TEMA 1- INTRODUCCIÓ

Model OSI de ISO

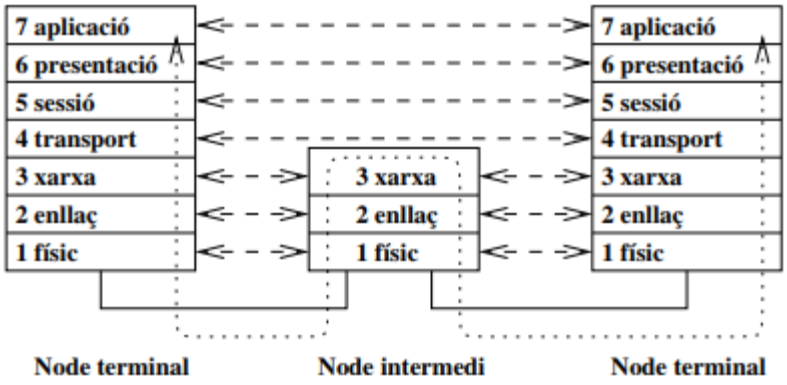
Divideix el conjunt de protocols que formen part d'una xarxa de computadors en **7 nivells**;

Els **nodes terminals** representen els dos sistemes que es comuniquen a través de la xarxa (p.e: **client** i servidor).

Els **nodes intermedis** tenen la funció d'encaminar l'informació (p.e: els **routers**).

Cada nivell utilitza una estructura de dades anomenada ***Protocol Data Unit*** **(PDU)** per a intercanviar informació amb el seu nivell parell. Que conte un **header i un payload**.

Cada nivell estableix un diàleg al corresponent amb les línies discontínues.



- Nivell 1 - **Físic**: Defineix les **característiques físiques i elèctriques** d'un dispositiu de xarxa. En aquest nivell només es parla de **bits o caràcters**.

- Nivell 2 - **Enllaç** *(Data link)*: Fa **d'interfície entre el nivell de xarxa i el físic**. Les PDUs que utilitza s'anomenen **"trames" *(frames)*.** Entre les seves funcions hi pot haver:
    - ***Framing***: Per a descobrir on comença i on acaba una trama dintre del flux de bits o caràcters que llegeix del nivell físic.
    - **Detecció d'errors**: Per a detectar si la trama rebuda té algun error.
    - **Recuperació d'errors**: per exemple, demanant la retransmissió d'una trama errònia al seu nivell parell.

- Nivell 3 - **Xarxa** *(Network)*: Les PDUs que fa servir s'anomenen **"paquets"** (**datagrames per Internet**). La seva principal funció és **l'encaminament de PDUs**: És a dir, aconseguir que les PDUs segueixin **el camí correcte entre la font i la destinació**.

- Nivell 4 - **Transport**: Les PDUs utilitzades es diuen **"*segments*"** (TCP) i ***datagrames*** en (UDP), . És un nivell extrem-a-extrem. La seva funció és establir **un canal entre les dues aplicacions que es comuniquen**. Pot oferir serveis de **segmentació o recuperació d'errors**.

- Nivell 5 - **Sessió**: La seva funció és el ***login***, permetre tornar a un punt conegut després d'un tall de connexió, etc.

- Nivell 6 - **Presentació**: Proporciona protocols de presentació de dades (**ASCI, MPEG**, etc.)

- Nivell 7 - **Aplicació**: Són els protocols de les aplicacions que fan servir la xarxa. Per exemple: **http, smtp, ftp, telnet**, etc.

Arquitectura TCP/IP

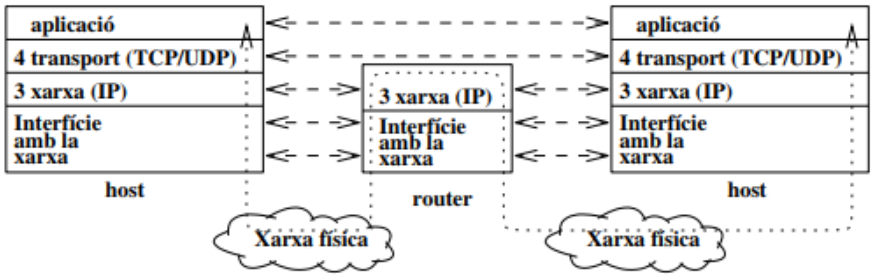


Figura 1.5: Arquitectura de TCP/IP.

Els **protocols més importants** d'Internet es corresponen amb el nivell de **xarxa: el *Internet Protocol* (IP) i** el nivell de **transport: El *Transmission Control Protocol* (TCP)**.

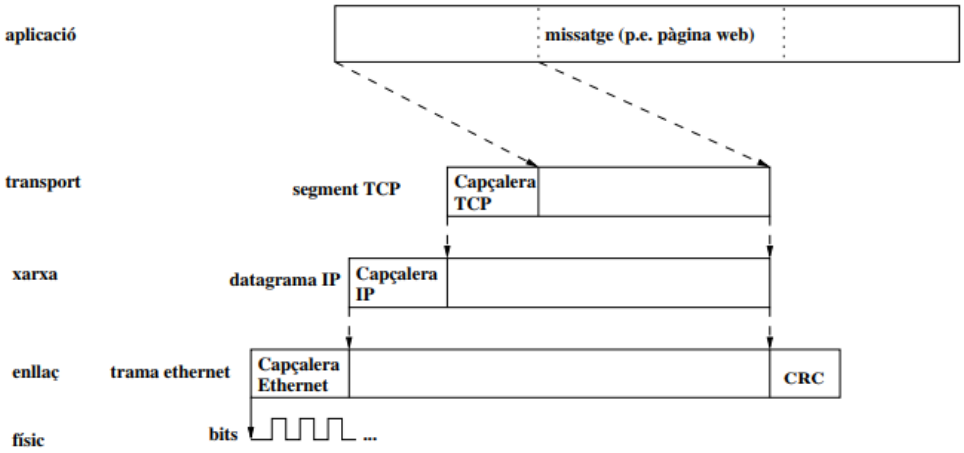
Per ser més exactes, pel que fa al **transport** hi ha **dos** protocols:

- **TCP,** que implementa una **transmissió fiable** (assegura una transmissió lliure d'errors),
- ***User Datagram Protocol* (UDP)**, que no assegura el **lliurament correcte de l'informació**.

Encapsulament de la informació

En la transmissió d'informació a través de la xarxa, **cada nivell** afegeix **una capçalera** amb **l'infromació necessària** per a comunicar-se amb **el nivell parell**.

Cada nivell **afegeix una capçalera** abans de passar **la PDU al nivell inferior** i **elimina la capçalera** abans de passar **la PDU al nivell superior**. Aquest procés s'anomena **encapsulament**:



Paradigma Client-Servidor

És el **model utilitzat per a les aplicacións a Internet per intercanviar-se informació**.

L'aplicació que **inicia l'intercanvi** d'informació s'anomena "**client**". Així doncs, el client envia el primer datagrama cap al servidor. Tal com hem vist, **el nivell de xarxa (IP)** porta els datagrames **fins a la destinació**. Un cop a la destinació**, cal lliurar l'informació a l'aplicació on van dirigits** (p.e: un servidor web). El responsable d'aquesta tasca és el nivell de transport (TCP o UDP).

Per poder **identificar les aplicacions** que es comuniquen, TCP/UDP fan servir els anomenats "**ports**": Un port identifica l'aplicació client, i l'altra aplicació servidor. Aquests ports **formen part de la capçalera de protocol TCP o UDP**.

Aquest servidor té associat un **port *well-known***. Cada port *well-known* **identifica un servei estàndard d'internet**, com ara ***ftp (port 21), telnet (port 23), web (port 80)***, etc. Els ports *well-known* tenen un valor en l'interval **[0, ..., 1023**] i estan estandarditzats per un organisme d'Internet anomenat **IANA**.

Quan el **client** inicia l'intercanvi d'informació, el SO li assigna un **port "efímer"** (que té un **valor >= 1024**). Aquest port identifica el client mentre dura l'intercanvi d'informació. Així doncs, per identificar completament una connexió en TCP/IP es fa servir la **tupla**: (port font, adreça IP font; port destinació, adreça IP destinació). Les **adreces IP** identifiquen **els hosts,** **els ports** identifiquen **les aplicacions dels hosts**.

