και προσκολλάτε στη διεύθυνση 1522.
- Μια διαδικασία στον υπολογιστή 1 από το TSAP 1208 ζητά να συνδεθεί με την 1522 του εξυπηρετητή 2.
- Αφού συνδεθούν (**connect**), αρχίζουν να ανταλλάζουν μηνύματα
- Αφού τελειώσουν «ελευθερώνουν» (**release**) την σύνδεση.

Διευθύνσεις και Πολυπλεξία

- Πως ένα πρόγραμμα ξέρει σε ποια διεύθυνση να συνδεθεί;
 - Σταθερές διευθύνσεις (TSAP), π.χ.,
 - http: port 80
 - ftp: port 21
 - telnet: port 23
 - smtp: port 25
 - Δυναμικές Διευθύνσεις (Initial Connection Protocol)
 - Name or directory server



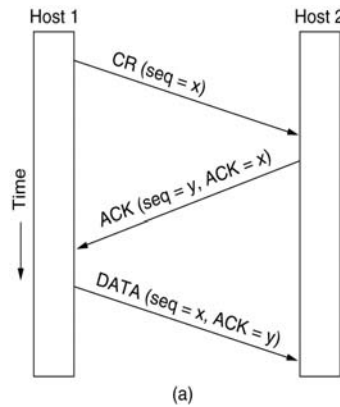
Δημιουργία Συνδέσεων (Connection Establishment)

- Υπηρεσίες με ή χωρίς σύνδεση
- Για να επιτευχθεί μια σύνδεση μεταξύ δύο hosts,
 - Ο host 1 στέλνει το μήνυμα ότι θέλει να δημιουργήσει μια σύνδεση (connection request (CR ή SYN))
 - Ο host 2 απαντά με την επαλήθευση (acknowledgement (ACK))
 - Ο host 1 αρχίζει να στέλνει τα πρώτα δεδομένα (DATA).
- Πρόβλημα: Τι θα συμβεί αν κάποια μηνύματα χαθούν, καθυστερήσουν ή φτάσουν περισσότερες από μία φορές (duplicates)
 - Π.χ., Σε μια συνδιαλλαγή με τράπεζα, ανοίγει μια σύνδεση, μετά γίνεται μεταφορά χρημάτων και τέλος κλείνει η σύνδεση.
 - Τι θα συμβεί αν η ίδια συνδιαλλαγή ξαναεμφανιστεί (πως μπορεί να ξαναεμφανιστεί);

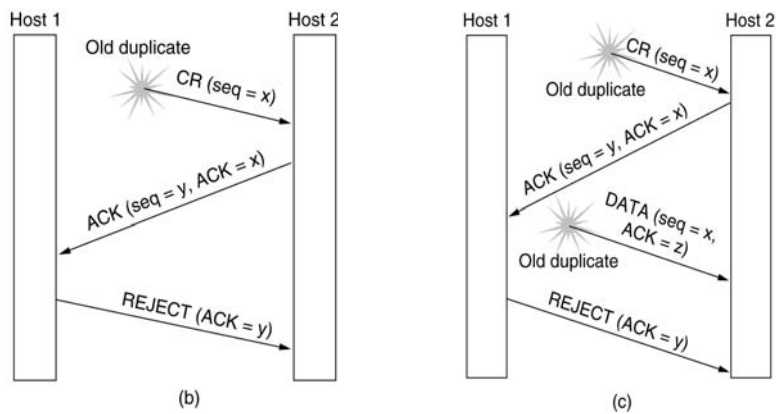
Δημιουργία Συνδέσεων (Connection establishment)

- Ας υποθέσουμε πως υπάρχει μέγιστος χρόνος T μετά από τον οποίο ένα πακέτο δεν μπορεί να συνεχίσει να υπάρχει.
 -
- Για να αποφευχθούν όσο το δυνατό περισσότερα προβλήματα, χρησιμοποιείται το “three way handshake” για την δημιουργία οποιασδήποτε σύνδεσης. Τα χαρακτηριστικά της μεθόδου είναι
 - Ο αρχικός αύξων αριθμός (seq) είναι τυχαίος (low order bits from a clock)
 - Ο Παραλήπτης διατηρεί τον δικό του αύξων αριθμό (ack)
 - Κάποιοι αριθμοί είναι απαγορευμένοι για να αποφεύγονται προβλήματα όταν ένας υπολογιστής αρχικοποιηθεί (reset).

