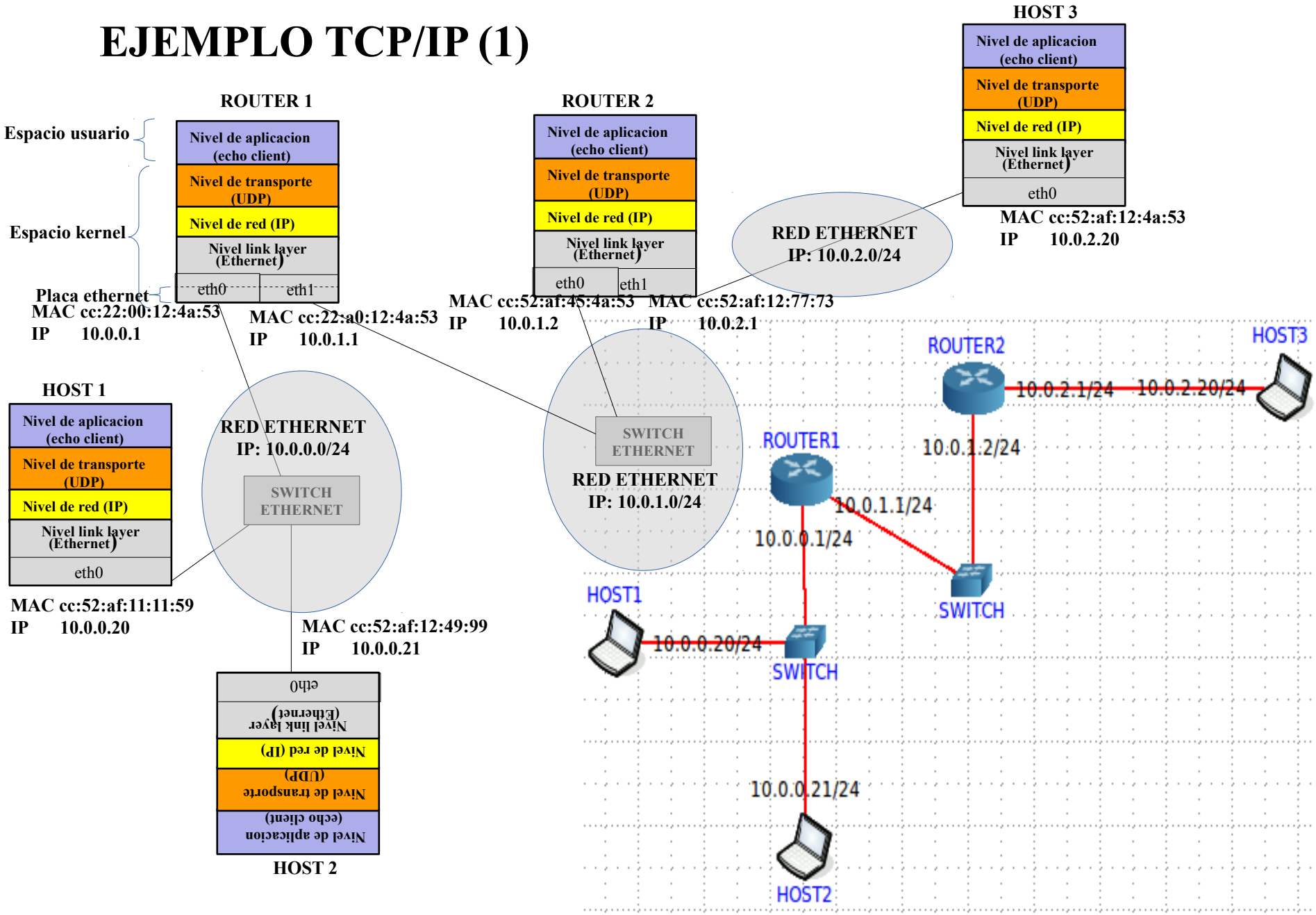
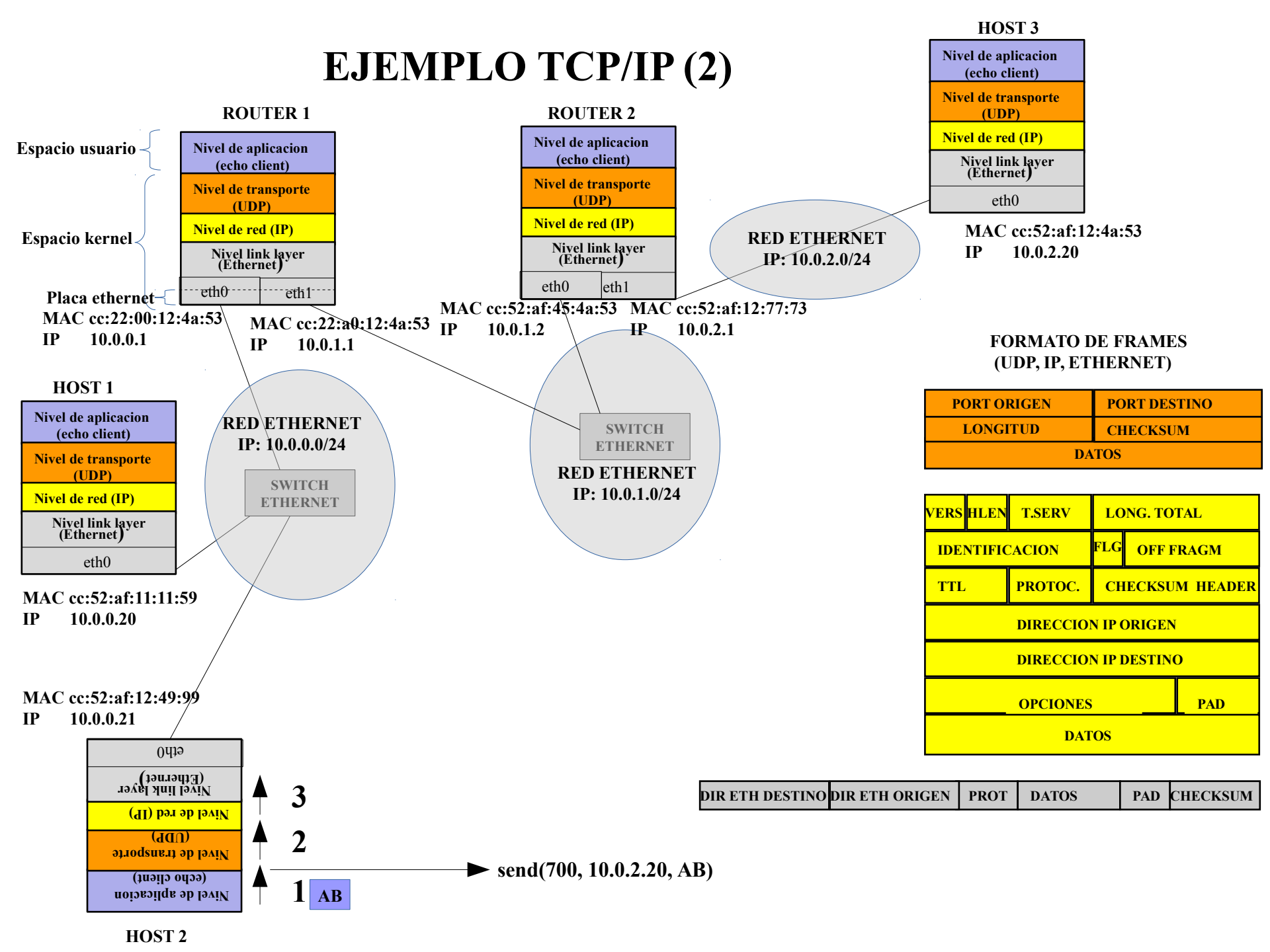
EJEMPLO TCP/IP (1)



EJEMPLO TCP/IP (2)



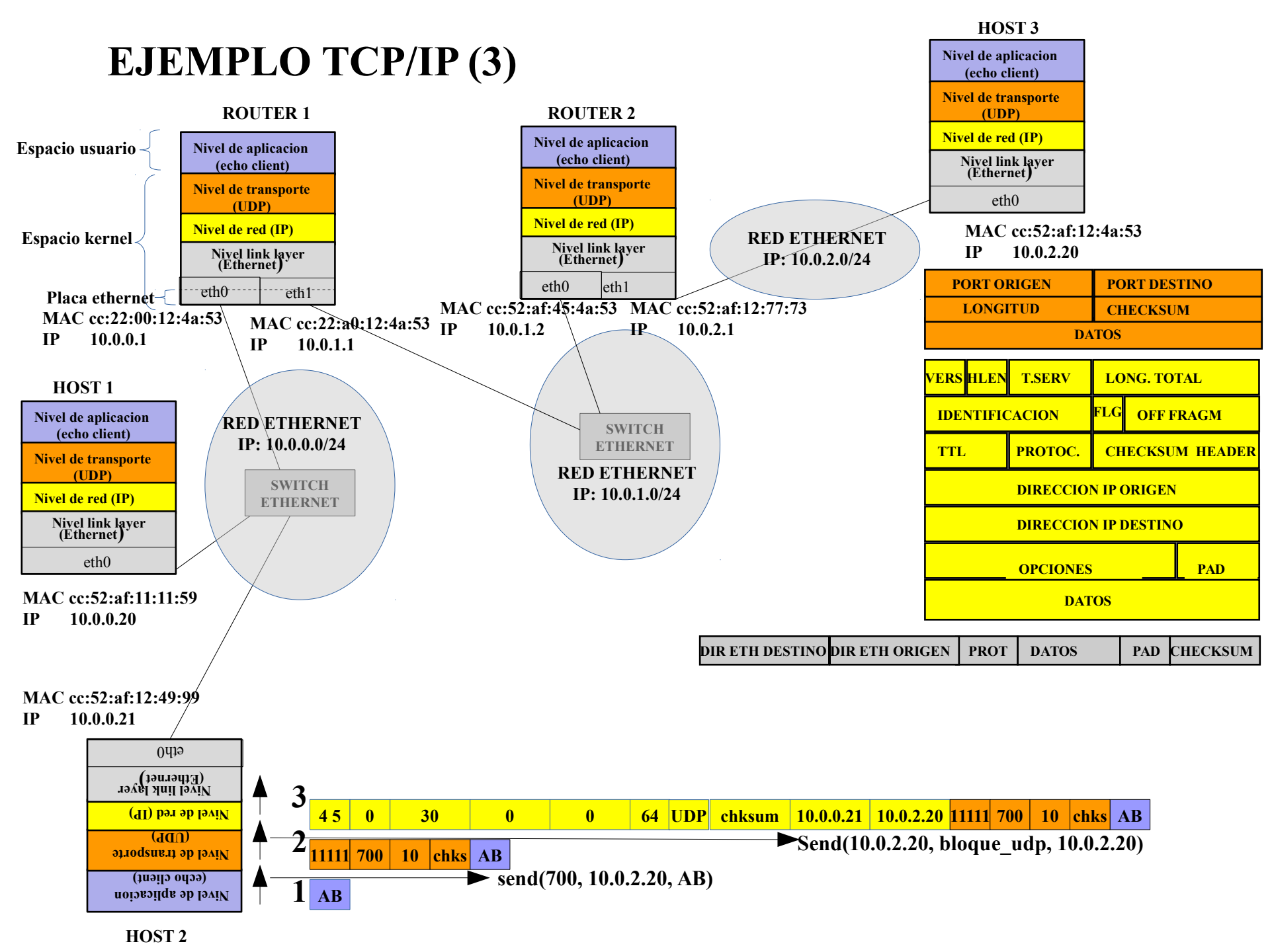
FORMATO DE FRAMES (UDP, IP, ETHERNET)

PORT ORIGEN	PORT DESTINO
LONGITUD	CHECKSUM
DATOS	

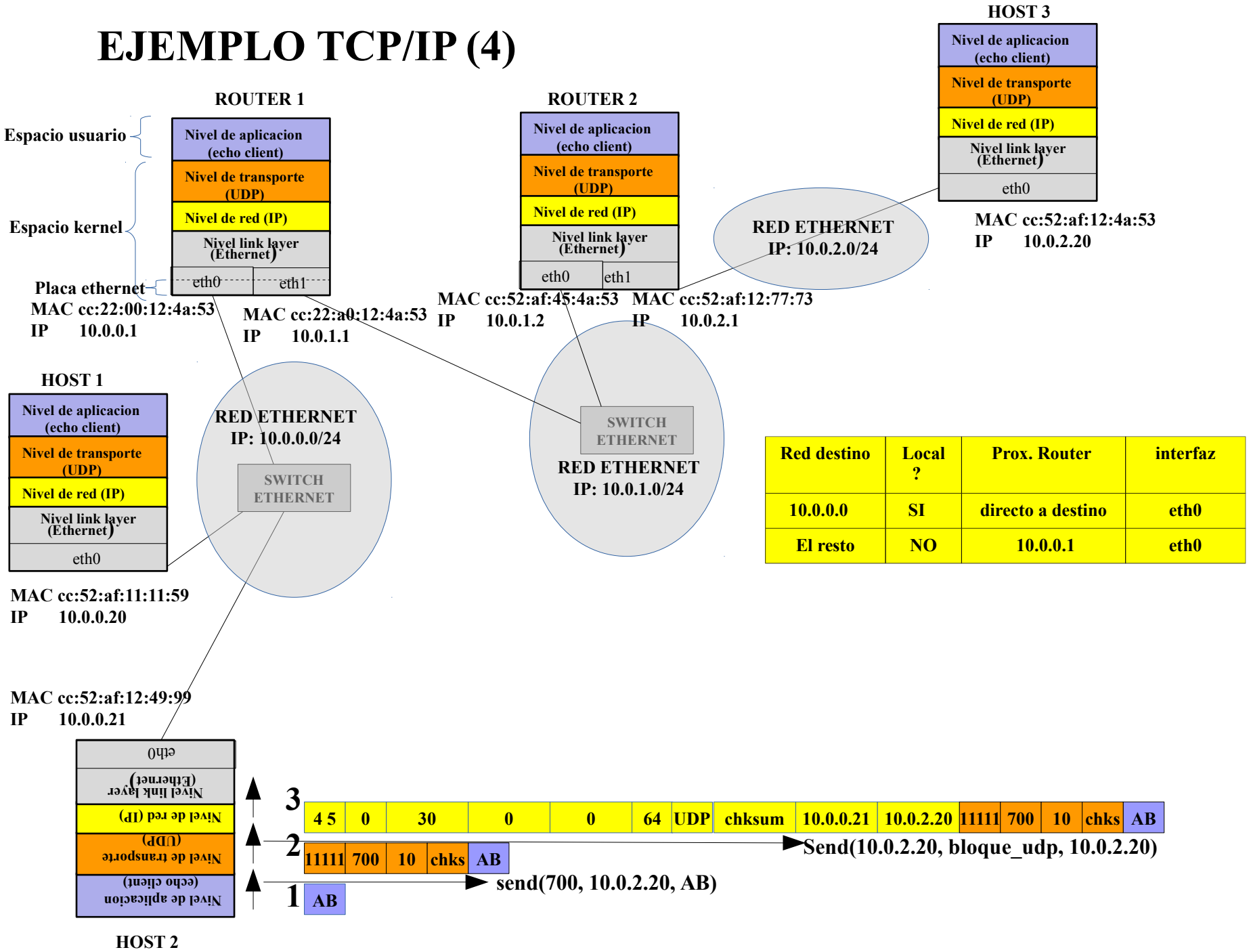
VERS	HLEN	T.SERV	LONG. TOTAL	
IDENTIFICACION		FLG	OFF FRAGM	
TTL	PROTOC.	CHECKSUM HEADER		
DIRECCION IP ORIGEN				
DIRECCION IP DESTINO				
OPCIONES				PAD
DATOS				