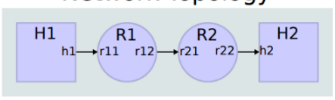
Example

- Discussion: Just Ethernet header contents are modified over links (hops)
- Description
  - Host 1 is connected to Router 1, Router 1 to Router 2, and Router 2 to Host2
  - Host 1 sends a test packet (ICMP Echo Request) to Host 2 over the network
  - Host 2 replies with 1 packet (ICMP Echo Reply)
  - Physical network interfaces identifiers (Ethernet addresses)
    - Host 1: h1
    - Router 1: r11 and r12
    - Router 2: r21 and r22
    - Host 2: hi
  - Network identifiers (IP addresses)
    - Host 1: H1
    - Host 2: H2
- Question
  - Describe the evolution of the Ethernet headers and IP headers of both packets

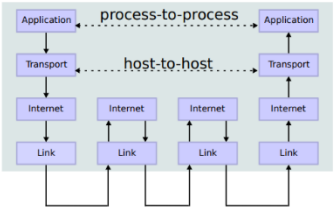
Example (cont.)

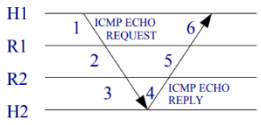
- Solution

Network Topology



Data Flow





Link	Packet	Header			
		Ethernet		IP	
		src	dst	src	dst
H1-R1	1	h1	r11	H1	H2
R1-R2	2	r12	r21	H1	H2
R2-H2	3	r22	h2	H1	H2
H2-R2	4	h2	r22	H2	H1
R2-R1	5	r21	r12	H2	H1
R1-H1	6	r11	h1	H2	H1

Exercici resolt: 2020t-c1-sol.pdf

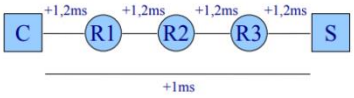
Duració: 1h 30 minuts. El test es recollirà en 25 minuts. ⟶ ~8 preguntes

Test (3,5 punts). Les preguntes valen la mitat si hi ha un error i 0 si hi ha més d'un error a la resposta.

1. El temps de transmissió d'un paquet de 1500 octets a 10 Mbps és 1,2 ms. En un enllaç determinat, el temps de propagació extrem a extrem entre un client i un servidor és d'1 ms. En aquest cas, el retard total extrem a extrem quan no hi ha cap node intermedi és de 2,2 ms.
Si afegim tres routers entre el client i el servidor:

- ☐ El retard mínim extrem a extrem serà 2,2 ms.
- ☐ El retard extrem a extrem serà com a màxim 6,6 ms.
- ☐ El retard mínim extrem a extrem serà 5,8 ms.
- ☐ El retard mínim extrem a extrem serà 4,6 ms.

$$temps\ transmissió = 1500\ octets \frac{8\,b}{1\,octet} \frac{1\,Mb}{1000000\,b} \frac{1}{10\frac{Mb}{s}} = 1{,}2\,ms$$



$$retard\ mínim\ extrem\ a\ extrem = 4 * 1{,}2\,ms + 1\,ms = 5{,}8\,ms$$

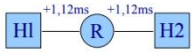
Exercici resolt: 2021t-c1-sol.pdf

2. Dos dispositius estan connectats a través d'un router. Suposem que el temps de propagació extrem a extrem és zero, que el router no afegeix retard a les cues i que la velocitat de transmissió dels enllaços és 10 Mbps.

- ☐ Si el paquet té 1400 octets (bytes) el temps de transmissió del paquet és 0'14ms
- ☐ Si el paquet té 1400 octets (bytes) el temps de transmissió del paquet és 1'12ms.
- ☐ Si el paquet té 1400 octets (bytes) el temps total fins que ha arribat a l'altre extrem és 2'24ms.
- ☐ Si es transmeten dos paquets de 700 octets (bytes) el temps total fins que el segon paquet arriba a l'altre extrem és 1'68ms.

1 pk de 1400 octets

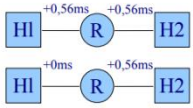
$$temps\ transmissió = 1400\ octets \frac{8\,b}{1\,octet} \frac{1\,Mb}{1000000\,b} \frac{1}{10\frac{Mb}{s}} = 1{,}12\,ms$$



$$temps\ total\ extrem\ a\ extrem = 2 * 1{,}12\,ms = 2{,}24\,ms$$

2 pk de 700 octets

$$temps\ transmissió = 700\ octets \frac{8\,b}{1\,octet} \frac{1\,Mb}{1000000\,b} \frac{1}{10\frac{Mb}{s}} = 0{,}56\,ms$$



⟶ 0ms pel primer hop del segon paquet perquè es fa en paral·lel al segon hop del primer paquet

$$temps\ total\ extrem\ a\ extrem = 3 * 0{,}56\,ms = 1{,}68\,ms$$

3. El model de referència ISO defineix 7 nivells: físic, enllaç de dades, xarxa, transport, sessió, presentació i aplicació.

- ☐ Tots els dispositius d'usuari i els routers de la xarxa gestionen (implementen) els 7 nivells.
- ☐ El model de referència TCP/IP agrupa els nivells de sessió, presentació i aplicació en un únic nivell d'aplicació.
- ☐ Tots els routers gestionen els nivells físic, enllaç de dades, xarxa i transport.
- ☐ El nivell de transport només el gestionen els dispositius d'usuari ("*hosts*").

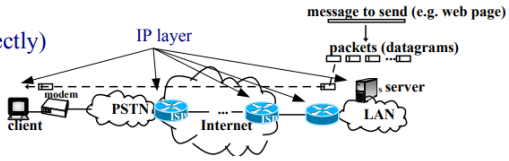
6. Sobre el model de comunicació client-servidor.

- ☐ Un host pot actuar a la vegada com a client i com a servidor.
- ☐ Els paquets d'una comunicació entre processos client i servidor s'identifiquen amb les adreces IP origen i destinació, els ports de client i de servidor, i el protocol.
- ☐ Un dispositiu pot establir moltes comunicacions com a client amb el mateix servidor i protocol.
- ☐ Un dispositiu amb una única adreça IP pot mantenir simultàniament moltes comunicacions client-servidor amb molts servidors diferents.

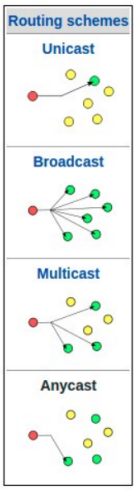
TEMA 2 - Xarxes IP

IP Layer Service

- Internet Protocol (IP)
  - Goal: routing packets from host to host
  - Design principle: interconnect hosts attached to LANs/WANs networks of different technologies
  - Characteristics
    - Connectionless (send/receive directly)
    - Stateless (no memory)
    - Best effort (no guarantee)



  - Design implications
    - Packets can be delivered out-of-order
    - Each packet can take a different path to the destination
    - No error detection (=> no correction) in payload
      - Left to higher layers, if any
    - No congestion control (beyond "drop")
  - Routing schemes: destinations
    - Unicast: one-to-one (multi-hop)
      - Ex. application layer: classic client-server
    - Broadcast: one-to-all (broadcast domain – data link layer)
      - Ex. link+IP layer: ARP request (upcoming slides)
    - Multicast: one-to-many
      - IP layer: Requires multicast routing: not supported by Internet
    - Anycast:
      - Application layer: DNS root nameservers clusters

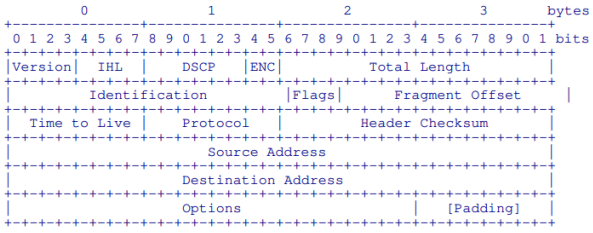


El procediment de **rebre els datagrames, processar-los i emmagatzemar-los fins que es transmeten **es coneix com a ***store and forward***. Cal destacar que les **taules d'encaminament** del router han d'estar **inicialitzades**. Si el router rep un datagrama dirigit a una **direcció desconeguda, el descarta**.

