Gestió de Sistemes i Xarxes

Configuració Bàsica de Xarxa

Dr Josep M. Banús

Dept. D'Enginyeria Informàtica i Matemàtiques

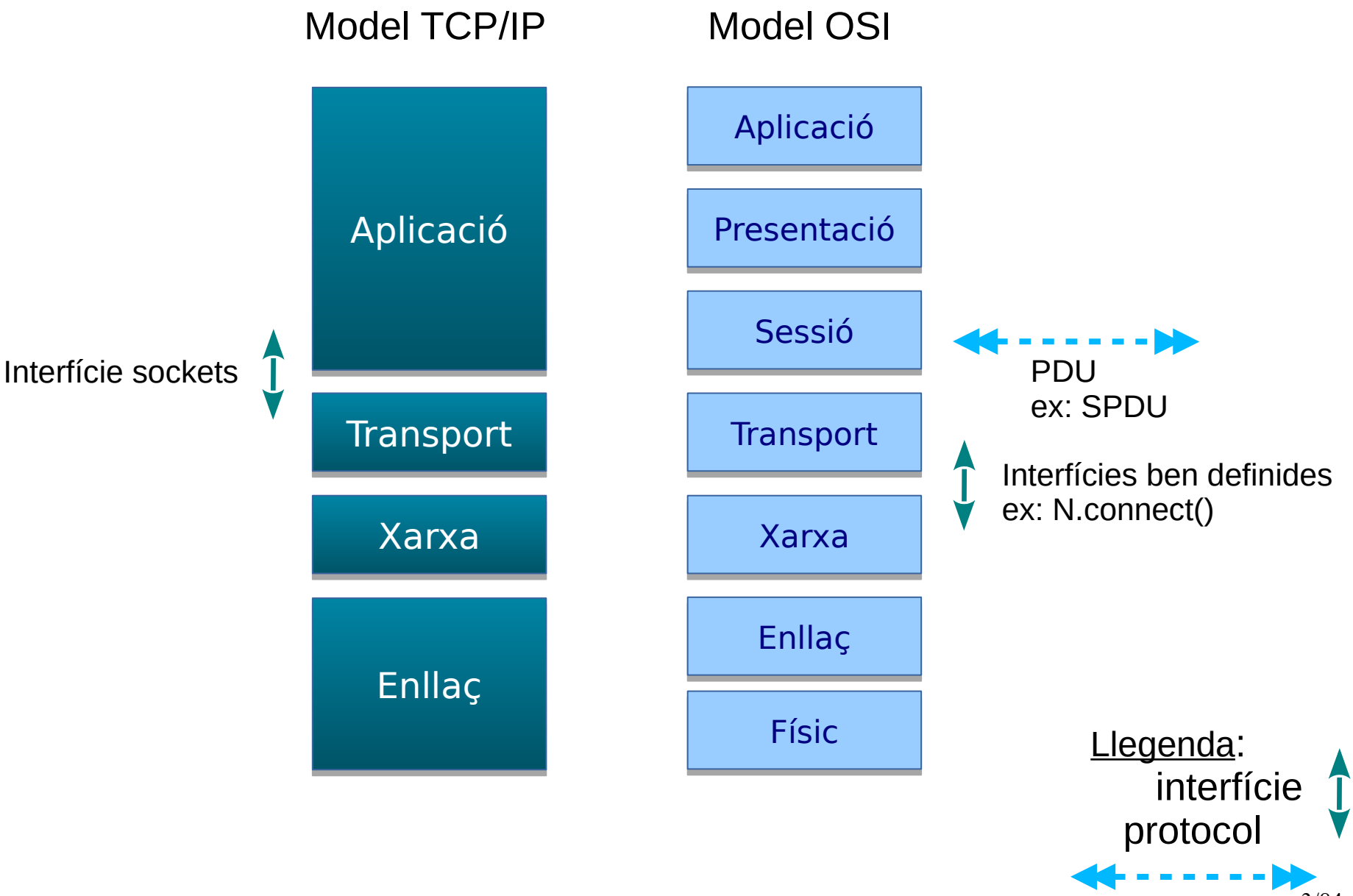
Universitat Rovira i Virgili

Contingut

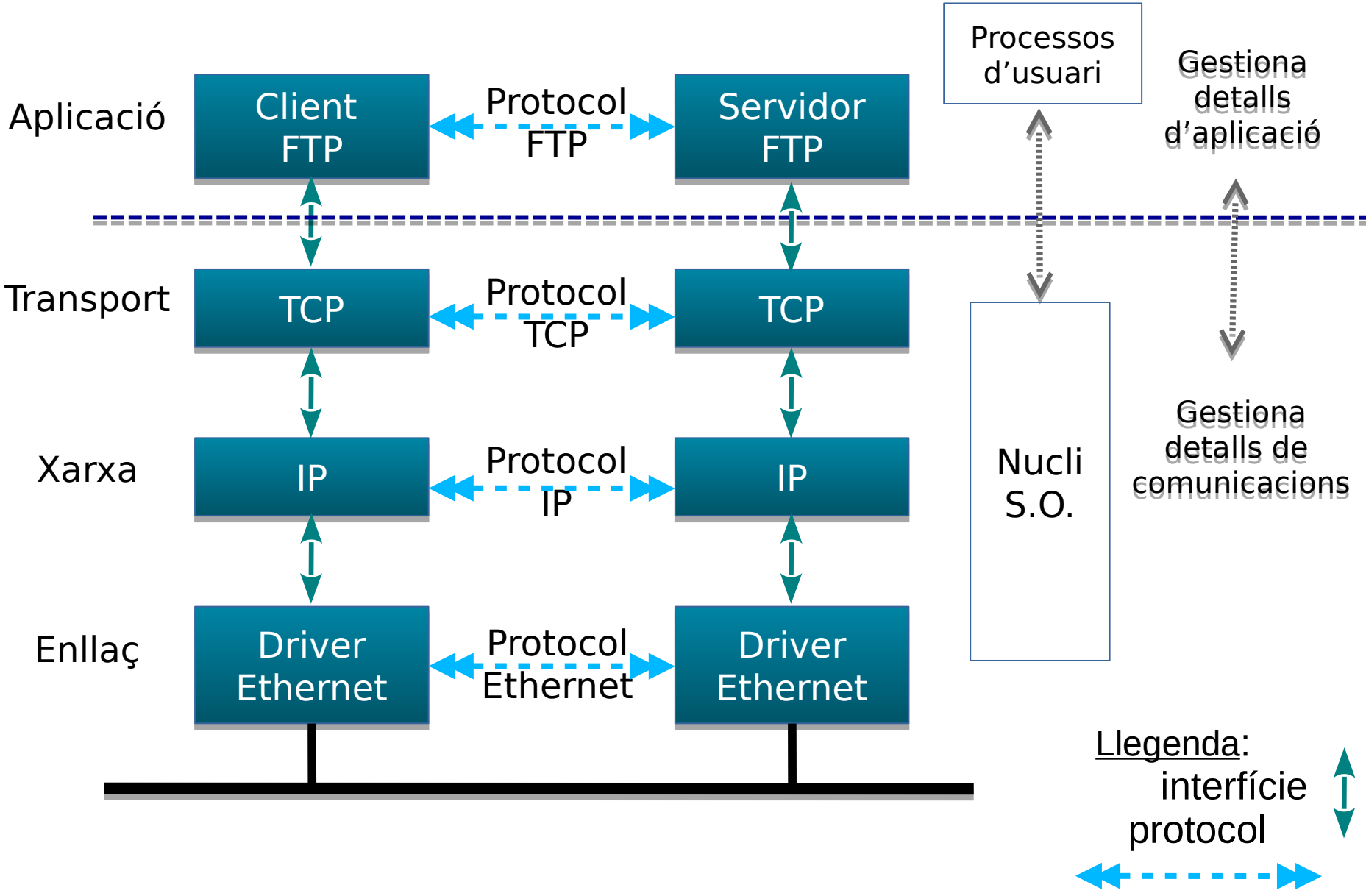
- Repàs
  - conceptes ✓
  - dispositius ✓
  - comandes bàsiques ✓
  - adreçament IP ✓
- Resum de fitxers importants
- Formes de configurar les interfícies
  - Dinàmica
  - Estàtica (manual) ✓
  - Amb gestors
  - Resum
- Referències Extres
- Annexes

efectives ~ 50

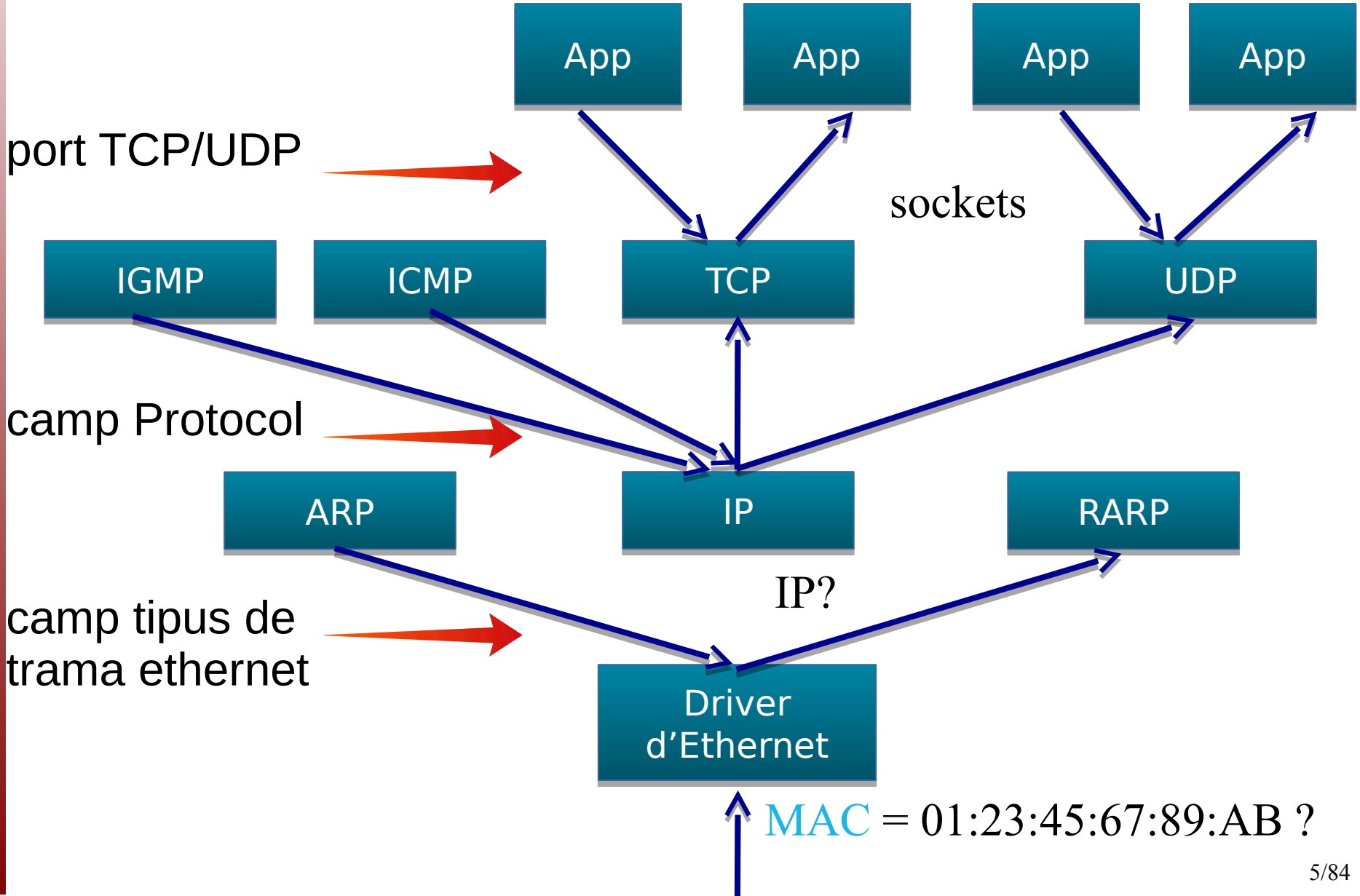
Repàs: Arquitectures



Repàs: Capes del TCP/IP

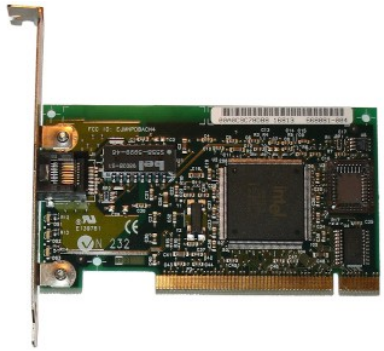


Repàs: demultiplexació



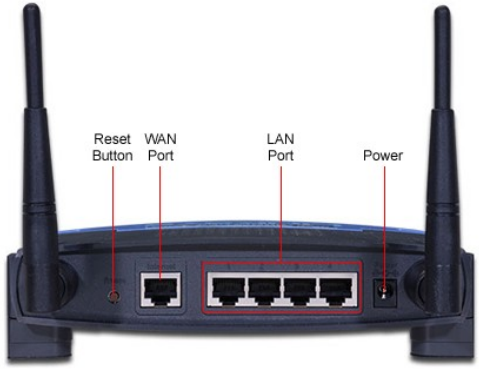
Repàs: dispositius bàsics

- NIC: network interface card
  - tipus: ethernet, token ring, etc
    → accés al medi aleatori o per torns
  - quantes per màquina ?
  - físiques o virtuals ?



