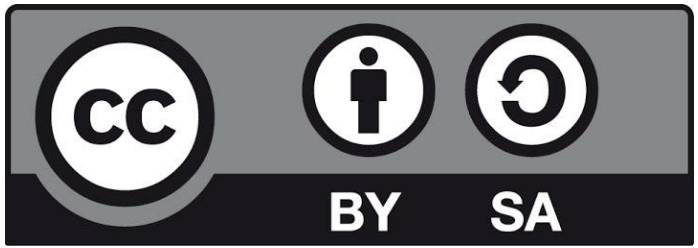
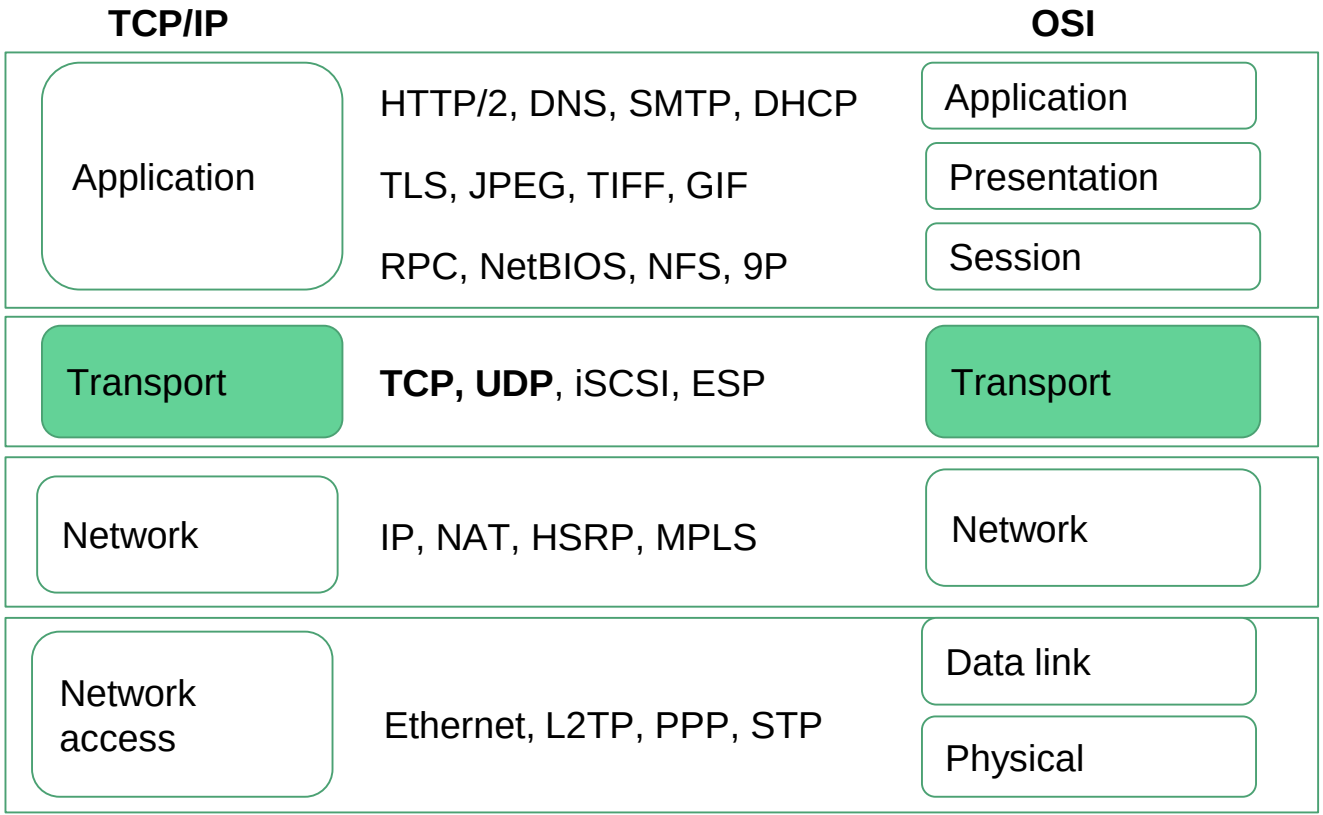
Το στρώμα μεταφοράς OSI