IPv4 Header (RFC 791 + updates)

Offsets	Octet	0							1							2								3						
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length														
4	32	Identification																Flags			Fragment Offset										
8	64	Time To Live								Protocol								Header Checksum														
12	96	Source IP Address																														
16	128	Destination IP Address																														
20	160	Options (if IHL > 5)																														
⋮	⋮																															
56	448																															

https://en.wikipedia.org/wiki/IPv4#Head



- 14 fields, 13 required
- `Version` – [4b] Version of the header: IPv4 vs. IPv6
  - `0100` for IPv4
  - `0110` for IPv6
- `IHL` (Internet Header Length) – [4b] Number of 32-bit words in the header
  - Min 5 (4*5 = 20 bytes); max 15 (4*15 = 60 bytes)
  - Length = [20, 24, 28, …, 60] bytes
- `DSCP` & `ECN` (previously `ToS`) – [6b] & [2b]
  - Differentiated Services Code Point – related to quality of service (QoS) management (e.g. voice over IP, VoIP)
  - Explicit Congestion Notification – related to end-to-end notification of network congestion
- `Total Length` – [16b] Packet size in bytes (header + data)
  - Min 20 bytes (header without data)
  - Max 65,535 (2^{16})
- `Identification` – [16b] Used for identifying the group of fragments of a single (fragmented) IP datagram
- `Flags` – [3b] Used to control fragmentation or identify fragments
  - bit 0: Reserved; always zero
  - bit 1: Don't Fragment (DF)
  - bit 2: More Fragments (MF)
- `Fragment offset` – [13b] Offset of a particular fragment relative to the beginning of the original unfragmented IP datagram in units of eight-byte blocks
- **`Time to Live`** – [8b] Each hop is decreased by one. IF zero, the router discards the packet and typically sends an ICMP time exceeded message to the sender.
- **`Protocol`** – [8b]
  - `0x01` (1) for ICMP
  - `0x04` (4) for IP-in-IP
  - `0x06` (6) for TCP
  - `0x11` (17) for UDP
- `Header checksum` – [16b] Used for error-checking of the header. Each router calculates the checksum of the header and compares it to the checksum field. If the values do not match, the router discards the packet.
- **`Source IP address`** – [32b]
- **`Destination IP address`** – [32b]
- Options – [max 40B] Options in 32-byte words

Fragmentació

IP **pot fragmentar** un datagrama quan la ***Maximum Transfer Unit* (MTU)** del nivell d'enllaç on ha d'encaminar-se té una **mida menor que la del datagrama.** Això sol passar en els següents casos:

- Quan un router ha d'encaminar un datagrama per una xarxa amb MTU menor que la d'on ha arribat.
- Quan es fa servir **UDP** i l'aplicació fa una **escriptura major que l'MTU de la xarxa**.

Quan es produeix la fragmentació, el reassemblatge es fa en la destinació dels datagrames. Per poder fer el reassemblatge es fan servir els camps:

- ***Identification*** *(16 bits)*: El nivell IP del *host* que genera els datagrames hi posa el valor d'un comptador que incrementa cada cop que es genera un nou datagrama. **Aquest número permet identificar fragments d'un mateix datagrama.**
- ***Flags*** *(3 bits)*: Són **0DM**. El primer bit no s'utilitza, els altres són:
  - D: **Flag de *Don't fragment***. Quan està a **'1' el datagrama no es pot fragmentar**. Si un router necessita fer-ho **el descartarà** i enviarà un missatge ICMP d'error. Es fa servir en el mecanisme MTU *path discovery*.
  - M: Flag de ***More fragments***. Si està a 1 vol dir que **hi ha més fragments**. Tots els datagrames el porten a '1' **excepte l'últim**.
- ***Offset*** *(13 bits)*: Posició del **primer byte** del fragment **en el datagrama original** (el primer val '0'). Es compta en unitats de 8 bytes.

MTU Path Discovery

El nivell de transport **TCP agrupa els bytes que escriu l'aplicació** fins a tenir un segment de **mida òptima** i després **l'envia**. La fragmentació no és desitjable perquè:

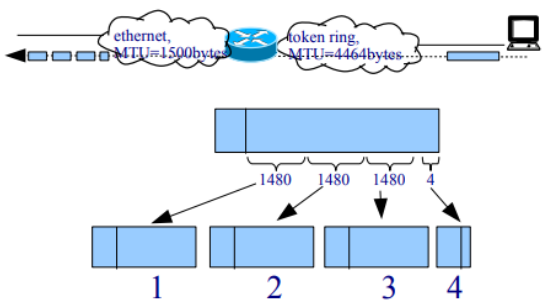
1. **Relantitza els routers**.
2. Pot provocar que hi hagi **fragments de mida petita** que redueixen **l'eficiència de la xarxa.**
3. Si es perd **un sol fragent**, s'han de **descartar tots els altres quan arriben a la destinació**.

MTU Path Discovery, consisteix en:

Primer proven d'**adaptar la mida dels segments a l'MTU de la xarxa on està connectat el *host***. Els datagrames s'envien amb el bit de ***Don't fragment*** **activat**. Si un router intermedi ha d'enviar-los per una **xarxa d'MTU menor, descartaran el datagrama** i enviaran un missatge ICMP d'**error *fragmentation needed but DF set***.. Quan el *host* rep el missatge d'error, **TCP redueix la mida dels segments** per adaptar-los en aquesta MTU.

IP Fragmentation - Example

- Original datagram = 4464 bytes (4Mbps Token Ring): 20 header + 4444 payload.
- Fragment size = $\left\lfloor \frac{1500-20}{8} \right\rfloor = 185$ 8-byte-words (1480 bytes) — Reminder: fragments sizes in bytes are multiple of 8
  - 1st fragment: offset = 0 , M = 1. 0~1479 payload bytes.
  - 2nd fragment: offset = 185, M = 1. 1480~2959 payload bytes.
  - 3rd fragment: offset = 370, M = 1 . 2960~4439 payload bytes.
  - 4th fragment: offset = 555, M = 0 . 4440~4443 payload bytes.



Adreces IP

Tenen **32 bits** (4 bytes). La següent figura mostra el format d'una adreça IP.

***Netid* identifica la xarxa i el *hostid* identifica un *host* dintre la xarxa**. El límit entre el netid i el hostid és variable. La notació que es fa servir es coneix com a *notació amb punts* i consisteix a expressar els **4 bytes de l'adreça en decimal separats per punts**, per exemple 147.83.34.25.

Hi ha diferents classes de ip; la classe **D és per adreces *multicast*** (per exemple la 224.0.0.1 identifica "*All systems on this Subnet*") i la **classe E són adreces reservades**. Tipus d'adreces:

Class	Netid [bytes]	Hostid [bytes]	Number of subnets	Netmask (slide 13)	Leading bits	Range
A	1	3	$2^7 = 128$	/8	0	0.0.0.0 – 127.255.255.255
B	2	2	$2^{14} = 16,384$	/16	10	128.0.0.0 – 191.255.255.255
C	3	1	$2^{21} = 2,097,152$	/24	110	192.0.0.0 – 223.255.255.255
D	-	-	-	-	1110	224.0.0.0 – 239.255.255.255
E	-	-	-	-	1111	240.0.0.0 – 255.255.255.255

L'assignació d'aquestes adreces es fa tenint en compte que:

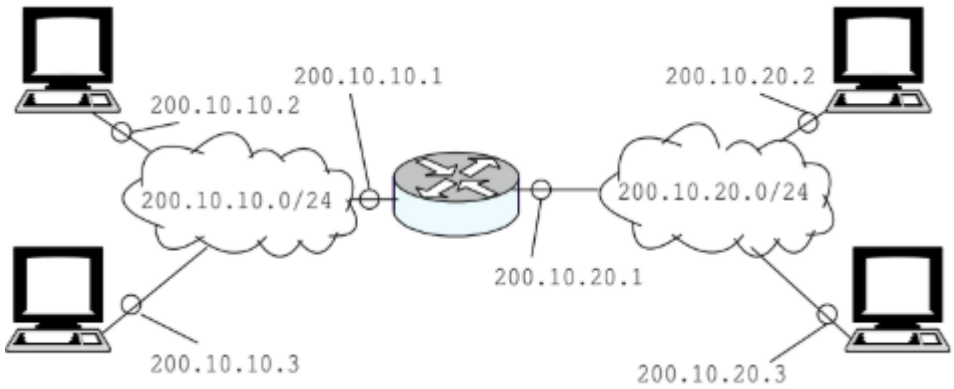
- **Una adreça identifica una interfície.**
- Totes les adreces d'una **mateixa xarxa IP** tenen el **mateix netid.**
- Totes les **adreces** assignades han de ser **diferents entre elles**

No totes les adreces es poden fer servir per numerar les interfícies. A continuació veiem les **adreces especials**:

netid	hostid	Significat
xxx	tot '0'	Identifica una xarxa. Es fa servir en les taules d'encaminament.
xxx	tot '1'	*Broadcast* en la xarxa xxx.
tot '0'	tot '0'	Identifica "aquest *host*" en "aquesta xarxa". Es fa servir com adreça origen en protocols de configuració (DHCP, vegeu la secció 2.7).
tot '1'	tot '1'	*Broadcast* en "aquesta xarxa". Es fa servir com adreça destinació en protocols d'autoconfiguració (DHCP, vegeu la secció 2.7).
127	xxx	*Loopback*: Comunicació entre processos en el mateix *host* amb TCP/IP.

A continuació veiem un exemple d'assignació d'adreces, destaquem que:

- Totes **les interfícies d'una mateixa xarxa tenen el mateix prefix** (netid).
- **No es poden fer servir adreces especials per a numerar interfícies**. Cada xarxa en té **dues**: la de la ***xarxa*** (amb el hostid tot a '0', que és la primera adreça disponible en el rang d'adreces de la xarxa) i la del ***broadcast*** en la xarxa (amb el hostid tot a '1', que és la ultima disponible en el rang d'adreces de la xarxa).
- El router ha de tenir assignada **una adreça en cada interfície**. Cada adreça ha de tenir **el netid de la xarxa on està connectada la interfície**.