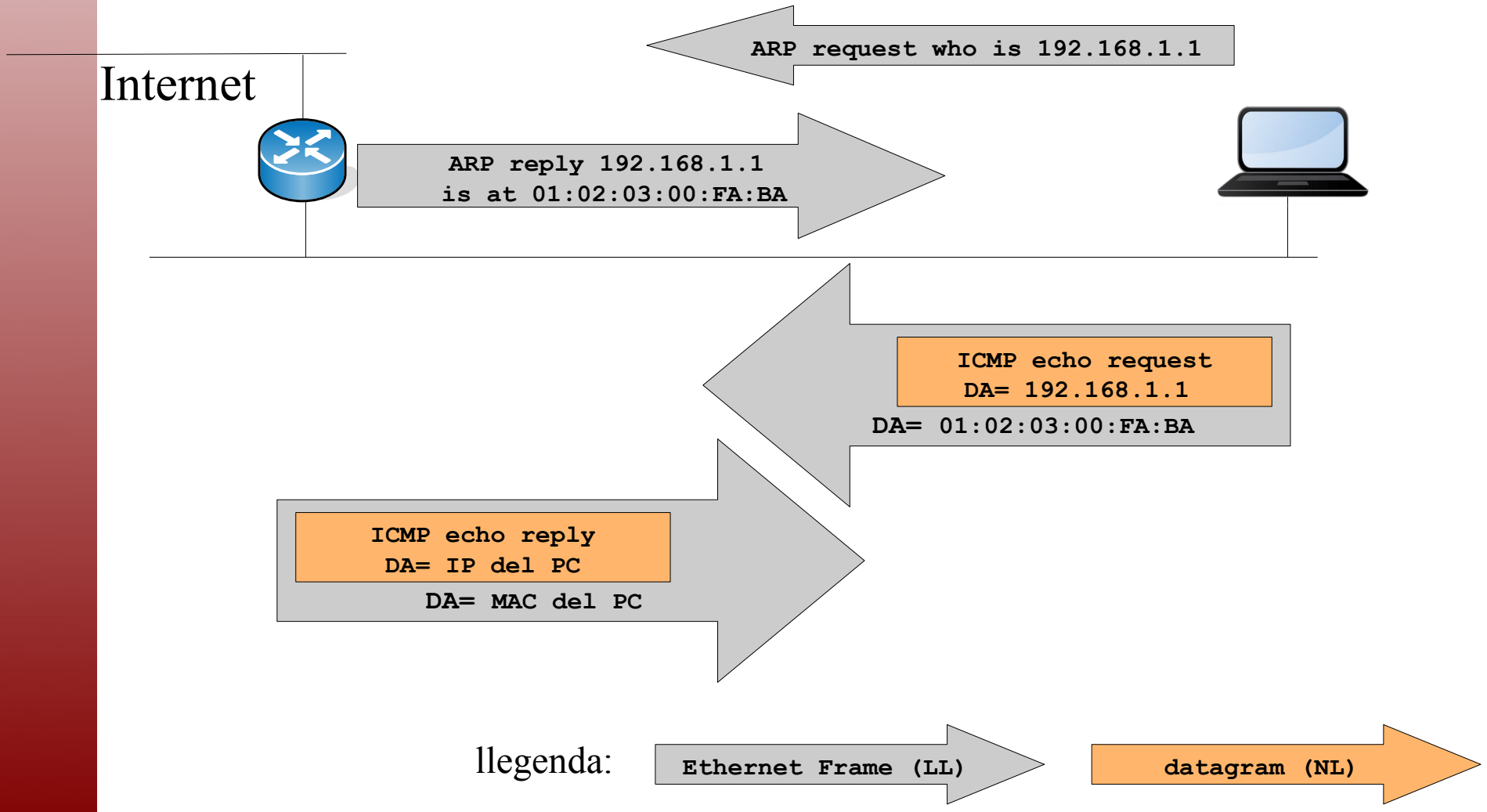
- Commutadors / switches:
  - ✓ forwarding de frames ethernet
    broadcast ?
  - ✓ separar tràfic / reduir col·lisions
  - vs hubs: repetidors
  - vs bridges: extensors de LAN

- Routers:
  - diverses NICs:
    - ✓ fan el routing i el forward de datagrames



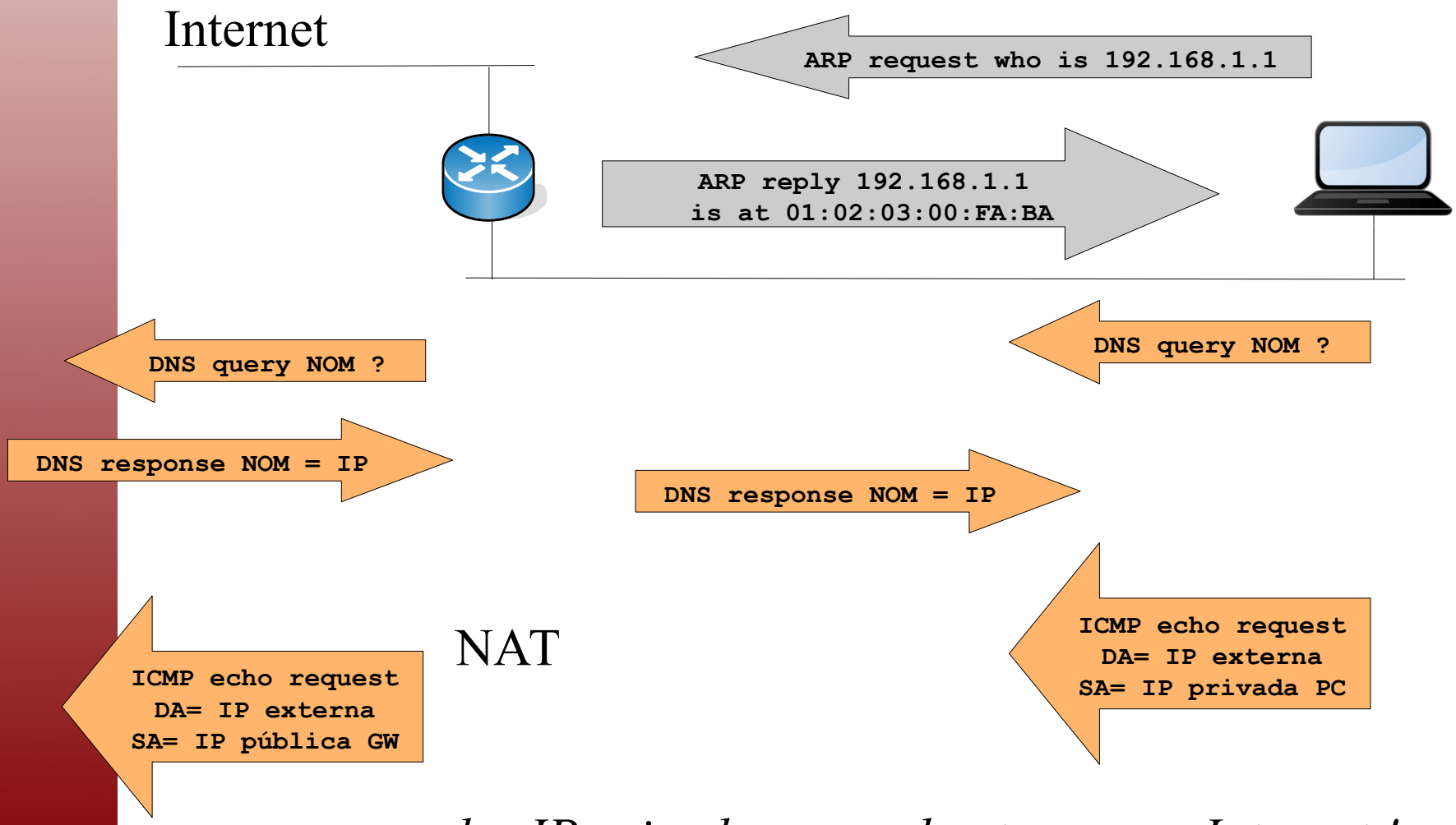
Exemple Repàs 1

- Un PC fa ping a la IP del *default gateway* (o un PC local)
  - → *sap la IP i te la connexió directa*
  - → *ha d'encapsular el datagrama IP dins una Frame Ethernet*



Exemple Repàs 2

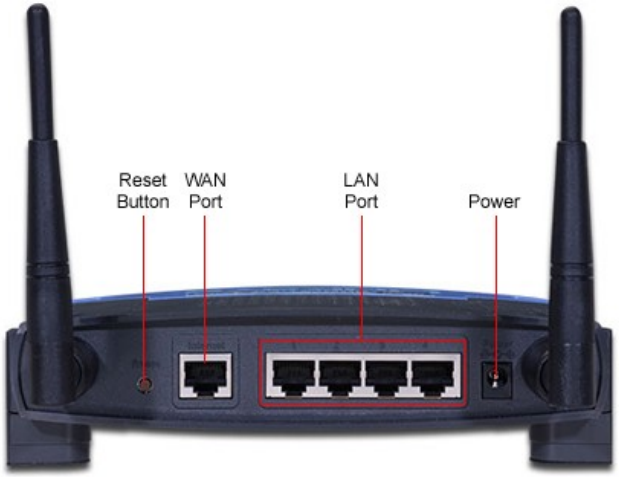
- Un PC fa ping a NOM d'una **màquina remota**
  → ha de passar per un router, doncs no té connexió directa amb el destí però sí amb el *default gateway*



les IP privades no poden travessar Internet !

Repàs: dispositius bàsics

- És un router aquest dispositiu ?



- ✓ commutador (capa 2, Link Layer)
- ✓ Access Point WiFi (capa 2, Link Layer)
- ✓ encamminador: (capa 3, Network Layer)

  LAN i WiFi ↔ WAN (Internet)
- ✓ SNAT (i possiblement DNAT)
- ✓ tallafocs (filtrat de datagrames) (capa 3 i 4)
- ✓ servidor DHCP intern (capa 7, Application Layer)
- ✓ servidor DNS intern opcional (capa 7, Application Layer)
- ✓ servidor web (per a la interfície de configuració)
- ✓ ...