Εισηγητής: Χρήστος Δαλαμάγκας
cdalamagkas@gmail.com

Άδεια χρήσης



Το στρώμα μεταφοράς στα δυο εννοιολογικά μοντέλα



Εισαγωγή στο στρώμα μεταφοράς

- Ορισμός: Σύνολο πρωτοκόλλων που παρέχουν υπηρεσίες μεταφοράς και συνδεσιμότητας από άκρο σε άκρο.
- Γιατί χρειάζεται το στρώμα μεταφοράς:
  - Το πακέτο IP είναι αυτοδύναμο, δηλαδή σκοπός του IP είναι να προωθηθεί ένα μεμονωμένο πακέτο στον προορισμό του μέσω του Διαδικτύου
  - Στο IP δεν μας ενδιαφέρει:
    - Αν ο παραλήπτης είναι έτοιμος να λάβει πακέτα
    - Αν τα πακέτα είναι μέρος μιας συνδιάλεξης
    - Αν τα πακέτα χάνονται στη διαδρομή τους προς τον προορισμό

Υπηρεσίες του στρώματος μεταφοράς

- Το στρώμα μεταφοράς καλύπτει τις ελλείψεις του IP για παροχή υπηρεσιών επικοινωνίας από άκρο σε άκρο. Συγκεκριμένα παρέχει:
  - Μηχανισμό για παράδοση ενός πακέτου στην κατάλληλη εφαρμογή (θύρες TCP/UDP)
  - Έλεγχο ροής των εισερχόμενων πακέτων, ώστε ο παραλήπτης να μην κατακλύζεται από πακέτα
  - Αποφυγή συμφόρησης
  - Έλεγχο σφαλμάτων στα πακέτα και αίτηση επανεκπομπής αν χρειάζεται
  - Πολύπλεξη συνομιλιών

