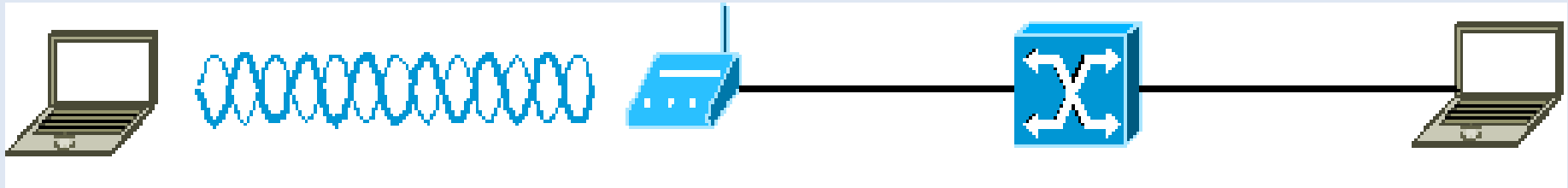


# ΗΥ335 4ο Φροντιστήριο

Δημιουργία ενός τοπικού δικτύου και επίδειξη προγραμμάτων τύπου client / server για TCP και UDP



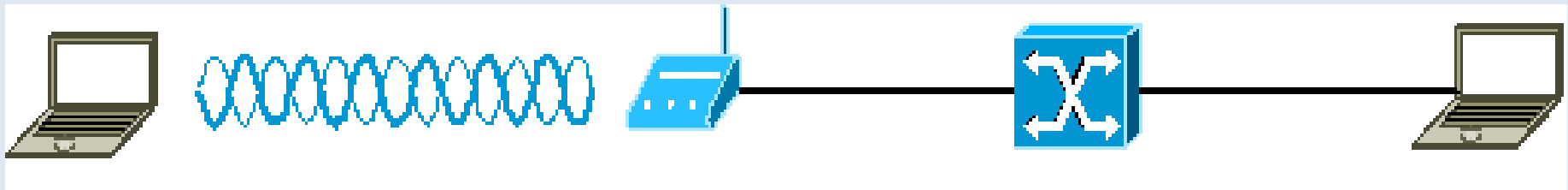
Fontas Fafoutis <fontas@csd.uoc.gr>  
Ilias Tsompanidis <itsomp@csd.uoc.gr>

# Δημιουργία ενός τοπικού Δικτύου

Μέρος Α

# Στοιχεία Δικτύου

- Wireless Access Point
  - Υλοποιεί το πρωτόκολλο IEEE802.11
- Hub
  - Προωθεί ότι λάβει από μία διεπαφή (interface) σε όλες τις άλλες
- Καλώδια UTP (συνεστραμμένου ζεύγους)



# Διευθυνσιοδότηση

- Διαθέσιμες IP διευθύνσεις για τοπικά δίκτυα
  - 10.x.x.x
  - 172.16.x.x
  - 192.168.x.x
- Θέλω να φτιάξω ένα δίκτυο με χαρακτηριστικά
  - Subnet: 192.168.2.0
  - Mask: 255.255.255.0
  - Host IPs: 192.168.2.1-255
  - Gateway: null (LAN only)

# Στατική Διευθυνσιοδότηση

- Σε Unix

- `$ ifconfig [interface] [IP Address] netmask [mask] up`
- `$ ifconfig eth0 192.168.2.1 netmask 255.255.255.0 up`
- `$ man ifconfig` (για περισσότερες πληροφορίες)

- Σε Windows

- LAN Connection properties → Internet Protocol (TCP/IP) properties → Use the following IP address

- Να δούμε αν όλα είναι όπως τα θέλουμε

- `$ ifconfig eth0`
- `$ ipconfig /all`

# Ασύρματη Σύνδεση

- iwlist
  - Λεπτομέρειες από ασύρματες διεπαφές
  - `$ iwlist [wireless interface] scan`
  - `$ iwlist wlan0 scan`
  - `$ man iwlist` (για περισσότερες πληροφορίες)

# Ασύρματη Σύνδεση

- iwconfig
  - Σύνδεση στο Access Point
  - `$ iwconfig [wireless interface] essid [my_AP_essid] mode managed ap any channel auto`
  - `$ iwconfig wlan0 essid "hy335_demo" mode managed ap any channel auto`
  - `$ man iwconfig` (για περισσότερες πληροφορίες)
- Μετα την επιτυχή σύνδεση με το AP ορίζουμε τις IP
  - `$ ifconfig wlan0 192.168.2.2 netmask 255.255.255.0 up`

# Είναι συνδεδεμένα;

- Με την εντολή ping μπορώ να μάθω αν το ένα laptop "βλέπει" το άλλο
  - `$ ping 192.168.2.1`
- Να θυμηθούμε τα ARP Tables
  - `$ arp -n`



