



# **Retele de calculatoare**

## **Protocolul IP**

**Sabin-Corneliu Buraga**

<http://www.infoiasi.ro/~busaco>



*“Learning French is trivial:  
the word for horse is cheval,  
and everything else follows in the same way.”*

**Alan J. Perlis**



# Cuprins

- Nivelul retea – Protocolul IP
  - Caracterizare
  - Adrese IP
  - Subretele
  - Datagrama
  - *Firewall & proxy*



# Protocolul IP

- Utilizat de sisteme autonome in vederea interconectarii
- Serviciu de transmitere de pachete (*host-to-host*)
- Translatare dintre diferite protocoale legatura de date
- Oferă servicii neorientate-conexiune, nesigure:  
**datagrame**
- Fiecare datagrama este independenta de celelalte
- Nu se garanteaza transmiterea corecta a datagramelor (pierdere, multiplicare,...)

# Protocolul IP

Adresele IP **nu sunt identice**  
cu cele ale nivelului MAC

(*e.g.*, adresele hardware ale placilor de retea)  
pentru ca IP trebuie sa suporte  
diferite implementari hardware (retele eterogene)

# Protocolul IP

- Foloseste doar adresele logice ale gazdelor
- Servicii:

**Transmitere** (*send*):

aplicatie (utilizator)  $\mapsto$  IP

**Distribuire** (*deliver*):

IP  $\mapsto$  aplicatie (utilizator)

**Raportare a erorilor** – optionala:

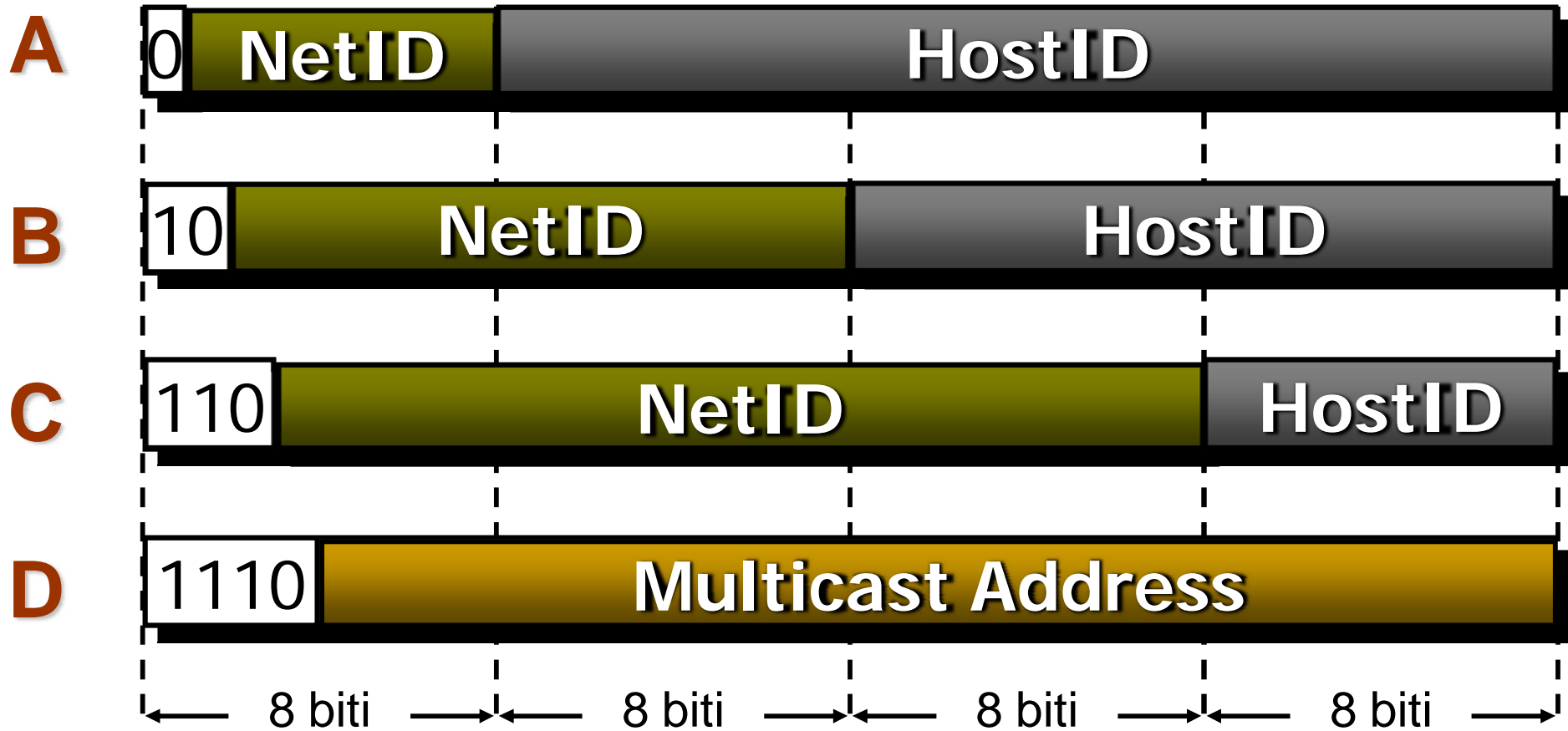
IP  $\mapsto$  aplicatie (utilizator)