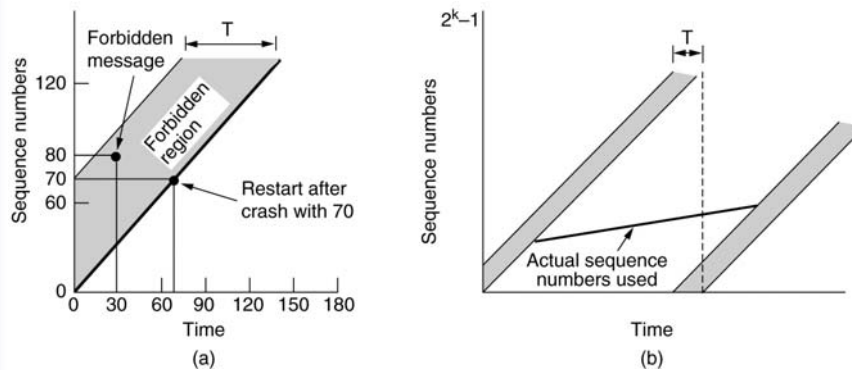
Επίτευξη Σύνδεσης



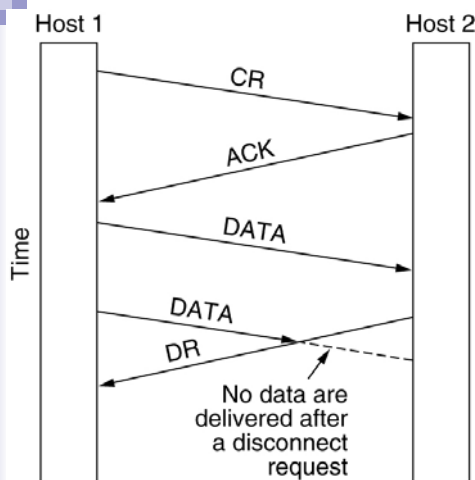
Αποφυγή Προβλημάτων



Συγχρονισμός μετά από Αρχικοποίηση



Ελευθέρωση Σύνδεσης



- Συντονισμός αποστολέα και παραλήπτη χρειάζεται και στην περίπτωση που θα ελευθερωθεί μια σύνδεση
 -
 - Χρήση πόρων (μνήμη) για την διατήρηση της κατάστασης κάθε σύνδεσης.

The diagram illustrates a battle scenario on a battlefield with a central depression. On the left, 'Blue army #1' is positioned on a raised area, featuring a flag with the letter 'B', two soldiers, and a cannon. On the right, 'Blue army #2' is similarly positioned on a raised area, also with a flag marked 'B', two soldiers, and a cannon. In the center, at the bottom of the depression, the 'White army' is located, marked by a flag with the letter 'W', three soldiers, and a cannon. The terrain is represented by a simple line drawing showing the relative positions of the forces.

The diagram illustrates the Stop-and-Wait protocol in two scenarios, (a) and (b), between Host 1 and Host 2.

Scenario (a): Successful Transmission

- Host 1:**
 - Send DR + start timer
 - Release connection
 - Send ACK
- Host 2:**
 - Send DR + start timer
 - Release connection
- Transmissions:**
 - DR (Data Request) from Host 1 to Host 2
 - DR (Data Request) from Host 2 to Host 1
 - ACK (Acknowledgment) from Host 2 to Host 1

Scenario (b): Packet Loss

- Host 1:**
 - Send DR + start timer
 - Release connection
 - Send ACK
- Host 2:**
 - Send DR + start timer
 - (Timeout) release connection
- Transmissions:**
 - DR (Data Request) from Host 1 to Host 2
 - DR (Data Request) from Host 2 to Host 1
 - ACK (Acknowledgment) from Host 1 to Host 2, which is marked as **Lost**.

(b)

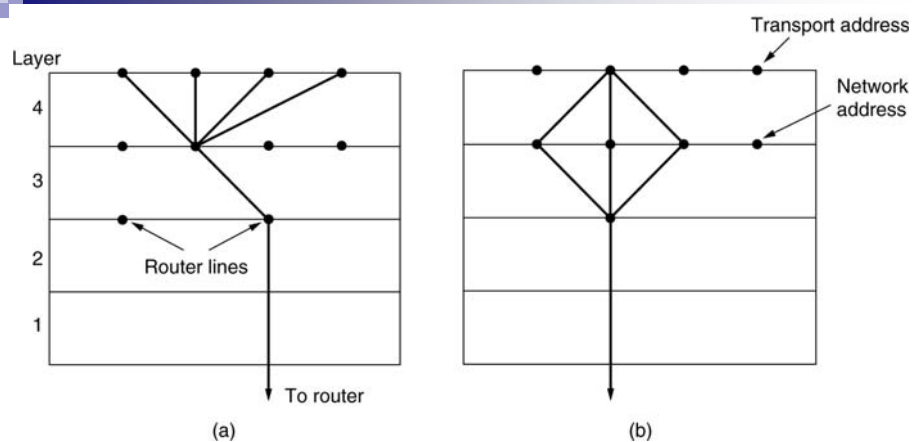
Παράδειγμα

	A	Message	B	Comments
1	→	< request 8 buffers>	→	A wants 8 buffers
2	←	<ack = 15, buf = 4>	←	B grants messages 0-3 only
3	→	<seq = 0, data = m0>	→	A has 3 buffers left now
4	→	<seq = 1, data = m1>	→	A has 2 buffers left now
5	→	<seq = 2, data = m2>	...	Message lost but A thinks it has 1 left
6	←	<ack = 1, buf = 3>	←	B acknowledges 0 and 1, permits 2-4
7	→	<seq = 3, data = m3>	→	A has 1 buffer left
8	→	<seq = 4, data = m4>	→	A has 0 buffers left, and must stop
9	→	<seq = 2, data = m2>	→	A times out and retransmits
10	←	<ack = 4, buf = 0>	←	Everything acknowledged, but A still blocked
11	←	<ack = 4, buf = 1>	←	A may now send 5
12	←	<ack = 4, buf = 2>	←	B found a new buffer somewhere
13	→	<seq = 5, data = m5>	→	A has 1 buffer left
14	→	<seq = 6, data = m6>	→	A is now blocked again
15	←	<ack = 6, buf = 0>	←	A is still blocked
16	...	<ack = 6, buf = 4>	←	Potential deadlock

→ Επιτυχής αποστολή μηνύματος,

... Απώλεια μηνύματος

Πολυπλεξία



Πολυπλεξία προς τα πάνω
(Upward multiplexing)

Πολυπλεξία προς τα κάτω
(downward multiplexing)

Απλό Πρωτόκολλο Μεταφοράς

■ Primitives

- LN: LISTEN (Αναμονή συνδέσεων)

-

■ Πακέτα

- CR: Call Request (Πρόθεση δημιουργίας σύνδεσης)
- CA: Call Accepted (Απάντηση στο CR)
- CLRR: Clear Request (Τερματισμός σύνδεσης)
- CLRC: Clear Confirmation (Απάντηση στο CLRR)
-
- CRT: Credit (Διαχείριση Παραθύρου)

