

Capa de Red y Protocolo IP

Mg. Gabriel H. Tolosa

tolosoft@unlu.edu.ar

“A name indicates what we seek. An address indicates where it is. A route indicates how we get there”.

Jon Postel

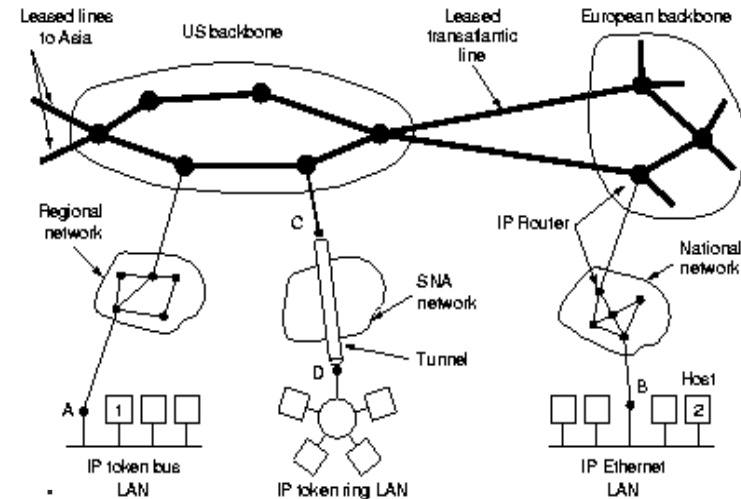
Capa de Red

Motivación

Diferentes tecnologías de redes LAN y WAN, las cuales:

- Diferentes medios
- Diferentes métodos de acceso
- Esquemas de direccionamiento propios
- Formato de trama propio (y MTU)
- Los protocolos definen temporizadores, utilizan técnicas de detección de errores y control de flujo y diferentes tipos de servicio (c/conexión, s/conexión)

Ninguna tecnología puede satisfacer todas las necesidades. La heterogeneidad es inevitable!!!



[Capa de Red]

- “Aunque es altamente deseable un servicio universal, las incompatibilidades entre el hardware de red y el direccionamiento físico impiden a una organización construir una red con vinculada con puentes que incluya tecnologías arbitrarias”

Douglas Comer

Capa de Red

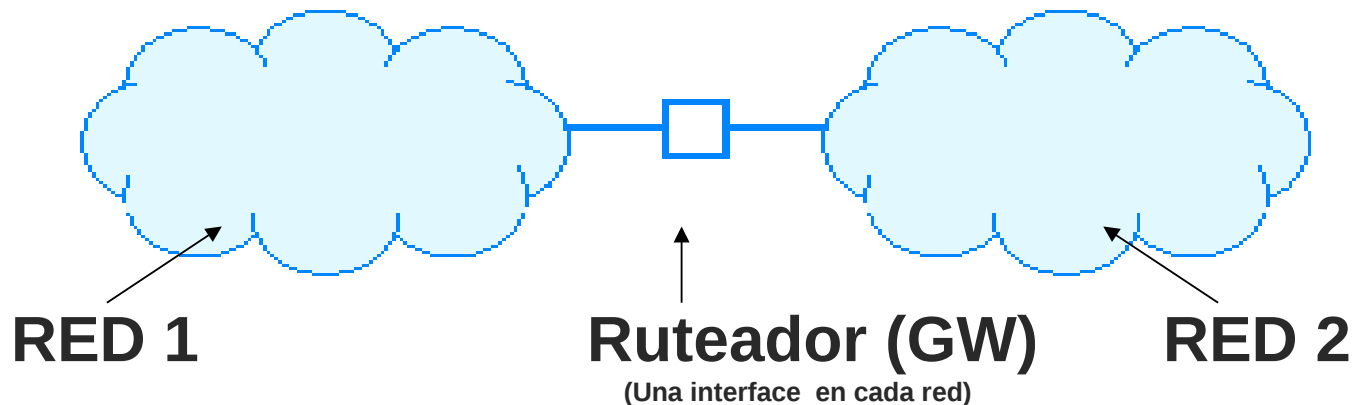
■ Funciones

- Proveer una abstracción de la tecnología subyacente, ocultando la heterogeneidad
- Crear una red “virtual” con:
 - Esquema de direccionamiento propio
 - Esquema de nombres (si hay) propio
- Pasar datos de una red a otra (encaminamiento o ruteo)
- Mantener información de estado y contabilidad (accounting)
- Gestionar las congestiones

Capa de Red – Internetworking

■ Interredes o Internets (Internetworks):

“Conjunto de redes heterogéneas conectadas mediante sistemas intermedios (ruteadores/gateways)”



