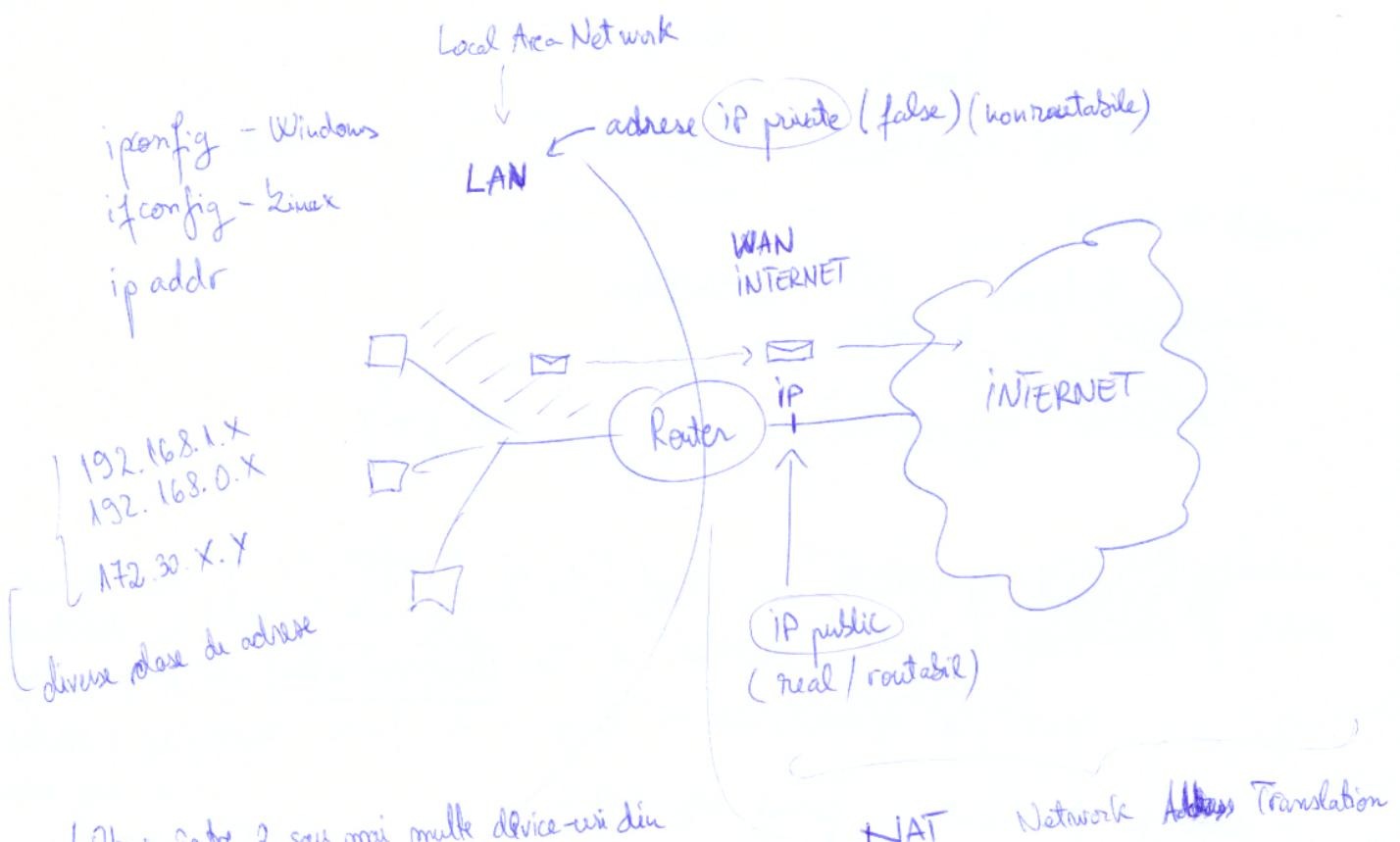
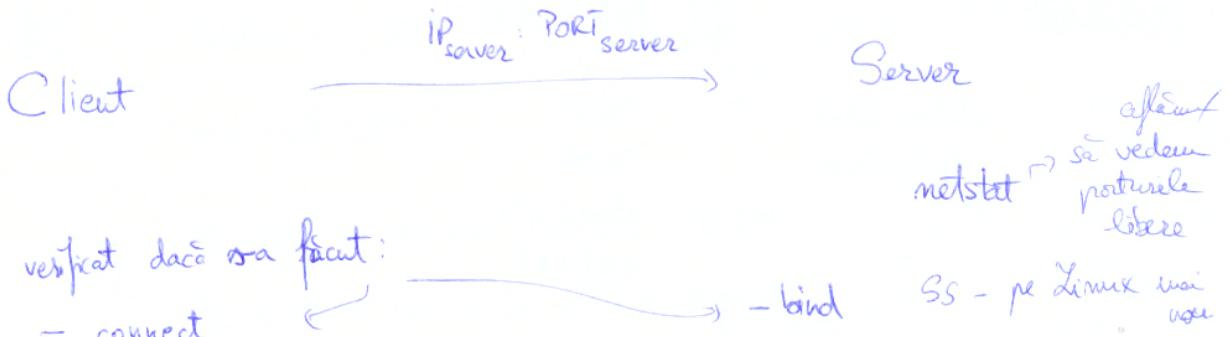