# Προγράμματα Client / Server

Μέρος Β

# Client / Server Εφαρμογές

- Προγράμματα σε επίπεδο εφαρμογής (Application Layer)
- Ο εξυπηρετητής (server) παρέχει μία υπηρεσία
  - Διαφημίζει την IP διεύθυνση και την port
- Οι πελάτες (clients) ζητάνε μία υπηρεσία
  - Πρέπει να γνωρίζουν την IP διεύθυνση και την port του εξυπηρετητή
- Χρησιμοποιούν TCP/IP ή UDP/IP sockets που τους παρέχει το λειτουργικό σύστημα

# Sockets

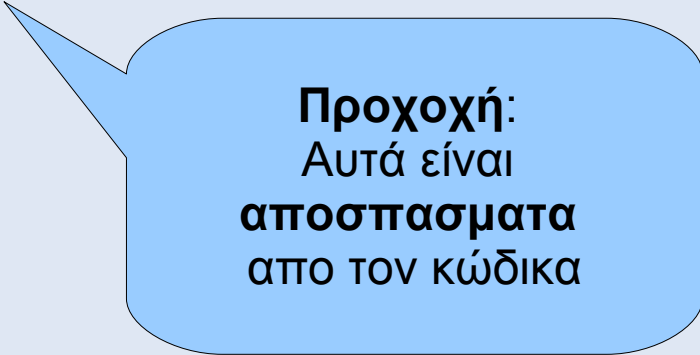
- Socket είναι μία διεπαφή (interface) ανάμεσα σε μία εφαρμογή και το TCP/IP protocol stack
  - Παρέχεται από το λειτουργικό σύστημα
  - Χαρακτηρίζεται μοναδικά από
    - Protocol
    - Local IP
    - Local port
    - Remote IP
    - Remote port

# Προτόκολλα μεταφοράς

- Ανάλογα με τις απαιτήσεις της εφαρμογής
- TCP
  - Απαιτεί σύνδεση
  - Παρέχει εγγυήσεις σωστής μετάδοσης, με την σωστή σειρά
  - κτλ
- UDP
  - Χωρίς σύνδεση

# TCP Server

- // Set a new ServerSocket on a **port**
- ServerSocket serverSocket = new ServerSocket(port);
- // Wait for **connections**
- Socket clientSocket = serverSocket.accept();
- // Get the input and output streams of the clientSocket
- PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
- BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
- // Read from and write to the socket
- String inputLine = in.readLine();
- out.println(outputLine);
- // Close the sockets
- clientSocket.close(); serverSocket.close();



**Προχολή:**  
Αυτά είναι  
**αποσπασματα**  
απο τον κώδικα

# TCP Client

- `// Attempts connection to the host on the specified port`
- `Socket socket = new Socket(host, port);`
- `// Gets the input and output streams of the client socket`
- `out = new PrintWriter(socket.getOutputStream(), true);`
- `in = new BufferedReader(new InputStreamReader(socket.getInputStream()));`
- `// Read and write to the socket`
- `String inputLine = in.readLine();`
- `out.println(outputLine);`
- `// Close the socket`
- `socket.close();`

# TCP Server / Client

- Server
  - Compile: `$ javac TCPServer.java`
  - Run: `$ java TCPServer 12321`
- Client
  - Compile: `$ javac TCPClient.java`
  - Run: `$ java TCPClient 192.168.2.1 12321`

# Telnet

- Πρωτόκολλο επιπέδου εφαρμογής
- TCP συνδέσεις σε Servers
- Επικοινωνία με αποστολή και λήψη κειμένου
- Δεν κρυπτογραφεί τα δεδομένα
  - Καλύτερα να χρησιμοποιούμε ssh
- Μπορούμε να το χρησιμοποιήσουμε σαν client για τον server μας
- `$ telnet 192.168.2.1 12321`



# UDP Server

- `// Waiting for UDP packets on a specified port`
- `DatagramSocket socket = new DatagramSocket(port);`
- `// Receiving packets..`
- `DatagramPacket packet = new DatagramPacket(buf, buf.length);`
- `socket.receive(packet);`
- `// In order to answer we must read the remote IP and port from the received packet`
- `InetAddress raddress = packet.getAddress();`
- `int rport = packet.getPort();`
- `packet = new DatagramPacket(buf, buf.length, raddress, rport);`
- `socket.send(packet);`
- `// Closing the socket`
- `socket.close();`

# UDP Client

- `// Open a socket and send packets containing the remote IP and port`
- `DatagramSocket socket = new DatagramSocket();`
- `DatagramPacket packet = new DatagramPacket(buf, buf.length, address, port);`
- `socket.send(packet);`
- `// Receiving packets..`
- `packet = new DatagramPacket(buf, buf.length);`
- `socket.receive(packet);`
- `// Closing the socket..`
- `socket.close();`

# UDP Server / Client

- Server
  - Compile: `$ javac UDPServer.java`
  - Run: `$ java UDPServer 12322`
- Client
  - Compile: `$ javac UDPClient.java`
  - Run: `$ java UDPClient 192.168.2.1 12322`

# Τελος

Ερωτήσεις;