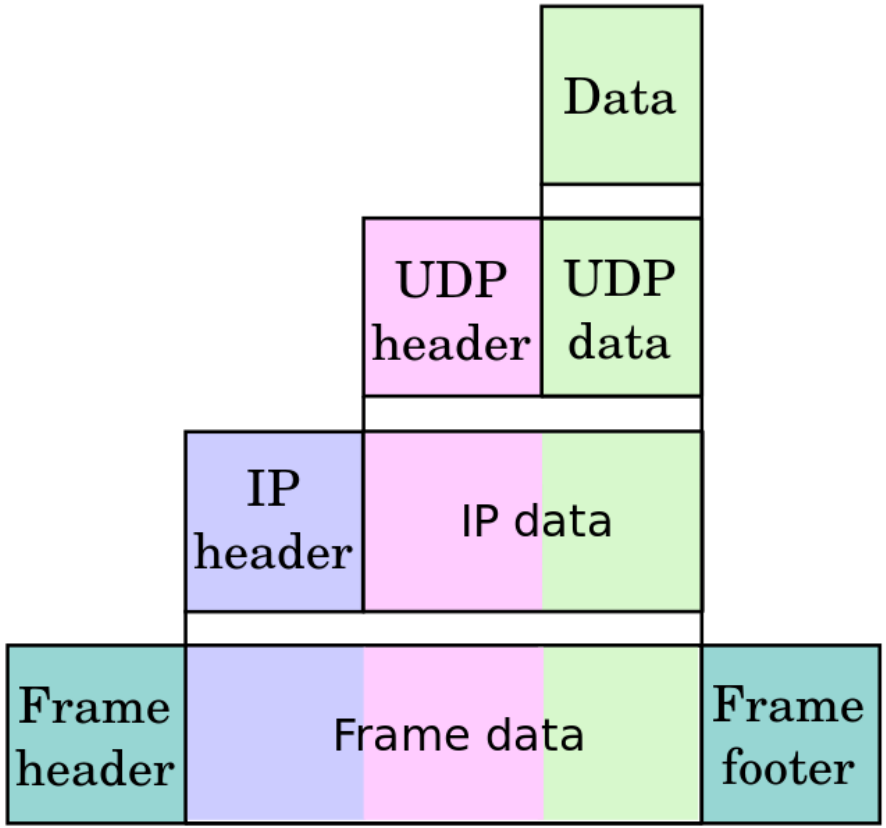
Ενθυλάκωση

Message / Μήνυμα

Segment / Τμήμα

Packet / Πακέτο

Frame / Πλαίσιο



Application

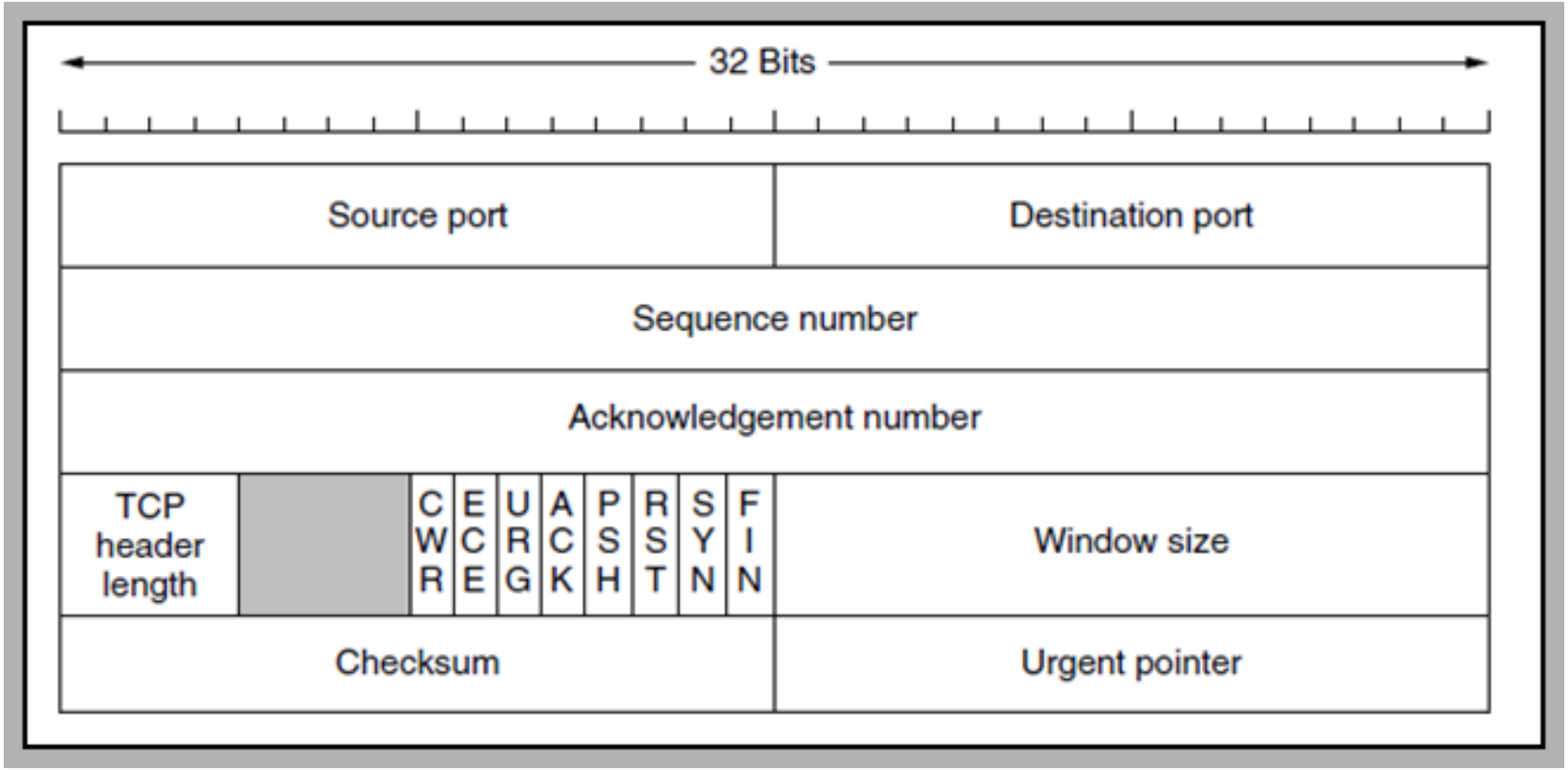
Transport

Internet

Link

Το πρωτόκολλο TCP

Η κεφαλίδα TCP



Η κεφαλίδα TCP

TCP Header

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port															
4	32	Sequence number																															
8	64	Acknowledgment number (if ACK set)																															
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size														
16	128	Checksum																Urgent pointer (if URG set)															
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																															