DIR ETH DESTINO	DIR ETH ORIGEN	PROT	DATOS	PAD	CHECKSUM

EJEMPLO TCP/IP (3)



EJEMPLO TCP/IP (4)



Red destino	Local ?	Prox. Router	interfaz
10.0.0.0	SI	directo a destino	eth0
El resto	NO	10.0.0.1	eth0

EJEMPLO TCP/IP (5)

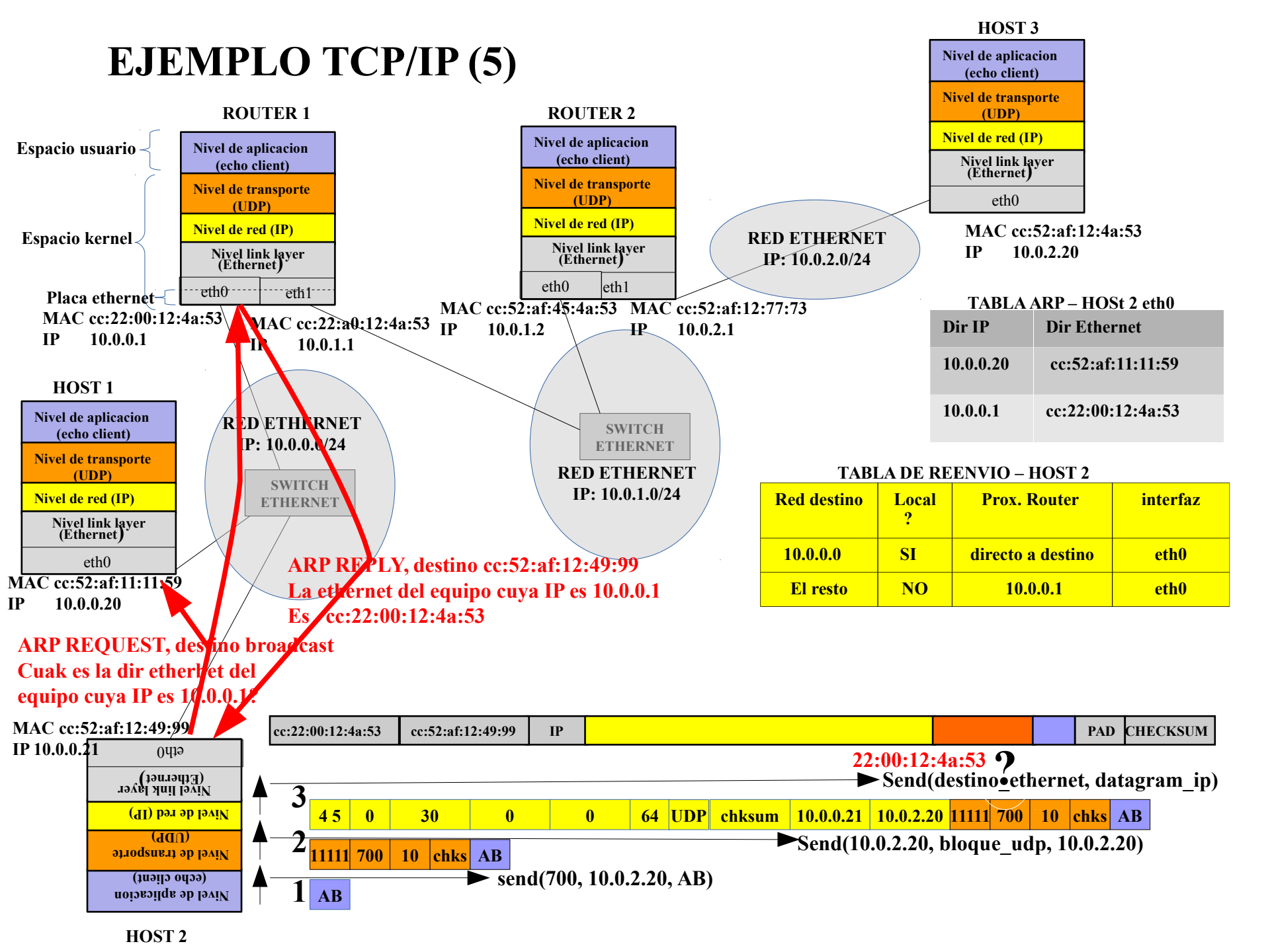


TABLA ARP – HOSt 2 eth0

Dir IP	Dir Ethernet
10.0.0.20	cc:52:af:11:11:59
10.0.0.1	cc:22:00:12:4a:53

TABLA DE REENVIO – HOST 2

Red destino	Local ?	Prox. Router	interfaz
10.0.0.0	SI	directo a destino	eth0
El resto	NO	10.0.0.1	eth0

EJEMPLO TCP/IP (6)

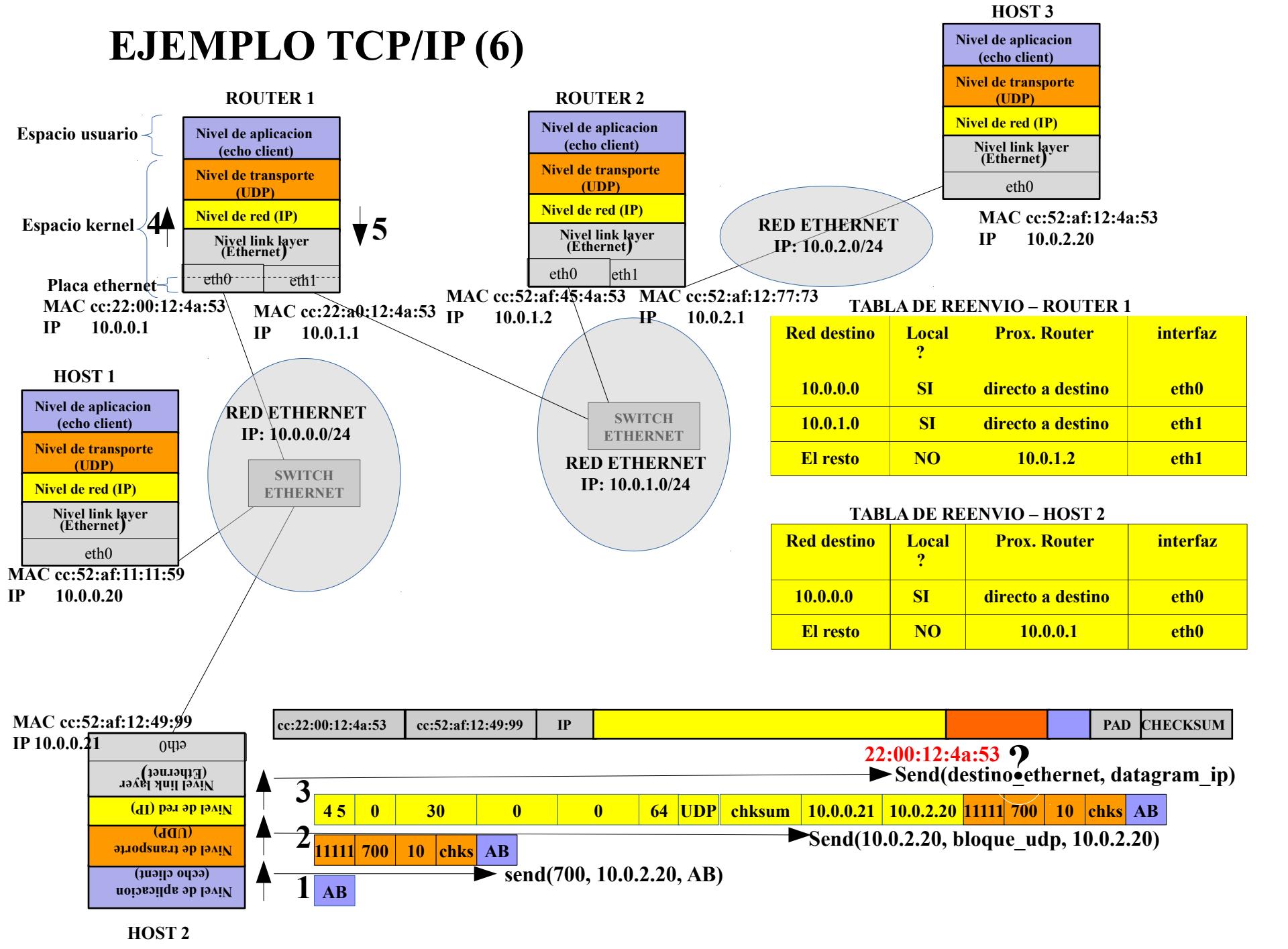


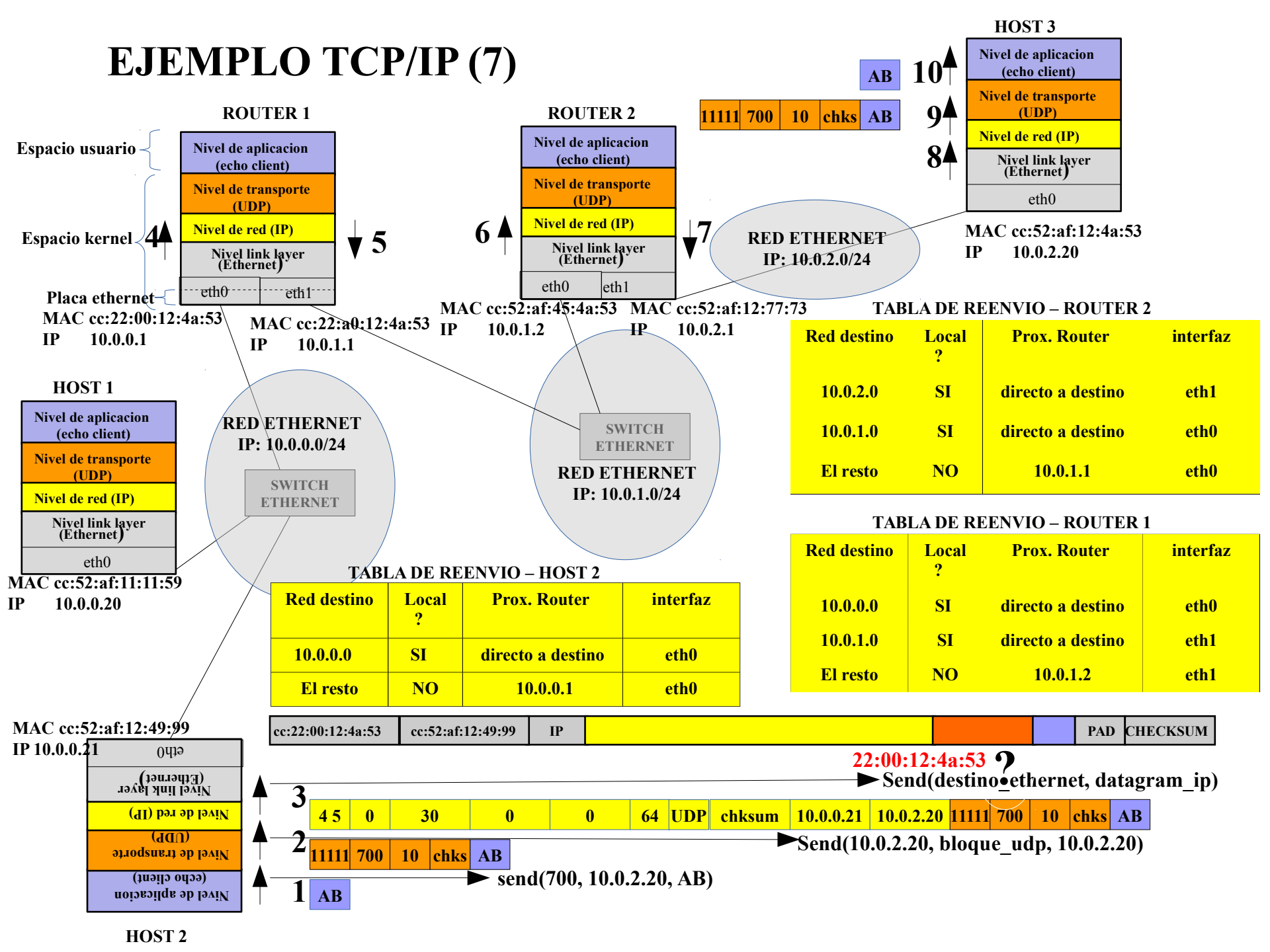
TABLA DE REENVIO – ROUTER 1

Red destino	Local ?	Prox. Router	interfaz
10.0.0.0	SI	directo a destino	eth0
10.0.1.0	SI	directo a destino	eth1
El resto	NO	10.0.1.2	eth1