# Protocolul IP

- Adrese:
  - logice, nu fizice – 32 biti: **x.x.x.x**
  - includ un identificator de retea (**NetID**) si un identificator de gazda (**HostID**)
  - fiecare gazda trebuie sa aiba o adresa IP unica
  - adresele IP sunt asignate de o autoritate centrala (**NIC – Network Information Center**)
  - sunt divizate in clase de adrese: A, B, C, D, E
  - clasa E nu este utilizata (experimentală)

# Adresele IP

Clasa



# Protocolul IP

- Adrese:
  - **clasa A**: 128 retele posibile, peste 4 milioane de gazde/retea
  - **clasa B**: 16K retele posibile, 64K gazde/retea
  - **clasa C**: peste 2 milioane de retele, 256 gazde/retea
  - **NetID**-ul este asignat unei organizatii de o autoritate centrala
  - **HostID**-ul e asignat local de administratorul retelei
  - ambii identificatori se utilizeaza pentru dirijare
  - exemplu:  
193.231.30.197 – clasa C

# Protocolul IP

- Adrese:
  - o interfata (placa) de retea are asignata o unica adresa IP – numita *adresa host*
  - o gazda poate avea mai multe placi de retea, deci mai multe adrese *host* (adrese IP)
  - gazdele unei aceleasi retele vor avea aceeasi *adresa network* (acelasi **NetID**)
  - adresele de *broadcast* au ca **HostID** toti bitii 1
  - adresa IP care are ca **HostID** toti bitii 0 se numeste *adresa retelei* – refera intreaga retea
  - exemplu: **193.231.30.0** (*adresa network* a masinilor **193.231.30.197** si **193.231.30.231**)
  - **127.0.0.1** – adresa de *loopback* (“eu” – localhost)

# Protocolul IP

- Din spatiul de adrese ce pot fi alocate efectiv, sunt rezervate urmatoarele – **RFC 1918**:

0.0.0.0 – 0.255.255.255

10.0.0.0 – 10.255.255.255 (adrese private)

127.0.0.0 – 127.255.255.255 (pentru *loopback*)

172.16.0.0 – 172.31.255.255 (adrese private)

192.168.0.0 – 192.168.255.255 (adrese private)