Resum de comandes: xarxa

- Comandes per a veure l'estat de la xarxa i/o comprovar-lo:
  - ethtool
  - ip link
  - ip neigh (~ arp : MAC+IP)
  - ip address, ip route
  - **ping** (paquet iputils-ping, hping3)
    - ✓ si no tornen: potser estan filtrats, o falla la ruta o el destí
  - **traceroute / tracepath**
    - ✓ pels core routers useu una web amb *lookinglass*
  - **tcpdump / wireshark**
  - **nslookup** o dig o host
    - ✓ consultes al DNS triat localment o a un d'extern explícit
  - **nmap:** escanejar ports per a veure els serveis oberts
  - **ssh/scp** / telnet /netcat (nc)
- *cal mirar el manual i provar amb els paràmetres (ex: -4 o -6)*

FL

LL

NL

TL

AL

Resum de comandes: serveis

- Amb **System V**: més antic
  - sudo **service** $servei { start | stop | status | reload ... }

    service networking **start**
  - És important monitoritzar els missatges als logs:

    tail /var/log/syslog
- Amb **systemd**: més modern i versàtil
  - sudo **systemctl** { start | stop | **status** | reload ... } $unit

    systemd **enable** sshd # o disable

    systemctl **start** sshd
  - Monitoritza millor els logs amb **journalctl**:

    systemctl list-units | grep $servei

    journalctl --unit=sshd

    journalctl -xe --unit=networking # explicacions i ves al final

    jornalctl --follow # ~ tail -f /var/log/syslog

Contingut

- Repàs
  - conceptes
  - dispositius
  - comandes bàsiques
  - adreçament IP
- Resum de fitxers importants
- Formes de configurar les interfícies
  - Dinàmica
  - Estàtica (manual)
  - Amb gestors
  - Resum
- Referències Extres
- Annexes

Esquema d'adreçament IP

- Les màquines connectades en xarxa necessiten almenys una adreça IP.
  - en poden tenir més d'una: loopback, eth1, wifi0, virtuals, ...
- A cada empresa o infraestructura de xarxa convé dissenyar un esquema d'adreçament:
  - ✓ Quines IPs usarem ?
  - ✓ Seran públiques o privades ?
  - ✓ Usarem IPv4 o IPv6 (o híbrid)
- Per a tenir diverses xarxes i separar el trànsit podem
  - Usar IPs diferents a cada subxarxa (ex: 10... i 192...)
  - Usar una IP i subdividir-la en diferents subxarxes (*subnetting*)
- *Cal dissenyar un esquema d'adreçament adient a l'empresa segons els requisits i les previsions*

Adreça IPv4

- A IPv4 les adreces de nivell de xarxa són de 32 bits.
- Es representen en 4 grups de 8 bits en notació decimal.

```
00001010.00100000.10000001.01100110  (binari)
10.32.129.102 (decimal)
```

- Cada estació connectada a Internet ha de disposar d'una adreça IP diferent (**única**, tret de NATs) o no es podria saber com arribar-hi.
- Cada paquet en porta dues: destí (DA) i origen (SA)
- El camí/ruta cap a tots els dispositius connectats a una xarxa és el mateix:
  - Es poden **agrupar** les seves adreces ?

Adreça IP: parts

- Els bits d'una adreça IP es divideixen en dues parts:
  - ✓ bits d'**identificació de xarxa**
  - ✓ bits d'**identificació de dispositiu** (host)
- Aquesta divisió es fa amb la màscara: 11.....000
  - ✓ els bits amb valor 1 indiquen la part de xarxa.
- Aquesta distinció en parts ens permet **agrupar** tots els dispositius d'una xarxa
- En el encaminament a una IP destí de host s'ha de **distingir** a quina xarxa destí pertany:
  - ✓ aplicant una **AND bit-a-bit** amb una màscara
- Aquest mètode permet reduir la mida de les taules d'encaminament, augmentant el rendiment

Màscara IPv4: formats

- Exemple: màscara de xarxa en diferents formats

```
11111111.11111111.11111111.00000000    binary format
   ≡ 255.255.255.0                   dotted decimal format
   ≡ /24                               mask bits format
   bitwise_not(mask) ≡ 0.0.0.255       wildcard format
```

→ *sempre son molts 1 seguits i després tot 0*

- **prefix length**: nombre d'uns (24 a l'exemple)
- Contra-exemples:

```
11111111.11111111.11111111.00100000 ~ 255.255.255.32
11111111.01111111.11111111.00000000 ~ 255.127.255.0
```

IPv4: Classes i rangs

- Classful:

Classful

Tipus	Classe	Rangs	Màscara de bloc	Número de Blocs	Adreces Per bloc
Unicast	A 0.0.0.0/1	0.0.0.0 – 127.255.255.255	255.0.0.0 ≡ /8	128	16.777.216
	B 128.0.0.0/2	128.0.0.0 – 191.255.255.255	255.255.0.0 ≡ /16	16.384	65.536
	C 192.0.0.0/3	192.0.0.0 – 223.255.255.255	255.255.255.0 ≡ /24	$2^{24-3} = 2^{21}$ 2.097.152	256
Multicast	D 224.0.0.0/4	224.0.0.0 – 239.255.255.255	-	-	-
Reservada	E 240.0.0.0/5	240.0.0.0 – 255.255.255.255	-	-	-
Qualsevol	0.0.0.0/0	0.0.0.0 – 255.255.255.255	0.0.0.0 ≡ 0	1	4.294.967.296

- Tots els blocs ja estan assignats !

Exemple de classe C

- La màscara 255.255.255.0 indica que dels 32 bits de l'adreça IP, els primers 24 són de xarxa i els 8 darrers són de host.
  - Adreça: 195.133.165.12 → pertany a una **tarja** d'un host (el 12è)
  - Màscara: 255.255.255.0
  - and bit a bit: 195.133.165.**0** → **adreça de la xarxa** (no assignada a cap tarja)
  - Adreça de **broadcast remot**: 195.133.165.**255**
  - Assignables a hosts/interfícies: 192.133.165.1 ... 195.133.165.254

  → és el host 12 de la xarxa 195.133.165.0
- A les configuracions 195.133.165.12/24 és incorrecte !!

→ la part de host ha d'estar a zero

IP de dispositiu: 195.133.165.12/**32**
IP de xarxa: 195.133.165.0/24

Classless IP: sense classes

- La distribució en classes malgasta moltes IPs
  - Ex: una classe A assignada a una empresa pot malgastar de l'ordre de 16 milions d' IPs !
- Fan falta moltes adreces IPv4

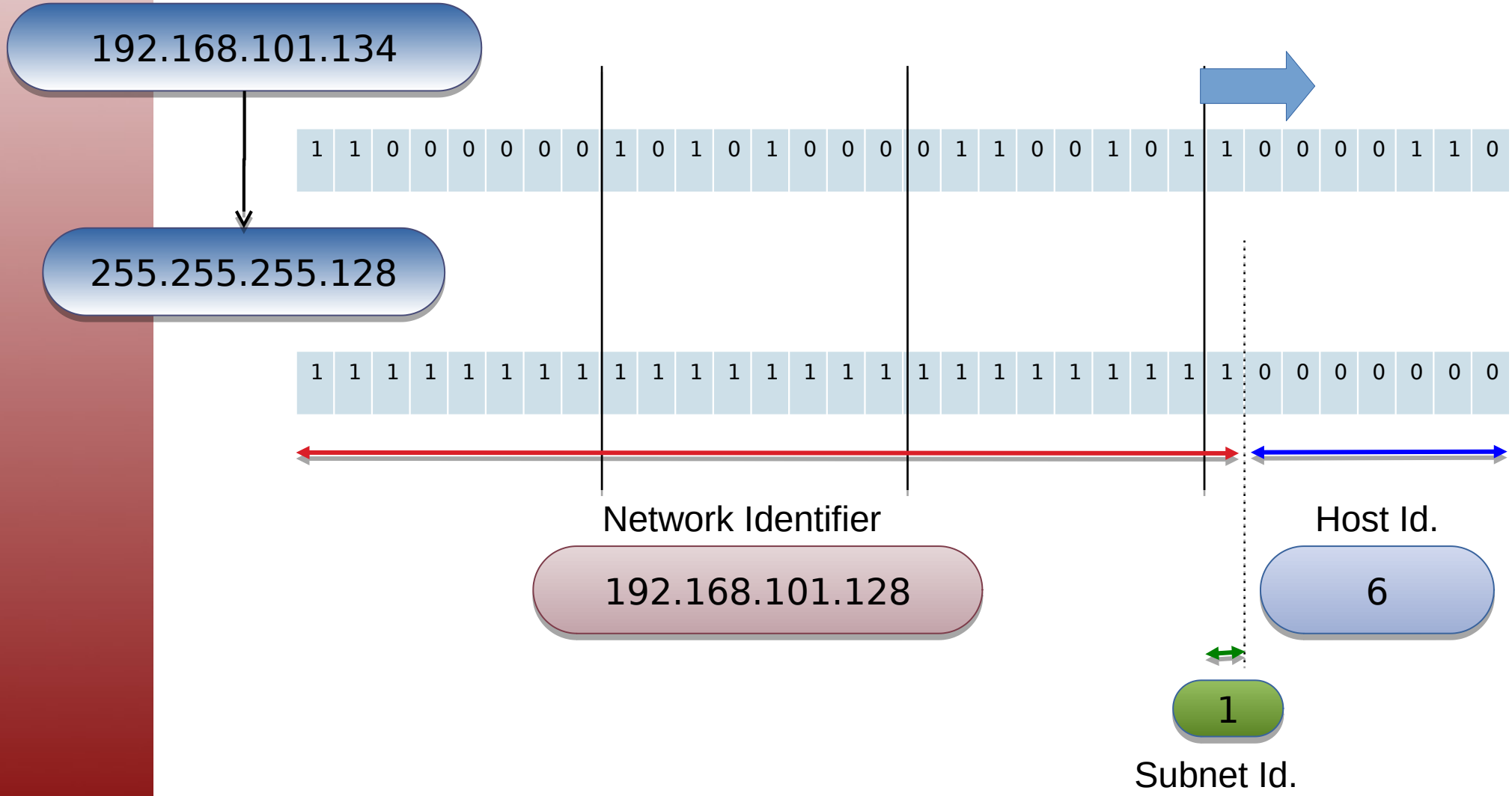
  → Solucions: Classless IPs, IP privades + NAT, IPv6
- **CIDR**: *Classless Inter-Domain Routing*
  - No usa les màscares tradicionals /8 /16 i /24
  - Pot usar tot l'espai d'adreces unicast: 0.0.0.0 a 223.255.255.255
- **FLSM**: *Fixed Length Subnet Mask*
  - Tota la empresa usa la mateixa *classless mask*
- **VLSM**: *Variable Length Subnet Mask*
  - Cada subxarxa usa la seva pròpia màscara (de longitud diferent)

Classless IP

- Les noves màscares ens permeten fer:
  - *subnetting*:
    - ✓ Incrementem el nombre de 1's
    - ✓ Permet diferenciar en diverses subxarxes locals
      → permet separar el trànsit entre elles
  - *supernetting*:
    - ✓ Decrementem el nombre de 1's
    - ✓ Permet tenir una xarxa més gran, amb més hosts
  - *aggregation*:
    - ✓ Decrementem el nombre de 1's
    - ✓ Permet tenir reduir la informació de routing
    - ✓ S'aplica al routers a la hora de anunciar les rutes.