# Cursul 3



| Obs: între 2 sau mai multe device-uri din LAN nu este necesar routerul

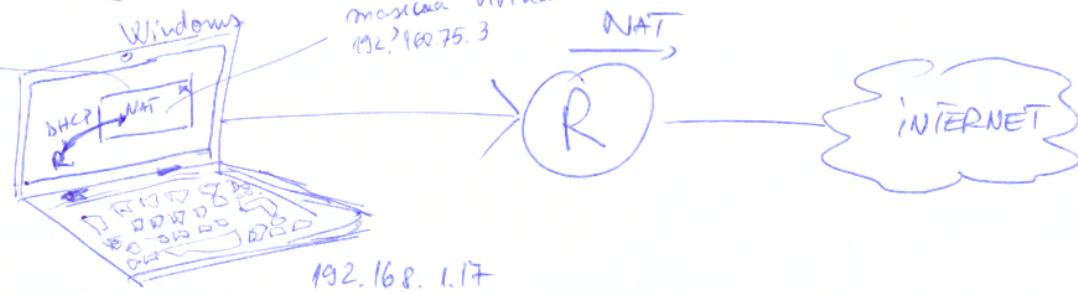
→ dacă merge ping între doi ip-uri, va trea conexiunea

IP C: 192.168.1.14

IPS: 192.168.1.37

→ Nu e nevoie să fie în același rețea, doar observăm că fac parte din același clasa de adrese IP.

