Δίκτυα Υπολογιστών

Το επίπεδο εφαρμογής (application layer)

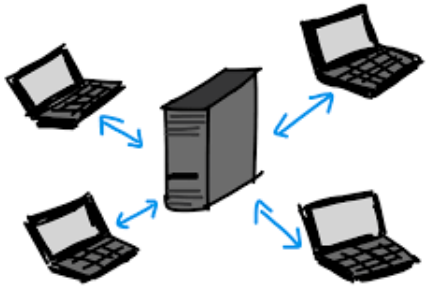
Κ. Βασιλάκης

Μεγάλο μέρος του περιεχομένου, προέρχεται από τις παρουσιάσεις (στα αγγλικά) των συγγραφέων του βιβλίου: Δικτύωση Υπολογιστών, προσέγγιση από πάνω προς τα κάτω, 6η έκδοση, (Computer Networking: A Top Down Approach), Jim Kurose & Keith Ross.

CC BY SA

Περίγραμμα – ενότητες που εξετάζονται

- Αρχές δικτυακών εφαρμογών.
- Αρχιτεκτονικές.
  - Μοντέλα υπηρεσιών επιπέδου μεταφοράς
  - Μοντέλο πελάτη εξυπηρετητή (client-server)
  - Μοντέλο ομότιμων (peer-to-peer)
- Επικοινωνία εφαρμογών.
  - Διεπαφές (sockets)
  - Διεργασίες (process)
  - Διεθυνσιοδότηση διεργασιών
- Χαρακτηριστικά υπηρεσιών επιπέδου μεταφοράς.
- Πρωτόκολλα επιπέδου εφαρμογών.