# Protocolul IP

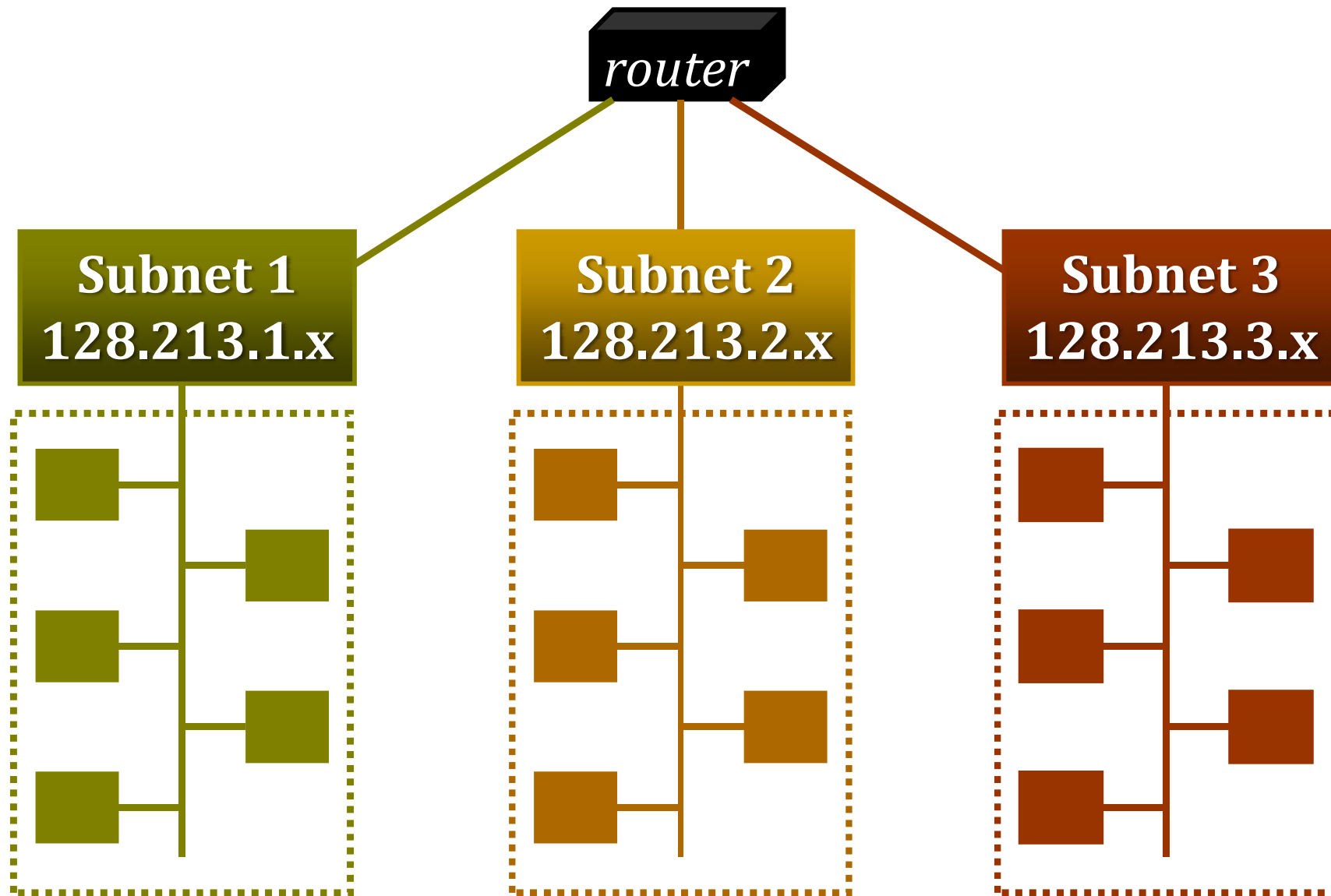
- Adrese de subretea:
  - spatiul de adrese host se poate divide in grupuri numite **subretele**
  - **identificatorul subretelei (SubnetID)** utilizat in general sa grupeze calculatoare pe baza topologiei fizice



- se poate simplifica dirijarea
- e posibil sa avem acelasi cablu pentru subretele multiple (*e.g.*, domeniul **.info.uaic.ro**)