Les **adreces han de ser úniques en tot internet**. Per això, l'organisme **IANA** assigna blocs d'adreces als ***Regional Internet Registers (RIR)***: ***RIEPE*** (EUR), ***ARIN*** (USA), ***APNIC*** (ASIA), ***LACNIC*** (LATAM).

A la vegada els RIR assignen blocs d'adreces als ***ISP*** i aquests als seus abonats..

Adreces Privades - RFC 1918

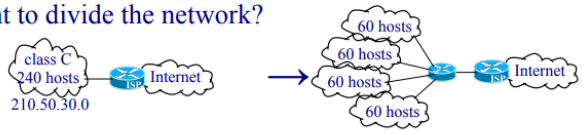
- **Private addresses** has been reserved for devices not using public addresses. These addresses are not assigned to any RIR (are not unique). There are addresses in each class:
  - 1 class A network: 10.0.0.0
  - 16 class B networks: 172.16.0.0 ~ 172.31.0.0
  - 256 class C networks: 192.168.0.0 ~ 192.168.255.0

Note: 169.254.0.0 – 169.254.255.255 is also a private range (used for IP autoconfiguration)

Subnetting

Per a expressar quina és la mida del netid es fa servir una màscara. La màscara és un número de 32 bits amb els bits més significatius que es corresponen amb el netid a '1' i els altres a '0'.

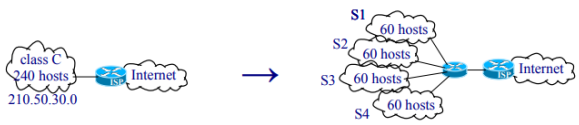
- Initially the netid was given by the address class: A with 2^{24} addresses, B with 2^{16} addresses and C with 2^{8} addresses.
- What if we want to divide the network?



- Subnetting allows adding bits from the hostid to the netid (called subnetid bits).
- Example: For the ISP the network prefix is 24 bits. For the internal router the network prefix is 26 bits. The 2 extra bits allows 4 "subnetworks".
- A **mask** (*netmask*) is used to identify the size of the netid+subnetid prefix.
- Mask notations:
  - dot notation, as 255.255.255.192
  - slash notation, giving the mask length (number of bits) as 210.50.30.0/26 (=> mask length: 26 bits)

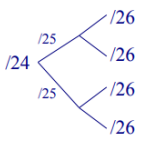
IP Addresses – Subnetting Example

- We want to subnet the address 210.50.30.0/24 in 4 subnets



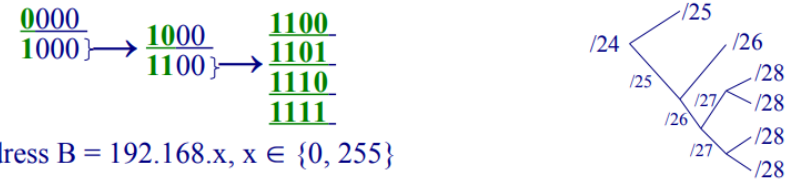
Base address B = 210.50.30

subnet	subnetid	IP net. addr.	range	broadcast	available
S1	00	B.0/26	B.0 ~ B.63	B.63	$2^6 - 2 = 62$
S2	01	B.64/26	B.64 ~ B.127	B.127	$2^6 - 2 = 62$
S3	10	B.128/26	B.128 ~ B.191	B.191	$2^6 - 2 = 62$
S4	11	B.192/26	B.192 ~ B.255	B.255	$2^6 - 2 = 62$



IP Addresses – Variable Length Subnet Mask (VLSM)

- Subnetworks of different sizes.
- Example, subnetting a class C address:
  - We have 1 byte for subnetid + hostid.
  - subnetid is green, chosen subnets addresses are underlined.



Base address B = 192.168.x, x ∈ {0, 255}

subnet	subnetid	IP net. addr.	range	broadcast	available
S1	0	B.0/25	B.0 ∼ B.127	B.127	$2^7 - 2 = 126$
S2	10	B.128/26	B.128 ∼ B.191	B.191	$2^6 - 2 = 62$
S3	1100	B.192/28	B.192 ∼ B.207	B.207	$2^4 - 2 = 14$
S4	1101	B.208/28	B.208 ∼ B.223	B.223	$2^4 - 2 = 14$
S5	1110	B.224/28	B.224 ∼ B.239	B.239	$2^4 - 2 = 14$
S6	1111	B.240/28	B.240 ∼ B.255	B.255	$2^4 - 2 = 14$

- In order to reduce routing tables size, CIDR proposed a "rational" geographical-based distribution of IP addresses to be able to "aggegate routes", and use masks instead of classes.
- Aggregation example:

200.1.10.0/24
200.1.11.0/24 ⟶ 200.1.10.0/23

Example

- We have been assigned the 192.169.1.0/24 (public) address block
- We want to split it into two equal subnetworks
- Secondly, we want to split in the same manner the two subnetworks
- Challenge: for each of the three cases find the network address, the network mask, the total number of IPs available for hosts, the lowest and the highest host IP and the broadcast IP

1) Whole block (/24)

```
user@dac:~$ ipcalc 192.169.1.0/24
Address:   192.169.1.0          11000000.10101001.00000001. 00000000
Netmask:   255.255.255.0 = 24   11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255            00000000.00000000.00000000. 11111111
=>
Network:   192.169.1.0/24       11000000.10101001.00000001. 00000000
HostMin:   192.169.1.1          11000000.10101001.00000001. 00000001
HostMax:   192.169.1.254        11000000.10101001.00000001. 11111110
Broadcast: 192.169.1.255        11000000.10101001.00000001. 11111111
Hosts/Net: 254                   Class C
```

2) 2 subnets (two /25)

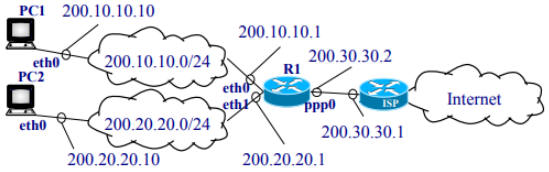
```
user@dac:~$ ipcalc 192.169.1.0/25
Address:   192.169.1.0          11000000.10101001.00000001.0 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.169.1.0/25       11000000.10101001.00000001.0 0000000
HostMin:   192.169.1.1          11000000.10101001.00000001.0 0000001
HostMax:   192.169.1.126        11000000.10101001.00000001.0 1111110
Broadcast: 192.169.1.127        11000000.10101001.00000001.0 1111111
Hosts/Net: 126                   Class C

user@dac:~$ ipcalc 192.169.1.128/25
Address:   192.169.1.128        11000000.10101001.00000001.1 0000000
Netmask:   255.255.255.128 = 25 11111111.11111111.11111111.1 0000000
Wildcard:  0.0.0.127            00000000.00000000.00000000.0 1111111
=>
Network:   192.169.1.128/25     11000000.10101001.00000001.1 0000000
HostMin:   192.169.1.129        11000000.10101001.00000001.1 0000001
HostMax:   192.169.1.254        11000000.10101001.00000001.1 1111110
Broadcast: 192.169.1.255        11000000.10101001.00000001.1 1111111
Hosts/Net: 126                   Class C
```

(cont.) 3) 4 subnets (four /26)

```
user@dac:~$ ipcalc 192.169.1.0/26
Address:   192.169.1.0          11000000.10101001.00000001.00 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.1.0/25       11000000.10101001.00000001.00 000000
HostMin:   192.169.1.1          11000000.10101001.00000001.00 000001
HostMax:   192.169.1.62         11000000.10101001.00000001.00 111110
Broadcast: 192.169.1.63         11000000.10101001.00000001.00 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.64/26
Address:   192.169.1.64         11000000.10101001.00000001.01 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.1.64/25      11000000.10101001.00000001.01 000000
HostMin:   192.169.1.65         11000000.10101001.00000001.01 000001
HostMax:   192.169.1.126        11000000.10101001.00000001.01 111110
Broadcast: 192.169.1.127        11000000.10101001.00000001.01 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.128/26
Address:   192.169.1.128        11000000.10101001.00000001.10 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.10 111111
=>
Network:   192.169.1.128/25     11000000.10101001.00000001.10 000000
HostMin:   192.169.1.129        11000000.10101001.00000001.10 000001
HostMax:   192.169.1.190        11000000.10101001.00000001.10 111110
Broadcast: 192.169.1.191        11000000.10101001.00000001.10 111111
Hosts/Net: 62                    Class C

user@dac:~$ ipcalc 192.169.1.192/26
Address:   192.169.1.192        11000000.10101001.00000001.11 000000
Netmask:   255.255.255.192 = 26 11111111.11111111.11111111.11 000000
Wildcard:  0.0.0.63             00000000.00000000.00000000.00 111111
=>
Network:   192.169.192/25       11000000.10101001.00000001.11 000000
HostMin:   192.169.1.193        11000000.10101001.00000001.11 000001
HostMax:   192.169.1.254        11000000.10101001.00000001.11 111110
Broadcast: 192.169.1.255        11000000.10101001.00000001.11 111111
Hosts/Net: 62                    Class C
```

Routing Table

- ip_output() kernel function queries the routing table for each datagram.
- Routing can be:
  - Direct: The destination is directly connected to an interface.
  - Indirect: Otherwise. In this case, the datagram is sent to a router.
- Default route: Is an entry where to send all datagrams with a destination address to a network not present in the routing table. The default route address is 0.0.0.0/0.
- Hosts routing tables usually have two entries: The network where they are connected and a default route.