http:pixabay.com

CC BY SA

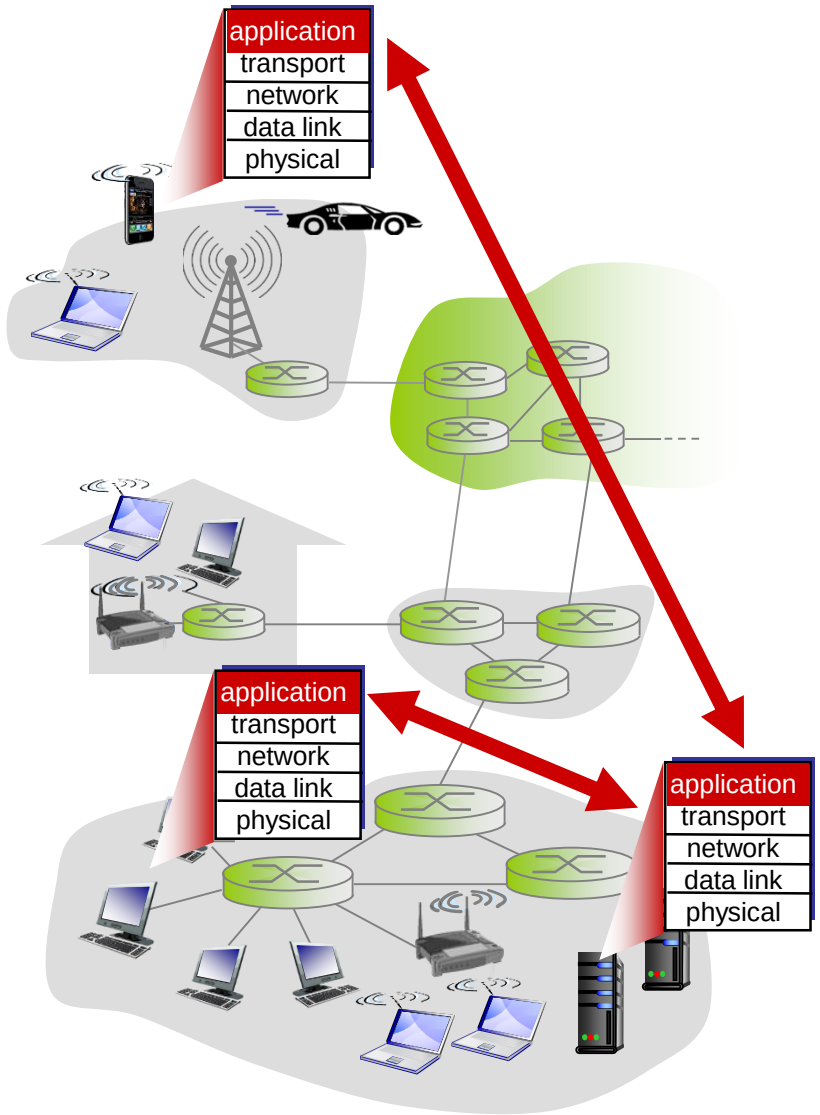
Κάποιες γνωστές δικτυακές εφαρμογές

- Ηλεκτρονικό ταχυδρομείο (e-mail).
- Παγκόσμιος Ιστός (world wide web -www).
- Κοινωνική δικτύωση (social networking).
- Ροές αποθηκευμένου βίντεο (streaming stored video clips).
- Διαδικτυακή τηλεφωνία (voice over IP).
- Ροές ζωντανές μεταδόσεων (real-time media stream delivery).
- Τηλεσυνδιάσκεψη πραγματικού χρόνου (real-time video conferencing).
- Δικτυακά παιχνίδια πολλών χρηστών (multi-user network games).
- Κοινή χρήση αρχείων μεταξύ ομότιμων (P2P file sharing).
- Στιγμιαία μηνύματα (instant messaging, chatting).

CC BY SA

Δημιουργώντας μια δικτυακή εφαρμογή

- Ανάπτυξη προγραμμάτων που
  - εκτελούνται σε (διαφορετικά) τερματικά συστήματα (hosts),
  - επικοινωνούν πάνω από τη υποδομή του δικτύου (π.χ. το λογισμικό του εξυπηρετητή web επικοινωνεί με το λογισμικό του browser).
- Δεν υπάρχει ανάγκη να γραφεί λογισμικό για συσκευές του πυρήνα του δικτύου.
  - Οι συσκευές του πυρήνα του δικτύου δεν τρέχουν εφαρμογές του χρήστη.
  - Το ότι οι εφαρμογές βρίσκονται στα τερματικά συστήματα επιτρέπει την ταχεία ανάπτυξη και διάδοσή τους.