# Protocolul IP

- Adrese de subretea:
  - Divizarea in subretele se face via **masca de retea** (*netmask*): bitii **NetID** sunt 1, bitii **HostID** sunt 0
  - Adresa IP: 192.78.2.213  
 11000000 01001110 00000010 11010101
  - Masca de retea: 255.255.255.192  
 11111111 11111111 11111111 11000000
  - Adresa retea: 192.78.2.192  
 11000000 01001110 00000010 11000000
  - Adresa retelei = masca de retea *AND* adresa IP



*Broadcast*-urile de subretea vor avea ca **HostID** toti bitii 1

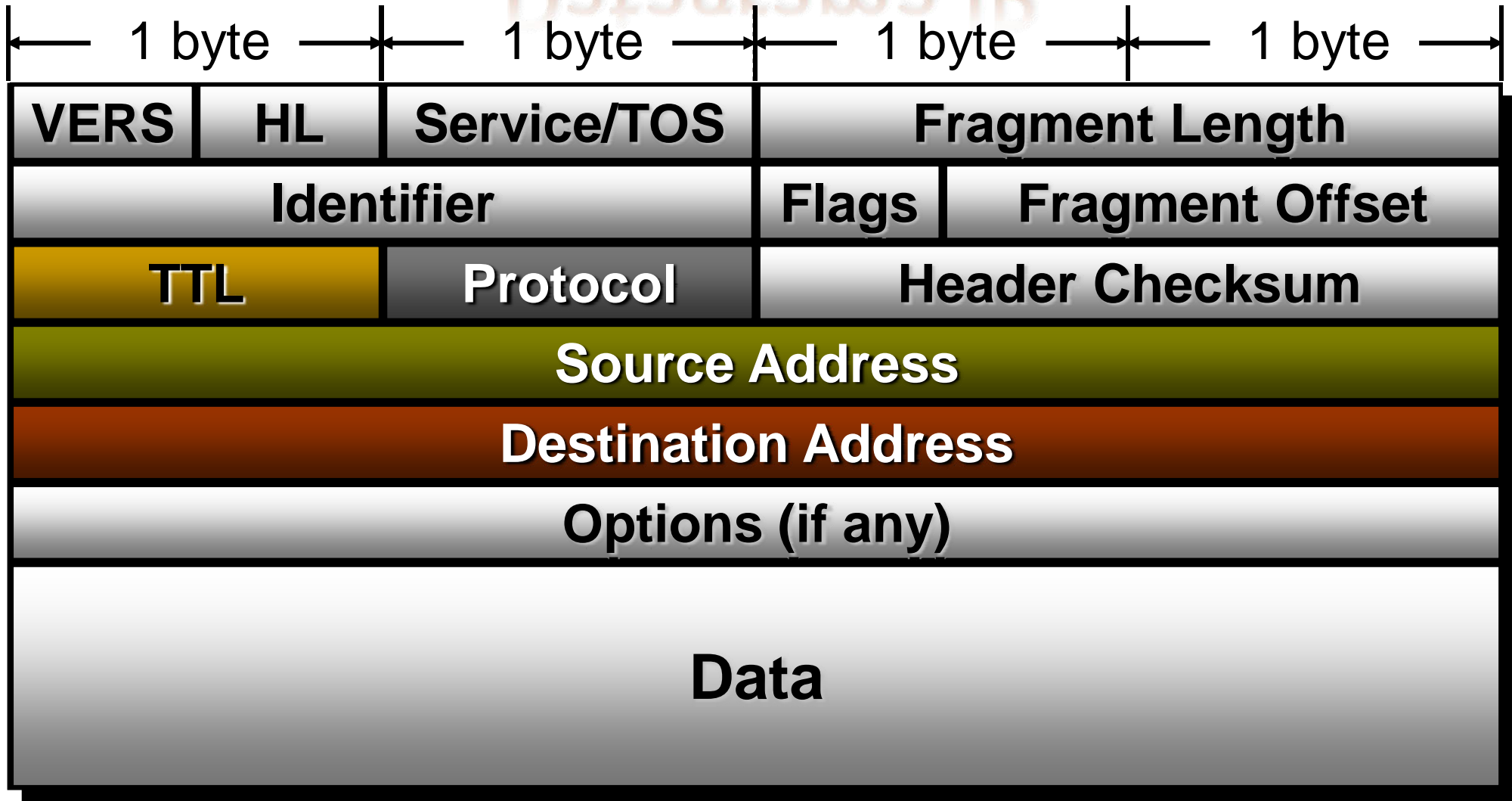
# Protocolul IP

- Masti de subretea implicite:
  - 255.0.0.0                      clasa A
  - 255.255.0.0                    clasa B
  - 255.255.255.0                clasa C

# Protocolul IP

- Conventii de notare:  $x.x.x.x/m$  inseamna ca se aplica o masca de  $m$  biti adresei IP precizata de  $x.x.x.x$
- Exemple:
  - $193.231.30.0/26$  – se aplica o masca de 26 biti adresei 193.231.30.0, selectindu-se ultimii 6 biti ( $=32-26$ ) ai adresei (rezulta  $2^6 = 64$  de valori distincte)
  - $10.0.0.0/12$  – se aplica o masca de 12 biti adresei IP 10.0.0.0, selectindu-se toate valorile posibile in ultimii 20 de biti ( $=32-12$ ) din adresa