## Modif. NAT cu model Bridge

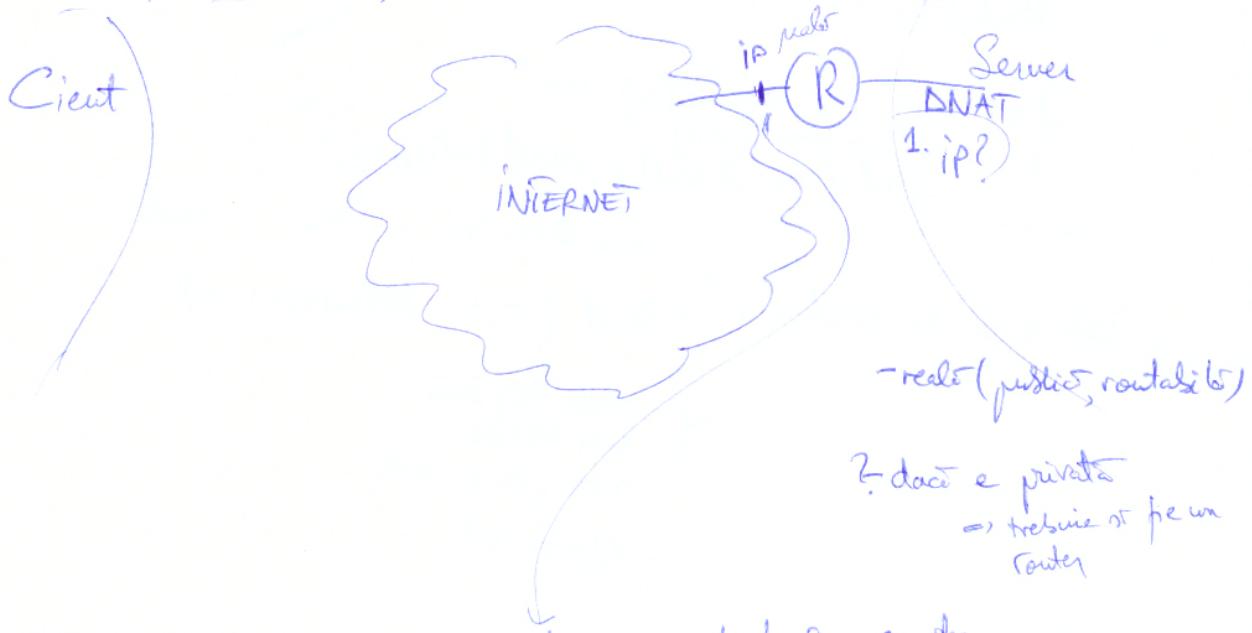


- Mașina fizică „joacă rolul de Router” pentru mașina virtuală
- Dacă avem model Bridge pt. mașina virtuală (nu NAT, ca mai sus), mașina fizică nu mai e router pt. aceasta, căci Poziția Routerului (R) îi va furniza adresa IP. (forma 192.168.1.X)
- Nu este obligatoriu ca serverul DHCP să ruleze pe router.

(când un device vrea să se conecteze la un router/server, trimite un mesaj de tip broadcast)

→ client și server în rețele diferite!

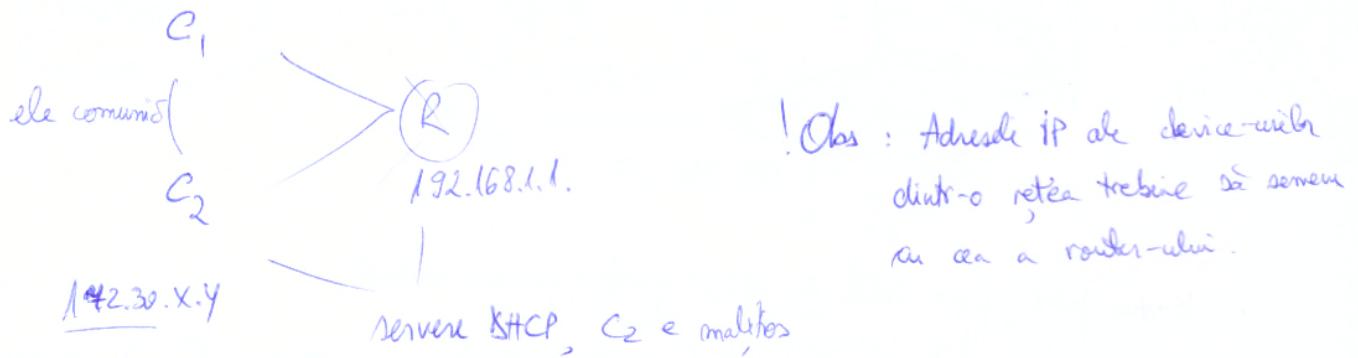
192.168.1.17



DNAT

Port forwarding  
Virtual Server

7.30.4.5 → nu va avea acces la internet!



C → S

↓

uint16\_t m;

n = htons(m)

send (c, &n, ...)

for (i=0; i < ntohs(m); i++)  
x[i] = htons(x[i])

send (c, &x[i], sizeof(x[i]), 0);

uint16\_t \*x;  
x = malloc(...)

send (c, x, n+sizeof(uint16\_t), 0)

m = strlen(x) → htons → send

Bacă e să de cădere: } Send (c, x, strlen(x)+1, 0)

sf recv (c, &m, ...)

m = ntohs

recv (c, x, n+1,

(x[n]=0)

! Important: cîte date trimitem, atâtceasă + de același tip de date  
(important pt. size)

ex:

```
int c;
char s[100];
uint16_t i,j;
int x;
c=socket ; connect --
scanf("%hu", &i)
    ↓
    se citește 2 octeti
    (hu - short unsigned)
```

// scanf("%d", &i)

! → pune la adresa lui i 4 octeti,  
înse i are doar 2 octeti  
⇒ se scrie 4 octetii care pot  
affecta alte date / variabile

strace / ltrace

/strace ./client