TABLA DE REENVIO – HOST 2

Red destino	Local ?	Prox. Router	interfaz
10.0.0.0	SI	directo a destino	eth0
El resto	NO	10.0.0.1	eth0

EJEMPLO TCP/IP (7)

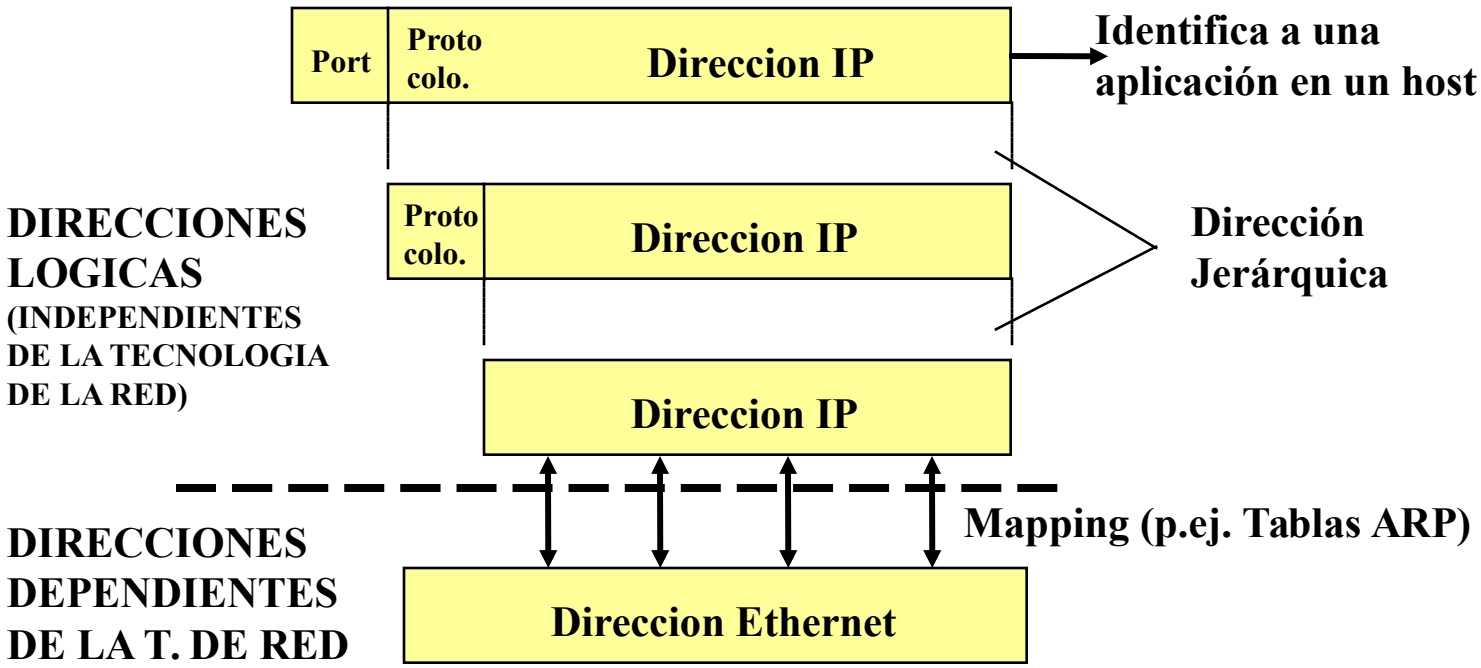


Direcciones IPv4

- **Identifican unívocamente un punto de acceso (interfaz) a la red. Un router o un host multi-homed tienen varias.**
- **Tienen un significado global en la Internet.**
- **Son asignadas por una autoridad central: InterNIC (Internet Network Information Center).**
- **Son números de 32 bits, expresados en notación decimal con puntos, byte a byte (p.ej. Ej. Representacion: 12.3.5.7, almacenada en la maquina (en hexa: 0C030507**
- **Un usuario de Internet, normalmente no las percibe, ya que para facilidad de los usuarios, se define un mapping estático de las direcciones IP con nombres "mas legibles" para las personas (DNS - Domain Name Server).**

Relación de las direcciones IPv4 con las de otros niveles

- Una dirección IP es independiente de las direcciones físicas de subred
- Las direcciones de niveles superiores contienen a la IP



Estructura de las direcciones IPv4

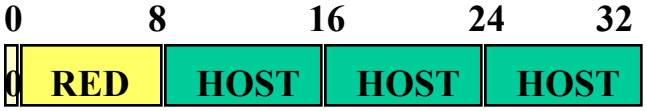
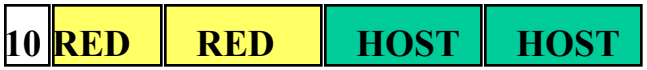
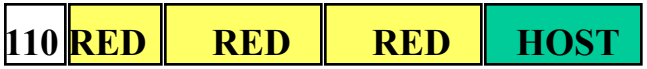
- **Esquema jerárquico, constan de una parte que indica de qué red física se trata, y otra que indica la interface o punto de conexión a la red (host).**

- **En 1984, se agrega una tercer elemento en la jerarquía para lograr mayor flexibilidad (subnets). (OBSOLETO)**

- **Los campos que componen la direción son de longitudes fijas predeterminadas; actualmente se elimina esta restricción (classless addressing).**

- **El componente RED de la dirección IP se utiliza para ubicar la red física de destino (ruteo) y el componente HOST se utiliza para identificar la interfaz dentro de esa red física**

Clases de direcciones IPv4 (obsoleto)

Clase	Formato	Rango	Redes/Hosts
A		0.0.0.0 a 127.255.255.255	126/16.777.214
B		128.0.0.0 a 191.255.255.255	16.382/65.534
C		192.0.0.0 a 223.255.255.255	2.097.150/254
D		224.0.0.0 a 239.255.255.255	
E		240.0.0.0 a 247.255.255.255	

Dirección especial: loopbak (127.0.0.0):

- **Para comunicaciones de procesos en la misma máquina.**
- **Nunca es propagada a la red**

Direcciones IPv4 con significado especial

Notación: <Red, Host>

<0, 0>	**este host en esta red**	**S**	**bootp**
<0, H>	**host H en esta red**	**S**	**host parcialmente inicializado**
<R, 0>	**un host en red R**	**S**	
<R, H>	**host H en red R**	**S/D**	
<R, -1>	**Directed broadcast todos los hosts de la red**	**D**	
<-1, -1>	**Limited broadcast**	**D**	**no propagada por los routers**

- **Significados especiales:**
  0: "este"
  -1: "todos"

No pueden usarse para identificar a un host o red en particular

Direcciones privadas

- **10.0.0.0 a 10.255.255.255 (una clase A)**
- **172.16.0.0 a 172.31.255.255 (16 clases B)**
- **192.168.0.0 a 192.168.255.255 (255 clases C)**

Problemas del esquema de direccionamiento original (clasfull)

- **Prefijos de red de longitud fija (clases), provoca un uso ineficiente en el espacio de direcciones: la capacidad en hosts de las redes predefinidas no se adapta a la realidad**

- **Crecimiento acelerado de la Internet, evidencia la falta de escalabilidad del esquema de direccionamiento (Agotamiento de clases B, incremento de tamaño de tablas de ruteo al utilizar direcciones de clase C).**

Soluciones

- **Estos problemas se solucionan a corto plazo en el contexto de IPv4.**
- **Definitivamente solucionados en IPv6.**

Clases, prefijos y mascaras

- **Cada clase IP define (según sus primeros bits) cuantos bits pertenecen a la parte de red y cuantos a la parte de host:**
  - **Clase A: 8 bits para red, 24 bits para host**
  - **Clase B: 16 bits para red, 16 bits para host**
  - **Clase C: 24 bits para red, 8 bits para host**

- **Tanto un prefijo como una mascara indican cuantos bits son para la red y cuantos para el host**
  - **Clase A: mascara:255.0.0.0 Prefijo:8**
  - **Clase B: mascara:255.255.0.0 Prefijo:16**
  - **Clase C: mascara:255.255.255.0 Prefijo:24**

- **Prefijo, mas comodo para la representacion**

- **Mascara:**
  - **usada internamente para obtener la red de una direccion IP dada:**

    Red = direccion_ip AND mascara

Direccionamiento IP

- **Direccionamiento jerárquico: <prefijo, host>**
  - **prefijo: utilizado por los routers para determinar paths para direcciones no locales**
  - **host: utilizado para ubicar el equipo local**

- **Prefijo**
  - **Compuesto por una dirección IP y una indicación de la cantidad de bits contiguos, a izquierda que lo componen**

  - **Longitud determinada por contexto**
    - **clase de dirección (A, B o C)**
    - **máscara de subred (extensión a derecha del prefijo de clase)**

  - **Indicado como una dirección IP, seguido de la cantidad de bits que lo componen**
    - **Clase C: 192.9.200.0/24**
    - **Clase B: 130.19.0.0/16**
    - **Clase A: 10. 0.0.0/8**

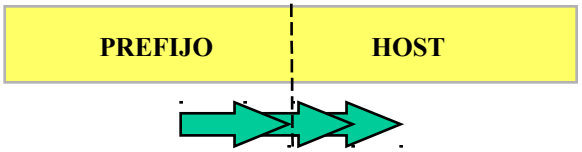
Clases de direccionamiento

- **Classful Addressing (OBSOLETO)**
  - **Los routers aceptan determinadas longitudes de prefijos (clases de direcciones IP): /8, /16 y /24**
  - **Para rutear un datagram, se busca en la tabla de rutas una dirección de red que coincida con el prefijo de la dirección de destino.**

- **Classless Addressing (UTILIZADO ACTUALMENTE)**
  - **Los routers aceptan longitudes de prefijo variables (/2 a /30)**
  - **Para rutear un datagram, se utiliza el criterio de ruta más específica (“longest match” al buscar en las tablas).**

Classless Addressing

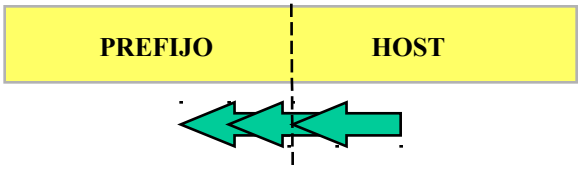
Subnetting (VLSM -Variable Length Subnet Masking-)



Extiende el prefijo hacia la derecha

Permite un mejor uso del espacio de direcciones, al soportar subredes de longitud variable que se adaptan mejor a casos particulares.

Supernetting (sumarización)



Reduce el prefijo hacia la izquierda

Permite reducir tamaño de tablas de ruteo y tráfico de intercambio de información de ruteo al posibilitar que un router anuncie y tenga una única entrada en la tabla para un conjunto de rutas.

VLSM (Variable Length Subnet Mask)

- **Uso más eficiente del espacio de direcciones**
  - **Posibilidad de asignar un número de direcciones acorde al número de equipos de la subred**

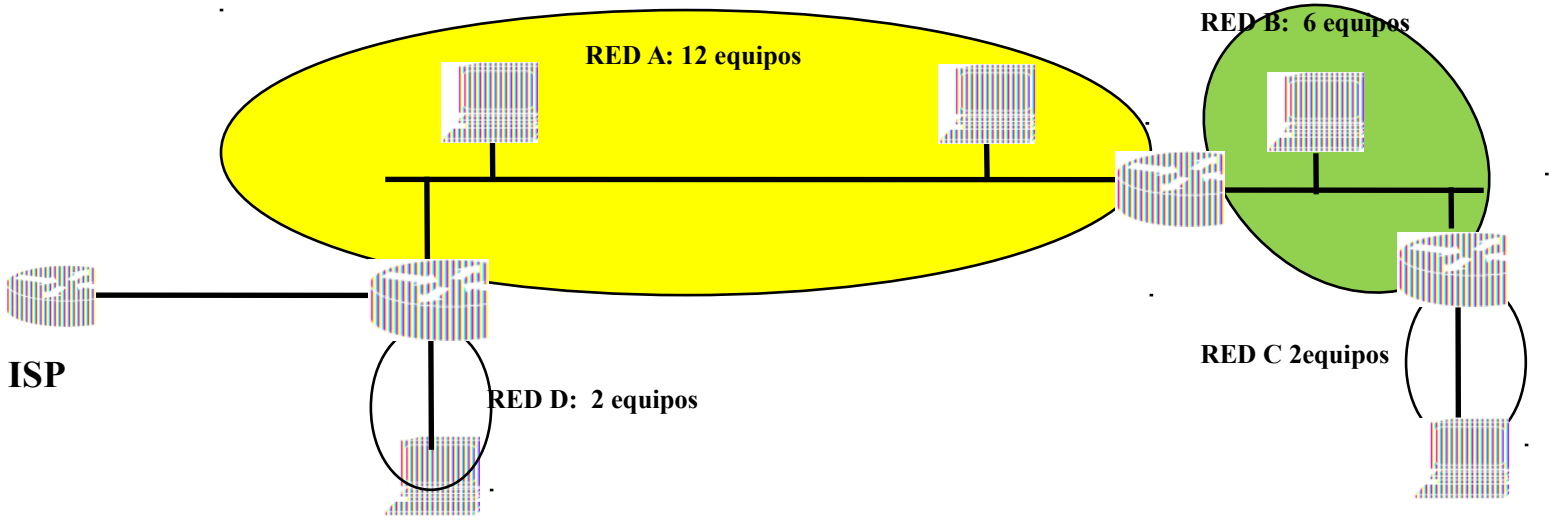
- **Requerimientos**
  - **Algoritmo “longest match prefix” para reenvío**
  - **Protcolos de ruteo que intercambien máscaras de red**
- **Reglas de asignación de direcciones**
  - **El espacio de direcciones asignado bajo una máscara no puede ser asignado bajo otra máscara (prefijo más largo).**