# Datagrama IP



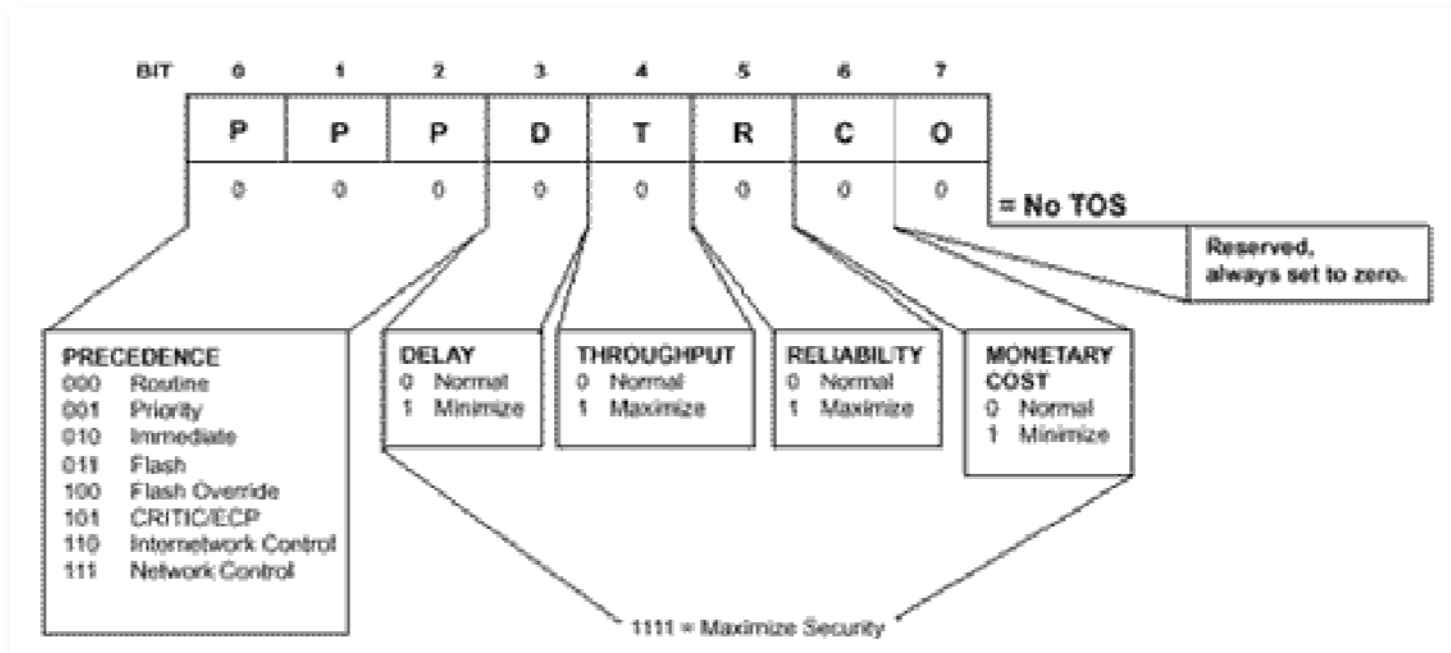


# Protocolul IP

- Valorile uzuale ale cimpului **VERS** sunt:
  - 4 – protocolul IP (RFC 791)
  - 6 – protocolul IPv6 (RFC 1883)

# Protocolul IP

- Cimpul **Service** specifica actiunile speciale ce pot fi efectuate asupra pachetului divizat in: **Precedenta** si **TOS** (*Type of Service*)



# Protocolul IP

- Cimpul **Protocol** specifica protocolul (de nivel superior) caruia ii este destinata informatia inclusa in datagrama:
  - 1 ICMP (*Internet Control Message Protocol*)
  - 2 IGMP (*Internet Group Management Protocol*)
  - 4 IP in IP
  - 6 TCP (*Transmission Control Protocol*)
  - 8 EGP (*Exterior Gateway Protocol*)
  - 17 UDP (*User Datagram Protocol*)
  - 46 RSVP (*Resource Reservation Protocol*)
  - 89 OSPF (*Open Shortest Path First*)etc.

# Protocolul IP

- Cimpul **TTL** (*Type To Live*) specifica “viata” pachetului (numarul va fi decrementat de fiecare *router* prin care trece pachetul)
  - valori uzuale: 64, 32 ori 15

# Protocolul IP

- Cimpul **Identifier**, in conjunctie cu **Flags** si **Fragment Offset**, identifica fragmentele de pachete (daca lungimea datelor depaseste valoarea **MTU** – *Maximum Transmission Unit*)