Genmask == Netmask

PC1 routing table:

Destination	Genmask	Gateway	Iface
200.10.10.0	255.255.255.0	0.0.0.0	eth0
0.0.0.0	0.0.0.0	200.10.10.1	eth0

PC2 routing table:

Destination	Genmask	Gateway	Iface
200.20.20.0	255.255.255.0	0.0.0.0	eth0
0.0.0.0	0.0.0.0	200.20.20.1	eth0

R1 routing table:

Destination	Genmask	Gateway	Iface
200.10.10.0	255.255.255.0	0.0.0.0	eth0
200.20.20.0	255.255.255.0	0.0.0.0	eth1
200.30.30.1	255.255.255.255	0.0.0.0	ppp0
0.0.0.0	0.0.0.0	200.30.30.1	ppp0

known destinations: network & mask

how to reach the destinations: (local) interface & gateway (next hop IP)

Routing Table – Another Unix Example

- Basic topology: Laptop connected to an access point (AP)

```
user@dac:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eht0
```

- Laptop – AP with 2 other connections

U – Route up
G – Gateway
H – Host

```
user@dac:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.1     0.0.0.0         UG    600    0        0 eth0
10.0.0.0        192.168.1.2     255.0.0.0       UG    0      0        0 eth0
10.0.0.2        192.168.1.3     255.255.255.255 UGH   0      0        0 eth0
192.168.1.0     0.0.0.0         255.255.255.0   U     600    0        0 eth0
```

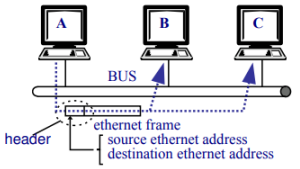
Default route
Network route
Host route
Local network

Warning: In GNU/Linux `route` is deprecated; `ip route show` (from the iproute2 package) must be used instead

Warning: routes with a longer prefix always take priority so in this case lower routes have priority (e.g. 10.0.0.2/32 over 10.0.0.0/8); however, routes shown by `route` may no respect this criterion; `ip route show` respects it.

Address Resolution Protocol, ARP (RFC 826)

- To send the datagram, IP layer may have to pass a "physical address" to the NIC driver. Physical addresses are also called MAC or hardware addresses.
- ARP translate IP addresses to "physical addresses" (used by the physical network).
- If needed, IP calls ARP module to obtain the "physical addresses" before the NIC driver call.
- Ethernet example:



Ethernet bus deployments are not used any more.

WiFi is a more up-to-date example: in a WiFi network all nodes receive all packages because they share physical layer => the MAC addresses are very meaningful. .

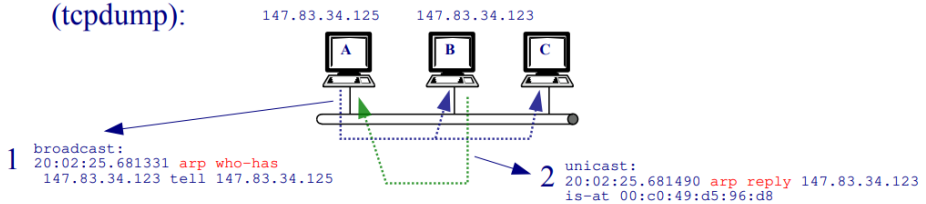
Note: In IPv6, the protocol that links IP to MAC addresses is the Neighbor Discovery (ND) Protocol.

Address Resolution Protocol, messages

- When IP calls ARP:
  - If ARP table has the requested address, it is returned,
  - otherwise:
    - IP stores the datagram in a temporal buffer, and a resolution protocol is triggered.
    - IP initiates a timeout and starts forwarding the next datagram in the transmission queue.
    - If the timeout triggers before resolution, the datagram is removed.
    - If ARP returns the requested address, IP calls the driver with it.
- ARP resolution in an ethernet network (broadcast network):
  - A broadcast "ARP Request" message is sent indicating the IP address.
  - The station having the requested IP address sends a unicast "ARP Reply", and stores the requesting address in the ARP table.
  - Upon receiving the "ARP Reply", the requesting station return the IP call with it.
  - ARP entries have a timeout refreshed each time a match occurs.

Address Resolution Protocol, messages - Example

- ARP messages (tcpdump):



- ARP tables:

```
A> /sbin/arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
147.83.34.123            ether   00:c0:49:d5:96:d8   C                     eth0

B> /sbin/arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
147.83.34.125            ether   00:14:F1:CC:59:00   C                     eth0
```

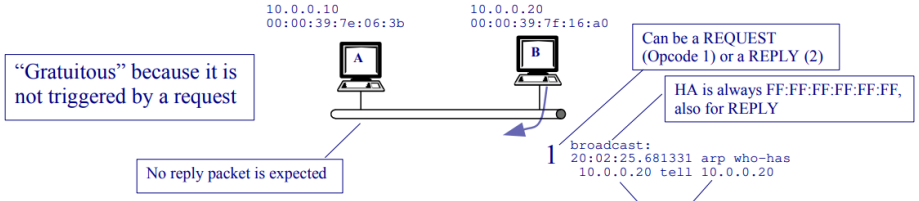
"Completed" flag

Warning: In GNU/Linux `arp` is deprecated; `ip link show` (from the iproute2 package) must be used instead (use 'ip l -s' to get statistics)

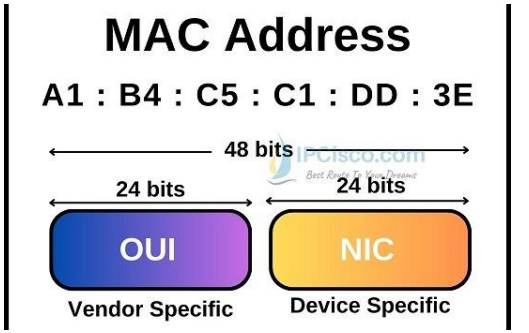
- Another example:

Time	No.	Source	Destination	Protocol	Length	Info	Pk len	Hardware src	Hardware dst
0.000000000	1	02:42:ac:1c:00:03	ff:ff:ff:ff:ff:ff	ARP	42	Who has 172.28.0.2? Tell 172.28.0.3	42	02:42:ac:1c:00:03	ff:ff:ff:ff:ff:ff
0.000148889	2	02:42:ac:1c:00:02	02:42:ac:1c:00:03	ARP	42	172.28.0.2 is at 02:42:ac:1c:00:02	42	02:42:ac:1c:00:02	02:42:ac:1c:00:03
0.000203120	3	172.28.0.3	172.28.0.2	ICMP	98	Echo (ping) request id=0x0014, seq=1/256, ttl=64 (reply in 4)	98	02:42:ac:1c:00:03	02:42:ac:1c:00:02
0.000304023	4	172.28.0.2	172.28.0.3	ICMP	98	Echo (ping) reply id=0x0014, seq=1/256, ttl=64 (request in 3)	98	02:42:ac:1c:00:02	02:42:ac:1c:00:03
5.159069240	5	02:42:ac:1c:00:02	02:42:ac:1c:00:03	ARP	42	Who has 172.28.0.3? Tell 172.28.0.2	42	02:42:ac:1c:00:02	02:42:ac:1c:00:03
5.159182181	6	02:42:ac:1c:00:03	02:42:ac:1c:00:02	ARP	42	172.28.0.3 is at 02:42:ac:1c:00:03	42	02:42:ac:1c:00:03	02:42:ac:1c:00:02
2315.53197…	7	fe80::42:66ff:fe2a:…	ff02::fb	MDNS	210	Standard query 0x0000 PTR _ipps._tcp.local, "QM" question PTR _p…	210	02:42:66:2a:02:ba	33:33:00:00:00:fb

Address Resolution Protocol – Gratuitous ARP



- Goals:
  - Detect duplicated IP addresses.
  - Update MAC addresses in ARP tables after an IP or NIC change.