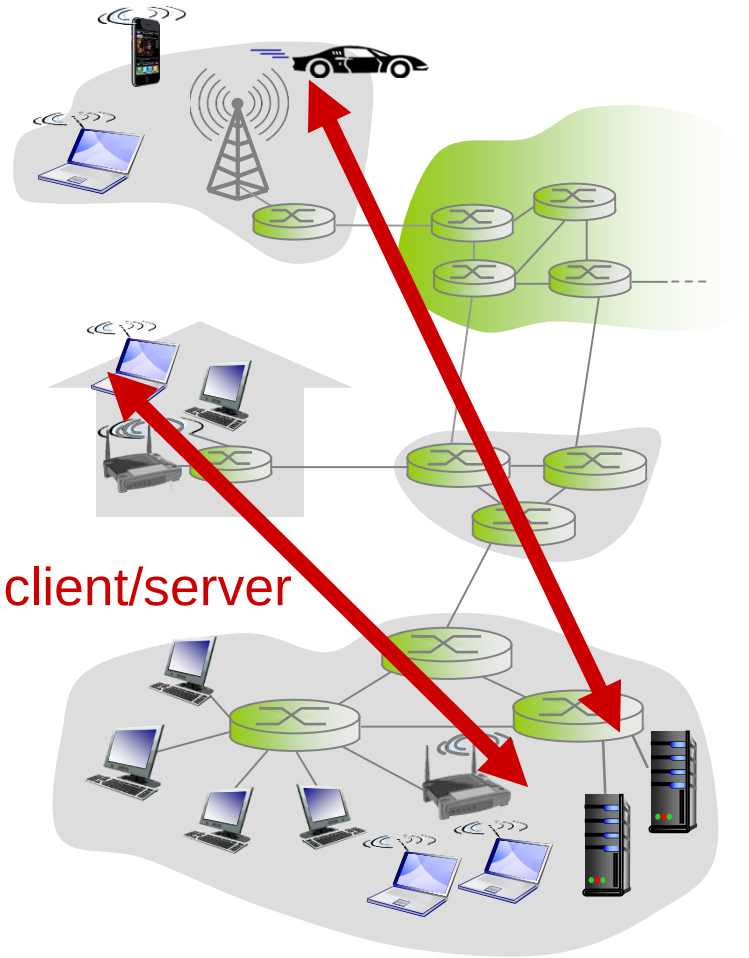


Αρχιτεκτονικές δικτυακών εφαρμογών

- Η αρχιτεκτονική μιας εφαρμογής σχεδιάζεται από κάποια ομάδα ανάπτυξης του λογισμικού.
- Οι εφαρμογές στηρίζονται στη σταθερή αρχιτεκτονική του δικτύου (5ο επίπεδο) που προσφέρει ένα συγκεκριμένο σύνολο υπηρεσιών.
- Σήμερα χρησιμοποιούνται κυρίως δυο αρχιτεκτονικές ανάπτυξης εφαρμογών δικτύου:
  - *Πελάτη-εξυπηρετητή* (client-server)
  - *Μεταξύ ομότιμων* (peer-to-peer -P2P)
- Υπάρχουν όμως και εφαρμογές που χρησιμοποιούν και τις δύο αυτές αρχιτεκτονικές ταυτόχρονα (υβριδική αρχιτεκτονική) π.χ. οι εφαρμογές ανταλλαγής άμεσων μηνυμάτων (instant messaging), η εφαρμογή Skype κλπ.

Αρχιτεκτονική πελάτη-εξυπηρετητή

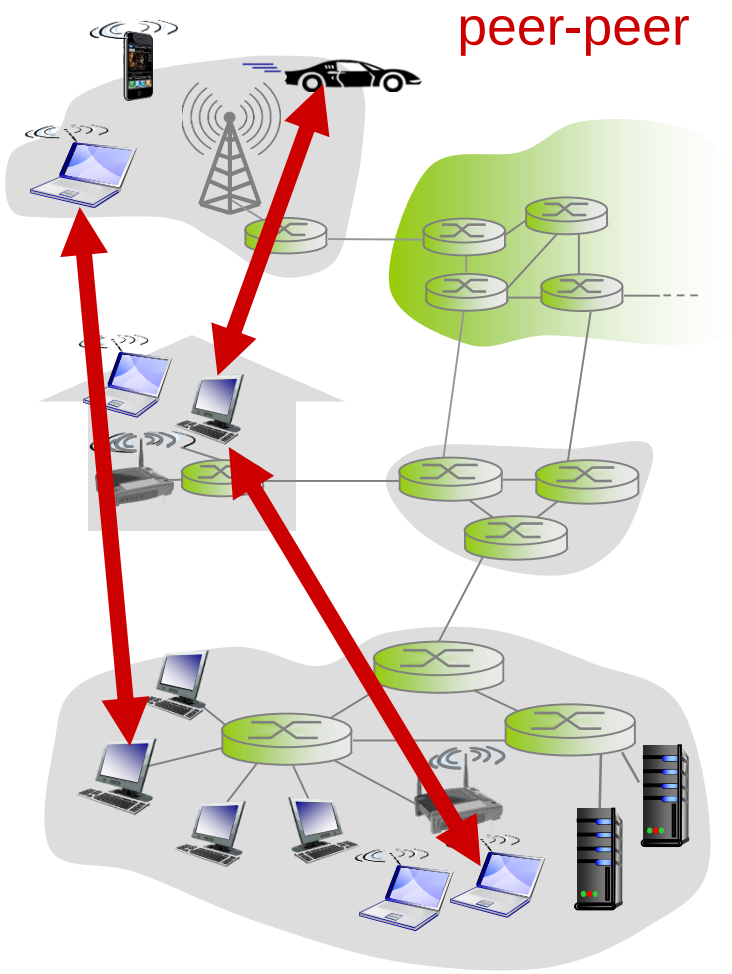
- Εξυπηρετητής (server):
  - *Εξυπηρετεί* αιτήσεις πελατών για υπηρεσίες.
  - Πρόκειται για ένα διαρκώς *ενεργό* τερματικό σύστημα (host) που συνήθως ανήκει στο φορέα που παρέχει την υπηρεσία.
  - Με *μόνιμη* γνωστή διεύθυνση (IP address).
  - Ομάδες εξυπηρετητών (server farms, datacenters) για κλιμάκωση.
- Πελάτες (clients):
  - Επικοινωνούν με τον εξυπηρετητή και *αιτούνται* υπηρεσίες.
  - Ενδεχομένως έχουν διακοπτόμενη σύνδεση.
  - Ενδεχομένως έχουν δυναμική διεύθυνση IP.
  - *Δεν επικοινωνούν* απευθείας μεταξύ τους.
- Γνωστές εφαρμογές: www, e-mail