Το πρωτόκολλο TCP

- Βασικό πρωτόκολλο μεταφοράς που παρέχει υπηρεσίες **αξιόπιστης** σύνδεσης δυο εφαρμογών
- Πριν οι εφαρμογές ξεκινήσουν την συνομιλία, εγκαθιδρύουν μια **συνεδρία TCP.** Η συνεδρία εγκαθιδρύεται με τη μέθοδο της τριπλής χειραψίας
- Αφού εγκαθιδρυθεί η συνεδρία, κάθε TCP segment συνοδεύεται από έναν αριθμό ακολουθίας (sequence number), ο οποίος αυξάνεται αναλογικά με το πλήθος των byte που μεταδίδονται
- Ο παραλήπτης χρησιμοποιεί τους αριθμούς ακολουθίας για να βάλει τα segment που λαμβάνει στη σωστή σειρά

Το πρωτόκολλο TCP

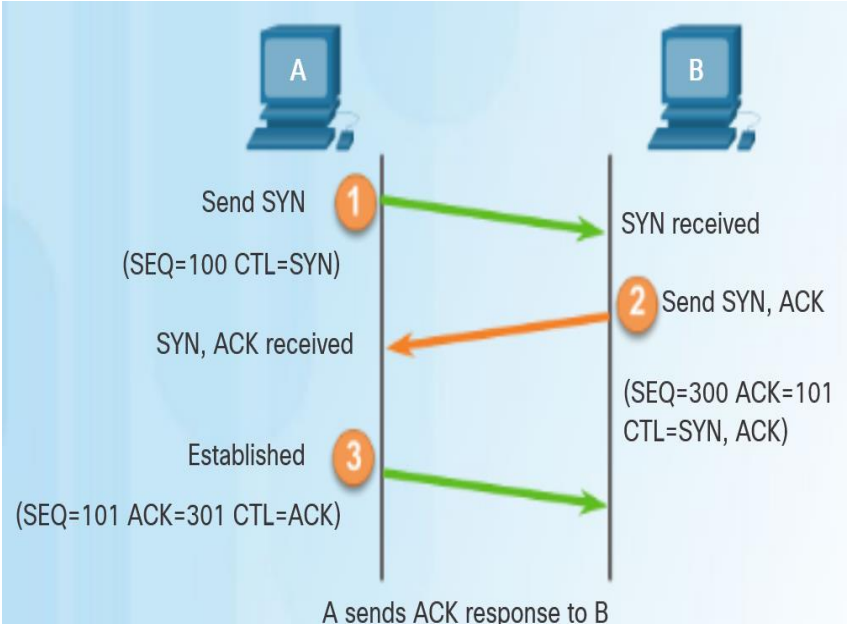
- Ο παραλήπτης χρησιμοποιεί τον αριθμό ακολουθίας για να διαπιστώσει αν κάποιο segment χάθηκε κατά τη μετάδοση
- Αν συμβεί κάτι τέτοιο, τότε ο παραλήπτης ζητά από τον αποστολέα να στείλει ξανά όλα τα segment από εκείνο το σημείο
- Για κάθε τμήμα δεδομένων που λαμβάνει ο παραλήπτης, στέλνει αντίστοιχο segment επιβεβαίωσης:
    - Στο πεδίο Acknowledgment Number μπαίνει το πιο πρόσφατο sequence number που επιβεβαιώνεται
    - Η σημαία ACK γίνεται 1