Address Resolution Protocol – Message format (ethernet) (cont.)

- HTYPE (hardware type) – [2B] Specifies the network link protocol type
  - `0x0001` for Ethernet
- PTYPE (protocol type) – [2B] Specifies the internetwork protocol for which the ARP request is intended. Same numbering as EtherType (see Ethernet II header, Unit 3)
  - `0x0800` for IPv4
- HLEN (hardware length) - [1B] Specifies the hardware address length in octets
  - 6 for Ethernet (MAC address)
- PLEN (protocol length) - [1B] Specifies the internetwork address length in octets
  - 4 for IPv4
- OPER (operation) - [2B] Specifies the operation that the sender is performing
  - 1 for request
  - 2 for reply
- SHA (sender hardware address) [6B for Ethernet] Media address of the sender
  - In a request indicates the address of the host sending the request
  - In a reply indicates the address of the host that the request was looking for
- SPA (sender protocol address) [4B for IPv4] Internetwork address of the sender
- THA (Target hardware address) Media address of the intended receiver
  - In a request is ignored
  - In a reply indicates the address of the host that originated the ARP request
- TPA (Target protocol) [4B for IPv4] Internetwork address of the intended receiver

Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

https://en.wikipedia.org/wiki/IPv4#Header

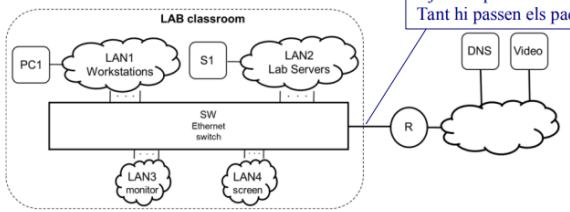
Exercici: 2021t-ef

Unit 2: IP Networks

Problema 1 (3,5 punts)

La figura mostra la configuració d'una aula de laboratori on hi ha llocs de treball (LAN1), servidors per donar suport als treballs dels laboratoris (LAN2), un PC de monitorització pel professor (LAN3) i una pantalla IP per a vídeo (LAN4). Cada laboratori disposa d'un commutador Ethernet (SW) on es configuren les 4 xarxes locals virtuals (VLAN) i l'adreçament proposat per a cada aula és 192.168.aula.0/24. El router R dona servei a més de 40 laboratoris amb la mateixa configuració.

Ajuda: aquest és l'enllaç que hem d'analitzar. Tant hi passen els paquets PC1-R com R-S1

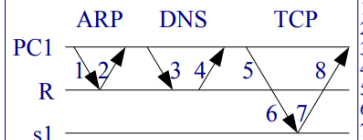


Ajuda: una connexió TCP comença amb el client enviat un SYN, al qual el servidor respon amb un SYN/ACK ...

c) (0,5 punts) La configuració que obté PC1 és: 192.168.1.2; router per defecte (gw): 192.168.1.1; DNS: 147.83.3.3. El PC1 inicia una connexió TCP amb el servidor s1-aula.fib.upc.edu. Completa la seqüència de trames i datagrames que passen per l'enllaç entre el commutador Ethernet i el router fins que PC1 rep el SYN/ACK. El router R ja té la informació a la taula ARP de tots els servidors.

Notació: majúscules per les adreces IP i minúscules per les adreces Ethernet (MAC). Exemple: PC1, pc1.

Ajuda: el cronograma d'intercanvi de paquets ens ajuda a visualitzar millor aquest intercanvi, I a comptar la quantitat de paquets



	Ethernet		ARP		IP			
	Origen	Destinació	Comanda	Missatge	Origen	Destinació	Protocol	Contingut
1	pc1	FF...FF	REQ	R?				
2	r	pc1	RESP	R -> r				
3	pc1	r			PC1	DNS	UDP	s1-aula.fib.upc.edu A?
4	r	pc1			DNS	PC1	UDP	DNS A=S1
5	pc1	r			PC1	S1	TCP	SYN
6	r	s1			PC1	S1	TCP	SYN
7	s1	r			S1	PC1	TCP	SYN/ACK
8	r	pc1			S1	PC1	TCP	SYN/ACK

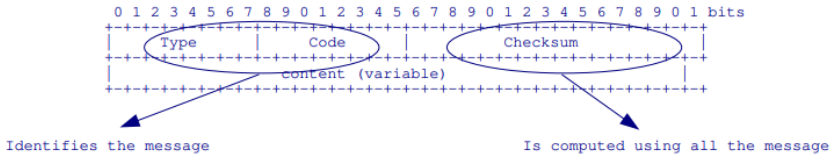
Advertència: no sempre cal omplir totes les cel·les de les taules dels anunciats

Internet Control Message Protocol, ICMP (RFC 792)

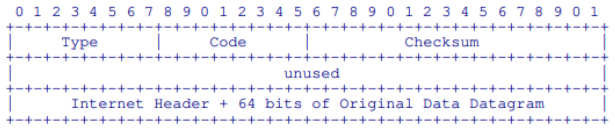
- Used for attention and error messages.
- Can be generated by
  - IP (e.g. TTL expiration)
  - ARP (e.g. resolution not possible)
  - Applications (e.g. ping)
- Are encapsulated into an IP datagram
  - (no UDP/TCP!)
  - It is an internet layer protocol
- Can be
  - i) query
  - ii) error
- Error messages are sent to the source IP of the datagram that generated the error condition
- An ICMP error message cannot generate another ICMP error message (to avoid loops)

Error messages only! Ping is a query message => it can trigger an ICMP error message (e.g. "network unreachable", "time exceed")

ICMP general format message (RFC 792)



- Query type messages have an identifier field, for request-reply correspondence.
- Error messages have a field where the first 8 bytes of the datagram payload causing the error are copied. These bytes capture the TCP/UDP ports. E.g. Destination Unreachable Message:



Code	Message	Use
1	DHCPDISCOVER	Client broadcast to locate available servers.
2	DHCPOFFER	Server to client in response to DHCPDISCOVER with offer of configuration parameters.
3	DHCPREQUEST	Client message to servers either (a) requesting offered parameters from one server and implicitly declining offers from all others, (b) confirming correctness of previously allocated address after, e.g., system reboot, or (c) extending the lease on a particular network address.
5	DHCPACK	Server to client with configuration parameters, including committed network address.

Common ICMP messages

Type	Code	query/error	Name	Description
0	0	query	echo reply	Reply an echo request
3	0	error	network unreachable	Network not in the RT.
	1	error	host unreachable	ARP cannot solve the address.
	2	error	protocol unreachable	IP cannot deliver the payload
	3	error	port unreachable	TCP/UDP cannot deliver the payload
	4	error	fragmentation needed and DF set	MTU path discovery
4	0	error	source quench	Sent by a congested router.
5	0	error	redirect for network	When the router send a datagram by the same interface it was received.
8	0	query	echo request	Request for reply
11	0	error	time exceeded, also known as TTL=0 during transit	Sent by a router when `--TTL=0`

Dynamic Host Configuration Protocol, DHCP (RFC 2131)

- Improves and can interoperate with previous BOOTP protocol.
- Used for automatic network configuration for assigning:
  - IP address and mask,
  - Default route,
  - Hostname,
  - DNS domain,
  - DNS servers,
  - etc.
- IP address configuration can be:
  - Dynamic: During a leasing time.
  - Automatic: Unlimited leasing time.
  - Manual: IP addresses are assigned to specific MAC addresses.
- It is an application layer protocol (server-client paradigm)
  - Uses UDP as transport protocol

DHCP – Message Fields (RFC 2131)

(informative slide, don't learn the message fields by heart!)

Only these 2 exist (not to be confused with `options`).
BOOTREQUEST: client to server
BOOTREPLY: server to client

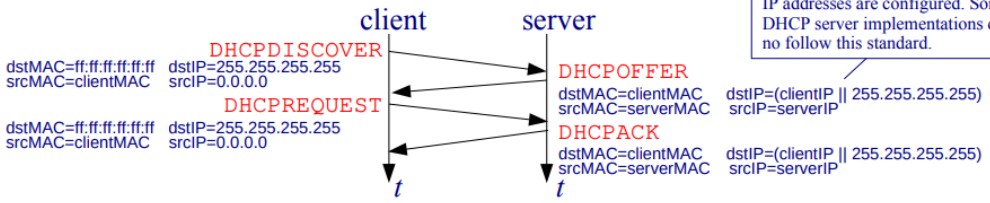
FIELD	OCTETS	DESCRIPTION
op	1	Message op code / message type. 1 = BOOTREQUEST, 2 = BOOTREPLY.