CLASSFUL

CLASSLESS

Ejemplo de asignación de direcciones conVLSM

- **Espacio de 32 direcciones IP asignado: 201.2.3.0/27**
- **Redes a definir: 12, 5, 2 y 2 equipos cada una**



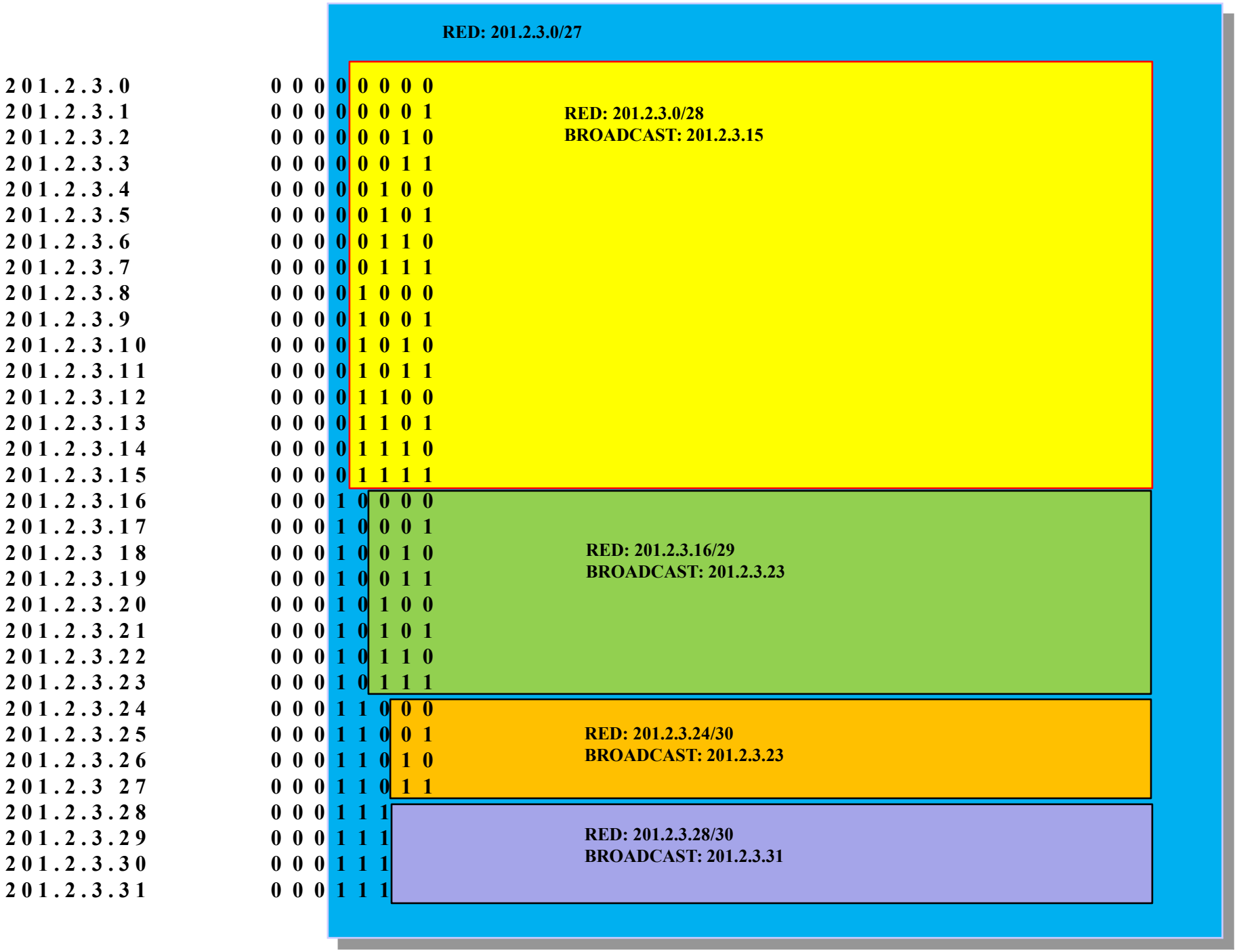
Asignacion de direcciones con VLSM

Considerar

- **No dividir en clases de direcciones:**
  en lugar de clase C 201.2.3.0
  bloque de 256 direcciones IP, a partir de la dirección 201.2.3.0
- **Se asignan bloques de 2**k direcciones, identificados por una red de prefijo (32 - k)**
- **La primera dirección es multiplo de 2**k, es la direccion de red**
- **El bloque de direcciones asignado, puede dividirse recursivamente, hasta llegar a bloques de prefijo 30**

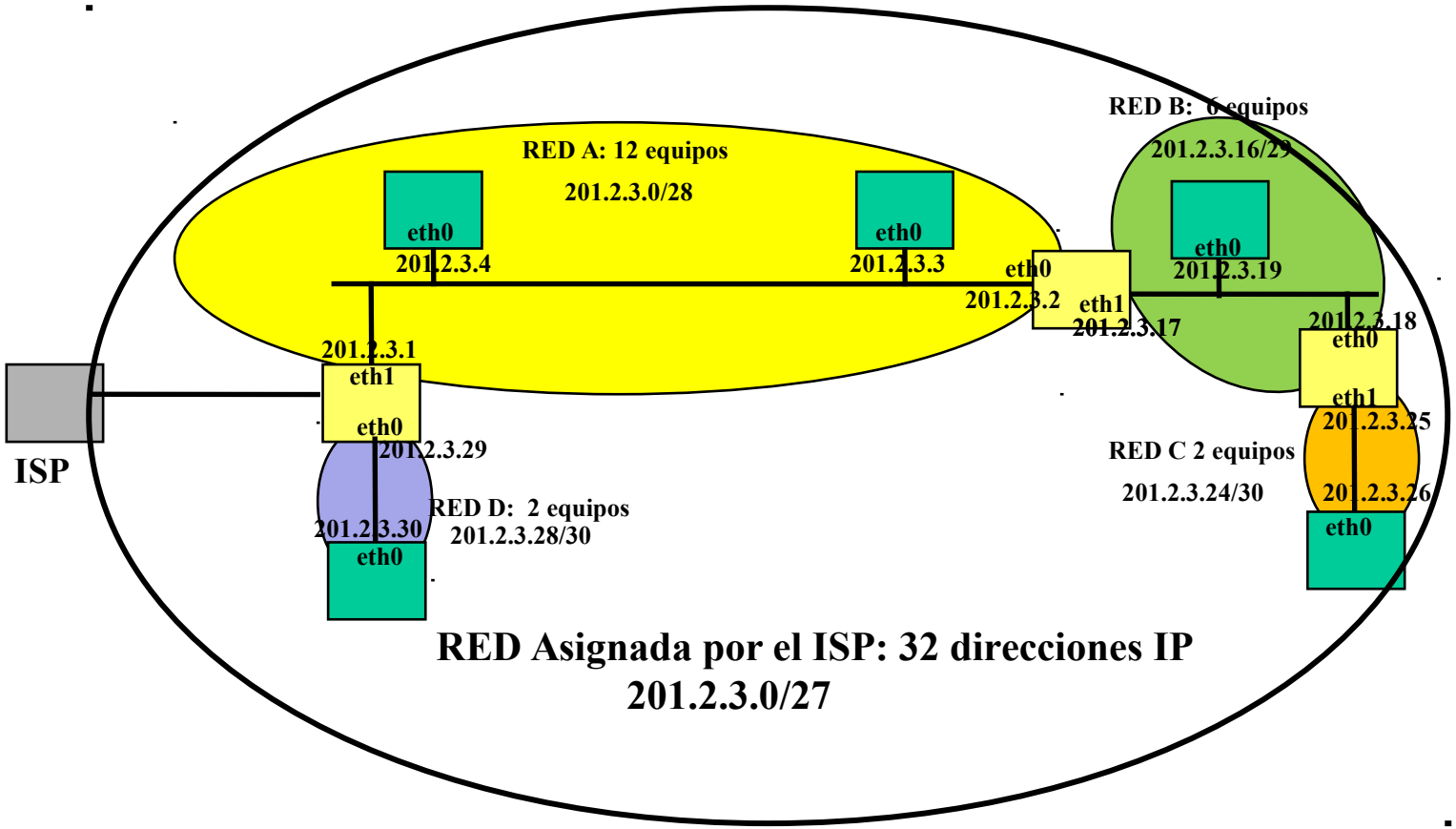
Procedimiento a seguir:

- **Ordenar las redes de mayor a menor cantidad de direcciones necesarias**
- **Desde la primer direccion del bloque, asignar una cantidad de direcciones contiguas a la primera red.Esta cantidad debe ser la minima potencia de dos tal que alcance para las direcciones que se necesitan en esa red**
- **Desde la siguiente direccion libre, realizar el mismo procedimiento para la siguiente red**
- **Tener en cuenta, si es posible, la topología de la red (de toda mi intranet)**


RED: 201.2.3.0/27
201.2.3.0 0 0 0 0 0 0 0 0
201.2.3.1 0 0 0 0 0 0 0 1
201.2.3.2 0 0 0 0 0 0 1 0
201.2.3.3 0 0 0 0 0 0 1 1
201.2.3.4 0 0 0 0 0 1 0 0
201.2.3.5 0 0 0 0 0 1 0 1
201.2.3.6 0 0 0 0 0 1 1 0
201.2.3.7 0 0 0 0 0 1 1 1
201.2.3.8 0 0 0 0 1 0 0 0
201.2.3.9 0 0 0 0 1 0 0 1
201.2.3.10 0 0 0 0 1 0 1 0
201.2.3.11 0 0 0 0 1 0 1 1
201.2.3.12 0 0 0 0 1 1 0 0
201.2.3.13 0 0 0 0 1 1 0 1
201.2.3.14 0 0 0 0 1 1 1 0
201.2.3.15 0 0 0 0 1 1 1 1
RED: 201.2.3.0/28
BROADCAST: 201.2.3.15
201.2.3.16 0 0 0 1 0 0 0 0
201.2.3.17 0 0 0 1 0 0 0 1
201.2.3 18 0 0 0 1 0 0 1 0
201.2.3.19 0 0 0 1 0 0 1 1
201.2.3.20 0 0 0 1 0 1 0 0
201.2.3.21 0 0 0 1 0 1 0 1
201.2.3.22 0 0 0 1 0 1 1 0
201.2.3.23 0 0 0 1 0 1 1 1
RED: 201.2.3.16/29
BROADCAST: 201.2.3.23
201.2.3.24 0 0 0 1 1 0 0 0
201.2.3.25 0 0 0 1 1 0 0 1
201.2.3.26 0 0 0 1 1 0 1 0
201.2.3 27 0 0 0 1 1 0 1 1
RED: 201.2.3.24/30
BROADCAST: 201.2.3.23
201.2.3.28 0 0 0 1 1 1
201.2.3.29 0 0 0 1 1 1
201.2.3.30 0 0 0 1 1 1
201.2.3.31 0 0 0 1 1 1
RED: 201.2.3.28/30
BROADCAST: 201.2.3.31

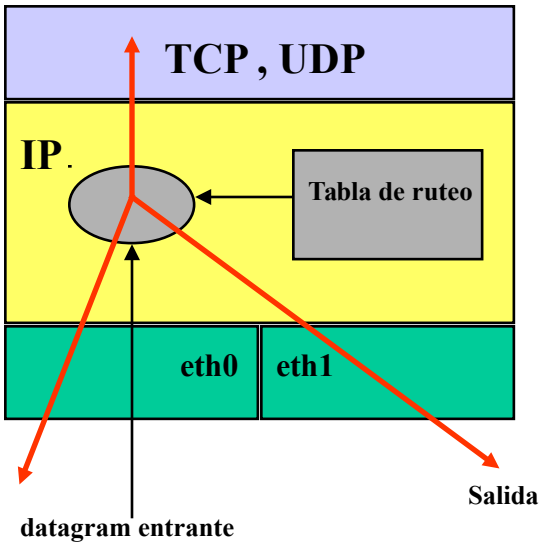
Ejemplo de asignación de direcciones conVLSM

- **Espacio de 32 direcciones IP asignado: 201.2.3.0/27**
- **Redes a definir: 12, 5, 2 y 2 equipos cada una**



Reenvío de paquetes

- Función correspondiente al nivel IP
- Para un datagram (originado en el equipo o entrante) debe decidirse, en base a su dirección de destino, hacia qué equipo enviarlo
- La decisión se toma en base a tablas de ruteo
- Las tablas pueden ser estáticas o dinámicas (si se utiliza un protocolo de ruteo)
- Un equipo que sólo funcione como host no reenvía datagrams



Ejemplo de reenvio VLSM (longest match prefix)