Το παράθυρο TCP

- Κανονικά, κάθε segment απαιτεί αντίστοιχη επιβεβαίωση
- Για να γίνονται πιο γρήγορα οι μεταδόσεις, υποστηρίζεται η δυνατότητα να επιβεβαιώνονται αθροιστικά πολλά segment
  - Δηλαδή, ένα acknowledgment segment να επιβεβαιώνει πολλά segment που ελήφθησαν μέχρι εκείνη τη στιγμή
  - Το πλήθος των byte για τα οποία δεν χρειάζεται ενημέρωση, ονομάζεται μέγεθος παραθύρου (window size)
  - Αν η σύνδεση είναι αξιόπιστη, τότε το μέγεθος του παραθύρου αυτού τείνει να αυξάνεται. Ο αλγόριθμος για την αύξηση του παραθύρου ονομάζεται «κυλιόμενο παράθυρο» (sliding window)
  - Αν η σύνδεση είναι αναξιόπιστη, τότε το μέγεθος παραθύρου μειώνεται, με αποτέλεσμα να μειώνεται η ροή των τμημάτων

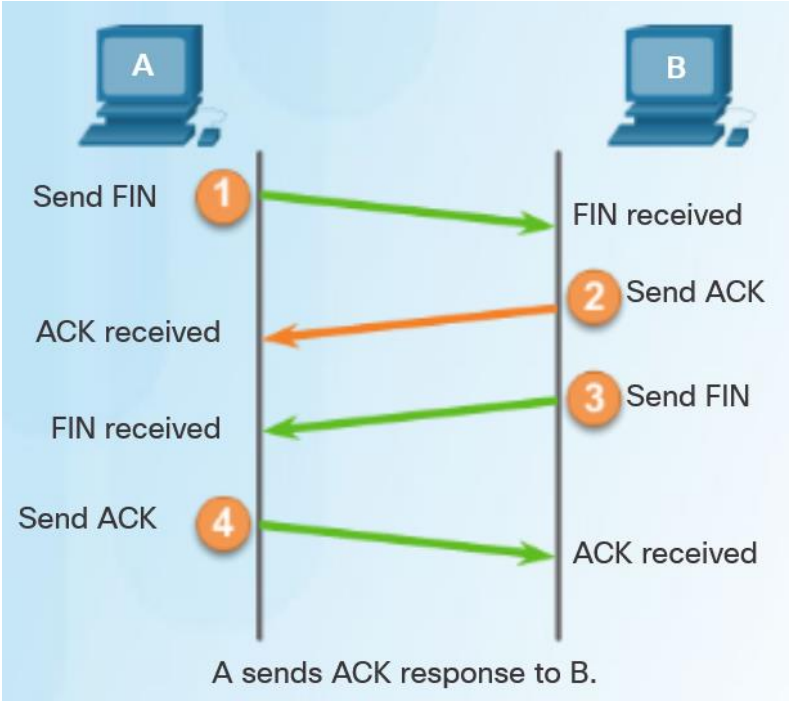
Εγκαθίδρυση συνεδρίας / Η χειραψία TCP

- Η εγκαθίδρυση μιας σύνδεσης / συνεδρίας TCP γίνεται με μια χειραψία TCP
- Σκοπός της χειραψίας είναι:
  - Να επαληθεύσει ότι ο απομακρυσμένος υπολογιστής είναι διαθέσιμος
  - Να προετοιμάσει τον απομακρυσμένο υπολογιστή για λήψη των segment
- Για τη χειραψία ενεργοποιούνται οι σημαίες SYN και ACK



Απόλυση συνεδρίας TCP

- Ο υπολογιστής που αποφασίζει τη λήξη της συνεδρίας, στέλνει ένα TCP segment με ενεργοποιημένη τη σημαία FIN
- Το άλλο άκρο στέλνει δυο segment, ένα επιβεβαίωσης και ένα με ενεργοποιημένη τη σημαία FIN
- Ο υπολογιστής απαντά με segment επιβεβαίωσης και η συνεδρία λήγει ομαλά