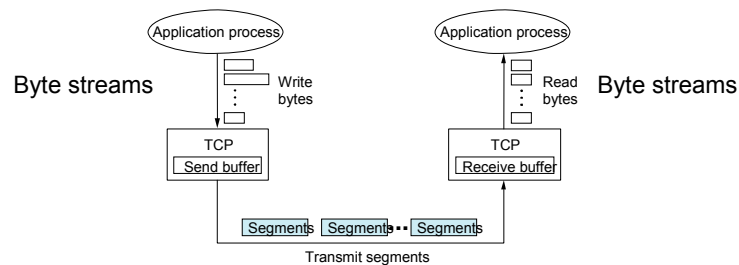
■ Καταστάσεις

- IDLE
- AWAIT (Αναμονή επαλήθευσης στο CR)
- PWAIT (Αναμονή Σύνδεσης)
- EST (established): Σύνδεση έχει επιτευχθεί
-
- REC: παραλαβή δεδομένων
- DISC: προσπάθεια για αποσύνδεση

Απλό Πρωτόκολλο Μεταφοράς

Πρωτόκολλα Διαδικτύου

- UDP (User Datagram Protocol)
- RPC (Remote Procedure Call)
-
- TCP (Transport Control Protocol)
- Άλλες μορφές του TCP



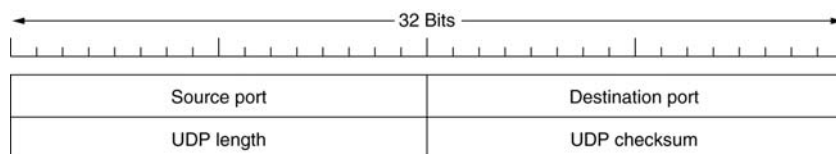
Μετάδοση Byte Streams

- Τα πρωτόκολλα UDP/TCP/... έχουν κάποιες ουρές (μνήμη) στην οποία οι εφαρμογές γράφουν ότι πληροφορία θέλουν να μεταδώσουν στον παραλήπτη.
- Τα πρωτόκολλα δεν χωρίζουν τα bytes σε μηνύματα.
- Πως αποφασίζει το πρωτόκολλο πότε να στείλει τα δεδομένα στο παραλήπτη;
 - Όταν μαζέψει αρκετά bytes (maximum segment size)
 - Όταν το ζητήσει η εφαρμογή (push operation)
 - Χρονόμετρο

User Datagram Protocol (UDP)

- Προσφέρει ελάχιστες υπηρεσίες
 - ☐ Ελάχιστο έλεγχο σφαλμάτων
- Δεν υποστηρίζει
 - ☐ Έλεγχο ροής ή συμφόρησης
 - ☐ Επαναμετάδοση σε περίπτωση σφαλμάτων
 - ☐ Παραλαβή δεδομένων στην σωστή σειρά
- Είναι απλό και ορισμένες εφαρμογές το χρησιμοποιούν για μετάδοση μη «ευαίσθητων» πληροφοριών, π.χ. φωνή ή video ή για ανταλλαγή «μικρών» μηνυμάτων.

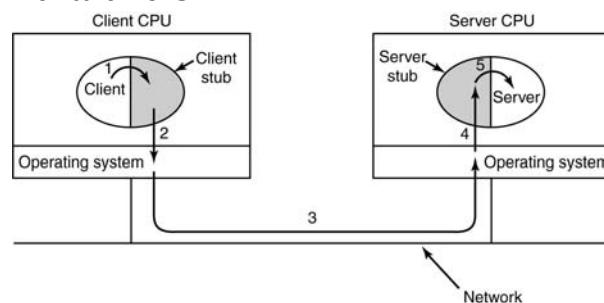
Επικεφαλίδα UDP



- UDP Length (συμπεριλαμβανομένων και των 8 bytes της επικεφαλίδας)
- Checksum (προαιρετικό): header, pseudoheader, data
 - ☐ Pseudoheader: IP source and destination addresses, protocol number and UDP Length

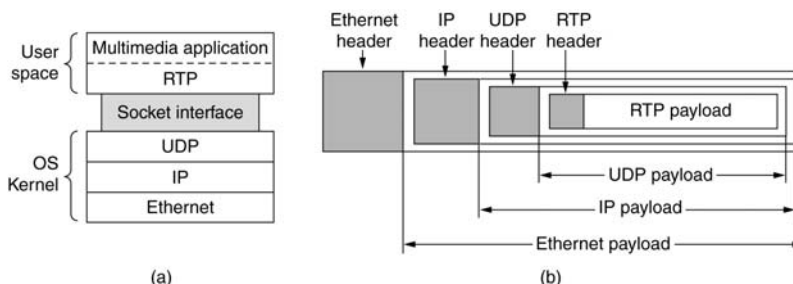
Remote Procedure Call (RPC)

- Εκτελεί μια διαδικασία (ρουτίνα) σε απομακρυσμένο υπολογιστή
- Η βασική ιδέα είναι να υπάρχει μια διαδικασία η οποία τρέχει σε απομακρυσμένο υπολογιστή αλλά να «φαίνεται» ότι τρέχει τοπικά.
- Τρέχει πάνω από UDP.