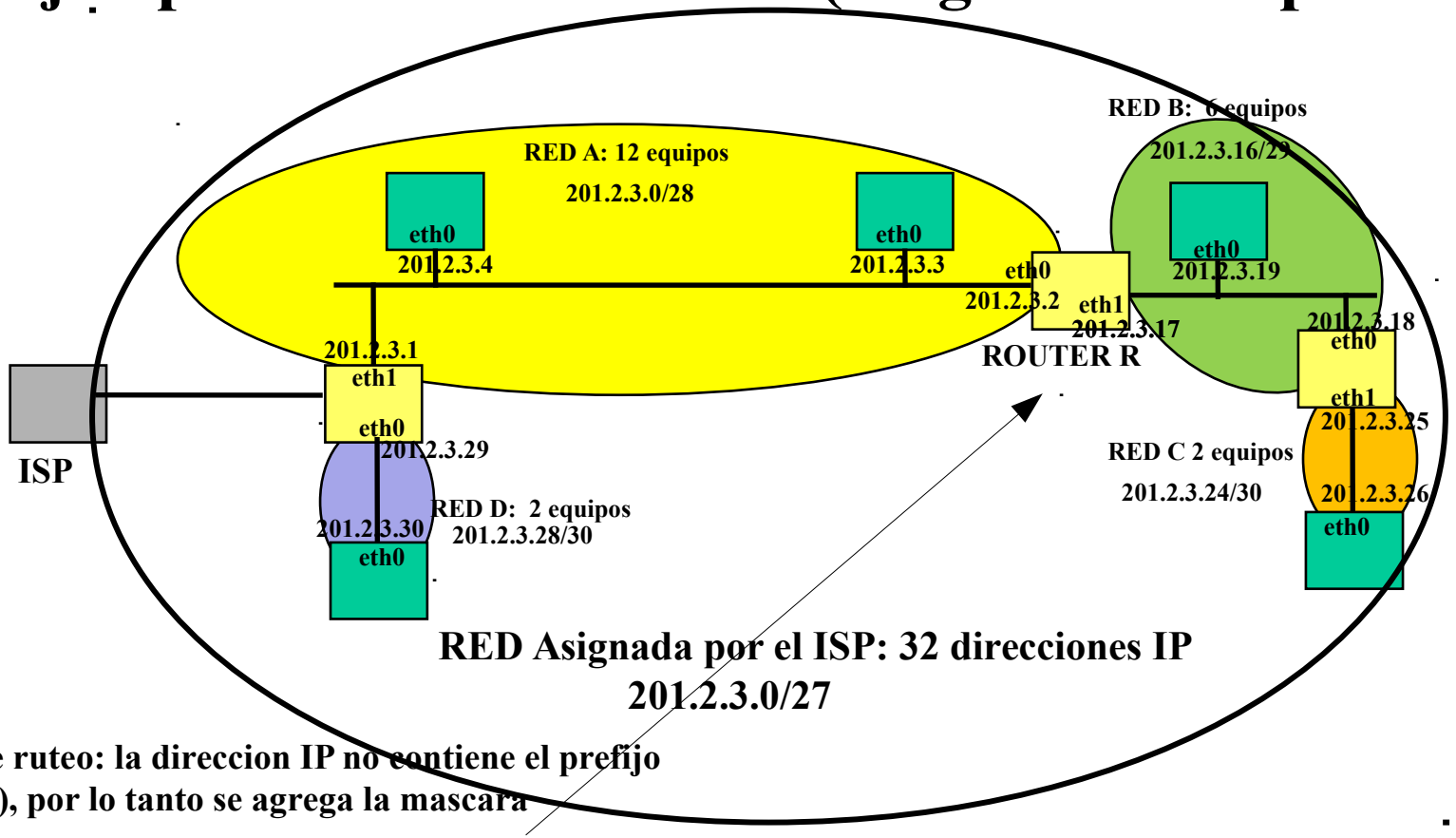


Tabla de ruteo: la direccion IP no contiene el prefijo (classles), por lo tanto se agrega la mascara

TABLA DE RUTEO (REENVIO) DE ROUTER R

Red destino	Mascara	Local?	Prox. Router	interfaz
201.2.3.16	255.255.255.248	SI	directo a destino	eth1
201.2.3.0	255.255.255.240	SI	directo a destino	eth0
201.2.3.28	255.255.255.252	NO	201.2.3.1	eth0
201.2.3.24	255.255.255.252	NO	201.2.3.18	eth1
0.0.0.0	0.0.0.0	NO	201.2.3.1	eth0

Ejemplo de reenvio VLSM (longest match prefix)

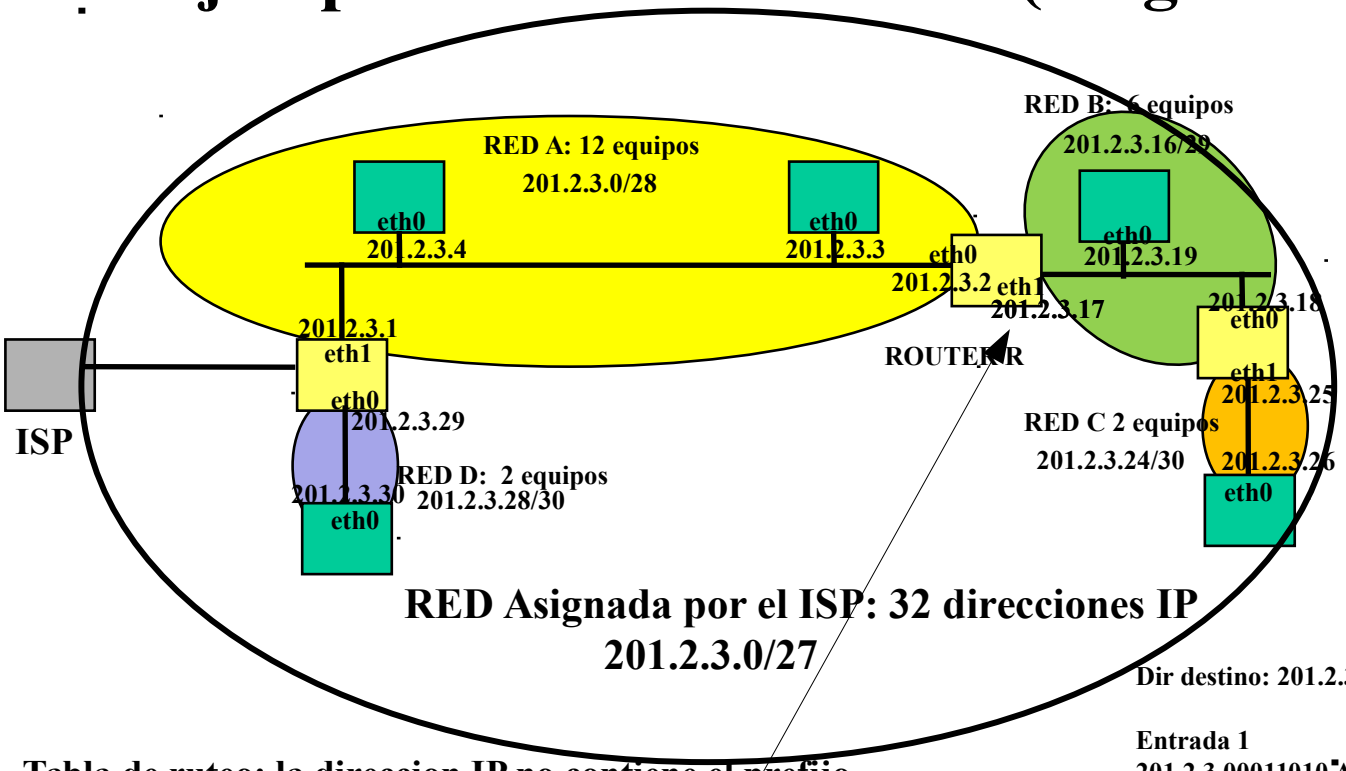


Tabla de ruteo: la direccion IP no contiene el prefijo (classles), por lo tanto se agrega la mascara

TABLA DE RUTEO (REENVIO) DE ROUTER R

Red destino	Mascara	Local?	Prox. Router	interfaz
201.2.3.16	255.255.255.11111000	SI	directo a destino	eth1
201.2.3.0	255.255.255.11110000	SI	directo a destino	eth0
201.2.3.28	255.255.255.11111100	NO	201.2.3.1	eth0
201.2.3.24	255.255.255.11111100	NO	201.2.3.18	eth1
0.0.0.0	0.0.0.0	NO	201.2.3.1	eth0

Dir destino: 201.2.3.26=201.2.3.00011010

Entrada 1
201.2.3.00011010 AND 255.255.255.11111000 = 201.2.3.00011000
201.2.3.24 |= 201.2.3.16 NO MATCH

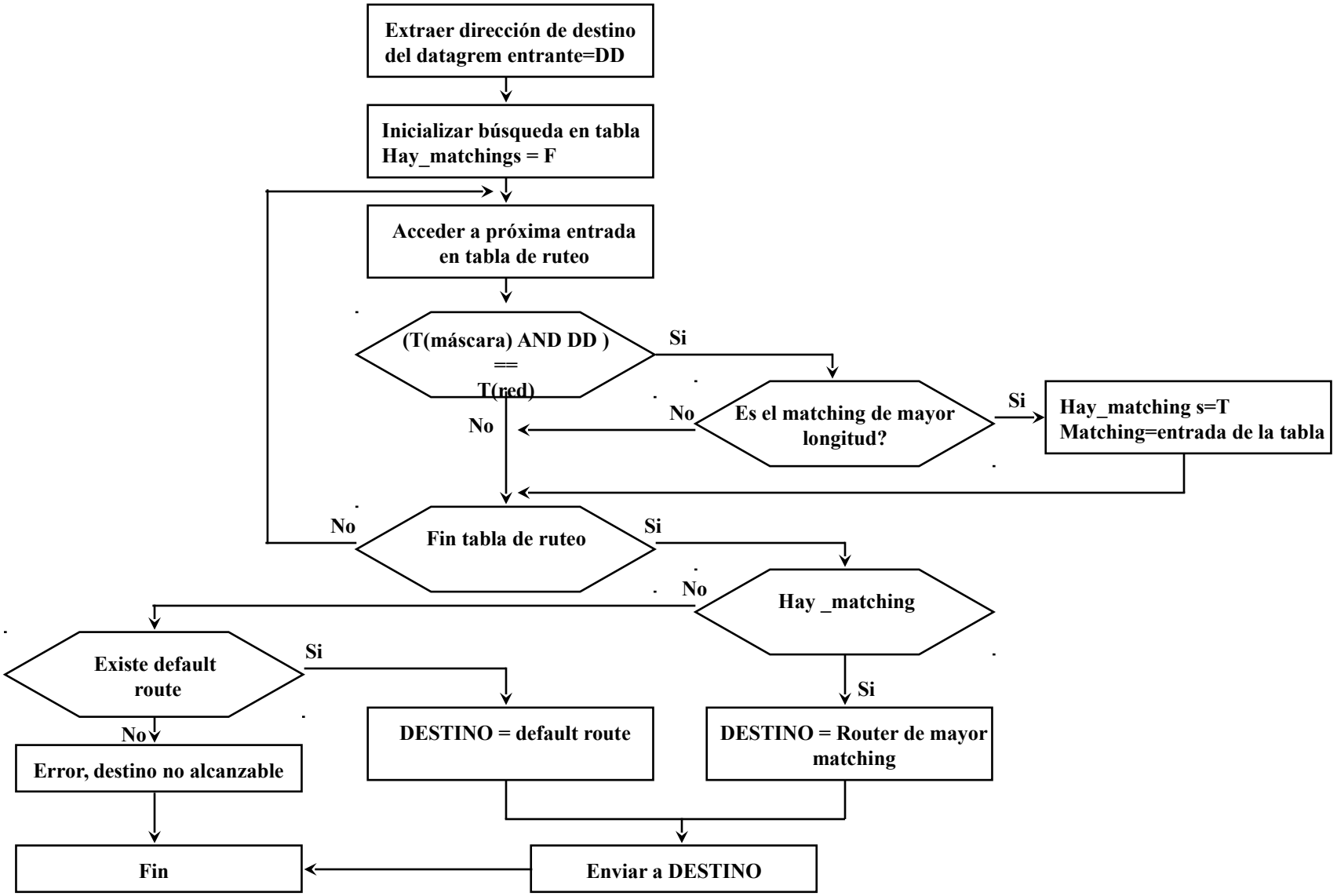
Entrada 2
201.2.3.00011010 AND 255.255.255.11110000 = 201.2.3.00010000
201.2.3.16 |= 201.2.3.0 NO MATCH

Entrada 3
201.2.3.00011010 AND 255.255.255.11111100 = 201.2.3.00011000
201.2.3.24 |= 201.2.3.28 NO MATCH

Entrada 4
201.2.3.00011010 AND 255.255.255.11111100 = 201.2.3.00011000
201.2.3.24 == 201.2.3.24 MATCH = 30

Entrada 5
201.2.3.00011010 AND 0.0.0.00000000 = 0.0.0.00000000
0.0.0.0 == 0.0.0.0 MATCH = 0

Algoritmo de búsqueda en tablas de ruteo con principio longest match prefix



CIDR

- **CIDR (RFC 1519, Nov 1992) propone:**
  - **Asignación jerárquica de grupos de direcciones de clase C**
  - **Direcciones classless: la división entre la parte de la dirección que corresponde a la red y al host es variable, indicada por una máscara (p.e. 200.2.2.2/24)**
  - **Los routers pueden "resumir" información respecto de un grupo de direcciones y propagar la información resumida (aggregation)**
  - **En las tablas de ruteo, se almacena la información resumida**
  - **Los protocolos de ruteo más nuevos lo soportan (BGP-4, OSPF, etc)**
  - **Los routers soportan el mecanismo de matching más específico (longest match) ya que es el utlizado en subnetting**