CC BY SA

Αρχιτεκτονική αμιγώς μεταξύ ομότιμων (pure P2P)

- Δεν στηρίζεται σε *αποκλειστικούς* servers.
- Δεν υπάρχει κάποιος διαρκώς ενεργός εξυπηρετητής, αλλά τυχαία τερματικά συστήματα που *επικοινωνούν απευθείας μεταξύ τους* (ομότιμα).
- *Δεν ανήκουν στον πάροχο* της υπηρεσίας.
- Οι ομότιμοι αιτούνται υπηρεσιών από τους ομότιμους τους και παρέχουν σε αυτούς υπηρεσίες όταν τους ζητηθεί.
- Διακοπτόμενη σύνδεση των ομότιμων και αλλαγή των διευθύνσεων IP.
- Υψηλή *δυνατότητα κλιμάκωσης*. Νέοι χρήστες ενδυναμώνουν τη υπηρεσία, αλλά δημιουργούν και νέες απαιτήσεις (αυτό-κλιμάκωση, self-scalability).
- *Μικρότερο κόστος* υλοποίησης.



CC BY SA

Προκλήσεις των P2P


http:pixabay.com

- Δυσκολίες στη διαχείριση, αλλά και στον *έλεγχο περιεχομένου*.
- Ο τρόπος λειτουργίας τους *δεν είναι συμβατός* με αυτόν των ISPs. Μεταφέρουν την κίνηση στα τερματικά συστήματα, όπου συνήθως εξυπηρετούνται από τις ασύμμετρες ζεύξεις των ISPs.
- Μπορεί να προκαλέσουν *προβλήματα ασφάλειας* λόγω της κατανεμημένης και ανοικτής φύσης τους.
- Η επιτυχία των μελλοντικών P2P εφαρμογών εξαρτάται από αν καταφέρουν να πείσουν τους χρήστες τους να παρέχουν *εθελοντικά πόρους* (εύρος ζώνης, αποθηκευτικούς χώρους, υπολογιστική ισχύ).

Επικοινωνία διεργασιών

Διεργασία (process)

Ένα πρόγραμμα που εκτελείται («τρέχει») σε ένα υπολογιστή.

- Μέσα στον ίδιο υπολογιστή (host), δύο διεργασίες επικοινωνούν χρησιμοποιώντας κάποια διαδιεργασιακή (inter-process) επικοινωνία, η οποία ορίζεται από το λειτουργικό σύστημα.
- Οι διεργασίες σε διαφορετικούς υπολογιστές στο δίκτυο, επικοινωνούν μεταξύ τους ανταλλάσσοντας μηνύματα (messages).

Διεργασία πελάτης

Μια διεργασία που ξεκινά την επικοινωνία

Διεργασία εξυπηρετητής

Μια διεργασία που αναμένει να επικοινωνήσουν μαζί της

- Εφαρμογές με αρχιτεκτονικές P2P έχουν ταυτόχρονα και διεργασίες πελάτες και διεργασίες εξυπηρετητές.