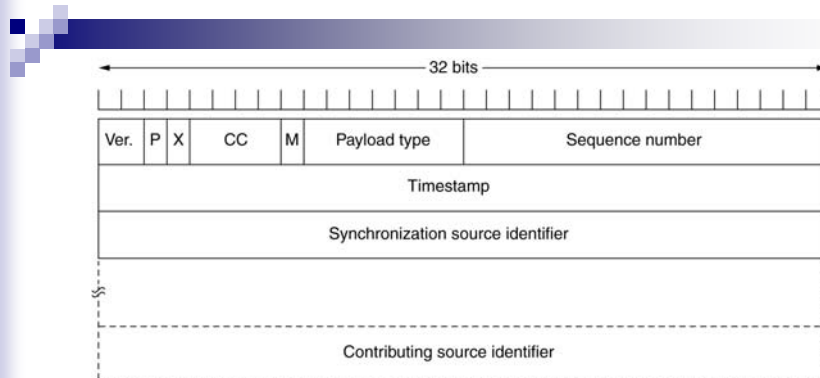


Real-Time Transport Protocol (RTP)

- Πρωτόκολλο το οποίο χρησιμοποιείται για εφαρμογές πραγματικού χρόνου (internet radio, internet telephony, music and video-on-demand, other multimedia).
- Τρέχει πάνω από το UDP
 - Πολυπλεξία διαφόρων πηγών σε ροή από UDP segments
 - Προσφέρει επιπρόσθετες υπηρεσίες



Η Επικεφαλίδα RTP



- P: Padding
- X: Extension Header
- CC: Contributing Sources
- M: Application specific
- Payload type: Encoding used
- SeqNum: detects lost packets
- Timestamp:
- Synchronization source identifier: multiplexing of sources
- Contributing source: mixers

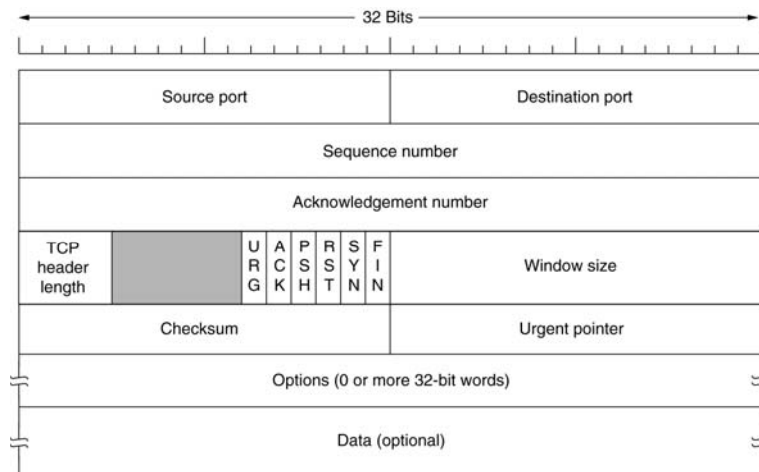
Transport Control Protocol (TCP)

- Προσφέρει υπηρεσίες αξιόπιστης μεταφοράς ρών από bytes.
 - Συνήθως χωρίζει τα δεδομένα σε segments με 1460 bytes (για να μην χρειάζεται να τεμαχιστούν από χαμηλότερα επίπεδα, π.χ., Ethernet μέγιστο πλαίσιο = 1500 bytes).
- Υποστηρίζει πολυπλεξία χρησιμοποιώντας *ports* (TSAP).
- Όλες οι συνδέσεις του TCP είναι διπλής κατεύθυνσης (full duplex)
- Υποστηρίζει επείγοντα δεδομένα (Urgent data).

Το Πρωτόκολλο TCP

- Κάθε TCP byte έχει τον δικό του 32-bit αύξων αριθμό (sequence number).
- Διαφορετικός αύξων αριθμός χρησιμοποιείται για τις επαληθεύσεις (ACKs).
- TCP Segments αποτελούνται από επικεφαλίδα με 20-bytes υποχρεωτικά καθώς και προαιρετικά πεδία.
- Χρησιμοποιεί τα πρωτόκολλα ολισθαίνοντος παραθύρου (sliding window: go back n , and selective repeat).
- Είναι σχεδιασμένο για να αποφεύγει αντίγραφα (duplicates) τα οποία έχουν καθυστερήσει πολύ ή κρυφτεί μέσα στο δίκτυο.

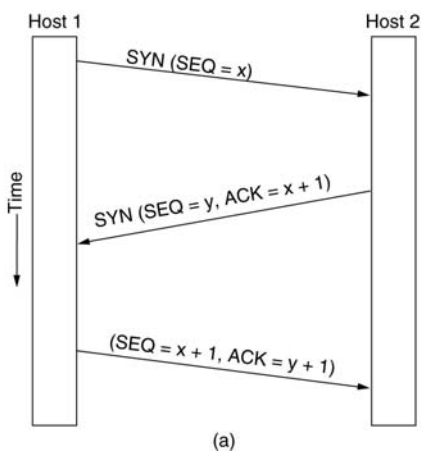
Η Επικεφαλίδα του TCP



Η Επικεφαλίδα του TCP

- TCP header length: Αριθμός των 32-bit words που υπάρχουν στην επικεφαλίδα TCP
- Flags
 - URG: επείγοντα δεδομένα που αρχίζουν από το σημείο στο οποίο δείχνει το Urgent Pointer
 - ACK: επαλήθευση
 - PSH: (push) Άμεση παράδοση στον παραλήπτη
 - RST: Αρχικοποίηση της σύνδεσης
 - SYN: Αίτημα για δημιουργία νέας σύνδεσης
 - FIN: Αίτημα για κλείσιμο της σύνδεσης
-
- Checksum: header, pseudoheader, data
 - Pseudoheader: IP source and destination addresses, protocol number and UDP Length
- Options: maximum TCP payload