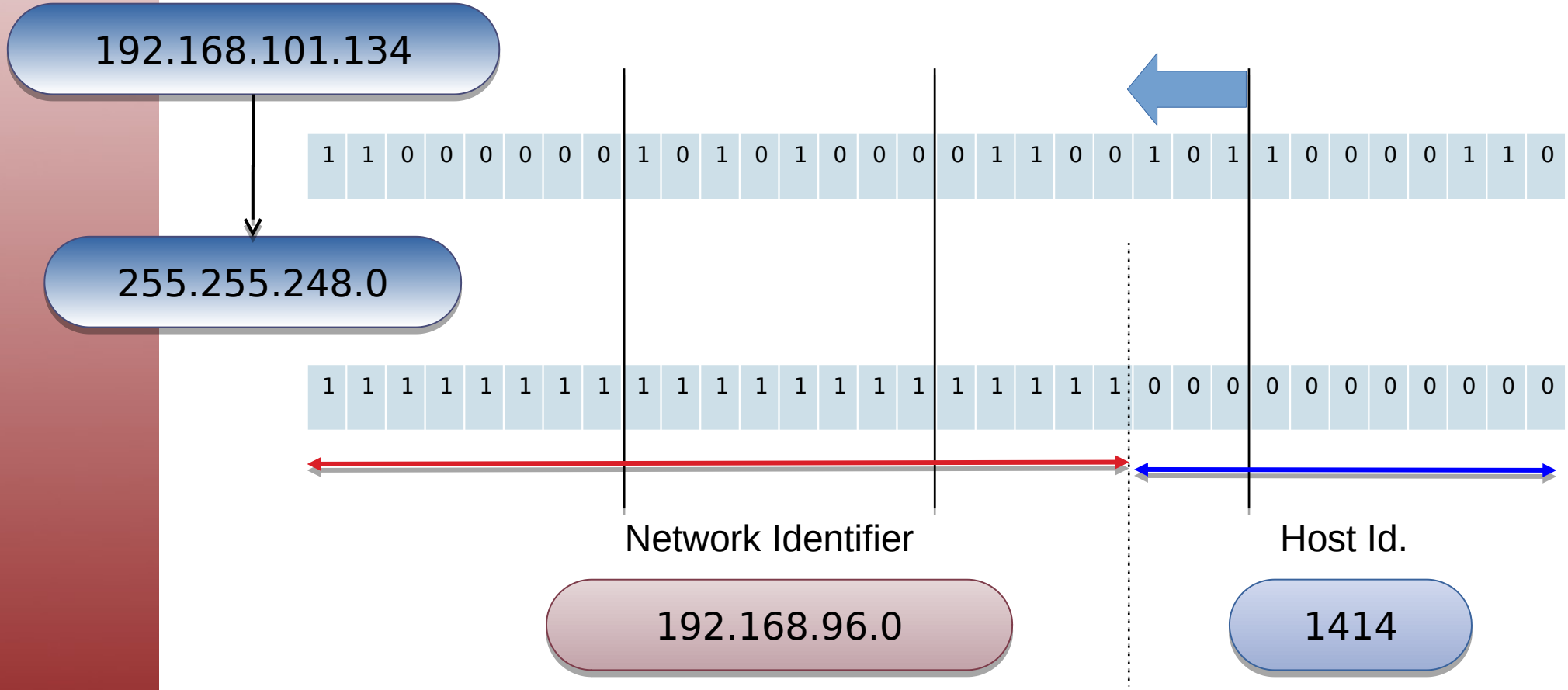
Exemple: subnetting

- Movem la màscara cap a la dreta (més uns):



Exemple: supernetting

- Movem la màscara cap a l'esquerra (menys uns):



- xarxes més grans
- route aggregation

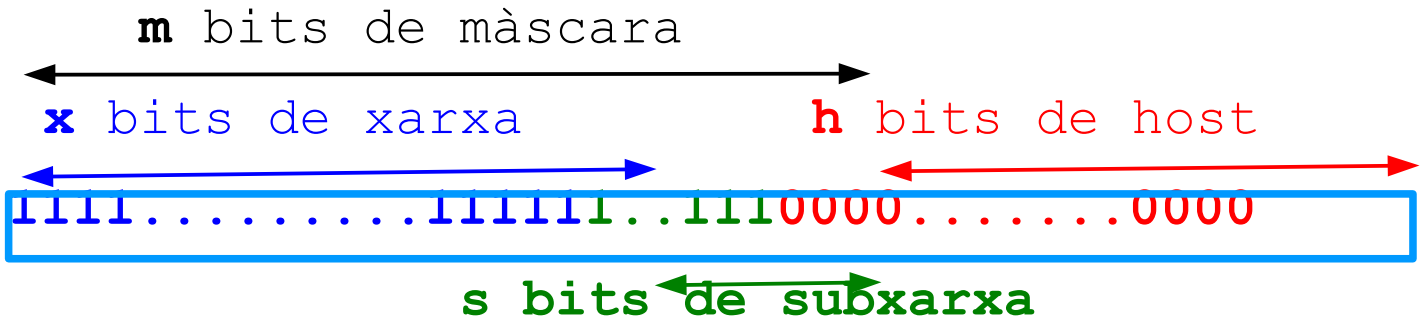
Problemes: cas 1

- Fórmules per a trobar les màscares /m:

Requisit: **necessitem nsubx subxarxes**
→ quants bits/subxarxa (s) hem d'usar?

$$s = \lceil \log_2(nsx) \rceil$$
$$si \;\; s < \; 32 - x - 1:$$
$$m = x + s$$

on:



x+s+h=32 → s ∈[0..32-x-1) , h ∈[2..32-x]

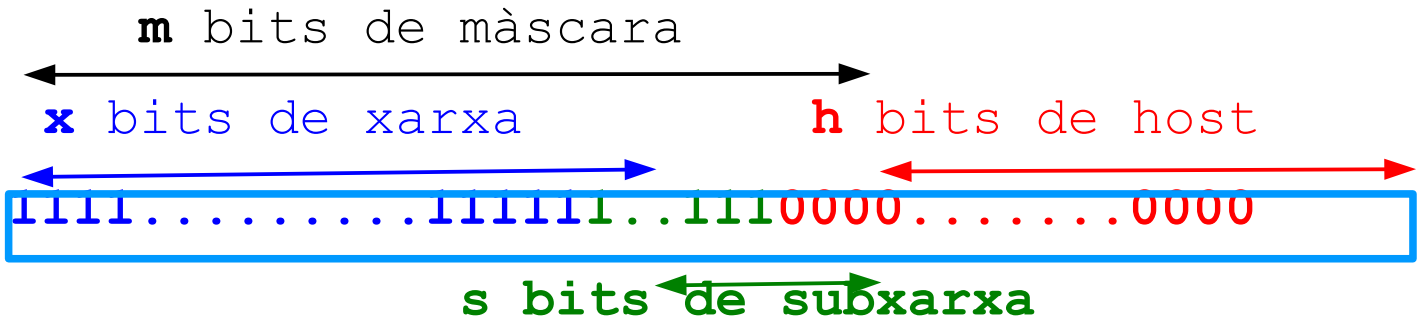
Problemes: cas 2

- Fórmules per a trobar les màscares /m:

Requisit: **necessitem *nhosts* a cada subxarxa**
→ quants *bits pels host* (h) hem d'usar ?

$$h = \lceil \log_2(nhosts+2) \rceil$$
$$si\ \ h < 32-x:$$
$$m = 32-h$$

on:



x+s+h=32 → s ∈ [0..32-x-1) , h ∈ [2..32-x]

Problema 1: híbrid

- Tenim 200 hosts distribuïts equitativament en 4 subxarxes i un router. Disposem de la IP pública de Classe C 194.224.210.0/24

4 xarxes → fan falta **2 bits** per identificar una xarxa → **/26**
tindrem **6 bits** per identificar 2^6-2 = 62 hosts > 51 hosts

X0: 11000010. 11100000. 11010010.**00hhhhhh** → 193.144.210.0/26

X1: 11000010. 11100000. 11010010.**01hhhhhh** → 193.144.210.64/26

X2: 11000010. 11100000. 11010010.**10hhhhhh** → 193.144.210.128/26

X3: 11000010. 11100000. 11010010.**11hhhhhh** → 193.144.210.192/26

Al router que les interconnecta se li poden assignar les següents IPs:

193.144.210.1 193.144.210.65

193.144.210.129 193.144.210.193

Problema 2: IP de xarxa o host

- Donada una IP i una màscara,
  la IP és de xarxa o de host?

  - Apliquem la màscara negada (wildcard)
    - ✓ si el resultat és 0 → és una IP de xarxa
    - ✓ sinó → és una IP de host

- Exemple:

La IP 192.168.1.124/28 és de host o de xarxa ?

- **124= 127 -3 → 0111 1111 -11 = 0111 1100**
- **wildcard= 0.0.0.15 → & 0000 1111**

0000 1100 → host

Problema 3: trobar la mask 1

- Donada una **IP de xarxa**, troba la màscara:
  - hem de buscar el primer 1 des de la dreta
- Exemple:

Troba la màscara per a la IP de xarxa 192.168.1.124

124= 127 -3 → 0111 1111 -11 = 0111 1100

com la id. de host ha de ser 0 llavors

la màscara maximitzant el nombre de hosts és

/30 ~ 255.255.255.252

Problema 4: trobar la mask 2

- Donada una **IP de host**, troba la màscara:
  - hem de buscar el primer 1 des de la dreta
    Com la id. de host ha de ser !=0 llavors
    → tria una màscara que inclogui (almenys) aquest 1
- Exemple:

Troba la màscara per a la IP de host 192.168.1.124

- 124= 127 -3 → 0111 1111 -11 = 0111 1100
- la màscara maximitzant el nombre de subxarxes és /29

Problemes IP/mask usant eines

- Podeu trobar moltes *cheatsheets* o eines
- N'hi ha moltes, per posar alguns exemples:
  - Linux: **ipcalc** i sipcalc
  - Solarwinds advanced subnet calculator
    - ✓ Web: www.calculator.net/ip-subnet-calculator.html
    - → També podeu trobar el codi (java o python)
- però com a estudiants primer convé
  - que ho sapigueu fer a mà i
  - que ho entengueu bé.

Adreces IPv4 privades

Important

- No totes les adreces són vàlides a Internet:
  els datagrames que les usen no passen del router perimètric
- Adreces privades:
  - 10.0.0.0-10.255.255.255 prefix: 10/8, una class A
  - 172.16.0.0-172.31.255.255 prefix: 172.16/12, 16 class B
  - 192.168.0.0-192.168.255.255 prefix: 192.168/16, 255 C
    - ✓ dispositius que no seran accessibles des d'Internet
    - ✓ amb NAT es canvien IP privades ↔ IPs públiques
- Usant adreçament privat podem prescindir del subnetting i del supernetting
- Ara bé, per a poder sortir cap a Internet necessitarem fer SNAT
- Els serveis proporcionats cap a fora hauran d'usar IPs públiques o bé haurem de fer DNAT (destination NAT)

Adreces Local-Link

- **Local-Link addressing** (IP4LL):
  - Usat per a la comunicació de dispositius locals
    - ✓ els datagrames no passaran cap router
  - Rang: 169.254.1.0 – 169.254.254.255 prefix: 169.254/16
  - No cal ni clients ni servidors DHCP (ex: IoT)
    - triar-ne una qualsevol (random)
    - comprovar amb ARP que cap altre veí l'usi
  - Si tenim un client DHCP però no troba el servidor temporalment es pot assignar una IP LL
  - també se'n diu APIPA (Automatic Private IP Addressing)

més IPv4 amb usos especials de la IANA

Introducció a les adreces IPv6

- Longitud de 128 bits:
  - 340 trilions de trilions d'adreces (uns 1000 milions amb IPv4)
  - representades en 32 dígits hexadecimals agrupades en 8 grups (words, quads, **hextets**) de 16 bits separades per ':'
- Classless
  - però els primers bits distingeixen el tipus o ***scope***
  - ja preveu el *subnetting* 'de sèrie'
- Exemples:
  - 2001:db8::1
  - 2001:db8:85a3::37:1 = 2001:0db8:85a3:0000:0000:0000:0037:1
  - fe80::7E89:F4ff:fe19:A95B
  - FF02::1 multicast: all local nodes i FF02::2 all local routers