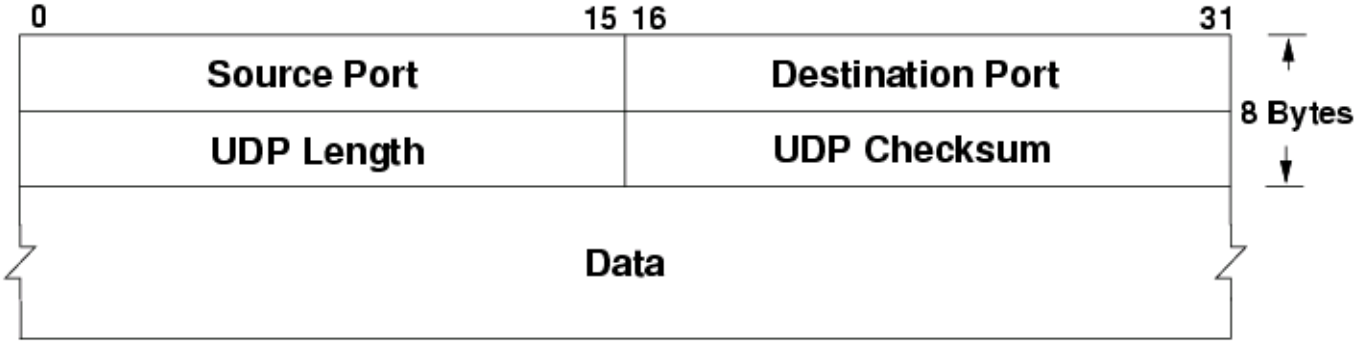
Σημαίες TCP

- **SYN**: Ενεργοποιείται για τη διαδικασία της χειραψίας
- **ACK**: Χρησιμοποιείται όταν το segment επιβεβαιώνει τη λήψη ενός ή περισσοτέρων τμημάτων. Μόνο όταν ACK=1 ο παραλήπτης διαβάζει το πεδίο Acknowledgment Number της κεφαλίδας
- **FIN**: Δηλώνει την επιθυμία ομαλού τερματισμού μιας συνεδρίας
- **RST**: Δηλώνει αδυναμία εγκαθίδρυσης συνεδρίας ή αδυναμία συνέχισής της
- URG
- PSH

Το πρωτόκολλο UDP

Το πρωτόκολλο UDP

- Ασυνδεσμοστραφές και αναξιόπιστο
- Δεν προσφέρει καμία από τις υπηρεσίες ελέγχου ροής και εγκαθίδρυσης συνεδριών του TCP
- Μόνη υπηρεσία η παράδοση των δεδομένων στην κατάλληλη εφαρμογή



Θύρες και sockets

Sockets

- Πώς ένας υπολογιστής μπορεί να εκτελεί πολλές εφαρμογές που «ακούν» στην ίδια διεύθυνση IP;
  - Κάθε υπολογιστής διαθέτει συνολικά 65.536 θύρες για TCP και άλλες τόσες για UDP
  - Οι θύρες δεν είναι φυσικές, αλλά λογικές, στο επίπεδο του πυρήνα του ΛΣ
  - Κάθε εφαρμογή ακούει σε μια από αυτές τις θύρες
  - Όταν θελήσουμε να στείλουμε ένα δικτυακό μήνυμα σε μια εφαρμογή, πρέπει να την προσδιορίσουμε συνδυάζοντας την IP του εξυπηρετητή με τη θύρα στην οποία ακούει η εφαρμογή
    - Δοκιμάστε να προηγηθείτε στη διεύθυνση **www.iekalfa.gr:443**
- Ο συνδυασμός της IP με μια συγκεκριμένη θύρα ονομάζεται socket. Πχ www.iekalfa.gr:443 ή 104.25.175.109:443

Πώς δουλεύει η επικοινωνία με socket (Παράδειγμα: web browsing)

- Ο υπολογιστής/πελάτης που έχει την διεύθυνση 83.212.112.9 θέλει να συνδεθεί στο socket iekalfa.gr:443
- Μέσω του DNS, το iekalfa.gr:443 μεταφράζεται σε 104.25.175.109:443
- Ο υπολογιστής ξέρει από πριν τι υπηρεσία θα χρησιμοποιήσει (HTTPS), άρα ξέρει ότι θα πρέπει να χρησιμοποιήσει TCP
- Πριν στείλει το πρώτο πακέτο προς τον server, ο υπολογιστής ανοίγει μια τυχαία θύρα TCP (έστω η 10543), η οποία ονομάζεται source port. Η θύρα αυτή μπαίνει στο κατάλληλο πεδίο του τμήματος TCP
- Το πακέτο φεύγει προς τον προορισμό από την IP του υπολογιστή και τη θύρα προέλευσης (source port)