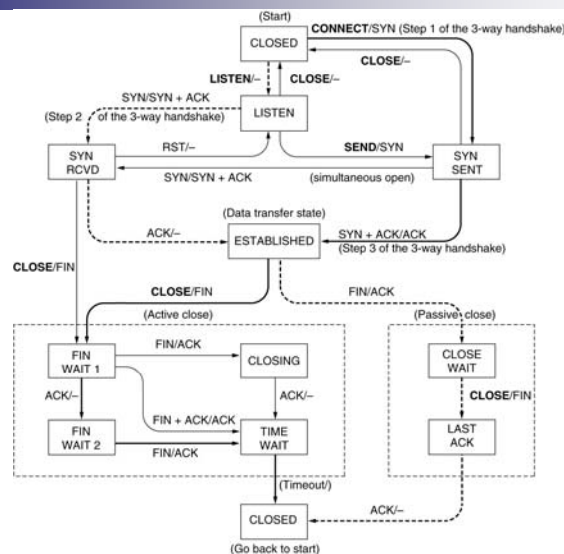
Σύνδεση TCP



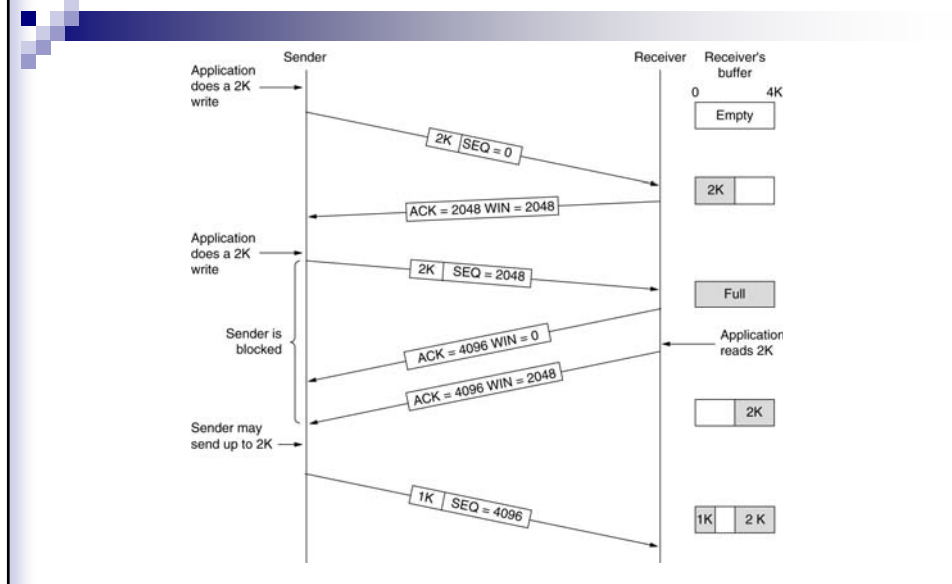
Οι Καταστάσεις (states) του TCP

- **CLOSED:** δεν υπάρχει ούτε αναμένεται σύνδεση
- **LISTEN:** Ο εξυπηρετητής (server) αναμένει σύνδεση
- **SYN RCVD:** Αίτημα για σύνδεση έχει παραληφθεί
- **SYN SENT:** Αίτημα για νέα σύνδεση έχει σταλεί
- **ESTABL:** Η σύνδεση έχει επιτευχθεί και ανταλλάσσονται δεδομένα
- **FIN WAIT 1:** Η εφαρμογή έχει στείλει αίτημα αποσύνδεσης
- **FIN WAIT 2:** Η άλλη πλευρά παρέλαβε το αίτημα αποσύνδεσης
- **TIMED WAIT:** Περιμένει μέχρις ότου όλα τα πακέτα να φύγουν από το δίκτυο
- **CLOSING:** Και οι δύο μεριές προσπαθούν να κλείσουν τη σύνδεση ταυτόχρονα.
- **CLOSE WAIT:** Η άλλη μεριά ενεργοποίησε αποσύνδεση
- **LAST ACK:** Περιμένει μέχρις ότου όλα τα πακέτα της σύνδεσης να φύγουν από το δίκτυο

Μηχανή Πεπερασμένων Καταστάσεων για Σύνδεση και Αποσύνδεση TCP.



Μετάδοση Δεδομένων Χρησιμοποιώντας Συνδέσεις TCP



Προβλήματα του TCP

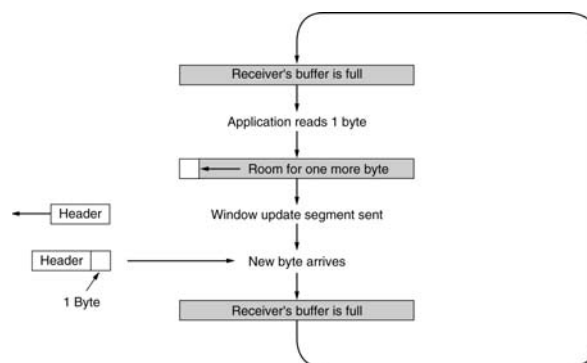
- Μετάδοση ενός Byte κάθε φορά
- Σύνδρομο Χαζού Παραθύρου (silly window syndrome).
- Επανάληψη του αύξοντος αριθμού (sequence number overflow)
- Απόδοση (keep the pipe full)

Μετάδοση Ενός Byte Κάθε Φορά

- Πρόβλημα στην απόδοση αφού αποστέλλονται 41-byte πακέτα με μόνο ένα byte πραγματικά δεδομένα!
- Λύση: Nagle's Algorithm
 - Όταν φτάνουν δεδομένα 1 byte κάθε φορά, τότε
 -
 - Αφού φτάσει η επαλήθευση, όλα τα bytes που έχουν συσσωρευτεί στην μνήμη αποστέλλονται μαζί στο επόμενο πακέτο και όλα τα υπόλοιπα φυλάγονται στη μνήμη μέχρι να επιστρέψει η επαλήθευση.
 - Η πιο πάνω λύση δεν δουλεύει πάντα...
- Τι συμβαίνει σε εφαρμογές που υπάρχουν δεδομένα που δεν μπορούν να περιμένουν (π.χ., θέση του ποντικιού (mouse position)).