DHCP – Protocol Messages (RFC 2131)

Unit 2: IP Networks

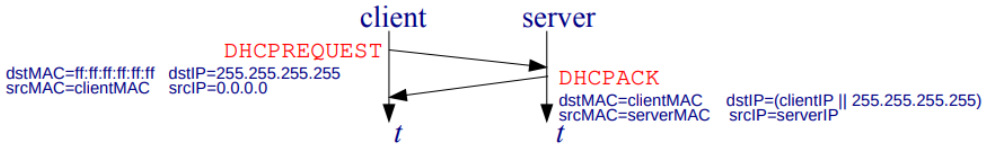
DHCP – Client-server interaction (RFC 2131)

- UDP, server port = 67, client port = 68.



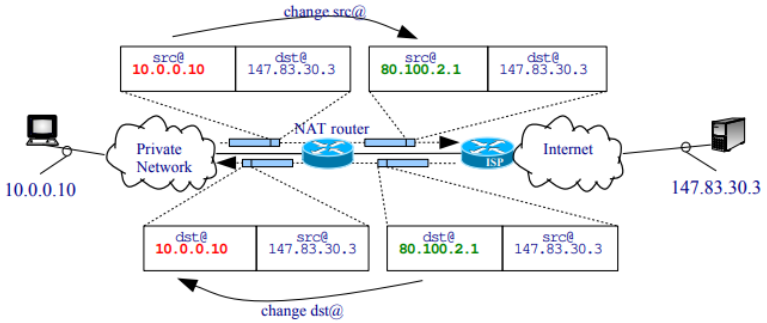
The client specifies whether in the reply (BOOTREPLY) the destination IP should be unicast or broadcast (i.e. clinetIP or 255.255.255.255). Unicast is the standard. Broadcast must be used only by those (old) hosts which cannot accept unicast packets before IP addresses are configured. Some DHCP server implementations do no follow this standard.

- The client can directly send `DHCPREQUEST`:
  - After rebooting if it remembers and wishes to reuse a previously allocated network address.
  - Extending the lease on a particular network address.



Network Address Translation, NAT (RFCs 1631, 2663 3022)

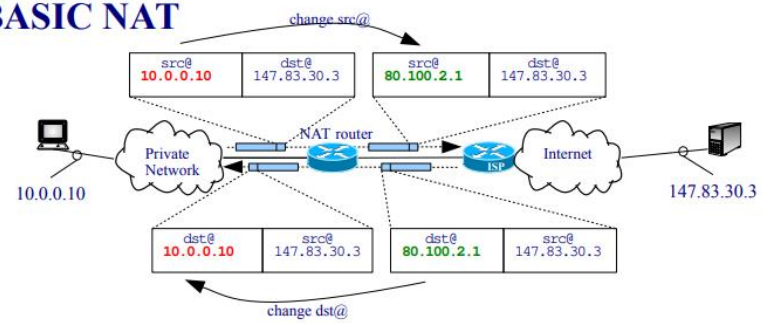
- Typical scenario: Private addresses (internal addresses) are translated to public addresses (external addresses).
- A NAT table is used for address mapping.
- Advantages:
  - Save public addresses.
  - Security.
  - Administration, e.g. changing ISP does not imply changing private network addressing.



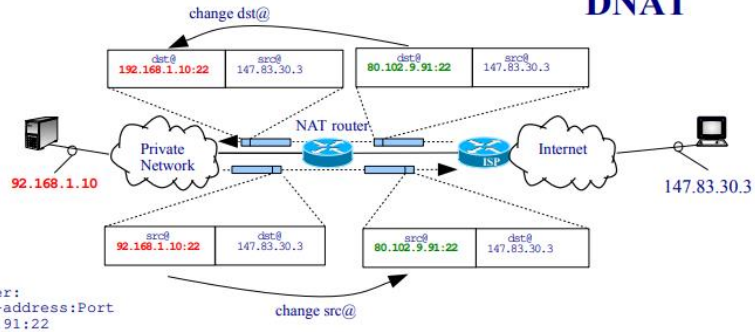
NAT – Types of translations

- Basic NAT (one-to-one):
  - A different external address is used for each internal address:
    - A different public IP address is needed for each hosts accessing Internet.
  - Each NAT table entry has the tuple: (internal address, external address).
  - Each host requires one NAT table entry.
  - Manually configured
- Port and Address Translation, PAT (one-to-many):
  - The same external address can be used for each internal address.
    - A unique public IP address can be used for all (internal) hosts accessing Internet.
  - Each NAT table entry has the tuple: (int. addr., int. port, ext. addr., ext. port)
  - Each connection requires one NAT table entry.
  - Entries are automatically added when an internal connection is initiated.
- Destination NAT, DNAT can be:
  - Enables external connections to internal servers.
  - The address translation is exactly the same as NAT, but, the connection is initiated from an external client.
  - Typically, some static configuration is needed to configure the server IP/port.

BASIC NAT



DNAT




```
Static entry in the NAT router:
Inside-address:Port  Outside-address:Port
192.168.1.10:22      80.102.9.91:22
```

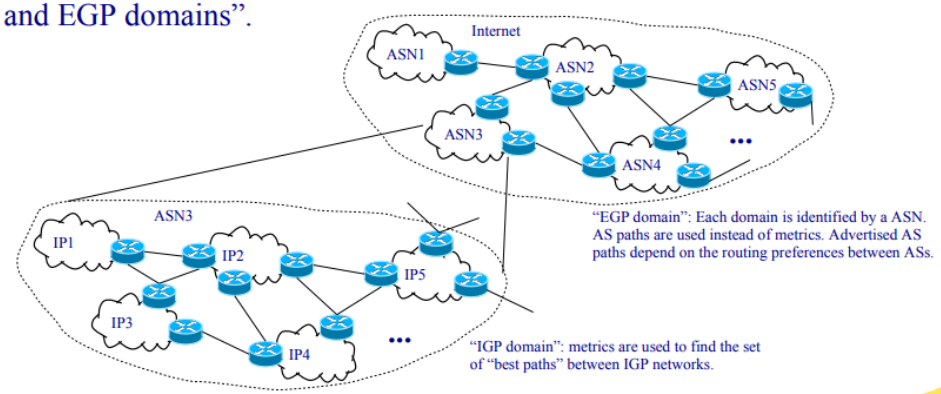
Routing algorithms

- Objective: add/update/remove entries to routing tables. Can be:
  - Static: Manual, scripts, DHCP.
  - Dynamic: Automatically update table entries, e.g. when a topology change occurs. This is done by a routing algorithm (also routing protocol).

- Internet is organized in Autonomous Systems (AS). In terms of ASs, routing algorithms are classified as:
  - Interior Gateway Protocols (IGPs): Inside the same AS. Examples:
    - RFC standards: RIP, OSPF.
    - Proprietary: CISCO IGRP.
  - Exterior Gateway Protocols (EGPs): Between different Ass.
    - Currently BGPv4.

Routing algorithms - Autonomous Systems (AS)

- AS definition (RFC 1930): "An AS is a connected group of one or more IP prefixes run by one or more network operators which has a SINGLE and CLEARLY DEFINED routing policy".
- Each AS is identified by a 16 or 32 bits AS Number (ASN) assigned by IANA.
- ASs facilitate Internet routing by introducing a two-level hierarchy: "IGP and EGP domains".

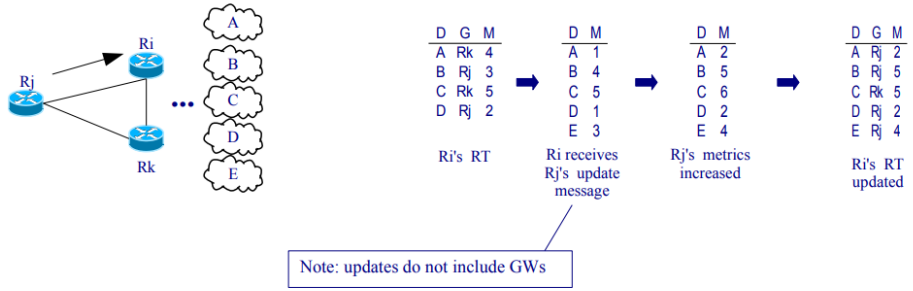


Routing Information Protocol, RIP (RFC 2453)

- The metric (distance) to a destination is the number of hops (i.e. transmissions) to reach the destination: 1 if the destination is attached to a directly connected network, 2 if 1 additional router is needed, etc.
- Routers send RIP updates every 30 seconds to the neighbors.
- RIP updates use UDP, with the same source and destination port: 520, broadcast dst. IP addr. (Version 1).
- RIP updates include destinations and metrics tuples.
- A neighbor is considered down if no RIP messages are seen during 180 seconds.
- Infinite metric is 16.
- Two versions of RIP: Version 2 i) allows variable masks (=> masks are added to the messages) and ii) uses the multicast dst. address 224.0.0.9 (all RIPv2 routers).
- This type of routing algorithms, where it is not known the whole topology but just the distance to each destination, are known as "distance-vector" and use a distributed variant of the "Bellman-Ford" algorithm for selecting the shortest path.