CC BY SA

Τα sockets

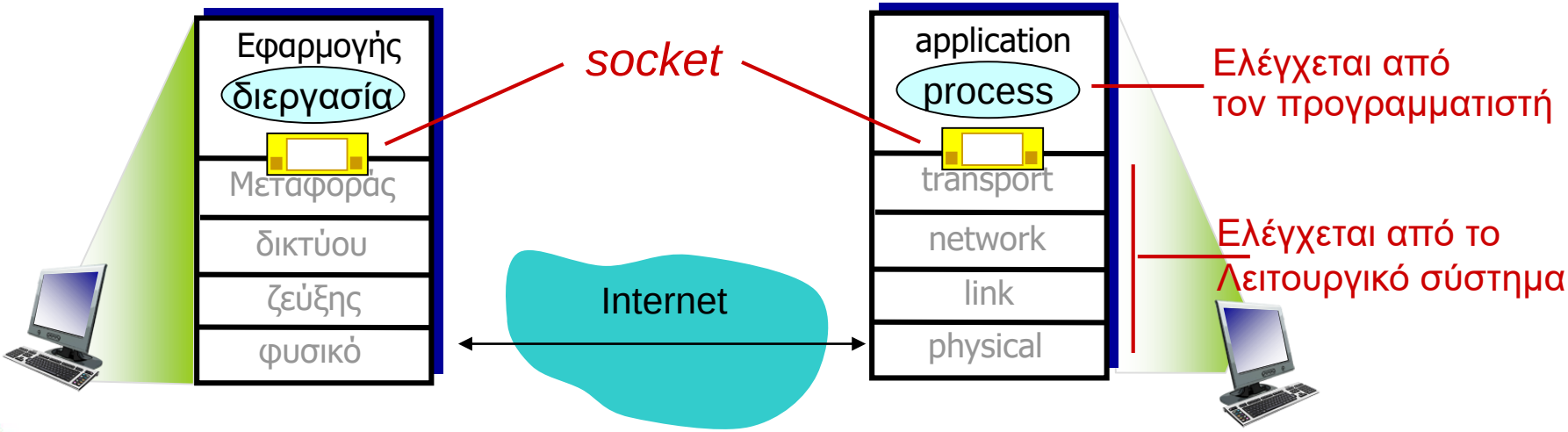
- Στο δίκτυο οι περισσότερες εφαρμογές αποτελούνται από *ζεύγη διεργασιών*, που επικοινωνούν μεταξύ τους ανταλλάσοντας μηνύματα.
- Κάθε μήνυμα από και προς το επίπεδο εφαρμογής θα πρέπει να περάσει από τα *υποκείμενα επίπεδα*.
- Μέρος κάθε διεργασίας (υπό-διεργασία) αναλαμβάνει την αποστολή και τη λήψη των μηνυμάτων (προς και από το επίπεδο μεταφοράς).
- Αυτό επιτυγχάνεται με μια *διεπαφή προγραμματισμού εφαρμογών* (API-Application Programming Interface) που καλείται "*socke*t".

socket

Η διεπαφή της διεργασίας ανάμεσα στην εφαρμογή και το δίκτυο (επίπεδο μεταφοράς) .