Σύνδρομο Χαζού Παραθύρου (silly window syndrome).

- Η εφαρμογή στον παραλήπτη διαβάζει τα δεδομένα από τη μνήμη ένα byte κάθε φορά.



- Clark's Algorithm:

Επανάληψη του αύξοντος αριθμού (Sequence number wrap-around)

- 32-bit Αύξων Αριθμός
- Ο απαιτούμενος χρόνος επανάληψης του ίδιου αύξοντος αριθμού για διάφορες ταχύτητες μετάδοσης δίνεται στον πίνακα.
- Για υψηλές ταχύτητες, ο χρόνος επανάληψης του ίδιου αύξων αριθμού είναι πολύ μικρός! Μικρότερος από την μέγιστη ζωή ενός πακέτου στο Διαδίκτυο.
-

Bandwidth	Time Until Wrap Around
T1 (1.5 Mbps)	6.4 hours
Ethernet (10 Mbps)	57 minutes
T3 (45 Mbps)	13 minutes
FDDI (100 Mbps)	6 minutes
STS-3 (155 Mbps)	4 minutes
STS-12 (622 Mbps)	55 seconds
STS-24 (1.2 Gbps)	28 seconds

Απόδοση – Μικρό μέγιστο παράθυρο (Keep the Pipe Full)

- Το μέγιστο παράθυρο με 16-bit πεδίο είναι 65536.
- Σε ένα δίκτυο με RTT περίπου 100ms, ο αριθμός των bits που χρειάζεται για να γεμίσει το κανάλι σε διάφορες ταχύτητες μετάδοσης δίνεται στον πίνακα.
- Για υψηλές ταχύτητες, ο αποστολέας πρέπει να περιορίσει τον ρυθμό μετάδοσης με αποτέλεσμα το κανάλι να υποχρησιμοποιείται
-

Bandwidth	Delay x Bandwidth
T1 (1.5 Mbps)	18KB
Ethernet (10 Mbps)	122KB
T3 (45 Mbps)	549KB
FDDI (100 Mbps)	1.2MB
STS-3 (155 Mbps)	1.8MB
STS-12 (622 Mbps)	7.4MB
STS-24 (1.2 Gbps)	14.8MB

Επέκταση του TCP Χρησιμοποιώντας τα Παιδιά των Επιλογών

- Οι επεκτάσεις προσπαθούν να επιφέρουν όσο το δυνατό μικρότερες αλλαγές στο υπάρχον πρωτόκολλο TCP.
- Τα επικοινωνούντα μέρη συμφωνούν στην αρχή της συνδιάλεξης κατά πόσο θα χρησιμοποιήσουν τις επιλογές ή όχι.
- Για να λυθούν τα προαναφερθέντα προβλήματα, χρησιμοποιούνται οι ακόλουθες επιλογές:
 - 32-bit timestamp για να ξεχωρίζει τα πακέτα του κάθε «κύκλου»
 - Πολλαπλάσιο της μικρότερης μονάδας δεδομένων (multiplier). Δηλαδή αντί ο αύξων αριθμός να μετρά bytes, μετρά 2 ή 4 ή περισσότερα bytes.

Έλεγχος Συμφόρησης στο TCP (Congestion Control)

- Το πρωτόκολλο έχει δύο αντιφατικούς στόχους
 - Να **ελέγξει (μειώσει) τη συμφόρηση** και τα χαμένα πακέτα.
 - Να κάνει όσο το δυνατό καλύτερη χρήση των πόρων του δικτύου και να **μεγιστοποιήσει τη «διεκπεραιωτική» ικανότητα (throughput)** του δικτύου.
- Για να πετύχει τους στόχους του
 - Μόλις ανιχνεύσει συμφόρηση τότε μειώνει το ρυθμό με τον οποίο αποστέλλει πακέτα.
 -
 - Εάν δεν υπάρχει συμφόρηση τότε προσπαθεί να στείλει περισσότερα πακέτα για να αυξήσει το throughput.
- Για να υλοποιήσει τους στόχους του, ελέγχει **δυναμικά** το παράθυρο συμφόρησης.
 - Το TCP διατηρεί μια μεταβλητή: *congestionWindow* (διαφορετική από το *flowWindow*)
 - Το ενεργό παράθυρο είναι πάντα το **ελάχιστο** των δύο.

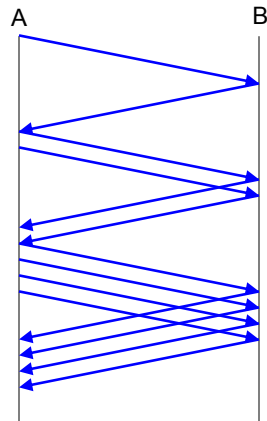
Φάσεις του Αλγορίθμου

- Το παράθυρο αυξάνεται σε δύο φάσεις
 -
 - *Congestion Avoidance*: Σε αυτή τη φάση το παράθυρο αυξάνεται γραμμικά με στόχο να αποφύγει τη συμφόρηση.
 - Οι δύο φάσεις χωρίζονται από ένα threshold.
 - Εάν το παράθυρο είναι μικρότερο από το threshold τότε χρησιμοποιεί το slow start
 - Εάν το παράθυρο είναι μεγαλύτερο από το threshold τότε χρησιμοποιεί το congestion avoidance

Slow Start vs. Congestion Avoidance

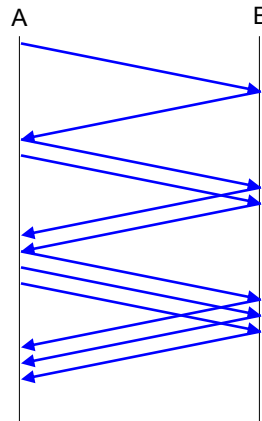
■ *Slow Start*:

- Για κάθε επαλήθευση (ACK) που παραλαμβάνεται, το παράθυρο αυξάνεται κατά ένα.