    No hi ha ARP sinó NDP

    ✓ Network Discovery Protocol (RFC4861)

Prefix IPv6

- Ús: Adreça IPv6/prefix
  - el prefix típic és /64 (rang del prefix: [0 .. 128])
- Principals prefixes segons l'àmbit (*scope*):
  - **2000::/3** Provider-Based **Global Unicast** ≈ **públiques**
    - ✓ **010** +45b ISP + 16b subnet + 64b interface ID
  - **FC00::/7**..FD::/8 **Unique Local** ≈ **privades**, routing intern
  - **FE80::/10** **Link-Local** Unicast → no encaminades
  - **FF::/8** Multicast (FFtx::/12 on t:temporal, x:scope)

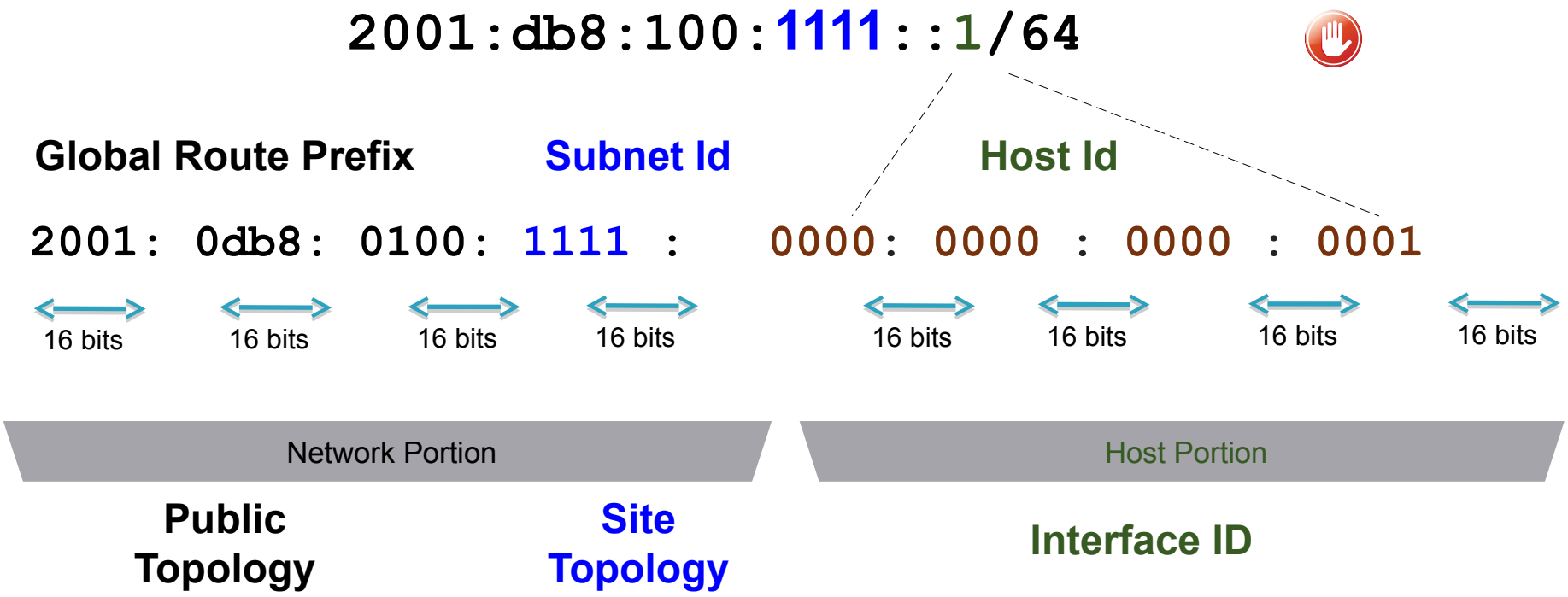
  → no existeixen els broadcast !

  ex: FF02::1:2 tots els DHCP servers → no encaminades

  ex: FF05::1:3 ídem però *site-local scope*
- Una interfície tindrà al menys una LL i opcionalment IP unicast, anycast i multicast per als altres àmbits

Adreces IPv6: parts

- Prefix: Part de Network o topologia
  → Inclou la part de “subnet” amb 16 bits



Adreces IPv6

- Casos especials:
  - interfície loopback ::1/128 (o simplement **::1**)
  - ruta per defecte **::/0**
  - ruta no especificada ::/128 (o simplement ::)
- link-local **FE80::/10** (*not routed)*
  - manual

    exemple: FE80::7
  - EUI-64 (usant la MAC address)

    ex: MAC=7C:89:F4:19:A9:5B

    → fe80::7e89:f4ff:fe19:a95b
  - random: cal comprovar que cap altre dispositiu la usi
- IPv4 mapped **::FFFF/96** pel software i a fitxers de config.

  exemple ::ffff:192.1.2.128/128

IANA IPv6 address space

Contingut

- Repàs
  - conceptes
  - dispositius
  - comandes bàsiques
  - adreçament IP
- Resum de fitxers importants
- Formes de configurar les interfícies
  - Dinàmica
  - Estàtica (manual)
  - Amb gestors

Fitxers de configuració

- En la configuració de serveis, normalment a les transparències apareixerà la ubicació principal dels fitxers, però cal que sabeu les diferents possibilitats que hi ha.
- Exemples genèrics:
  - /etc/nomservei/ (sobre)escrits a l'instal·lar el paquet
  - /etc/nomservei/**conf.d**/ els fitxers de lloc, personalitzats
  - /etc/**default**/nomservei per a establir algunes vars globals
  - **/usr/lib**/nomservei/ path alternatiu per als paquets
  - **/var/lib/**nomservei/ en execució hi posa dades (estable)
  - /var/lib/**cache**/nomservei/ cache i configuracions manuals
  - **/run/**nomservei/conf.d en execució (volàtil)
- Si als diferents dirs hi ha el mateix fitxer cal saber **quin preval**
- Els fitxers d'un dir es llegeixen en ordre i els darrers poden reconfigurar coses dels primers (es sol posar noms 10-, 20-..)

Resum de fitxers importants

- /etc/hosts
  - ✓ indica a cada IP el seu nom (sense el servei DNS)
  - ✓ és estàtic però còmode en alguns casos
- **/etc/resolv.conf**: molt important i vital
  - ✓ llista de IP del servidors de noms a consultar
  - ✓ el domini DNS, etc
- /etc/nsswitch.conf: Name Service Switch (NSS)
  - ✓ ordre de preferència d'on buscar la informació dels noms
- /etc/hostname: nom de la màquina al boot
- /etc/protocols: nom i número (ip=0, tcp=6, udp=17)
- /etc/services: llista de noms i ports dels serveis (sockets)

```
$ getent services http
http                    80/tcp www
```

Resum de fitxers importants

- Exemple de **/etc/hosts**

C:\Windows\System32\drivers\etc\hosts

```
# /etc/hosts
127.0.0.1        localhost
192.168.1.1      router      sortida

# The following lines are desirable for IPv6 capable hosts
::1              ip6-localhost ip6-loopback
fe00::0          ip6-localnet
ff00::0          ip6-mcastprefix
ff02::1          ip6-allnodes
ff02::2          ip6-allrouters
```

- Exemples de **/etc/resolv.conf**

```
domain   labdeim.net
search   urv.cat urv.net urv.es
nameserver 10.40.1.2
```

```
nameserver 8.8.8.8   # dns.google
nameserver 1.1.1.1   # Cloudflare
```

- /etc/networks: similar al /etc/hosts

```
loopback     127.0.0.0
localnet     192.168.0.0    cameua
#link-local   169.254.0.0
```

Resum de fitxers importants

- /etc/**sysctl.conf**: paràmetres del *kernel*
- /etc/sysctl.d/10-network-security.conf

```
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks (reverse-path filter)
net.ipv4.conf.default.rp_filter=1
net.ipv4.conf.all.rp_filter=1
#net.ipv4.tcp_syncookies=1

net.ipv4.ip_forward=0
#net.ipv6.conf.all.forwarding=1

# Do not accept ICMP redirects (prevent MITM attacks)
net.ipv4.conf.all.accept_redirects = 0
net.ipv6.conf.all.accept_redirects = 0

# Do not send ICMP redirects (we are not a router)
net.ipv4.conf.all.send_redirects = 0

# Do not accept IP source route packets (we are not a router)
net.ipv4.conf.all.accept_source_route = 0
net.ipv6.conf.all.accept_source_route = 0

# Log Martian Packets
net.ipv4.conf.all.log_martians = 1

# IPv6
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

Contingut

- Repàs
  - conceptes
  - dispositius
  - comandes bàsiques
  - adreçament IP
- Resum de fitxers importants
- Formes de configurar les interfícies
  - Dinàmica
  - Estàtica (manual)
  - Amb gestors
  - Resum

Configuració de xarxa Unix

- El **mínim** que es necessita:
  - un client DHCP (dhclient) per a obtenir la configuració de xarxa de forma automàtica des d'un servidor DHCP
    - ✓ amb IPv6 això és opcional
  - O bé estàticament amb un scripts i/o comandes
    - ✓ dels paquets iproute2 o net-tools
- Requisits:
  1) una adreça IP per a les interfícies de xarxa (NICs)
  2) la IP de la passarel·la per defecte (**default gateway**)
  3) configurar com accedir a un servidor DNS per a poder traduir noms de servidors a les seves IPs

Configuració de xarxa Unix

- Exemple de configuració estàtica mínima:

  1) una adreça IP per a les interfícies de xarxa (NICs)

```
sudo ip address add 192.168.1.3 dev eth0
sudo ip link set dev eth0 up
```

  2) la IP de la passarel·la per defecte (**default gateway**)

```
sudo ip route add default 192.168.1.1 via eth0
```

  3) configurar com accedir a un servidor DNS per a poder traduir noms de servidors a les seves IPs

```
echo "nameserver 8.8.8.8" | sudo tee -a /etc/resolv.conf
```

- Taula d'encaminament del PC:

```
ip route show
default via 192.168.1.1 dev eth0 onlink
169.254.0.0/16 dev eth0 scope link metric 1000
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.3
```

Configuració de les NICs

- 1- Automàtica/dinàmica:
  - dhclient
  - zeroconf (LinkLocal address. mDNS, DNS-sd)
- 2- Estàtica:
  1) manual: amb comandes a la CLI i/o a un script:
     - ✓ ex: ifconfig + route + arp + netstat... o iproute (iw per a WiFi)
  2) a fitxers de configuració que llegirà un executable
     - ✓ ex: /etc/network/interfaces + ifup o ifdown
       - → ifup crida a dhclient o aplica la estàtica del fitxer
- 3- Gestors a alt nivell: (criden a dhclient, ifup, iproute,...)
  - són millors per a ordinadors personals / desktop / laptop
    - ✓ systemd.networkd
    - ✓ NetworkManager (NM): menys usat per a servers, però ...
    - ✓ wicd , connman, netctl, etc

1- DHCP clients

- Obtenir la configuració de d'un servidor DHCP local
  - ✓ IP, net mask, IP del router, nom, domini, IP del servidor de noms, etc
- Hi ha alternatives:
  - ✓ dhcpcd: daemon, sempre executant-se
  - ✓ dhclient: en aixecar una interfície, queda en *background*
- No cal configurar-li res, simplement funciona
  - ✓ però se li poden configurar molts paràmetres.
- Els altres paquets els solen cridar (ifupdown, NM, etc)
- Exemples:

