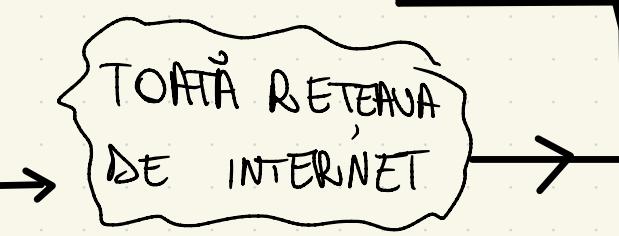
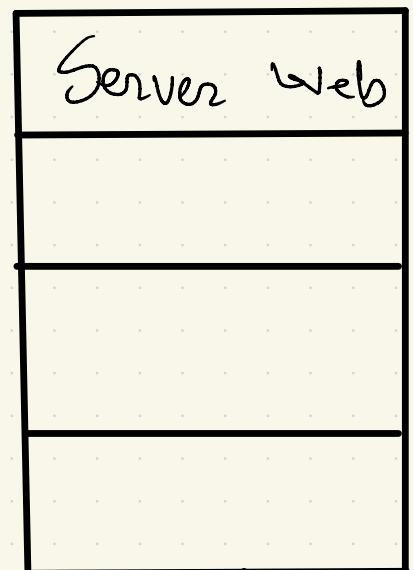
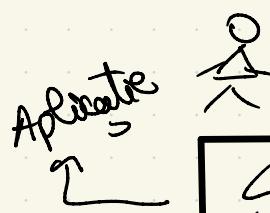
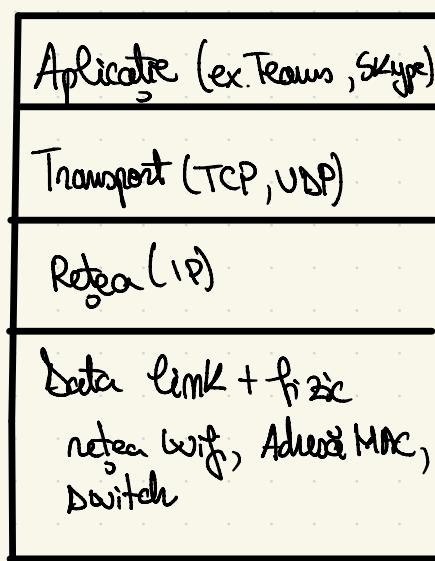


- stiva de protocoale TCP / IP
↳ o stivă
- adresația IP (IPv4, IPv6)
- mecanismul de direcție (rutează)
- mecanismul DNS = traducerea numeelor de calculator în adrese IP și invers
↳ Domain Name System
- paradigma client - server

4 octetii = 32 băti

2 Stiva de protocoale

datele sunt codificate la nivel



- TCP = transport control protocol
- server = un program în execuție
- programul devine proces când rulează
↳ o instanță în execuție
- protocolul HTTP
- servicii program

- torrent - se conecteaza la altii

↳ ex. descarcam un film - nu se mai intampina prim tracker, ci alt client devine server (info nu vine de la tracker ci de la alt client)

- NTP - network time protocol (afara cat e ceva)

• RFE

- VPN = ne ia un packet / info de la un nivel și ne împachetează în alt nivel și altădată

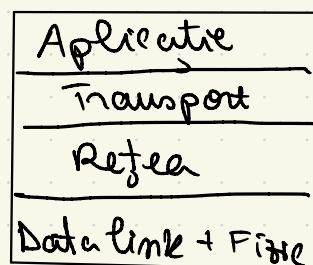
- client TCP/IP → stivă

- client, server = aplicații
↳ adresa IP



Curs 2

- Socket-uri =
- TCP



TCP UDP
IP

CLIENT

Socket

connect(c, IP Server
POPserver)

SERVER

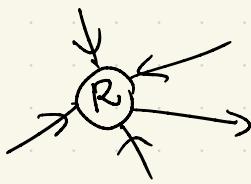
s = Socket

bind(0, -8080
-INADDR_ANY)

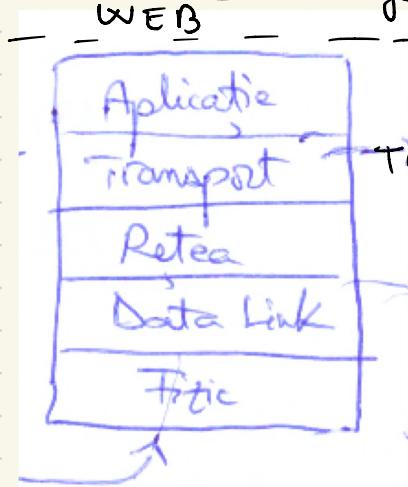
↳ 0.0.0.0

↳ -1 (erșat), deoarece portul este ocupat
accept(>),

↳ -1 (erșat)



- de căutat despre copiile de internet
- TCP - face modif necesare la nivel transport pt a se asigura că datele ajung la destinat
- dacă la apel sistem \rightarrow s-a stabilit conexiunea
 - ↳ eroare



- UDP**
- nu păstrează info despre stare emisitor (trimitere date)
 - avantaj \rightarrow mai rapid (TCP - întârziat de verificarea stării de trimitere date)
 - ?