■ *Congestion Avoidance*:

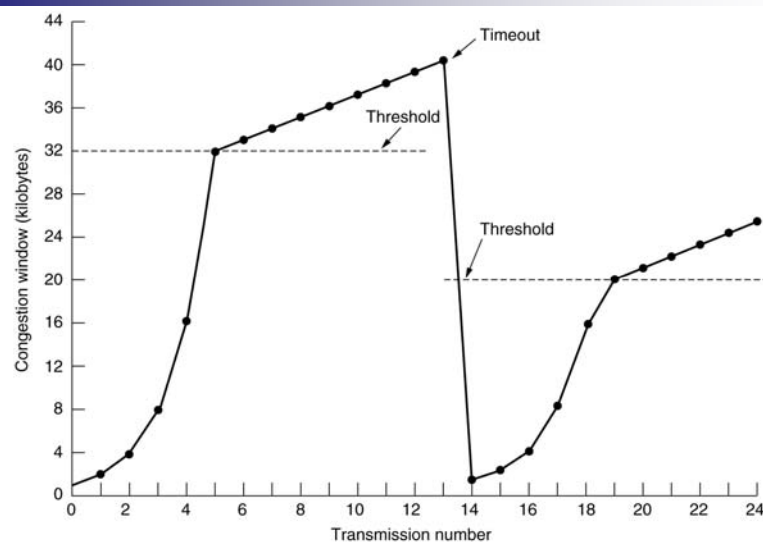
- Για κάθε παράθυρο που παραλαμβάνεται επιτυχώς, το παράθυρο αυξάνεται κατά ένα.



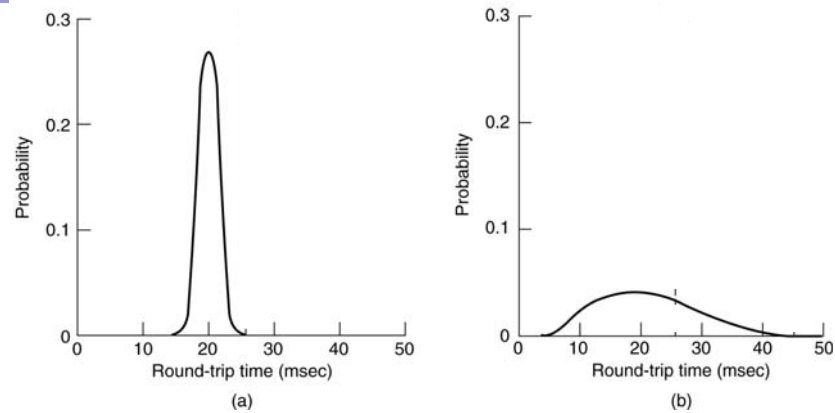
Απώλεια Πακέτου

- Σε περίπτωση απώλειας πακέτου, τότε το TCP αυτόματα υποθέτει συμφόρηση και συνεπώς μειώνει το ρυθμό μετάδοσης
 - Απώλεια πακέτου ανιχνεύεται αφού λήξει το ανάλογο χρονόμετρο (timer).
- Σε περίπτωση συμφόρησης:
 - $\text{Threshold} := \text{congestionWindow}/2$
 -
- Η απόδοση του πρωτοκόλλου είναι πολύ **ευαίσθητη στον κατάλληλο καθορισμό του χρονόμετρου** (γιατί;)
 - Ο καθορισμός του χρονόμετρου είναι δύσκολο πρόβλημα (γιατί;)

Παράδειγμα του Παραθύρου του TCP



Καθορισμός του Χρονομέτρου



Υπολογισμός του RTT

- Το TCP εκτιμά το RTT χρησιμοποιώντας τον αλγόριθμο
 - M είναι το τελευταίο δείγμα RTT
 - $\alpha = 7/8$
- Το χρονόμετρο καθορίζεται σαν
 - Στο αρχικό TCP, $\beta = 2$.
- **Πρόβλημα:** Δεν υπολογίζει την απόκλιση (variance) της κατανομής

Υπολογισμός του RTT

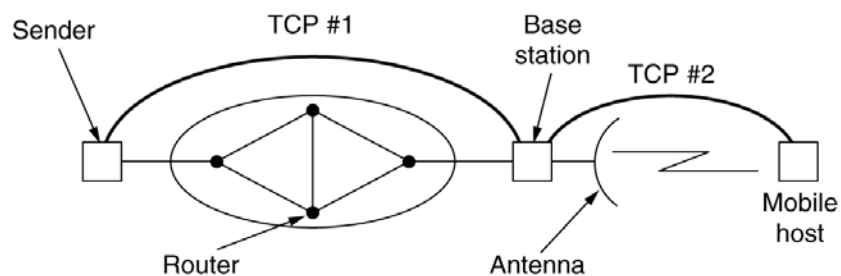
- Εκτίμηση της απόκλισης D (variance)
 - όπου
 - M είναι το τελευταίο δείγμα RTT
 - $\alpha = 7/8$
 - Το χρονόμετρο καθορίζεται σαν
- **Πρόβλημα:** Τι συμβαίνει όταν το δείγμα έχει σταλεί περισσότερες από μία φορές;
- **Karn's Algorithm:** Πακέτα που έχουν σταλεί πέραν της μίας φορές **δεν** υπολογίζονται στο RTT ή D