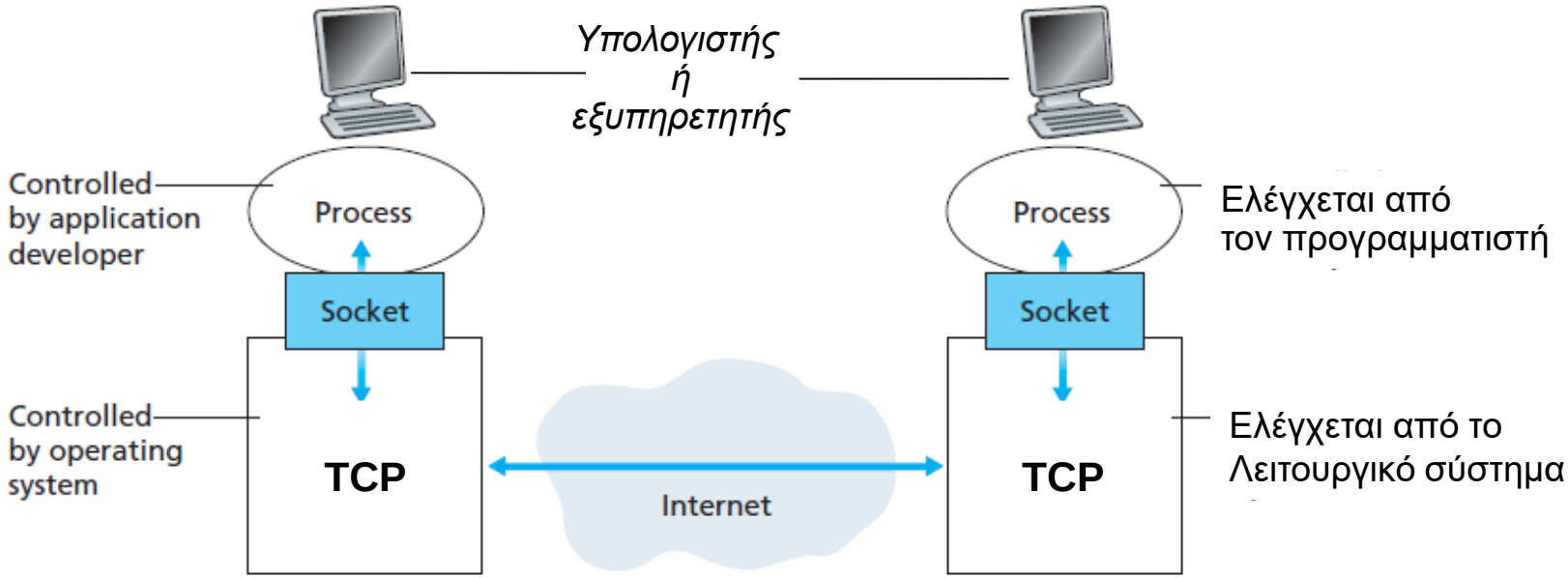
Πως γίνεται η διαδικασία με τα sockets

- Μια διεργασία στέλνει/λαμβάνει μηνύματα στο/από το socket της:
  - Η *διεργασία αποστολέας* στέλνει μήνυμα προς το socket.
  - Η *υποδομή του επιπέδου μεταφοράς* παίρνει το μήνυμα από το socket αποστολής και το στέλνει στο socket της διεργασίας προορισμού.
  - Η *διεργασία λήψης* παραλαμβάνει το μήνυμα από το socket προορισμού και το παραδίδει στην εφαρμογή.
- Η όλη διαδικασία στηρίζεται στη ύπαρξη της υπηρεσίας μεταφοράς που υλοποιείται με κάποιο πρωτόκολλο.



Socket: μεταξύ εφαρμογής και επιπέδου μεταφοράς

- Ο προγραμματιστής ελέγχει πλήρως την εφαρμογή, αλλά έχει μικρό έλεγχο στη υπηρεσία μεταφοράς.
- Είναι αυτός που επιλέγει το *πρωτόκολλο μεταφοράς*.
- Ενδεχομένως να έχει τη δυνατότητα ρύθμισης ορισμένων παραμέτρων.
- Συνήθως παρέχονται *περισσότερα από ένα* πρωτόκολλα μεταφοράς.
- Όταν αναπτύσσεται μια εφαρμογή *επιλέγεται ένα* από τα διαθέσιμα πρωτόκολλα.



Διευθυνσιοδότηση διεργασιών

- Σε μια αποστολή μηνύματος, για να λάβει μια διεργασία ένα μήνυμα θα πρέπει να έχει κάποιο *αναγνωριστικό (*identifier).
- Θα μπορούσαμε να χρησιμοποιήσουμε την μοναδική *32-bit διεύθυνση IP* που έχει ο υπολογιστής προορισμού…
- Όμως αρκεί η IP διεύθυνση του υπολογιστή προορισμού, όπου τρέχει η διεργασία λήψης, για την αναγνώριση αυτής της διεργασίας?
- Όχι! Διότι σ' ένα υπολογιστή μπορεί να «τρέχουν» πολλές διεργασίες δικτυακών εφαρμογών.
- Πρέπει με κάποιο τρόπο να *καθοριστεί η ταυτότητα* της διεργασίας που αφορά στην σύνδεση μας.
- Αυτό καθίσταται δυνατόν με έναν ***αριθμό θύρας*** (port number) που ορίζεται στον υπολογιστή προορισμού.

Ταυτότητα μιας διεργασίας (identifier)

- Για να προσδιοριστεί μια διεργασία, η ταυτότητα της περιλαμβάνει τόσο τη *διεύθυνση IP* όσο και τον *αριθμό θύρας* (port number) που σχετίζεται με τη διεργασία στον υπολογιστή.
- Σε δημοφιλείς δικτυακές εφαρμογές έχουν εκχωρηθεί συγκεκριμένοι αριθμοί θυρών. Δείτε το:
  - http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers
- Παραδείγματα αριθμών θύρας:
  - Εξυπηρετητής HTTP: 80
  - Εξυπηρετητής e-mail: 25
- Για να σταλεί ένα HTTP μήνυμα στον εξυπηρετητή web *http://www.teicrete.gr* θα πρέπει να δοθούν τα στοιχεία που χαρακτηρίζουν την διεργασία υποδοχής:
  - Διεύθυνση IP: 147.95.40.60
  - Αριθμός θύρας: 80