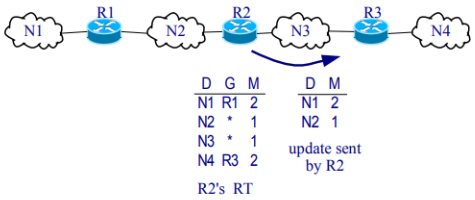
RIP – Routing Table (RT) Update Example

- Upon receiving an update
  1) Increase the message metrics (+1 to all metrics received)
  2) change the routing table if:
     - There is a better path (lower metric) towards a destination
     - The gateway being used changes the metric
     - There is a new route
- Example: When Ri receives an update message from Rj:



RIP – Count to Infinity: Solutions

- Split horizon: When the router sends the update, removes the entries having a gateway in the interface where the update is sent:



- Split horizon with Poisoned Reverse: the same as split horizon except that the entries with M=16 are not removed. The poison reverse rule overwrites split horizon rule: *poisoned routes* are also sent ("back") to the neighbor from which were learnt.
- Triggered updates: Consists of sending the update before the 30 seconds timer expires when a metric change in the routing table.
- Hold down timer (CISCO): When a route becomes unreachable (metric = 16), the entry is placed in *holddown* during 180 seconds. During this time, the entry is not updated.

To allocate time for poisoned routes to propagate.

Security in IP

- Goals:
  - Confidentiality: Who can access.
  - Integrity: Who can modify the data.
  - Availability: Access guarantee.
- Vulnerabilities:
  - Technological: Protocols (e.g. ftp and telnet send messages in "clear text") and networking devices (routers...)
  - Configuration: Servers, passwords, ...
  - Missing security policies: Secure servers, encryption, firewalls, ...

Security in IP – Attacks

- Reconnaissance: Previous to an attack.
  - Available IP addresses.
  - Available servers and ports.
  - Types of OSs, versions, devices...
  - Eavesdropping
- Access: Unauthorized access to an account or service.
- Denial of Service: Disables or corrupts networks, systems, or services.
- Viruses, worms , trojan horses...: Malicious software that replicate itself.

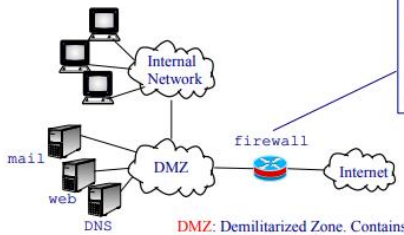
Security in IP – Basic Solutions

- Firewalls.
- Virtual Private Networks (VPN) with encrypted payload.

Security in IP – Firewalls

- Firewall: System or group of systems that enforces an access control policy to a network.
- There are many firewall types:
  - From simple packet filtering based on IP/TCP/UDP header rules,
  - to state-full connection tracking
  - and application-based filtering (packet inspection)

Middlebox is a generic term to refer devices installed between routers (frequently without knowledge of one of both communicating parties). They include: firewalls, traffic shapers, proxies, etc.



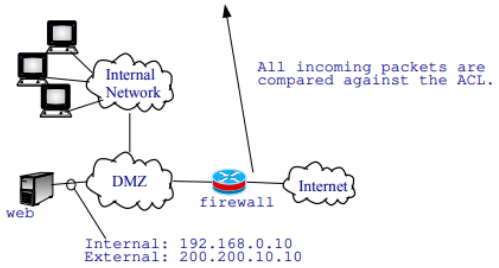
DMZ: Demilitarized Zone. Contains servers exposed to the Internet.

Security in IP – Basic Firewall Configuration

- NAT
- Access Control List, ACL

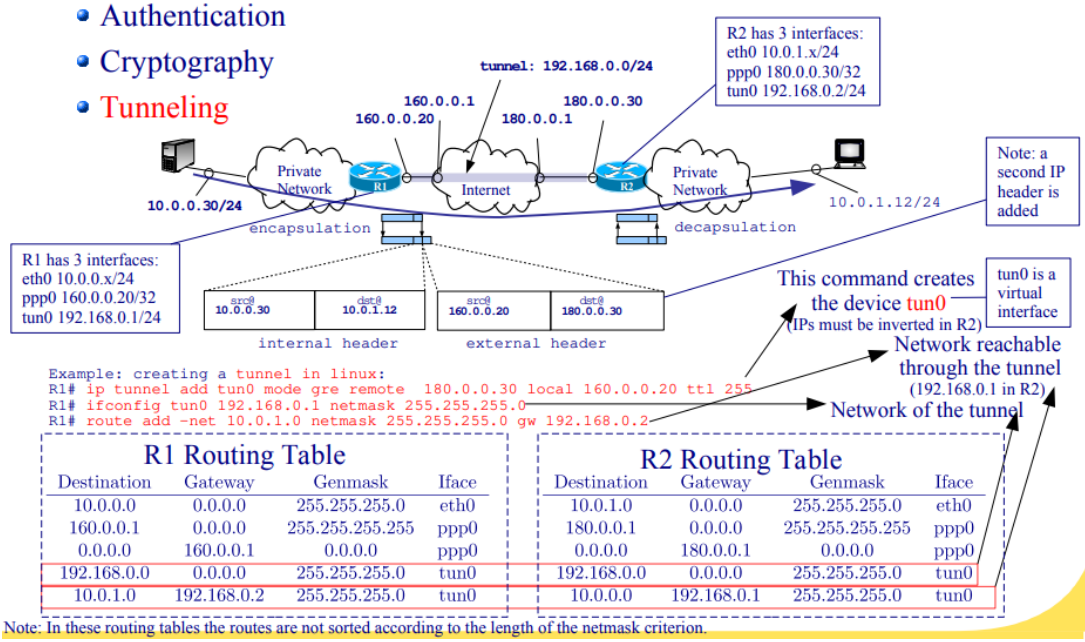
The order in which the entries appear in the list is crucial: once a packet matches a rule the corresponding action action is taken and no further entries are processed

Protocol	IP-src	IP-dst	Port-src	Port-dst	Action
TCP	*any*	200.200.10.10/32	*any*	80	*accept*
TCP	*any*	*any*	< 1024	≥ 1024	*accept*
ICMP	*any*	*any*	–	–	*accept*
IP	*any*	*any*	–	–	*deny*



Security in IP – VPN Security

- Authentication
- Cryptography
- Tunneling




```
Example: creating a tunnel in linux:
R1# ip tunnel add tun0 mode gre remote  180.0.0.30 local 160.0.0.20 ttl 255
R1# ifconfig tun0 192.168.0.1 netmask 255.255.255.0
R1# route add -net 10.0.1.0 netmask 255.255.255.0 gw 192.168.0.2
```

R1 Routing Table

Destination	Gateway	Genmask	Iface
10.0.0.0	0.0.0.0	255.255.255.0	eth0
160.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	160.0.0.1	0.0.0.0	ppp0
192.168.0.0	0.0.0.0	255.255.255.0	tun0
10.0.1.0	192.168.0.2	255.255.255.0	tun0

R2 Routing Table

Destination	Gateway	Genmask	Iface
10.0.1.0	0.0.0.0	255.255.255.0	eth0
180.0.0.1	0.0.0.0	255.255.255.255	ppp0
0.0.0.0	180.0.0.1	0.0.0.0	ppp0
192.168.0.0	0.0.0.0	255.255.255.0	tun0
10.0.0.0	192.168.0.1	255.255.255.0	tun0

Note: In these routing tables the routes are not sorted according to the length of the netmask criterion.

Security in IP – VPN Tunneling Problems

- Fragmentation inside the tunnel will use the external header, thus, the exit router of the tunnel may reassemble fragmented datagrams.
- ICMP messages sent inside the tunnel are addressed to the tunnel entry.
- MTU path discovery may fail.
- Solution: the router entry maintains a "tunnel state", e.g. the tunnel MTU, and generate ICMP messages that would be generated inside the tunnel. Furthermore, the tunnel entry router typically fragment the datagrams, if needed, before encapsulation, to avoid the exit router having to reassemble fragmented datagrams.

Security in IP – VPN Tunneling

Types of tunnels:

- IP over IP (RFC 2003): Basic encapsulation.
- Generic Routing Encapsulation, GRE (RFC 1701): There is an additional GRE header: allows encapsulating other protocols (not only IP).
- Point-to-Point Tunneling Protocol, PPTP (RFC 2637): Add the ppp functionalities.
- IPsec (RFC 2401): Standards to introduce authentication and encryption and tunneling to IP layer.