# Protocolul IP

- Cimpul **Flags** stocheaza 3 biti:
  - 1 bit – nefolosit
  - Don't Fragment* (DF) bit – datagrama nu poate fi fragmentata (daca routerul nu poate transmite pachetul nefragmentat, il va distruge)
  - More Fragments* (MF) bit – semnaleaza ca pachetul este un fragment, urmat de altele (ultimul fragment are MF=0)

# Protocolul IP

- Fragmentarea datagramelor
  - fiecare fragment (pachet) are aceeași structură ca datagrama IP
  - reasamblarea datagramelor se face la destinatar
  - dacă un fragment al unei datagrame e pierdut, acea datagramă este distrusă (se trimite la expeditor un mesaj **ICMP** – *Internet Control Message Protocol*)
  - mecanismul de fragmentare a fost folosit pentru unele atacuri (un fragment “special” e considerat ca fiind parte a unei conexiuni deja stabilite, astfel încât îi va fi permis accesul via *firewall*) – *firewall piercing*

# Protocolul IP

- Controlul fluxului, detectia erorilor
  - daca pachetele ajung prea rapid, receptorul acestora va elimina pe cele in exces (va trimite si un mesaj ICMP la destinatar)
  - daca apare o eroare (*checksum error*), pachetul este distrus