Asignación propuesta para las direcciones clase C

Direcciones 194.0.0.0 a 195.255.255.255 Europa
Direcciones 198.0.0.0 a 199.255.255.255 América del Norte
Direcciones 200.0.0.0 a 201.255.255.255 América Central y América del Sur
Direcciones 202.0.0.0 a 203.255.255.255 Asia y el Pacífico

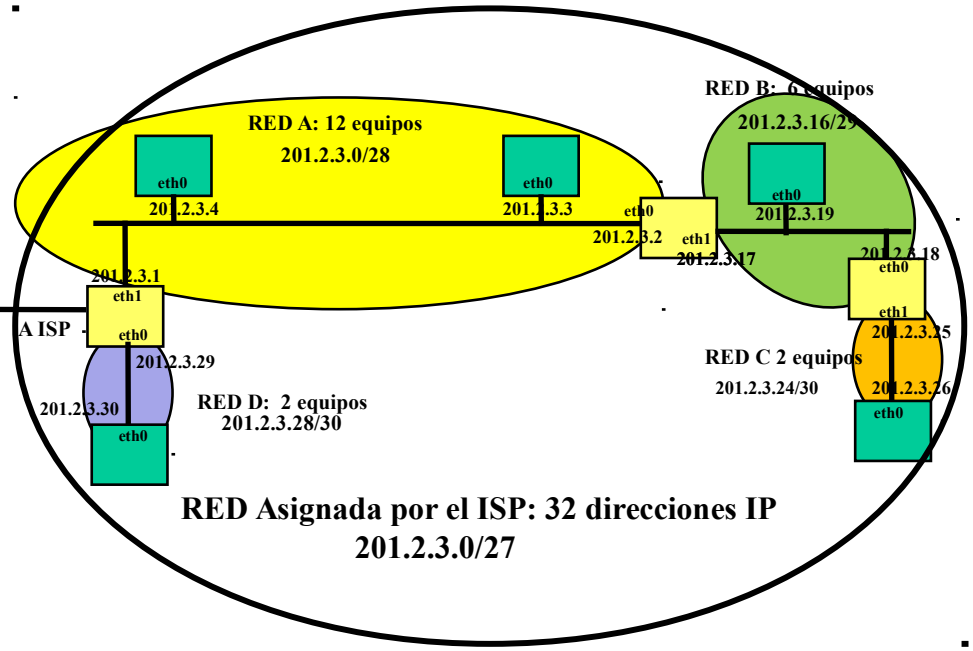
Configuracion de interfaces y direcciones

- **ifconfig (man ifconfig): Se utiliza para observar y/o configurar caracteristicas de las interfaces de red (p. ej. eth0, etc)**
  - **Permite (entre otras cosas):**
    - **Observar que interfaces hay y cuales son sus caracteristicas**
    - **Activar o desactivar una interfaz**
    - **Asignar a una interfaz una direccion IP y definir la red a la que la conectamos**

- **route (man route): Se utiliza para indicar rutas a diferentes redes.**

- **Ip 2 (man ip): reemplaza a los anteriores a partir del kernel.. Lo utilizaremos parcialmente para:**
  - **Configuracion de interfaces (ip link)**
  - **Asignacion de direcciones (ip address)**
  - **Configuracion de rutas (ip route)**

Ejemplo ifconfig y route



root@ROUTER-R: /tmp/pycore.60790/ROUTER-R.conf

Archivo Editar Ver Buscar Terminal Ayuda

```
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# ifconfig eth0 201.2.3.2/28
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# ifconfig eth1 201.2.3.17/29
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# ifconfig
eth0      Link encap:Ethernet  direcciónHW 00:00:00:aa:00:00
          Direc. inet:201.2.3.2  Difus.:201.2.3.15  Másc:255.255.255.240
          Dirección inet6: fe80::200:ff:feaa:0/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:61 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:8 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:10084 (10.0 KB)  TX bytes:888 (888.0 B)

eth1      Link encap:Ethernet  direcciónHW 00:00:00:aa:00:02
          Direc. inet:201.2.3.17  Difus.:201.2.3.23  Másc:255.255.255.248
          Dirección inet6: fe80::200:ff:feaa:2/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:61 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:8 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:10104 (10.1 KB)  TX bytes:888 (888.0 B)

root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route
Tabla de rutas IP del núcleo
Destino         Pasarela        Genmask         Indic Métric Ref    Uso Interfaz
201.2.3.0       *               255.255.255.240 U     0      0        0 eth0
201.2.3.16      *               255.255.255.248 U     0      0        0 eth1
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf#
```

Archivo Editar Ver Buscar Terminal Ayuda

```
oot@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# ifconfig
th0       Link encap:Ethernet  direcciónHW 00:00:00:aa:00:00
          Dirección inet6: fe80::200:ff:feaa:0/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:55 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:8 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:9442 (9.4 KB)  TX bytes:888 (888.0 B)

th1       Link encap:Ethernet  direcciónHW 00:00:00:aa:00:02
          Dirección inet6: fe80::200:ff:feaa:2/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
          Paquetes RX:55 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:8 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colaTX:1000
          Bytes RX:9462 (9.4 KB)  TX bytes:888 (888.0 B)

oot@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route
abla de rutas IP del núcleo
estino         Pasarela        Genmask         Indic Métric Ref    Uso Interfaz
oot@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf#
```

root@ROUTER-R: /tmp/pycore.60790/ROUTER-R.conf

Archivo Editar Ver Buscar Terminal Ayuda

```
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route add -net 201.2.3.28/30 gw 2
01.2.3.1
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route add default  gw 201.2.3.1
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route add -net 201.2.3.24/30 gw 2
01.2.3.18
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf# route
Tabla de rutas IP del núcleo
Destino         Pasarela        Genmask         Indic Métric Ref    Uso Interfaz
default         201.2.3.1       0.0.0.0         UG    0      0        0 eth0
201.2.3.0       *               255.255.255.240 U     0      0        0 eth0
201.2.3.16      *               255.255.255.248 U     0      0        0 eth1
201.2.3.24      201.2.3.18      255.255.255.252 UG    0      0        0 eth1
201.2.3.28      201.2.3.1       255.255.255.252 UG    0      0        0 eth0
root@ROUTER-R:/tmp/pycore.60790/ROUTER-R.conf#
```

IP, ARP e ICMP (1)

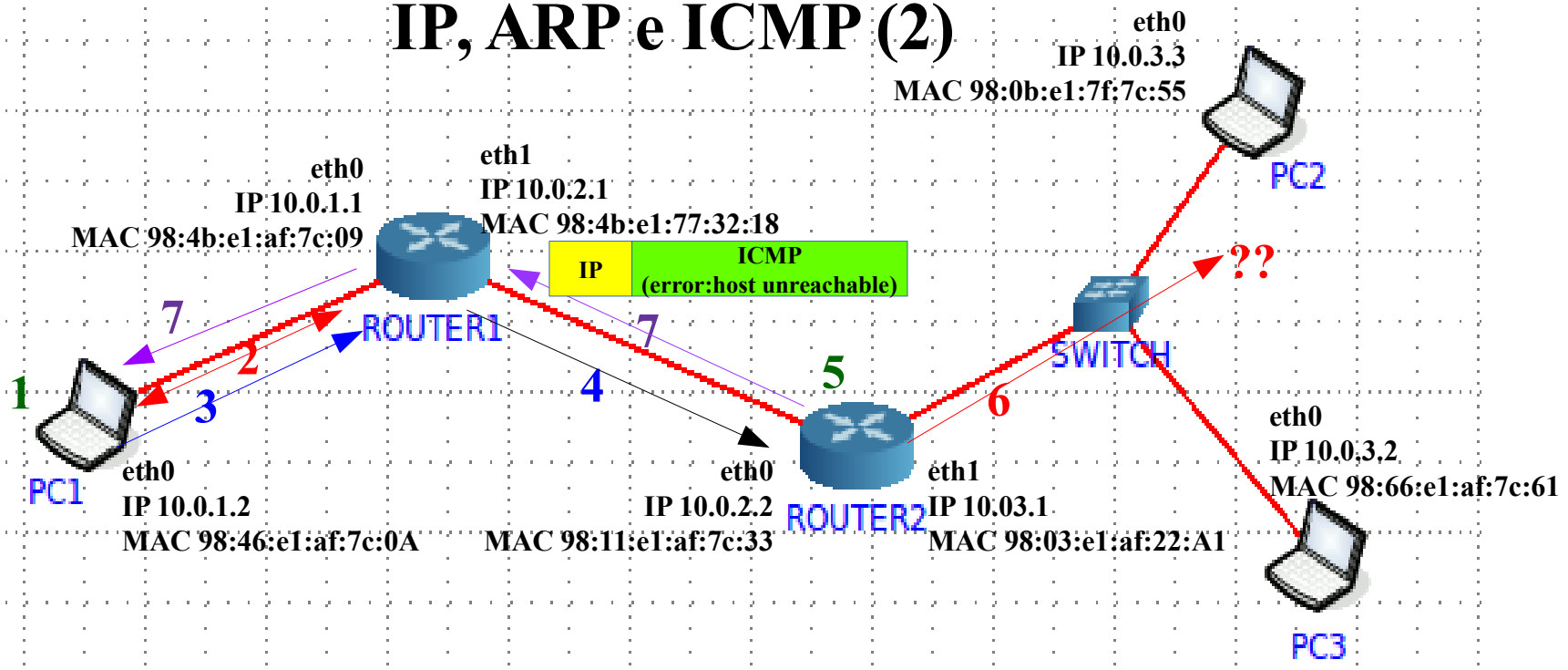
Funcion de IP:
transportar los datos desde el origen al destino (p.ej de 10.0.1.2 a 10.0.3.2)

Funcion de ARP:
Informar a IP acerca de la direccion Ethernet (MAC) de unn equipo en la red (mapping IP/Ethernet)

Funciones de ICMP:
a- Notificacion de errores o situaciones anormales. Por ejemplo, si ocurre un error en el envio de un datagram IP, ICMP lo informa al origen
b- Permite chequear conectividad entre dos equipos (uso de ping, que involucra echo request y echo reply)
c- Configuracion de equipos (por ejemmplo, posibilita a un host aprender rutas)

Encapsulacion

IP, ARP e ICMP (2)



IP en PC1 envia un datagram a la direccion 10.0.3.4 (no existente)

1- Consulta tabla de reenvio: proximo router a la red 10.0.3.0/24 es 10.0.0.1. Para hacerlo encapsula el datagram en un frame ethernet.

2- Para saber cual es la MAC del equipo que tiene IP = 10.0.0.1 usa ARP (request y reply)

3- Envia a ROUTER1

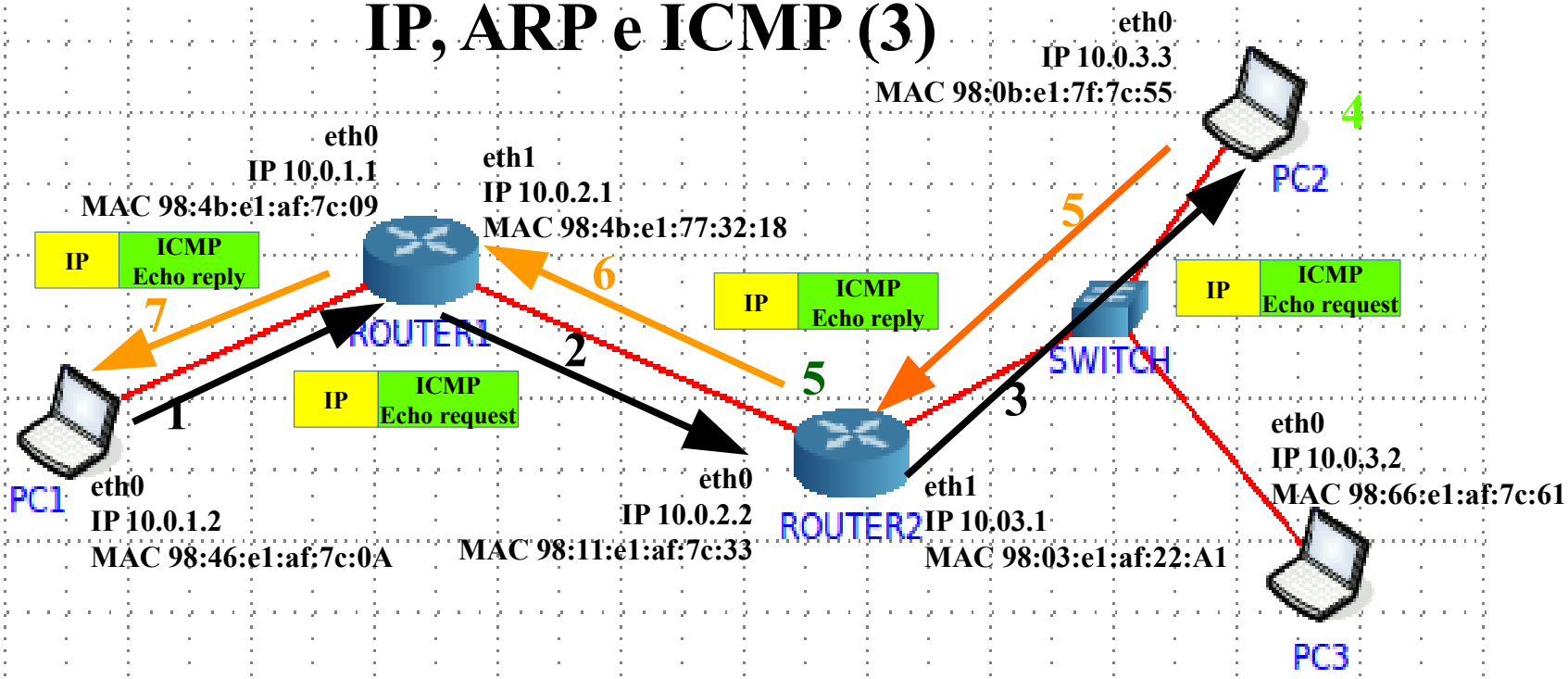
4- ROUTER1 realiza el mismo proceso (1 y 2) y envia a ROUTER2