Τι υπηρεσίες μεταφοράς απαιτούν οι εφαρμογές

- Αξιόπιστη μεταφορά (ακεραιότητα) δεδομένων:
  - κάποιες εφαρμογές (π.χ. ήχου, video), ανέχονται κάποιες *απώλειες* (loss-tolerant applications),
  - άλλες εφαρμογές (π.χ. e-mail, www, μεταφορά αρχείου) απαιτούν 100% *αξιόπιστη μεταφορά* δεδομένων (reliable data transfer).
- Διεκπεραιωτική ικανότητα (εγγύηση για ρυθμό μετάδοσης):
  - κάποιες εφαρμογές (ιδιαίτερα οι πολυμεσικές) απαιτούν κάποιο *ελάχιστο throughput* για να είναι «αποτελεσματικές» (bandwidth sensitive),
  - άλλες εφαρμογές «βολεύονται» με όσο throughput πάρουν (ελαστικές – elastic) π.χ. e-mail, www κλπ.
- Χρονισμός (εγγύηση για χρόνο άφιξης):
  - κάποιες εφαρμογές (πραγματικού χρόνου, π.χ. VoIP, διαδραστικά παιχνίδια) απαιτούν χαμηλή καθυστέρηση για να είναι «αποτελεσματικές».
- Ασφάλεια:
  - Κρυπτογράφηση (encryption), ακεραιότητα δεδομένων (data integrity).

CC BY SA

Απαιτήσεις επιλεγμένων εφαρμογών

Εφαρμογή	Απώλεια δεδομένων	Throughput	Ευαισθησία ως προς το χρόνο
Μεταφορά αρχείου	Όχι απώλειες	ελαστική	όχι
e-mail	Όχι απώλειες	ελαστική	όχι
Έγγραφα Web	Όχι απώλειες	ελαστική	όχι
Ήχος/ βίντεο πραγματικού χρόνου	Ανοχή στις απώλειες	ήχος: 5kbps-1Mbps, βίντεο:10kbps-5Mbps	ναι, 100-δες msec
Αποθηκευμένος ήχος/βίντεο	Ανοχή στην απώλειες	ήχος: 5kbps-1Mbps, βίντεο:10kbps-5Mbps	ναι, λίγα secs
Διαδραστικά παιχνίδια	Ανοχή στην απώλειες	ως λίγα kbps	ναι, 100-δες msec
Στιγμιαία μηνύματα	Όχι απώλειες	ελαστική	ναι και όχι

CC BY SA

Υπηρεσίες του επιπέδου μεταφοράς

- Για την εξυπηρέτηση των εφαρμογών το επίπεδο μεταφοράς του Διαδικτύου διαθέτει υπηρεσίες που βασίζονται σε 2 κυρίως πρωτόκ
  - το *TCP* και
  - το *UDP*.

https://en.wikipedia.org/

- Μια από τις πρώτες αποφάσεις των προγραμματιστών διαδικτυακών εφαρμογών είναι να *επιλέξουν* ένα από τα δύο αυτά πρωτόκολλα.
- Αυτά παρέχουν διαφορετικό σύνολο υπηρεσιών στις εφαρμογές που τα καλούν.

Υπηρεσίες του TCP

- *Συνδεσιστρεφής* (connection-oriented) υπηρεσία:
  - απαιτείται αρχικοποίηση (setup) μεταξύ διεργασιών πελάτη και εξυπηρετητή – *διαδικασία χειραψίας*.
- *Αξιόπιστη μεταφορά* δεδομένων μεταξύ διεργασιών αποστολής και λήψης.
- Περιλαμβάνει μηχανισμό *ελέγχου ροής* (flow control):
  - ο αποστολέας δεν υπερφορτώνει τον παραλήπτη.
- Κάνει έλεγχο *συμφόρησης* (congestion control):
  - επιβράδυνση αποστολής πακέτων όταν το δίκτυο είναι υπερφορτωμένο.
- Δεν παρέχει:
  - εγγυήσεις ως προς χρόνο,
  - εγγυήσεις ελάχιστου throughput (διεκπεραιωτική ικανότητα) ,
  - ασφάλεια (εκτός αν εμπλουτιστεί με το SSL - Secure Socket Layer).

CC BY SA

Υπηρεσίες του UDP

- *Ασυνδεσιστρεφής* (connection-less) υπηρεσία:
  - δεν υπάρχει διαδικασία «χειραψίας» πριν την ανταλλαγή μηνυμάτων.
- *Αναξιόπιστη μεταφορά* δεδομένων μεταξύ διεργασίας αποστολής και λήψης.
- Δεν παρέχει εγγυήσεις :
  - άφιξης των μηνυμάτων,
  - ούτε καν με την σωστή σειρά!
- Επίσης δεν παρέχει:
  - αρχικοποίηση σύνδεσης,
  - αξιοπιστία,
  - έλεγχο ροής,
  - έλεγχο συμφόρησης,
  - εγγύηση ως προς το χρόνο ή το throughput
- Είναι όμως *απλό και «ελαφρύ»* πρωτόκολλο.