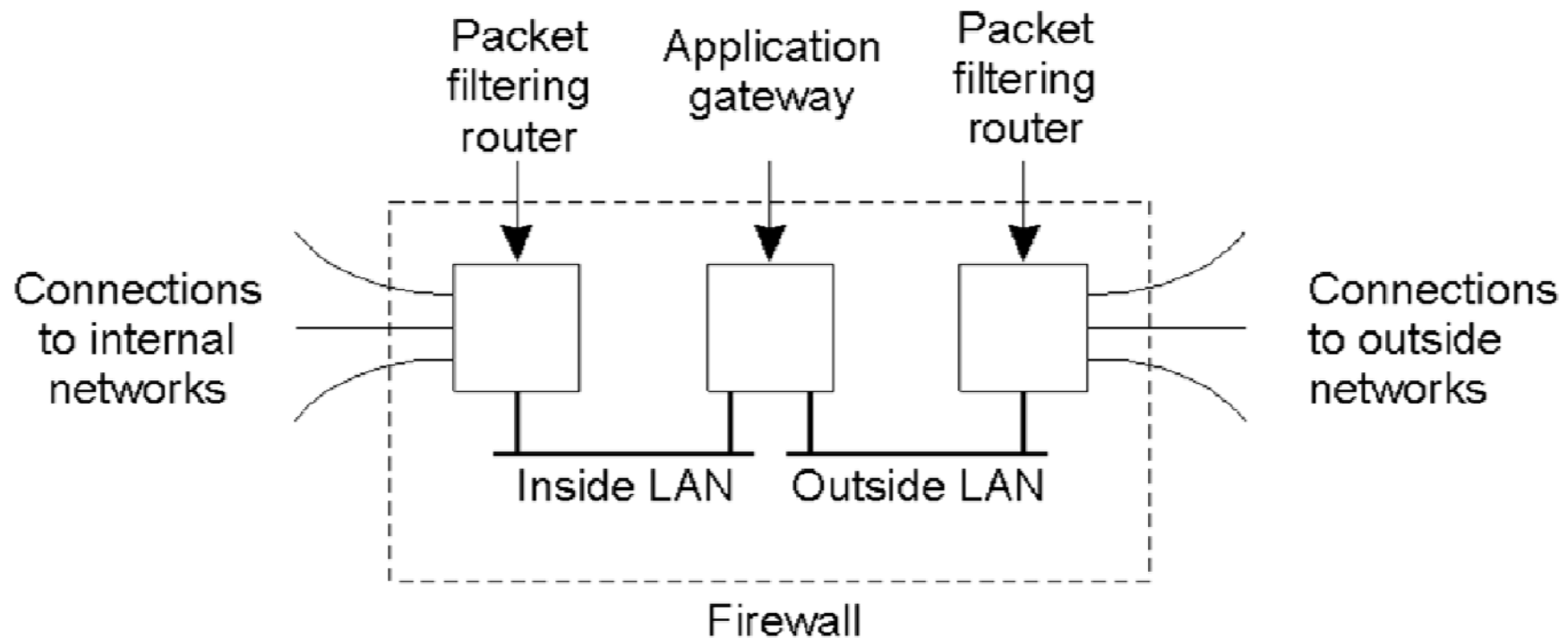
# Protocolul IP

- Filtrarea pachetelor (datagramelor)
  - Se realizeaza de un **firewall**: ofera accesul din exterior in retea internă, conform unor politici (reguli) de acces, doar pentru anumite tipuri de pachete (utilizate de anumite protocoale/servicii)
    - preintimplina anumite atacuri vizind securitatea
  - *Firewall*-ul poate fi software (iptables, ISA Server, ZoneAlarm etc.) sau hardware (e.g., Cisco PIX, Cyberoam, NetScreen)
  - *Firewall*-ul poate juca rol de *proxy* sau *gateway*