```
sudo dhclient -4 -1 eth0
sudo dhclient -r eth0
```

→ UP,IPv4, un intent

→ release (DOWN)

2-1 Configuració Manual

- A baix nivell, des de la CLI o amb scripts:
  - paquet **net-tools**
    - ✓ el més clàssic, quasi obsolet
  - paquet **iproute2**
    - ✓ més actual, més possibilitats (pot semblar més complexe)
    - ✓ distingeix més entre nivells (Link Layer, Network Layer)

net-tools	iproute2	ús
ifconfig	ip link + ip address	canviar IP, MAC, ...
route	ip route	veure o afegir rutes
netstat	ss	veure estat o estadístiques
arp	ip neigh	veure les MACs de la cache

- Les *wireless* van a banda:
  - iwconfig (paquet wireless-tools, més vell)
  - iw (més modern)

2-1 Configuració amb iproute2

- Comanda **ip**: algunes objectes que pot tractar:
  - **link**: enllaços de diversos tipus
  - **address**
  - **route**
  - **neigh**: taula d'adreces MAC
  - tuntap: crear una interfície túnel (tun és L3 mentre que tap és L2)
  - tunnel: mode { ipip | gre | sit | ipip6 | vti ... etc }
  - **rule**: regles per a triar una taula d'encaminament o una altra
  - netns: network namespace
- Altres comandes interessants del paquet iproute2:
  - per a configurar commutadors virtuals (amb moltes interfícies):
    - ✓ **bridge** --json -d link [show dev brname] **-d : details**
    - ✓ bridge fdb show → veure les MACs apreses pel pont
      → alternativa al paquet **bridge-utils** (comanda brctl)
  - tc: traffic control (FYI: classificar, filtrar, disciplines de cues)

2-1 Configuració amb iproute2

- Tipus d'enllaços:
  - bridge: commutador virtual on connectar-hi altres interfícies:

    ex: ip link set dev enp0s3 master br0
  - veth:
    - ✓ ethernets virtuals per a connectar-les a un bridge o a una altra veth (peer)
  - dummy: similar a loopback (per posar IP de 'gestió')
  - bond: per a combinar 2 o més NICs com si fossin una
  - vlan
    - ✓ una mateixa NIC pot portar trànsit de diferents VLANs
    - ✓ el commutador (real) separa les trames de cada VLAN
    - ✓ ex: enp0s3.10 → les trames ethernet porten el TAG=10
  - macvlan
    - ✓ semblant a l'anterior, però a més separa per MAC (aïlla més)
  - etc

2-1 Manual amb comandes

- Mostrar informació de les NIC:

```
ip link [show ifname]
ip address [show ifname]
```

- Exemples: nivell d'enllaç (L2) i nivell xarxa (L3)

```
$ ip -4 link show eth0
3: eth0: <BROADCAST,MULTICAST,PROMISC,UP> mtu 1400 qdisc pfifo_
link/ether 48:54:e8:2a:47:16 brd ff:ff:ff:ff:ff:ff

$ ip -4 address show eth0
3: eth0: <BROADCAST,MULTICAST,PROMISC,UP> mtu 1400 qdisc pfifo_
link/ether 48:54:e8:2a:47:16 brd ff:ff:ff:ff:ff:ff
inet 10.0.0.1/8 brd 10.255.255.255 scope global eth0
```

- Configurar l'adreça IPv4 a una NIC:

```
sudo ip address add 192.168.100.100/24 broadcast \
 192.168.100.255 dev eth0

sudo ip link set eth0 up
```

2-1 Manual amb comandes

- Mostrar informació la/les taula/es d'encaminament:

```
ip route [show IP/mask]
ip route list table main
```

- Consultar com es va a un destí:

```
ip route get 8.8.8.8
```

- Configurar la taula d'encaminament:

primer cal saber com es va al 172.20.0.1 !

```
ip route add 195.96.96.0/24 via 10.0.0.1

ip route add default via 172.20.0.1 dev eth2

ip route add blackhole $address/$mask  #descartar paquets
```

- Adreces a la taula de MACs (apreses amb ARP):

```
ip neigh [show $IP]
```

  - o estàtiques afegides a mà:

```
ip neigh add 192.168.1.1 lladdr 00:0c:0c:c0:cc:01 dev eth0
```

2-1 Estadístiques amb iproute2

- A la major part de comandes: opció **-s**
- Convé comprovar l'estat dels sockets amb **ss** (~ netstat)

→ eina per a investigar els sockets

ss --all massa info (tots els ~ 17 tipus de sockets)

ss [-4|-6] --tcp connexions TCP **establertes**

ss --listen [--tcp | --udp] ports que estem **escoltant**

- També es pot monitoritzar en viu conforme s'estableixen:

  ss -E
- Per a saber les estadístiques dels protocols TCP/IP:

  ss --summary

  nstat Tcp*

  nstat --json

2-1 Estadístiques iproute2

FYI

- Estadístiques de xarxa (locals al OS): ss -f inet -s

```
Total: 618
TCP:   15 (estab 6, closed 2, orphaned 0, timewait 1)

Transport Total     IP        IPv6
RAW       0         0         0
UDP       3         2         1
TCP       13        9         4
INET      16        11        5
FRAG      0         0         0

Netid  State  Recv-Q  Send-Q  Local Address:Port   Peer Address:Port
tcp    ESTAB  0       0       192.168.1.132:60356  40.77.226.250:https
...
```

- Estadístiques detallades TCP/IP (del OS): nstat -s

```
$ nstat Icmp*
IcmpInMsgs                      153                0.0
IcmpInErrors                    8                  0.0
IcmpInDestUnreachs              152                0.0
IcmpInEchoReps                  1                  0.0
...
```

2-1 Estat amb iproute2

FYI

- Per interfície o adreça: ip -s link|address [show intfname]

```
...
    RX: bytes  packets  errors  dropped overrun mcast
    8910361509 7457630  0       1572    0       24507
    TX: bytes  packets  errors  dropped carrier collsns
    609517648  3704976  33      0       0       0
```

- Per a veure els sockets que s'està escoltant: ss -4lntu

```
State   Recv-Q  Send-Q  Local Address:Port  Peer Address:Port
LISTEN  0       32            0.0.0.0:53         0.0.0.0:*
LISTEN  0       128           0.0.0.0:22         0.0.0.0:*
LISTEN  0       32               [::]:53            [::]:*
LISTEN  0       128              [::]:22            [::]:*
LISTEN  0       128                 *:80               *:*
```

- Mirar les connexions establertes: ss -4t

```
State   Recv-Q  Send-Q  Local Address:Port    Peer Address:Port
ESTAB   0       0      192.168.1.44:38116    40.101.137.98:https
ESTAB   0       0      192.168.1.44:52104  142.250.178.165:https
......
```

2-2 Manual amb ifupdown

- No cal fer tantes comandes:
  - ✓ es posa la configuració en un fitxer
- Aquest fitxer es llegeix en temps de boot
- Des de la línia de comandes es pot aixecar o baixar una de les NICs que hi hem definit;

```
ifquery --list
sudo ifdown eth0
sudo ifup eth0
ifquery --state eth0
```

- o bé reengegant el servei de xarxa:

```
systemctl restart networking
# service networking restart
```

però això afecta a tota la configuració de xarxa !

2-2 ifupdown: Boot

- Fitxer de configuració principal: /etc/network/interfaces

```
auto     lo
iface     lo inet  loopback

auto     eth0
iface     eth0 inet  static
    address       192.168.0.1
    network       192.168.0.0
    netmask       255.255.255.0
    broadcast     192.168.0.255
    gateway       192.168.0.111
    hostname       arrakis
    dns-domain example.com
    dns-nameservers 192.168.11.1
    post-up  iptables-restore < /etc/regles.fw
```

al boot ja configurarà les auto

pre-up
up
post-up
pre-down
post-down

- Quan hi diu *allow-hotplug* no les aixeca al boot sols si:
  - es fa ifup manualment o
  - es connecta la tarja en calent

```
allow-hotplug wlan0
iface     wlan0 inet  dhcp
  wpa-conf /etc/wpa_supplicant/wlan0.conf
```

feu: man interfaces

2-2: fitxers ifupdown

- Configuració al fitxer /etc/network/interfaces

```
/etc/network
├── if-down.d
│   └── openvpn
├── if-post-down.d
│   └── bridge -> /lib/bridge-utils/ifupdown.sh
├── if-pre-up.d
│   ├── bridge -> /lib/bridge-utils/ifupdown.sh
│   └── uml-utilities
├── if-up.d
│   ├── openvpn
│   └── uml-utilities
├── interfaces
└── interfaces.d
```

hook scripts: per a totes les interfícies

```
source /etc/network/interfaces.d/*
```

- però es pot posar a altres llocs:

```
$ ifup -i /home/milax/gsx/interfaces
```

2-2 Configuració de perfils

- Pot ser útil treballar amb perfils
  - ✓ permet configurar una mateixa interfície de formes diferents, segons el lloc on ens connectem.
- Utilització de perfils:
  - sense “network interface name”
  - sense “auto”
  - sense “allow-hotplug”

```
auto    lo
iface    lo inet  loopback

iface    Fora  inet  dhcp

iface    Casa  inet  static
    address       192.168.0.100
    netmask      255.255.255.0
    gateway      192.168.0.111
    dns-domain example.com
    dns-nameservers 192.168.11.1
```

- Connexió PC a LAN amb DHCP

  $ ifup eth0=Fora

  $ ifdown eth0=Fora

- Connexió PC a LAN amb static IP

  $ ifup eth0=Casa

  $ ifdown eth0=Casa

3- Gestors a alt nivell

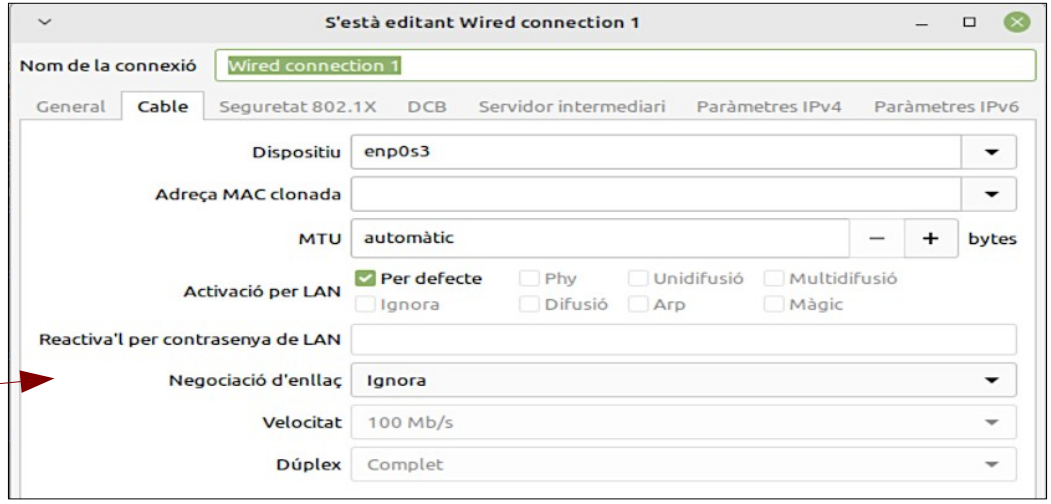
- Algunes de les distribucions modernes incorporen mecanismes nous de configuració
  - amb una sintaxi de les dades més estructurada,
  - unes comandes amb més funcionalitat
  - uns mecanismes més adients de comunicació amb el kernel i l'usuari
- Són els gestors de xarxa a alt nivell
  - ✓ NetworkManager
  - ✓ systemd.networkd

→ és difícil dir quin és millor o quin prevaldrà

- En tot cas, poden coexistir en un mateix sistema i cal anar molt en compte amb les **interferències** que es pugui provocar.

3-1: NetworkManager

- NetworkManager Server:
  - rep notificacions de udev via el system D-Bus
  - envia notificacions a les apps del desktop via el user D-Bus
- Cal anar en compte: interfereix amb els altres mètodes
  - ✓ normalment és recomanable usar un sol tipus de gestor
  - ✓ o bé anar en compte qui gestiona cada interfície
- Configurat via:
  - fitxers
  - script o CLI:
    - ✓ **nmcli**
    - ✓ nmtui
  - si disposem d'entorn gràfic:
    - ✓ nm-tray | nm-applet
    - ✓ nm-connetion-editor



3-1 NetworkManager

- Distingeix entre '**devices**' i '**connections**':

  diverses connexions poden usar un dispositiu però sols una d'elles pot estar activa.

- Les comandes **nmcli** més bàsiques són:
  - nmcli general {status | hostname | permissions | logging}
  - nmcli device (~ nmcli dev)
  - nmcli connection (~ nmcli con) # similar als perfils
  - nmcli networking { on | off }
  - nmcli networking check
    - ✓ retorna : none, portal, limited, full, unknown
- Sense paràmetres retornen un resum.
- **Va molt bé per als shell scripts (ex: --terse)**

3-1 NetworkManager

- Fitxer principal: /etc/NetworkManager/NetworkManager.conf