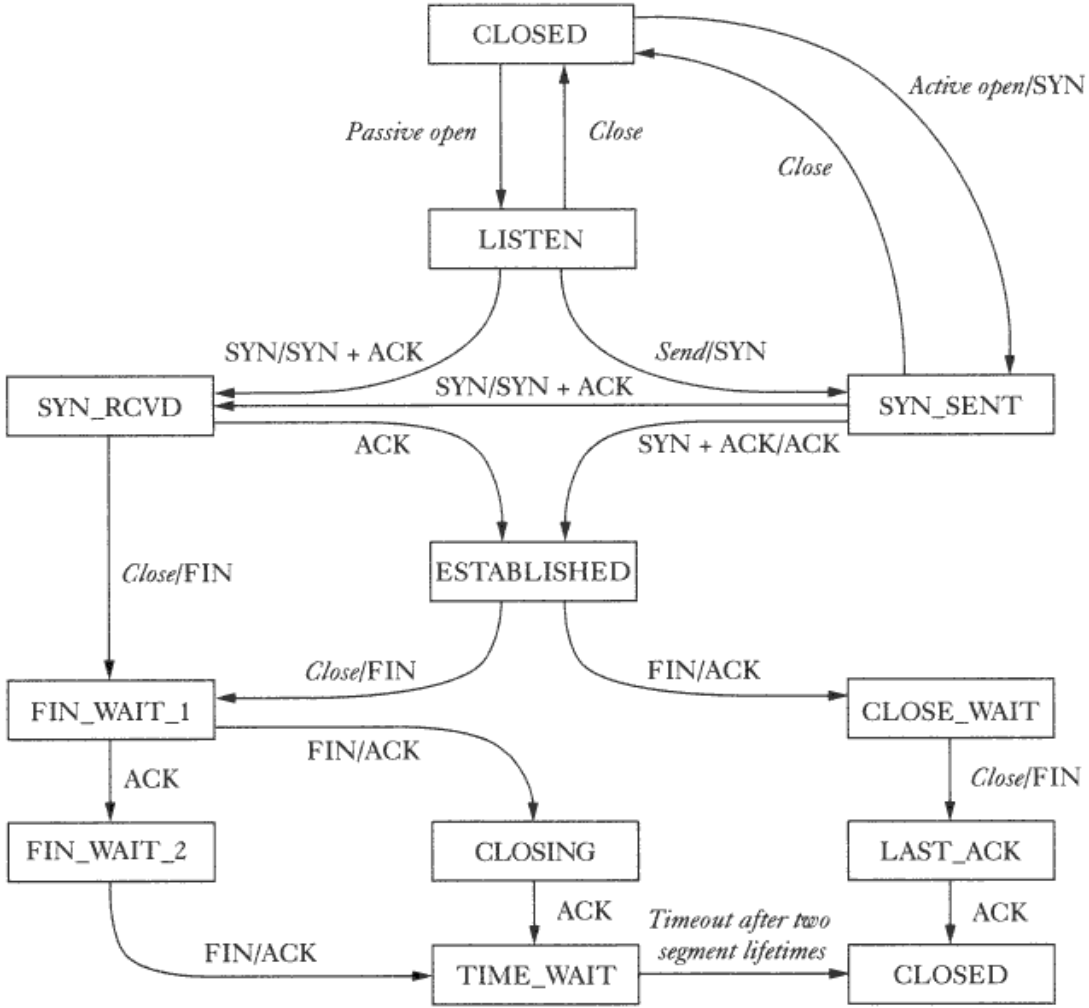
Πώς δουλεύει η επικοινωνία με socket (Παράδειγμα: web browsing)

- Ο server λαμβάνει ένα πακέτο IP, το οποίο μέσα του έχει ένα τμήμα TCP
- Διαβάζοντας το τμήμα TCP, ο server διαπιστώνει ότι πρέπει να το παραδώσει στη θύρα 443, στην οποία ακούει ο web server
- Ο web server λαμβάνει την ωφέλιμη πληροφορία του τμήματος, την επεξεργάζεται και προετοιμάζει την απάντηση
- Η απάντηση έχει IP προορισμού την IP του υπολογιστή και τη θύρα προέλευσης. Δηλαδή την 83.212.112.9:10543
- Η απάντηση παραδίδεται στη θύρα που ακούει ο web browser του πελάτη και έτσι η συνομιλία συνεχίζεται