Docket (AF_INET, SOCK_DGRAM, 0)

la UDP \rightarrow fără connect

\hookrightarrow folosim send(c, &i, sizeof(i), 0, structura cu datele de identificare parametrului,

struct cu cîte ne trimiti, sizeof(struct))

D = socket (AF_INET, SOCK_DGRAM, 0)

bmd = (D, {
- INADDR_ANY
- port})

fără: accept

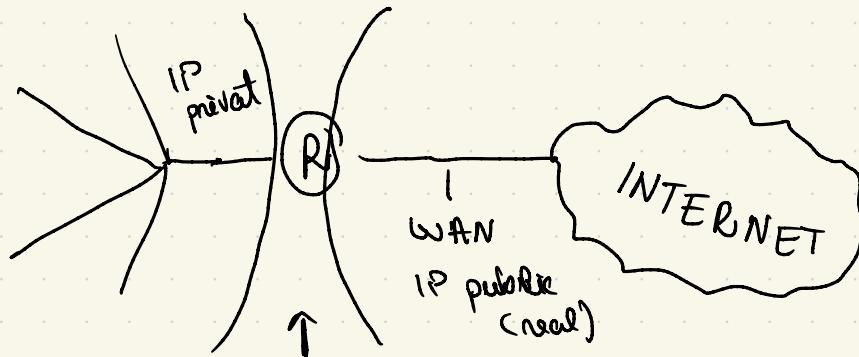
recvfrom (D, &i, sizeof(i), MSG_WAITALL,

struct cu cîte ne trimiti, sizeof(struct))

sendto (D, &i, sizeof(i), 0

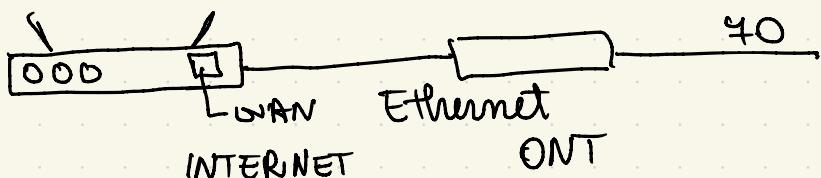


Curs 5 - 2 membri



interconectarea 2 retele locale
(router, switch)

- AP (acces point) (bridge) → interconectarea 2 retele
⇒ se comportă ca un switch
(arhitectură fizică dif.)



* OpenWRT

- broadcast - să se vadă între ele pînă la rețea

- interfațe de rețea

- IP public - IP-WL de la provider

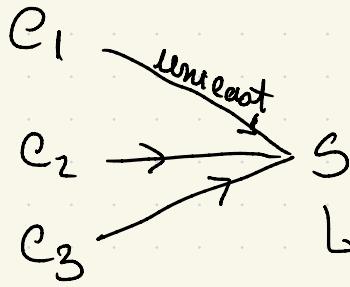
- IP privat - le putem seta noi

- de către despre VLAN. Windows se comportă ca un router și margină virtuală.

- configurație în mod NAT/BRIDGE

Tipuri de trafic

- pipe/fifo - unidirectional
- unicast (1:1)
- broadcast (1: la toti)
- multicast (1: la mai multe)
- anycast (1: oricare primă)

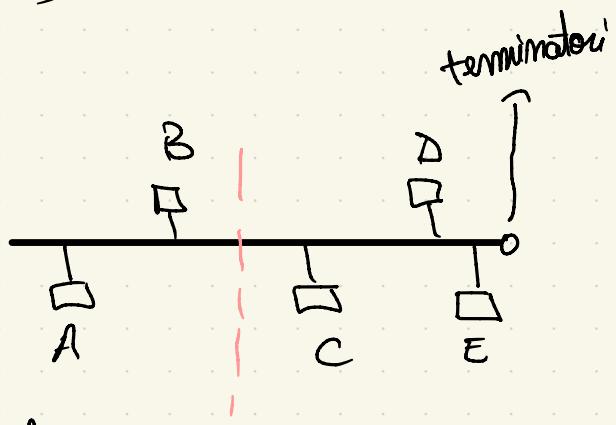


- dacă sunt în rețea locală, nu emis menajă de server (eu devin) și fac broadcast

↳ Serverul trimite la clienti \Rightarrow broadcast

Topologii de rețele

1. Rețele cu topologie bus (liniară)
nu mai sunt folosite astăzi



dacă se întrebuie nu comunică:
 $A-E, A-B$

Curs 7 - 16 noi.

- Adresele MAC - doar în rețea locală
- ARP, DHCP, RARP - ?
- ping - ?
- tcpdump - comandă Linux
- gateway - IP router
- ipconfig sau — /renew

CURS 8 - 23 noi VLANs

Curs 11 - 1h dec

103. 231. 20. 0/24

Subdivizare

Simpărtirea claselor