```
[main]
plugins=ifupdown,keyfile

[ifupdown]
managed=false

[device]
match-device=interface-name:wlan0
managed=1

[connection]
id=fils1
type=ethernet
autoconnect=false
interface-name=enp0s3

[ethernet]
mac-address=48:54:e8:2a:47:16
```

NM gestiona totes les NICs excepte les definides a /etc/network/interfaces

Si *true* les gestiona totes

Ull:
qui posa la ruta per defecte?

- Guarda les configuracions actuals a:

  /etc/NetworkManager/system-connections/ (ex: fils1.nmconnection)

feu: man NetworkManager.conf

3-1 Exemples nmcli

FYI

- Per a veure les connexions:

```
root@milax:~# nmcli connection show
NAME     UUID                                  TYPE      DEVICE
fils1    f0b8a07c-5b08-3b52-8349-06dba6c4afe5  ethernet  enp0s3
Wifi1    1a979eb3-93ce-4f60-a0a6-56024288ecec  wifi      --

root@milax:~# nmcli con show --active
NAME  UUID                                  TYPE      DEVICE
fils1 f0b8a07c-5b08-3b52-8349-06dba6c4afe5  ethernet  enp0s3

root@milax:~# nmcli -terse con show --active
fils1:f0b8a07c-5b08-3b52-8349-06dba6c4afe5:802-3-ethernet:enp0s3
root@milax:~# nmcli -g NAME con show --active
fils1
```

- Aixecar i baixar una connexió:

```
root@milax:~# nmcli conn down fils1
S'ha desactivat correctament la connexió «fils1» camí activa
     D-Bus: /org/freedesktop/NetworkManager/ActiveConnection/28)
root@milax:~# nmcli conn up fils1
La connexió s'ha activat correctament (camí actiu D-Bus: /org/freedeskt
```

feu: man nmcli-examples

3-1 Exemples nmcli

FYI

- device : amb informació L2, L3 i nm:

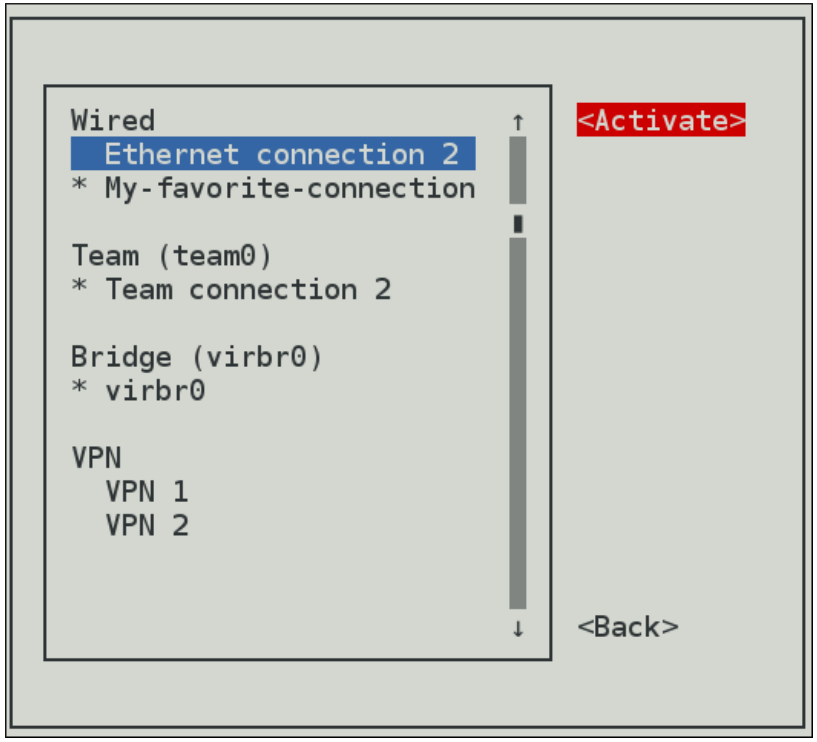
```
root@milax:~# nmcli dev show enp0s3
GENERAL.DEVICE:                         enp0s3
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         48:54:e8:2a:47:16
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     fils1
GENERAL.CON-PATH:                       /org/freedesktop/NetworkManager/ActiveConnecti
WIRED-PROPERTIES.CARRIER:               active
IP4.ADDRESS[1]:                         192.168.1.2/24
IP4.GATEWAY:                            192.168.1.1
IP4.ROUTE[1]:                           dst = 192.168.1.0/24, nh = 0.0.0.0, mt = 100
IP4.ROUTE[2]:                           dst = 169.254.0.0/16, nh = 0.0.0.0, mt = 1000
IP4.ROUTE[3]:                           dst = 0.0.0.0/0, nh = 192.168.1.1, mt = 100
IP4.DNS[1]:                             192.168.1.1
IP6.ADDRESS[1]:                         fe80::7b0c:2ace:beb6:e16/64
IP6.GATEWAY:                            --
IP6.ROUTE[1]:                           dst = fe80::/64, nh = ::, mt = 100

root@milax:~# nmcli -g GENERAL.STATE dev show enp0s3
100 (connected)
```

3-1 nmtui

- NetworkManager Text UI (amb ncurses)
  → **una bona opció per a servidors** (sense GUI)

```
┌──────────┤ TUI del NetworkManager ├──────────┐
│                                               │
│ Si us plau seleccioneu una opció              │
│                                               │
│ Edita una connexió                            │
│ Activa una connexió                           │
│ Estableix el nom de sistema de l'ordinador    │
│                                               │
│ Surt                                          │
│                                               │
│                                  <D'acord>    │
│                                               │
└───────────────────────────────────────────────┘
```



3-2 systemd.networkd

- Serveis: systemd-resolved, systemd-**networkd** (systemctl)
- Separa en diferents tipus de fitxers: .network .link .netdev
- Fitxers a: /etc/systemd/network/ | /usr/lib/systemd/network
  - Exemples .network: (similar als perfils)

10-dynamic.network

```
[Match]
Name=enp0s3
[Network]
DHCP=yes
```

20-static.network:

```
[Match]
Name=en*
[Network]
Address=192.168.0.15/24
Gateway=192.168.0.1
```

- CLI: **networkctl** {list,up,down,status,renew,reload,...}

feu: man systemd.network /EXAMPLES

systemd.networkd

FYI

- Exemples .link

```
[Match]
Path=*-usb-*

[Link]
NamePolicy=mac
```

```
[Match]
MACAddress=12:34:56:78:90:ab

[Link]
Description=USB to Ethernet Adapter
Name=ethusb0
```

- Exemple per a virtualització: .netdev

```
[NetDev]
Name=br0
Kind=bridge

# kind: bridge, bond, vlan, veth, etc
```

Scripting

- Podem consultar la informació amb qualsevol comanda i analitzar-la amb un ***shell script***:
  - ✓ grep | cut | tr | sed | head | tail ...
- És més fàcil extreure informació amb les noves eines.
  - Exemple amb nmcli amb shell:

```
milax:~$ nmcli --terse dev show enp0s3 | grep "IP4.ADDRESS"
IP4.ADDRESS[1]:10.0.0.1/16
IP4.ADDRESS[2]:192.168.1.44/24

milax:~$ nmcli -g IP4.ADDRESS dev show enp0s3
10.0.0.1/16 | 192.168.1.44/24
```

- Però encara poden portar a errors

  a l'exemple anterior, quina és la primera adreça ?

Scripting

- Les eines que poden extreure en format **json** el faciliten

```
[{"ifindex":2,"ifname":"enp0s3","flags":["BROADCAST","MULTICAS
,"LOWER_UP"],"mtu":1500,"qdisc":"pfifo_fast","operstate":"UP",
de":"DEFAULT","group":"default","txqlen":1000,"link_type":"eth
dress":"08:00:27:97:21:07","broadcast":"ff:ff:ff:ff:ff:ff"}]
```

- Exemple ip address i python:

```
import subprocess
import json
cercaINTF= "enp0s3"
ipv4= None
output= subprocess.check_output(["ip --json address"], shell=True)
NICs= json.loads(output)
for nic in NICs:
    if nic['ifname'] == cercaINTF:
        if nic['operstate'] == 'UP':
            mac= nic['address']
            for addr in nic['addr_info']:
                if addr['family'] == "inet":
                    ipv4= addr['local']
                    break
if not ipv4 is None:
    print(ipv4,'\t',mac)
```

Resum de config de NICs

- Configuració mínima:
  - ✓ dhclient + comandes iproute2 (o net-tools)
- Els *desktops* solen incorporar un gestor d'alt nivell i eines gràfiques per a usuaris no privilegiats
- Hi pot haver diversos gestors simultàniament i cal anar en compta en les **interferències** entre ells:
  - ✓ qui gestiona una interfície concreta ?
- NM és molt proactiu
  - la eina nmcli és molt potent
- systemd-networkd sol estar present però no intervé
  - la eina networkctl és poc versàtil (de moment...)
  - només es configura mitjançant fitxers

Resum de config de NICs

- Qui gestiona una interfície concreta ?

```
milax:~$ ifquery --state lo enp0s3 lxcbr0
lo=lo
lxcbr0=lxcbr0
```

ifupdown ?

```
milax:~$ nmcli dev
DEVICE          TYPE      STATE          CONNECTION
docker0         bridge    connectat      docker0
wlo1            wifi      no disponible  --
p2p-dev-wlo1    wifi-p2p  no disponible  --
lxcbr0          bridge    sense gestió   --
lxcbr1          bridge    sense gestió   --
enp3s0          ethernet  sense gestió   --
vboxnet0        ethernet  sense gestió   --
lo              loopback  sense gestió   --
```

NM?