5- ROUTER2 consulta su tabla de reenvio, la red 10:0:3.0/24 es local, debe enviar a la direccion 10.0.3.4

6- Para saber cual es la MAC del equipo que tiene IP = 10.0.3.4 usa ARP (request y reply). Nadie contesta

7- ROUTER2 envia a PC1 (10.0.1.2) un datagram IP que contiene un ICMP: host no encontrado

IP, ARP e ICMP (3)



IP en PC1 hace ping a PC2: ping 10.0.3.3

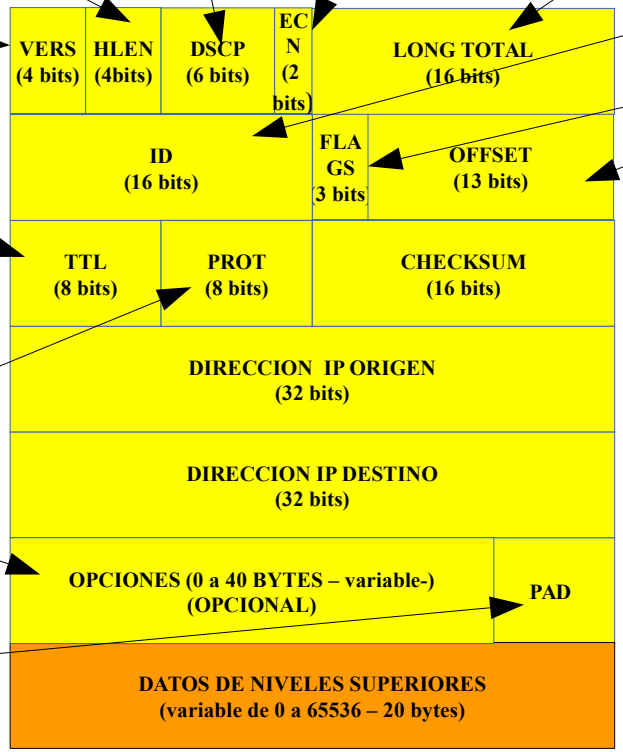
(solo se grafica el envio de los datagrams IP con los ICMP encapsulados)

1,2,3- PC1 envia ping (ICMP echo request) a PC2

4- PC2 recibe el echo request y contesta con echo reply

5,6,7- El ICMP echo replay es transportado (encapsulado en IP) al host PC1

Formato de frame IP



Longitud del header en multiplos de 4 bytes

Indica a los routers como tratar el datagram

Indicacion de router congestionado

Longitud total en bytes del datagram (hasta 65536)

Version: 4

Fragmentacion del datagram
ID: identificacion unica del datagram
FLAGS: control de fragmentacion
OFFSET: posicion relativa de los datos de este datagam

Cantidad maxima de routers que puede recorrer el datagram

Protocolo encapsulado en el datagram (TCP, UDP)

Opciones, longitud variable, hasta 40 bytes

Relleno para completar campo de opciones a multiplo de 4 bytes

HEADER FIJO (20 bytes)

OPCIONES HEADER (hasta 40 bytes)

DATAGRAM (hasta 65536 byres)

HEADER
- longitud variable 20 a 60 bytes
- procesado en los routers
- byte network order

DATOS
- procesado por niveles superiores
- lingitud variable

DSCP: Differentiated Services Code Point
ECN: Explicit Congestion Notification
TTL: Time to live

Fragmentación IP

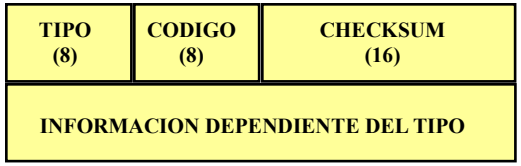
- **MTU (Maximum transmission Unit): Diferente para cada subred**
- **Fragmentación: en un router tal que MTU de salida < long. dg**
- **Puede refragmentarse un fragmento**
- **Reensamblado en el destino**

- **Fragmentación:**
  - **En cada fragmento se copia header, cambian campos Flag, Long total y fragment offset**
  - **Se copian o no las opciones, dependiendo del bit de copia**

- **Reensamblado:**
  - **Se reconoce los fragmentos por el campo identificación**
  - **Se reconoce el fin del datagram por el bit de flag de último fragmento**
  - **Se rearma el dg en base a los fragment offsets**
  - **Timer para reensamblado**

ICMP (Internet Control Message protocol)

- **Internet Control Message Protocol (RFC 792, Sep 1981)**
- **Objeto: Nivel IP de un router o host informa a nivel IP del origen de un datagram acerca de problemas**
- **Encapsulado en IP (Protocolo = 1), parte necesaria de toda implementación IP**
- **No se generan paquetes ICMP sobre**
  - **condiciones de error producidas por ICMP**
  - **datagrams multicast o broadcast**
  - **fragmentos de datagrams IP, excepto el primero**
- **Formato general**

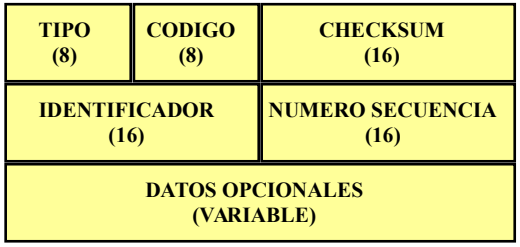


Tipo: Identifica el tipo de mensaje
Codig": Información adicional, dependiendo del tipo
Checksum: sólo para ICMP, calculado como en IP

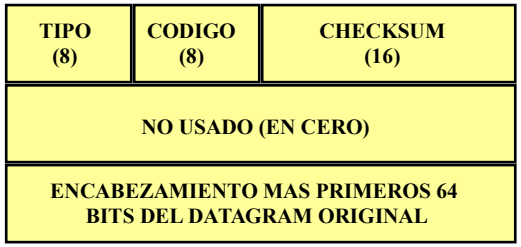
TIPOS
0: Echo reply
3: Destination Unreachable
4: Source Quench
5: Redirect
8: Echo Request
11:Time Exceeded
12:Parameter Problem
13:Timestamp Request
14:Timestamp Reply
15:Information Request (obsoleto)
16:Information Reply (obsoleto)
17:Address Mask Request
18:Address Mask Reply

ICMP

- **Echo Request y Reply (tipos 8 y 0)**
  - **Un equipo envía un request; el destino envía el reply con los mismos datos**
  - **Se comprueba el funcionamiento del destino y nodos intermedios**
  - **Campos Identificador y secuencia, para uso del emisor, se devuelven iguales**
  - **Codigo = 0**
  - **Utilizado por ping**



- **Destination Unreachable (tipo 3)**
  - **Enviado por un router o host (destino) que no puede entregar el dg.**
    - **0: Network Unreachable**
    - **1: Host Unreachable**
    - **2: Protocol Unreachable**
    - **3: Port Unreachable**
    - **4: Fragmantación y DF set**
    - **5: Falla en Source Route**



- **Time Exceeded (tipo 11)**
  - **Enviado por un router si TTL llega a 0**
  - **Códigos**
    - **0: TTL excedido (0)**
    - **1: Tiempo de reensamblado (fragmentación) excedido**

ARP: Address Resolution Protocol

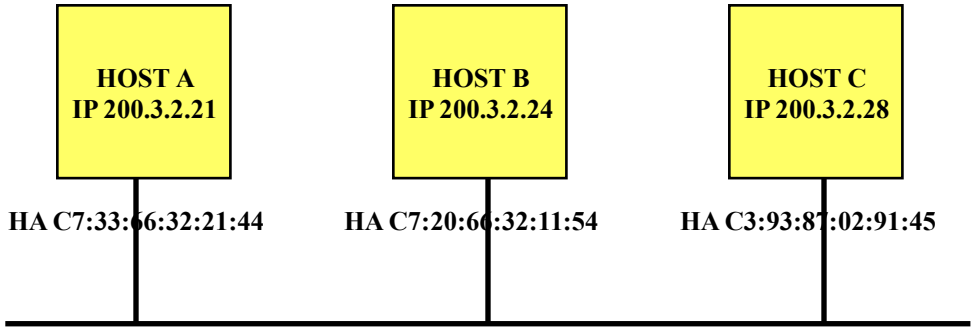
- **Surge como necesidad en las primeras redes Ethernet (broadcast múltiple acceso) (RFC 826, Nov. 1982)**

- **Una maquina conoce:**
  - **Su dirección IP (por ejemplo almacenada en disco)**
  - **Su dirección de hardware (grabada en la placa Ethernet)**
  - **La dirección IP del equipo a quien desea enviar un datagram**

  - **Para poder enviar el datagram, debe encapsularlo en un frame Ethernet, es decir, necesita conocer la direccion de hardware del equipo destino**
  - **Lo resuelve a través del protocolo ARP**

- **Características del medioambiente para ARP:**
  - **Direcciones de subred en hardware, de mayor longitud que las direcciones IP**
  - **Redes dinámicas, los equipos se conectan y desconectan, y cambian sus direcciones (placas) de subred**

ARP: Address Resolution Protocol

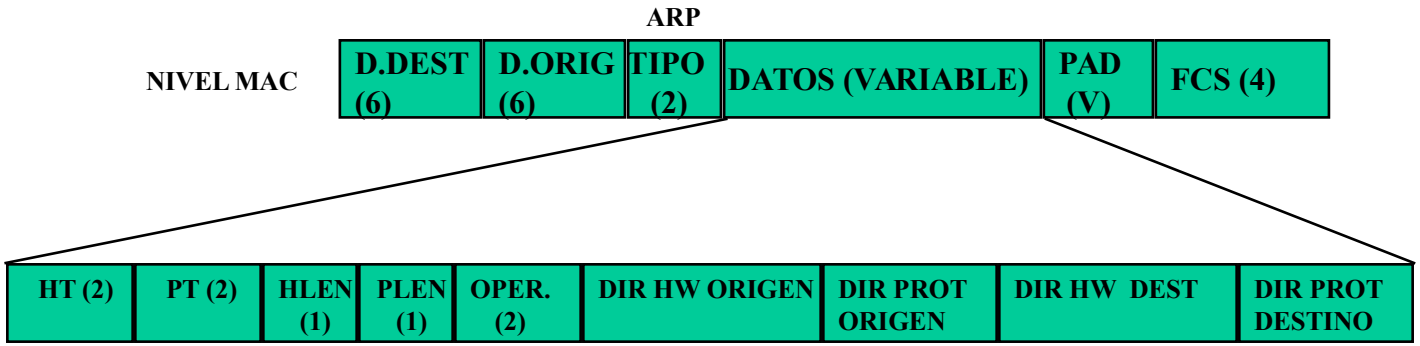


Operación del ARP

Host A (200.3.2.21) desea enviar datagram a Host B (200.3.2.24)
No conoce la dirección Ethernet de B

- **A (su ARP) envía un frame Ethernet, broadcast, con un frame ARP encapsulado preguntando: Que dirección Ethernet tiene el equipo cuya dir IP es 200.3.2.21 (ARP REQUEST)**
- **B (el equipo que reconoce su dir IP) responde(su ARP) enviando su dir Ethernet (ARP REPLY)**
- **A coloca el mapping 200.3.2.24 / C7:20:66:32:11:54 en su tabla ARP**
- **A está en condiciones de enviar el dg a B**

Formato de frame ARP



HT: Tipo de hardware de la subred (Ethernet = 1)
PT: Tipo de protocolo (IP = 0800 hex)
HLEN: Longitud de la dirección de subred
PLEN: Longitud de la dirección del protocolo
OPER: Tipo de PDU
- **ARP REQUEST**
- **ARP RESPONSE**
- **RARP REQUEST**
- **RARP RESPONSE**

DIR ORIGEN: Las del equipo que origina el request
DIR DESTINO: Las del equipo que contesta c/reply

El reply será enviado a la dirección Ethernet origen que va en el ARP, no a la dirección origen MAC

Frames request y replay intercambiados cuando equipo 1 hace un request a equipo 2