Εναλλακτικός Τρόπος Ανίχνευσης Απωλειών Λόγω Συμφόρησης

- Η απόδοση του αλγόριθμου «αποσυμφόρησης» του TCP βασίζεται σε μεγάλο βαθμό στην «ακρίβεια» του χρονόμετρου.
 - Εάν θέσουμε το χρόνο συντηρητικά (μεγάλη καθυστέρηση), τότε η πηγή θα **καθυστερεί υπερβολικά πριν αποκαταστήσει την επικοινωνία**.
 - Εάν θέσουμε το χρόνο πολύ μικρό, τότε θα έχουμε **αχρείαστες επαναμεταδόσεις**.
- Άλλες εκδόσεις του TCP έχουν επιπρόσθετους τρόπους ανίχνευσης απωλειών.
 -
 - Ο αποστολέας, όταν παραλάβει **n συνεχόμενες επαληθεύσεις** του ίδιου πακέτου, τότε υπολογίζει πως το πακέτο έχει χαθεί και ξεκινά επαναμετάδοση καθώς και το μηχανισμό αποσυμφόρησης (υπάρχει εισήγηση για $n=3$).
 - Ο αλγόριθμος χρησιμοποιεί επίσης το **χρονόμετρο** σαν επιπρόσθετη δικλίδα ασφαλείας.

TCP σε Ασύρματα Δίκτυα

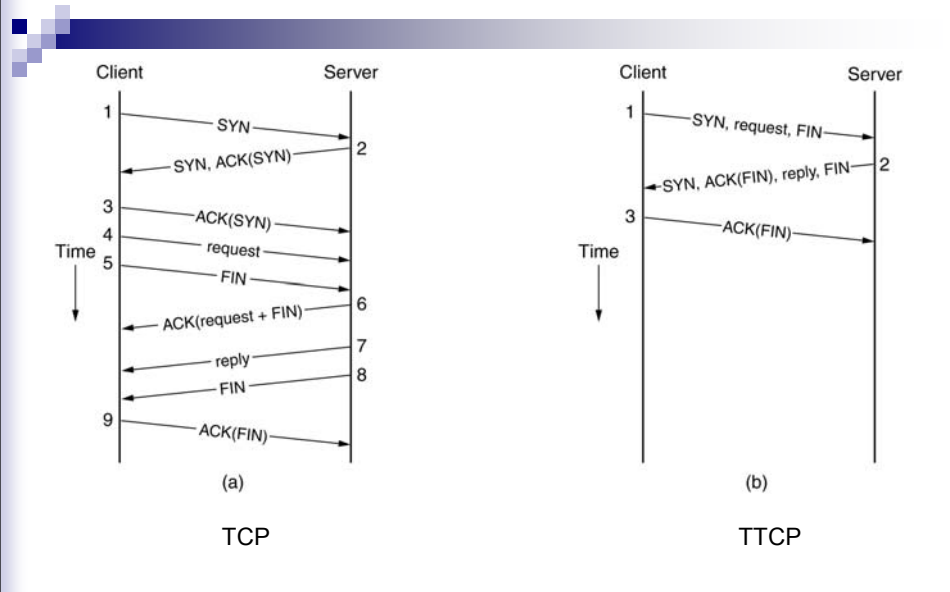
- **Πρόβλημα:** όταν η πηγή ανιχνεύσει απώλεια υποθέτει ότι προήλθε εξ αιτίας συμφόρησης, όμως σε ασύρματα δίκτυα οι απώλειες λόγω μετάδοσης είναι συχνό φαινόμενο!



Transactional TCP

- Ας υποθέσουμε πως ένας πελάτης θέλει απλά να ζητήσει μια πληροφορία από ένα εξυπηρετητή, να πάρει την απάντηση και να τελειώσει την σύνδεση. Επίσης, ο πελάτης αναμένει πως η απάντηση είναι αξιόπιστη (π.χ., μια μεταφορά χρημάτων).
- Με το TCP το πιο πάνω σενάριο χρειάζεται μια σειρά με **εννέα** μηνύματα το οποία ανοίγουν μια σύνδεση, ανταλλάσσουν πληροφορίες και στη συνέχεια κλείνουν την σύνδεση.
- Αυτή η διαδικασία δεν είναι καθόλου αποδοτική και χρειάζεται κάτι πιο απλό.
 - Το UDP θα ήταν καλή επιλογή (από πλευράς απόδοσης) αλλά δεν είναι αξιόπιστο (δεν υποστηρίζει εύρεση και αναμετάδοση σφαλμάτων)

Transactional TCP



Θέματα Απόδοσης

■ Προβλήματα Απόδοσης

- Συμφόρηση (φτάνουν περισσότερα πακέτα από όσα μπορεί να προσθήσει ο δρομολογητής)
- Μη ισοζυγισμένοι πόροι
 - Π.χ., gigabit γραμμή συνδεδεμένη με κάποιον αργό Η/Υ
- Συγχρονισμός (synchronous overloads)
 - Ένα γεγονός προκαλεί μια άλλη σειρά από αλυσιδωτά προβλήματα
- Ρύθμιση των πρωτοκόλλων (bandwidth-delay product)

■ Μετρήσεις για την απόδοση του δικτύου

- Ξεχωρίστε τις σχετικές παραμέτρους και μετρικές και προσπαθήστε να καταλάβετε τι συμβαίνει.