Καταστάσεις θυρών

Μια θύρα μπορεί να βρίσκεται σε μια από τις εξής καταστάσεις:

- **CLOSED**: Καμία εφαρμογή δεν ακούει σε αυτή την θύρα. Αν στείλουμε ένα TCP SYN σε αυτή τη θύρα, θα λάβουμε TCP RST από τον πυρήνα του ΛΣ
- **LISTEN**: Υπάρχει μια εφαρμογή που ακούει παθητικά σε αυτή τη θύρα
- **ESTABLISHED**: Έχει εγκαθιδρυθεί μια συνεδρία TCP
- Και πολλές άλλες ενδιάμεσες ...


CLOSED
Active open/SYN
Passive open
Close
Close
LISTEN
SYN/SYN + ACK
Send/SYN
SYN/SYN + ACK
SYN_RCVD
SYN_SENT
ACK
SYN + ACK/ACK
Close/FIN
ESTABLISHED
Close/FIN
FIN/ACK
FIN_WAIT_1
CLOSE_WAIT
FIN/ACK
ACK
Close/FIN
FIN_WAIT_2
CLOSING
LAST_ACK
ACK
ACK
Timeout after two segment lifetimes
FIN/ACK
TIME_WAIT
CLOSED

Κατηγοριοποίηση θυρών

Ομάδα θυρών	Περιγραφή
0 μέχρι 1023	Πολύ γνωστές θύρες
1024 μέχρι 49151	Καταχωρημένες θύρες από την IANA
49152 μέχρι 65535	Δυναμικές θύρες

- Πολύ γνωστές θύρες: Τις χρησιμοποιούν πολύ γνωστές εφαρμογές του Διαδικτύου. Για να «ακούσει» μια εφαρμογή σε μια από αυτές τις θύρες, πρέπει να εκτελεστεί με δικαιώματα διαχειριστή / root
- Οι καταχωρημένες θύρες αφορούν διάφορα πρωτόκολλα και εφαρμογές που έχουν καταχωρηθεί στην IANA
- Οι δυναμικές θύρες ανοίγονται από εφαρμογές/πελάτες που ξεκινούν μια μετάδοση TCP/UDP

Well-Known Port Numbers

Port Number	Protocol	Application	Acronym
20	TCP	File Transfer Protocol (data)	FTP
21	TCP	File Transfer Protocol (control)	FTP
22	TCP	Secure Shell	SSH
23	TCP	Telnet	–
25	TCP	Simple Mail Transfer Protocol	SMTP
53	UDP, TCP	Domain Name Service	DNS
67	UDP	Dynamic Host Configuration Protocol (server)	DHCP
68	UDP	Dynamic Host Configuration Protocol (client)	DHCP
69	UDP	Trivial File Transfer Protocol	TFTP
80	TCP	Hypertext Transfer Protocol	HTTP
110	TCP	Post Office Protocol version 3	POP3
143	TCP	Internet Message Access Protocol	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Hypertext Transfer Protocol Secure	HTTPS

Σύγκριση TCP / UDP

Συγκριτικός πίνακας TCP / UDP

Κριτήριο	TCP	UDP
Τύπος σύνδεσης	Συνδεμοστραφές	Χωρίς σύνδεση (connectionless)
Καταλληλόλητα χρήσης	Βολικό για εφαρμογές που απαιτούν υψηλή αξιοπιστία, χωρίς ο χρόνος παράδοσης να είναι κρίσιμος	Βολικό για εφαμογές που χρειάζονται γρήγορη μετάδοση και που απαντούν σύντομα ερωτήματα από πολλούς πελάτες
Ταξινόμηση δεδομένων		
Ταχύτητα	Χαμηλή	Υψηλή
Αξιοπιστία	Απόλυτη εγγύηση για την παράδοση όλων των segment στη σωστή σειρά	Χωρίς εγγυήσεις για την παράδοση των segment
Επιβάρυνση	Υψηλή	Χαμηλή
Έλεγχος ροής	Ναι	Όχι
Έλεγχος σφαλμάτων	Ναι, ζητείται επανεκπομπή των segment που έχουν σφάλματα	Ναι, απορρίπτεται το segment που έχει σφάλμα