Formato:
<MAC dest, MAC orig,oper,H orig, P orig, H dest, P dest>

Request:
<M Broad, M1, request, H1, IP 1, ??, IP 2>

Reply
<H1, H2, reply, H1, IP1, H2, IP2>

ARP

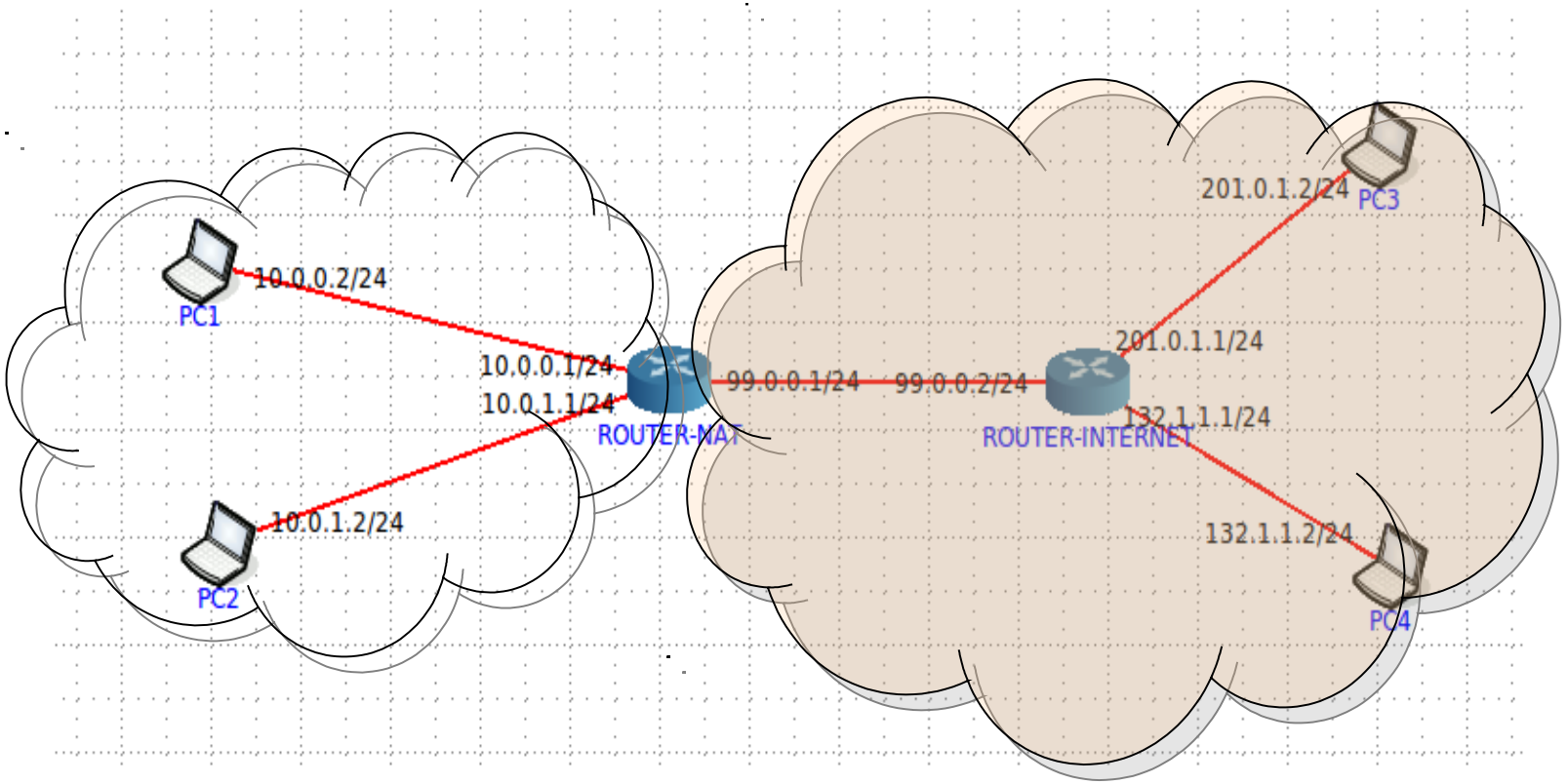
- **Mejoras**
  - **Los ARP en cada equipo almacenan una tabla dinámica con los mappings**
  - **Las entradas se eliminan luego de un cierto tiempo**
  - **Se utilizan los ARP request (broadcast) para generar entradas en las tablas para las direcciones origen**
  - **Gratuitous ARP (ARP preguntando por su propia dirección)**
    - **Permite determinar direcciones IP duplicadas**
    - **Permite que los hosts actualicen sus tablas ARP**
  - **Control de envío de ARP requests**
  -
- **Comandos ARP**
  - **arp**
  - **ip neigh**

NAT (Network address translation)

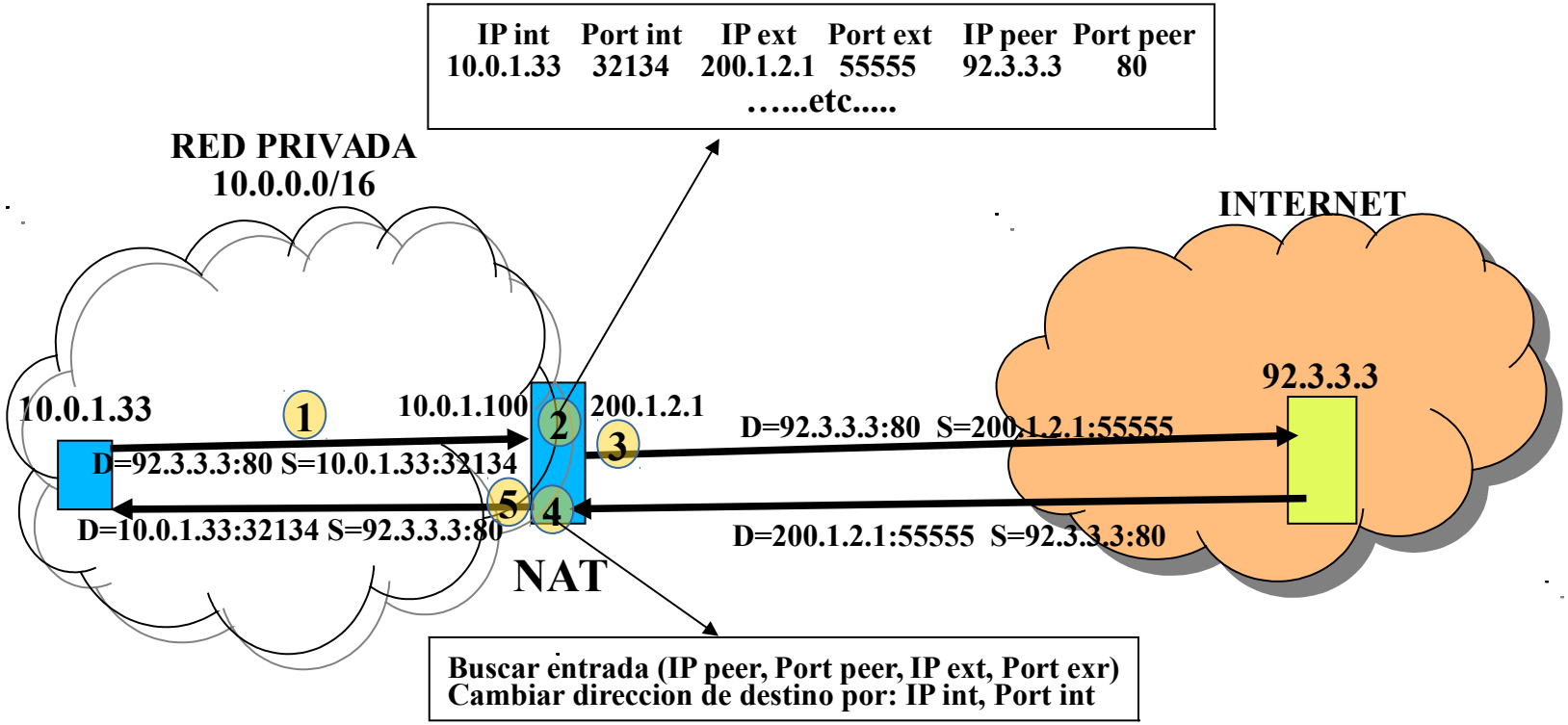
La función NAT se sitúa en el router de salida de una intranet hacia la Internet

Consiste en cambiar las direcciones (y ports) de los paquetes que salen o entran en la intranet

Surge debido a la escasez de direcciones IPv4



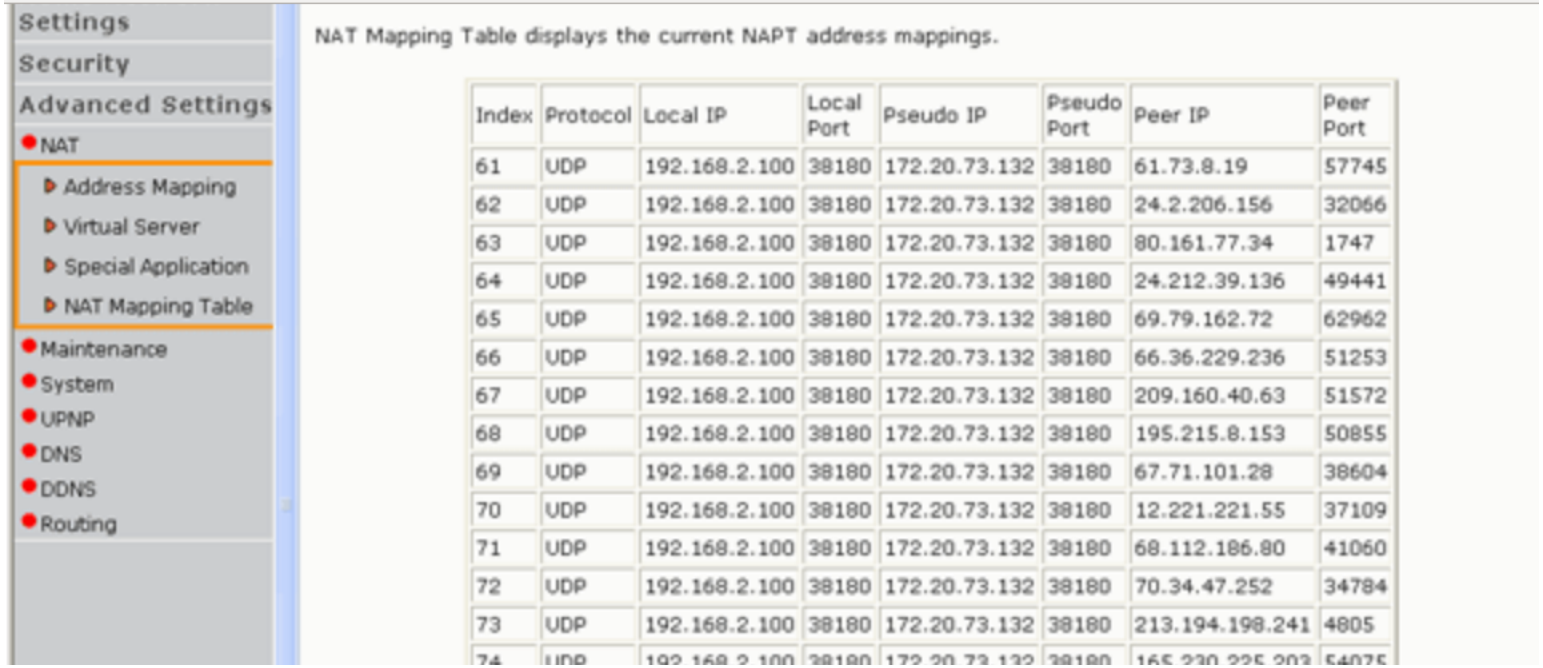
Funcionamiento simplificado de un NAT



Caso particular de NAT, se reemplaza direccion y port en el envio y en el arribo se exige que coincidan todos los campos
Si enviamos un segundo paquete al mismo destino y con el mismo port, no entra en la tabla de mapping (solo modifica)

1- Envio del host interno al externo
2- Creacion de una entrada enla tabla interna de mapping, selección de nueva IP y (en ciertos casos) nuevo port de origen
3-Envio del paquete modificado
4-Busqueda de entrada existente en la tabla, para filtrar y realizar conversion de direccion de destino
5-Envio del paquete modificado

Ejemplo de tabla de mapping en un NAT



Index	Protocol	Local IP	Local Port	Pseudo IP	Pseudo Port	Peer IP	Peer Port
61	UDP	192.168.2.100	38180	172.20.73.132	38180	61.73.8.19	57745
62	UDP	192.168.2.100	38180	172.20.73.132	38180	24.2.206.156	32066
63	UDP	192.168.2.100	38180	172.20.73.132	38180	80.161.77.34	1747
64	UDP	192.168.2.100	38180	172.20.73.132	38180	24.212.39.136	49441
65	UDP	192.168.2.100	38180	172.20.73.132	38180	69.79.162.72	62962
66	UDP	192.168.2.100	38180	172.20.73.132	38180	66.36.229.236	51253
67	UDP	192.168.2.100	38180	172.20.73.132	38180	209.160.40.63	51572
68	UDP	192.168.2.100	38180	172.20.73.132	38180	195.215.8.153	50855
69	UDP	192.168.2.100	38180	172.20.73.132	38180	67.71.101.28	38604
70	UDP	192.168.2.100	38180	172.20.73.132	38180	12.221.221.55	37109
71	UDP	192.168.2.100	38180	172.20.73.132	38180	68.112.186.80	41060
72	UDP	192.168.2.100	38180	172.20.73.132	38180	70.34.47.252	34784
73	UDP	192.168.2.100	38180	172.20.73.132	38180	213.194.198.241	4805
74	UDP	192.168.2.100	38180	172.20.73.132	38180	165.230.225.203	54075

Ventajas y desventajas de NAT

Desventajas

- **Necesidad de Application Level Gateways (ALG)**
  - **Especificos para cada protocolo**
  - **Aplicaciones que transportan direcciones IP (SNMP)**
  - **Aplicaciones con conexiones interrelacionadas (p.ej FTP)**
- **Performance**
  - **Recalculo de checksums si modifica direcciones**
  - **Recalculo de TCP checksum si se modifican ports**
  - **Controles adicionales si se fragmentan paquetes**
- **Perdida de visibilidad de los equipos internos**
- **Preservacion de ports (p.ej. port par-impar)**
- **Consideraciones sobre NATs sucesivos**
- **Incompatibilidades con IPSec**

Ventajas

- **Soluciona temporariamente la escacez de direcciones IPv4**
- **Elimina los costos de cambio de direccion de red**
- **Mejora la privacidad de la red**

NAT en iptables

- **Tabla “nat” de iptables**
- **Sólo consultada para el primer paquete que define una conexión**
- **Targets:**

DNAT

válido en PREROUTING y OUTPUT
Cambia dirección y/o port de destino

SNAT

Válido en POSTROUTING
Cambio dirección y/o port de origen

MASQUERADING

Válido en POSTROUTING
Cambio dirección y/o port de origen
Sólo recomendable para asignación dinámica de IP