Ποιες υπηρεσίες δεν παρέχονται

- Ούτε το TCP ούτε το UDP δεν παρέχουν:
  - *Εγγυήσεις διεκπεραιωτικής* ικανότητας.
  - Υπηρεσίες *χρονισμού* (μεταφορά σε συγκεκριμένο χρόνο).
- Γενικότερα, το Διαδίκτυο δεν παρέχει εγγυήσεις χρονισμού και διεκπεραιωτικής ικανότητας.
- Όμως εφαρμογές που είναι ευαίσθητες στη διεκπεραιωτική ικανότητα και στο χρονισμό, όπως οι εφαρμογές πραγματικού χρόνου (video, ήχος), φιλοξενούνται στο Διαδίκτυο.
- Πως;
  - Έχουν σχεδιαστεί για μπορούν να αντιμετωπίσουν αυτή την έλλειψη των εγγυήσεων (χρήση ειδικών τεχνικών).
  - Συχνά χρησιμοποιούν το UDP πρωτόκολλο επειδή είναι ελαστικές στις απώλειες (ή χρησιμοποιούν εφεδρικά το TCP).

CC BY SA

Δημοφιλείς εφαρμογές και πρωτόκολλα

Εφαρμογή	Πρωτόκολλο εφαρμογής	Πρωτόκολλο μεταφοράς
Ηλεκτρονικό ταχυδρομείο E-mail	SMTP [RFC 2821]	TCP
Απομακρυσμένη προσπέλαση τερματικού	Telnet [RFC 854]	TCP
Μεταφορά αρχείων	FTP [RFC 959]	TCP
Πολυμέσα συνεχούς ροής	HTTP (π.χ. Youtube), RTP[RFC 1889]	TCP ή UDP
Τηλεφωνία Διαδικτύου	SIP, RTP, ιδιοταγή (π.χ. Skype)	τυπικά UDP

Δημοφιλείς εφαρμογές, πρωτόκολλα επιπέδου εφαρμογής και πρωτόκολλα επιπέδου μεταφοράς.

CC BY SA

Πρωτόκολλα επιπέδου εφαρμογής

- Ένα πρωτόκολλο επιπέδου εφαρμογής (application-layer protocol) προδιαγράφει πως οι διεργασίες μιας κατανεμημένης εφαρμογής που εκτελούνται σε διαφορετικούς υπολογιστές ανταλλάσουν μεταξύ τους μηνύματα.
- Πρόκειται απλά για ένα συστατικό μιας δικτυακής εφαρμογής.
- Για παράδειγμα η εφαρμογή web περιλαμβάνει:
  - ένα πρότυπο για μορφοποίηση εγγράφων (HTML),
  - ένα πρόγραμμα περιήγησης web (πχ firefox, chrome, internet explorer),
  - έναν εξυπηρετητή web (πχ apache) και
  - ένα πρωτόκολλο επιπέδου εφαρμογής (HTTP) που ορίζει την μορφή και την αλληλουχία των μηνυμάτων που ανταλλάσσονται μεταξύ του περιηγητή web και του εξυπηρετητή web.

CC BY SA

Τι ορίζει ένα πρωτόκολλο επιπέδου εφαρμογής

- Τους *τύπους των μηνυμάτων* που ανταλλάσσονται:
  - π.χ. των αιτήσεων, των αποκρίσεων κλπ.
- Την *σύνταξη* διαφόρων τύπων μηνυμάτων:
  - τα πεδία στα μηνύματα & πως αυτά διαχωρίζονται/απεικονίζονται.
- Την *σημασιολογία* των πεδίων, δηλαδή:
  - τη σημασία της πληροφορίας μέσα στα πεδία
- *Κανόνες* για το πότε και πώς οι διεργασίες ανταλλάσσουν μηνύματα.
- Πρωτόκολλα *δημόσιας χρήσης* (πχ HTTP, SMTP) που:
  - ορίζονται στα *RFC*s
  - επιτρέπουν τη διαλειτουργικότητα (interoperability).
- *Ιδιοταγή* (proprietary) πρωτόκολλα:
  - Τα περισσότερα πρωτόκολλα είναι ανοικτά. Υπάρχουν όμως και ιδιόκτητα πρωτόκολλα εταιρειών που δεν διατίθενται δημόσια (κλειστά, πχ Skype).

CC BY SA