# Protocolul IP

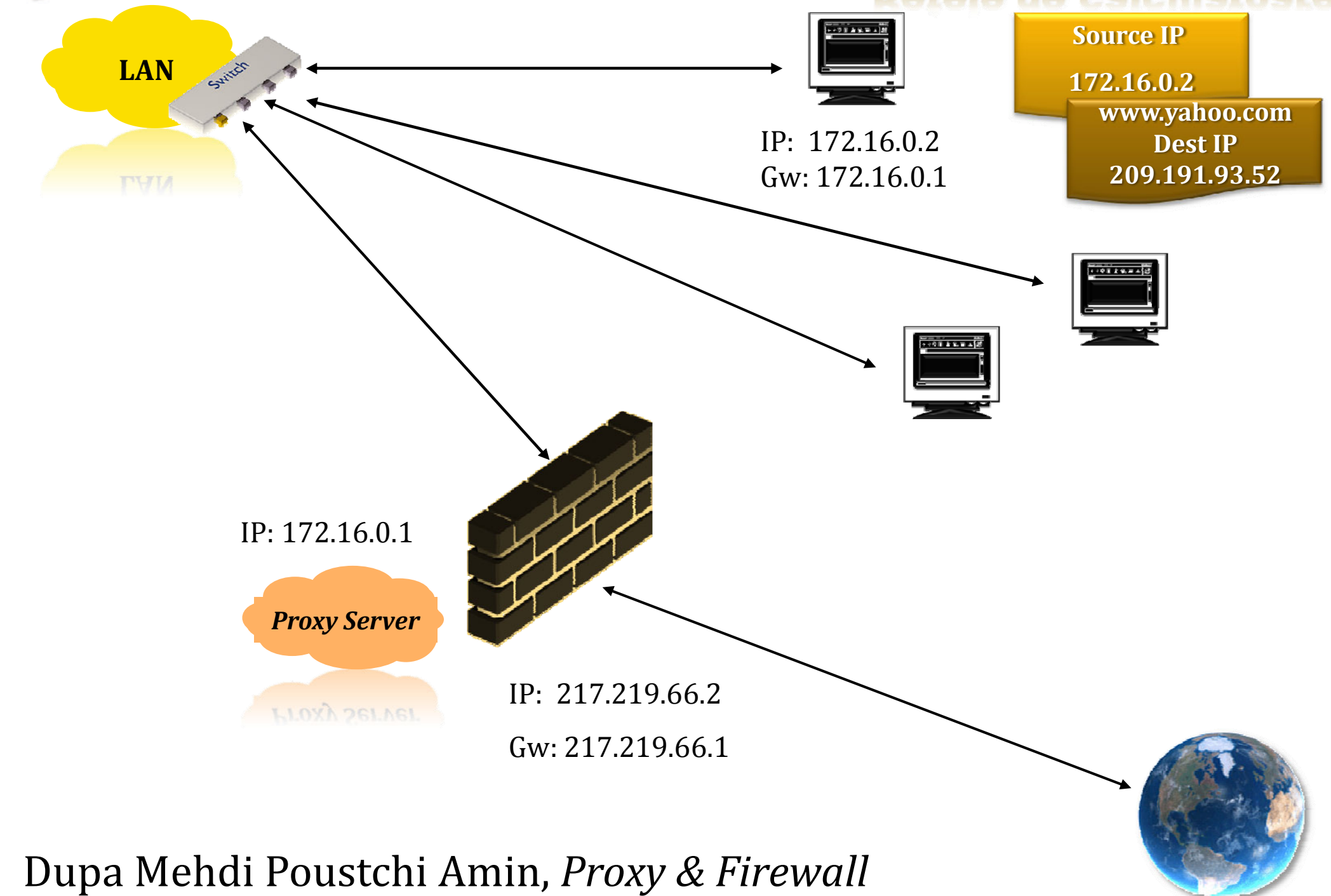
## Filtrarea datelor

Structura conceptuala a unui *firewall*



# Protocolul IP

- Rolul & arhitectura unui *proxy*
  - Acces indirect la alte rețele (Internet) pentru gazdele dintr-o rețea locală via *proxy*
  - *Proxy*-ul poate fi software ori hardware
  - Rol de poartă (*gateway*), de *firewall* sau de server de *cache*
  - *Proxy*-ul oferă partajarea unei conexiuni Internet
  - Utilizat la îmbunătățirea performanței (*e.g.*, *caching*, controlul fluxului), filtrarea cererilor, asigurarea anonimității



Dupa Mehdi Poustchi Amin, *Proxy & Firewall*



# Rezumat

- Protocolul IP
  - Caracterizare
  - Adrese IP
  - Subretele
  - Datagrama
  - *Firewall & proxy*



# Intrebari?