```
milax:~$ networkctl list
IDX LINK            TYPE     OPERATIONAL SETUP
  1 lo              loopback carrier     unmanaged
  2 enp3s0          ether    degraded    unmanaged
  3 wlo1            wlan     off         unmanaged
  5 docker0         bridge   no-carrier  unmanaged
 79 vboxnet0        ether    off         unmanaged
100 lxcbr0          bridge   routable    unmanaged
101 lxcbr1          bridge   degraded    unmanaged
```

networkd?

Referències Extres

- /proc/sys/net

  https://www.kernel.org/doc/html/latest/admin-guide/sysctl/net.html

  https://www.kernel.org/doc/html/latest/networking/ip-sysctl.html

- iproute2:
  - Task-centered iproute2 user guide

Annexos

- Diapositives extres, només com a curiositat (FYI)
  - Visió global de les diferents distribucions Linux
  - Canviant el nom de les interfícies de xarxa
  - Exemples amb MS Windows

Periodic Table of Linux Distros

FYI

Font: distrowatch.com

Popularity on DistroWatch → 1
Symbol: Mi
Name: Linux Mint
Stable Release Year: 2.006

13 Ar Arch Linux 2.002																	31 Sl Slackware Linux 1.994
3 Ma Manjaro 2.013	41 Ab ArchBang Linux 2.010												16 Ka Kali Linux 2.004	90 Ra Raspbian 2.015	69 Om OpenMandriva Lx 2.013	12 Ce CentOS 2.004	71 Ko Korora Project 2.011	82 Ab Absolute Linux 2.007
53 Re Revenge OS 2.017	70 Ch Chakra GNU/Linux 2.010												1 Mi Linux Mint 2.006	91 Si siduction 2.011	15 Li Linux Lite 2.014	7 Fe Fedora 2.003	58 Qu Qubes OS 2.012	72 Au AUSTRUMI 2.009
77 Pa Parabola 2.011	34 Ue Ultimate Edition 2.008	35 Bl Black Lab Linux 2.008	22 Ll LXLE 2.014	57 Us Ubuntu Studio 2.007	24 Um Ubuntu MATE 2.015	25 Ta Tails 2.010	26 Pe Peppermint OS 2.011	27 Du Devuan GNU+Linux 2.017	28 Pa Parrot Security OS 2.016	66 Ec ExTiX 2.008	30 Ue Ubuntu Budgie 2.016	18 Kn KDE neon 2.017	33 Xu Xubuntu 2.006	103 Gm GALPon MiniNo 2.010	86 Rs Rockstor 2.015	61 Or Oracle Linux 2.009	87 Ze Zenwalk Linux 2.005
5 An Antergos 2.013	102 Gp GParted Live 2.006	8 Dp deepin 2.004	89 Uk Ubuntu Kylin 2.013	4 Ub Ubuntu 2.004	42 Ne Netrunner 2.010	62 Ba BackBox Linux 2.012	44 Ku Kubuntu 2.005	100 Og OSGeo-Live 2.017	65 Cz Clonezilla Live 2.008	98 Sh SharkLinux 2.017	48 Qo Q4OS 2.015	49 Ug Ubuntu GNOME 2.013	32 Sp SparkyLinux 2.012	51 Kn KNOPPIX 2.003	17 Ma Mageia 2.011	68 Sc Scientific Linux 2.007	84 Sa Salix 2.009
55 Bs Bluestar Linux 2.016	11 El elementary OS 2.011		83 Mx Maui Linux 2.016	73 St SteamOS 2.013	74 El Elive 2.005	54 Vo Voyager Live 2.012	88 Em Emmabuntüs 2.014	37 Bo Bodhi Linux 2.012	78 So SolydXK 2.013	20 Ax antiX 2.010	80 Ro Robolinux 2.014	81 Tr Trisquel GNU/Linux 2.007	2 De Debian 1.996	36 Fo feren OS 2.016	14 Pc PCLinuxOS 2.005	39 Ro ROSA 2.011	75 Ve VectorLinux 2.003
50 Ba BlackArch Linux 2.014	117 Tp 3CX Phone System 2.016		104 Lk Linux Kodachi 2.017	105 Ne Neptune 2.017	59 Wa wattOS 2.010	107 Sk Skolelinux 2.004	108 Us Univention CS 2.016	109 Gr Grml 2.004	110 Ip IPFire 2.011	111 Un Untangle NG Firewall 2.008	19 Lu Lubuntu 2.010	21 Mx MX Linux 2.017	10 Zo Zorin OS 2.009	115 Bl BunsenLabs Linux 2.016	56 Co ClearOS 2.001	76 Re RHEL 1.995	85 Po Porteus 2.011

40 Ka KaOS 2.014	114 Bp blackPanther OS 2.003	106 Qu Quirky 2.010	60 Ra RancherOS 2.017	113 Ss SLES 2.004	101 Go GoboLinux 2.002	63 Ni NixOS 2.014	64 Nu NuTyX 2.010	47 Ap Alpine Linux 2.010	29 An Android-x86 2.009	67 Ge GeckoLinux 2.016	9 So Solus 2.016	38 Ge Gentoo Linux 2.002	43 Sa Sabayon 2.005	118 Pk Porteus Kiosk 2.014
23 Pu Puppy Linux 2.006	52 Fm 4MLinux 2.014	116 Dl Devil-Linux 2.002	92 Ay AryaLinux 2.015	6 Su openSUSE 1.998	94 Le LibreELEC 2.017	95 Oe OpenELEC 2.012	96 St SliTaz GNU/Linux 2.008	97 Vd Void 2.015	46 La Lakka 2.016	99 Ov OpenMediaVault 2.014	45 Tc Tiny Core Linux 2.009	112 Rc Redcore Linux 2.017	93 Sc SystemRescueCd 2.004	79 Ca Calculate Linux 2.009

Debian/Ubuntu | Arch | Red Hat/Fedora/Mandriva | Slackware | Gentoo | Other

altres Unix

Configuració de xarxa segons distribucions

- Debian like: ifupdown, NetworkManager , systemd (enllaç)
- Redhat like: (vegeu aquest enllaç)
  - NetworkManager i/o
  - ifupdown + /etc/sysconfig/network-scripts/
- Arch Linux: iproute + diversos (vegeu aquest enllaç)
- Slackware: /etc/rc.inet (vegeu aquest enllaç)
- Gentoo: ifconfig + net-setup (vegeu aquest enllaç)
- Suse: ifconfig + net-setup (/etc/sysconfig/network/ifcfg*)
- BSD, FreeBSD, OpenBSD:
  - ifconfig + /etc/rc.conf + /etc/netstart

Ubuntu netplan

FYI

- Usat en temps de *boot*, per a preparar la configuració
  - es posa a fitxers /etc/netplan/*.yaml

    Exemple:

```
network:
  version: 2
  renderer: networkd
  ethernets:
    eth_lan0:
      dhcp4: true
      match:
        macaddress: 00:11:22:33:44:55
      set-name: eth_lan0
```

genera la configuració per a:

systemd-networkd
o
NetworkManager

- També es pot aplicar en calent:

```
sudo netplan try
sudo netplan apply
```

Noms de les NICs

FYI

- Els noms de les targes de xarxa poden variar molt.
  - Exemple: part del **dmesg**:

```
[3.627496] kernel: r8169 0000:03:00.0 eth0: RTL8168h/8111h, 48:54:e8:2a:47
[3.629323] kernel: r8169 0000:03:00.0 enp3s0: renamed from eth0
```

- Poden ser canviades, a diversos llocs, però és important quan es fa.
- L'ordre de preferència està especificat a:

```
# File: /lib/systemd/network/99-default.link
[Match]
OriginalName=*
[Link]
NamePolicy=keep kernel database onboard slot path
AlternativeNamesPolicy=database onboard slot path
MACAddressPolicy=persistent
```

Noms de les NICs

FYI

- Per a esbrinar tota la informació:

```
$ udevadm info /sys/class/net/enp3s0
P: /devices/pci0000:00/0000:00:1c.7/0000:03:00.0/net/enp3s0
...
E: INTERFACE=enp3s0
E: IFINDEX=2
...
E: ID_NET_NAME_MAC=enx4854e82a4716
E: ID_NET_NAME_PATH=enp3s0
E: ID_BUS=pci
E: ID_VENDOR_ID=0x10ec
E: ID_MODEL_ID=0x8168
...
E: ID_PATH=pci-0000:03:00.0
...
E: ID_NET_LINK_FILE=/usr/lib/systemd/network/99-default.link
E: SYSTEMD_ALIAS=/sys/subsystem/net/devices/enp3s0
E: TAGS=:systemd:
... { vol dir: output omitted }...
```

Noms de les NICs

FYI

- Es pot evitar amb paràmetres del kernel, abans del boot, al grub

```
net.ifnames=0 and biosdevname=0
```

- o bé després del boot a les regles **udev**:

```
# File: /etc/udev/rules.d/70-persistent-net.rules
SUBSYSTEM=="net", ACTION=="add", KERNELS=="0000:03:00.0"
, NAME="nic0"
```

  - ✓ per PCI o per MAC:

```
ATTR{address}=="01:02:03:04:05:06"
```

- o bé als fitxers de configuració:
  - ✓ netplan.yaml, interfaces, NetworkManager.conf, systemd .link,
- o bé amb les comandes de **iproute**:

```
ip link set dev enp3s0 name nic0
```

2-1 Configuració Manual

FYI

- NET-TOOLS: antic, però usada a algunes distros

```
# ifconfig eth0 192.168.1.129 netmask 255.255.255.192 up
```

ifconfig interface [address [parameters]]

- ✓ up | down
- ✓ netmask mask
- ✓ pointopoint address
- ✓ broadcast address
- ✓ metric number
- ✓ mtu bytes
- ✓ arp | -arp
- ✓ ...

route [add | del] [-net | -host] target [if]

```
# route add default gw $IPDEFAULTGW
# route add $IPDESTNET gw $IPGATEWAY
```

Scripts Windows

- Hi ha diverses opcions:
  - Comandes del DOS (batch files: *.bat)
  - Comandes netsh
  - Comandes wmi

    (windows managment instrummentation commands)
  - Comandes PowerShell
    - ✓ el més modern i potent
    - ✓ la corba d'aprenentatge és gran

Scripts Windows: DOS

FYI

- Comandes DOS:
  - **ipconfig, ping, tracert, arp, getmac, nslookup**
  - Proveu des d'un cmd.exe:
    - ipconfig /h
  - fitxers a: C:\Windows\System32\drivers\etc
- Exemple .bat usat a les pràctiques (d'alguns cursos):
  - Llista les NICs d'un node de Virtualbox i obté les NICs connectades, els seus números i si estan connectats:

```
set info=""
for /F "delims=" %%N in ('%VBOXMNG% showvminfo %NODE1% ^| findstr /i "NIC
.*connected" ^| find %ZONA%') do set info=%%N

set nic=0
for /f "tokens=1 delims=:" %%N in ('echo "%info:)=\)%"') do set nic=%%N
set nic=%nic:~1%

set num=0
for /f "tokens=2 delims= " %%N in ('echo %nic%') do set num=%%N

set mode=""
for /f "tokens=3 delims=," %%N in ('echo "%info:)=\)%"') do set mode=%%N
for /f "tokens=2 delims=:" %%N in ('echo %mode: =%') do set mode=%%N
```

Scripts Windows: netsh

FYI

- **netsh** examples:

```
netsh interface show interface
pause
netsh interface set interface "Wi-Fi" Enable
netsh interface show interface
netsh wlan show profile
pause
netsh interface ipv4 set address name="Wi-Fi" source=dhcp
netsh interfave ipv4 show config
pause
rem o estàticament:
netsh interface ipv4 set address name="Wi-Fi"
                Static 192.168.1.3 255.255.255.0 192.168.1.1
netsh interface ipv4 set dns name="Wi-Fi" static 8.8.8.8
```

Scripts Windows: WNIC

FYI

- **wmic** examples:

```
:Get-Adapter-Index
REM  Adapter name "%~1"    Return variable "%~2"
  for /F %%I in ('wmic nicconfig  where "IPEnabled=TRUE" get index ^|
            findstr /r [0-9]') do (
   for /F %%N in ('wmic nic where "deviceid=%%I" get NetConnectionID ^|
            findstr /v /r "^$ NetConnectionID" ^|findstr /i /c:"%~1"') do
                        CALL set "%~2=%%I"
  )
goto :EOF
```

Scripts Windows: PS

FYI

- **PowerShell**:

```
New-NetIPAddress –InterfaceAlias “Wired Ethernet Connection”
        -IPv4Address “192.168.1.3” -PrefixLength 24
        -DefaultGateway 192.168.1.254
```

```
Set-DnsClientServerAddress -InterfaceAlias “Wi-Fi”
        -ServerAddresses 192.168.1.2, 192.168.2.2
```