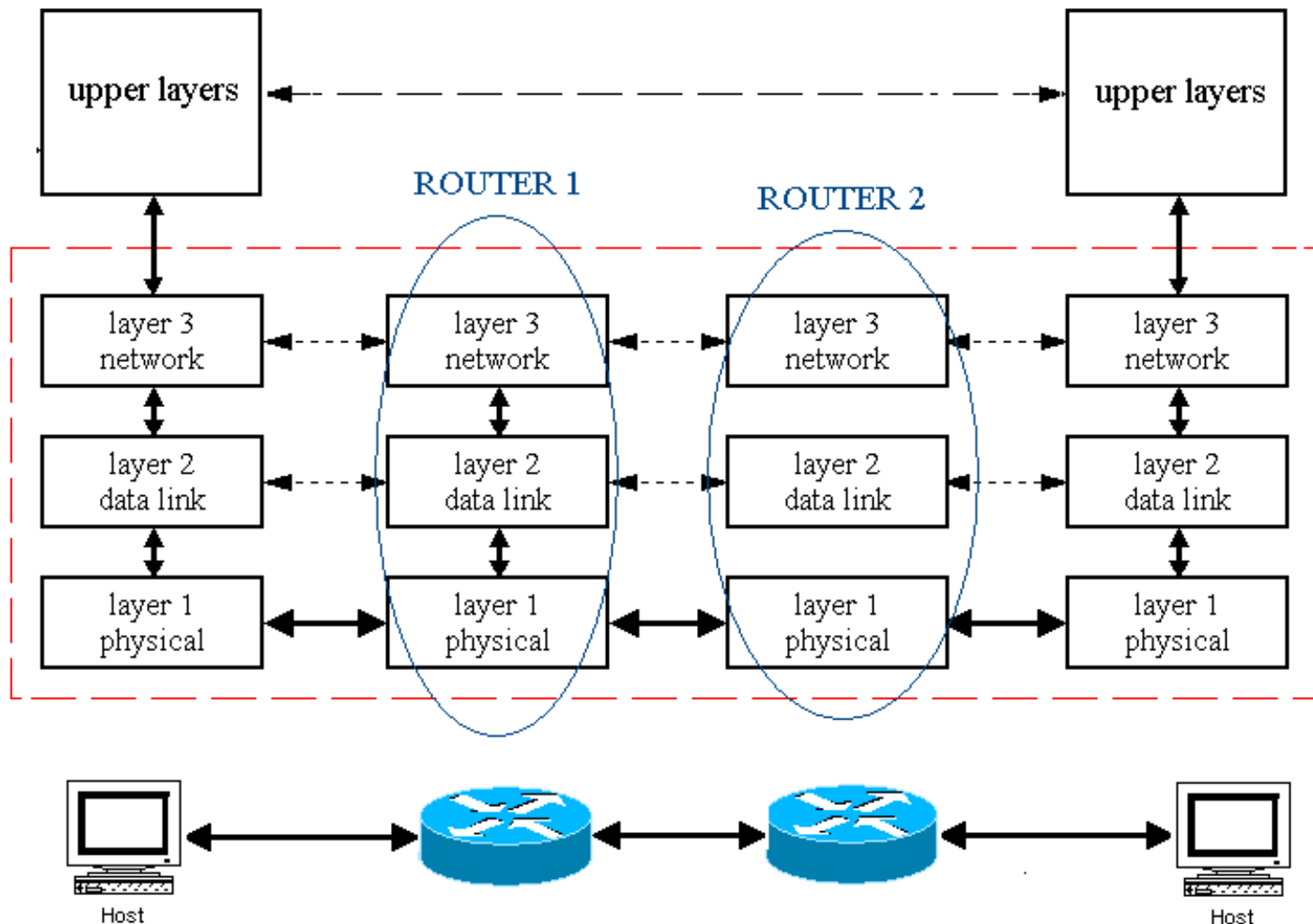
- Un ruteador puede interconectar redes con diferentes tecnologías de enlace, a través de la capa de red:
 - Se implementa por software
 - El sistema resultante aparece (y se comporta) como homogéneo.
 - Caso paradigmático: **Internet** (definida por TCP/IP)

Capa de Red – Internetworking

- **Objetivo:** Construir un sistema de comunicaciones que:
 - Oculte la heterogeneidad (a las capas superiores, y especialmente al usuario)
 - Aparezca como único (sin “costuras”)
 - De propósito general (independiente de las aplicaciones)
 - De alcance global
- **Solución:** Crear una red “virtual” que:
 - Defina un esquema de direccionamiento global
 - Defina un esquema de nombres
 - Se implemente mediante software (en hosts y en ruteadores)

Capa de Red

En el Modelo OSI



Capa de Red – Protocolo IP

- Protocolo de capa red de la pila TCP/IP (RFC 791)

“Define una única red virtual sobre las diferentes clases de plataformas de comunicaciones subyacentes”

- Especifica
 - Esquema de direccionamiento
 - Formato de mensaje
 - Mecanismo de encaminamiento

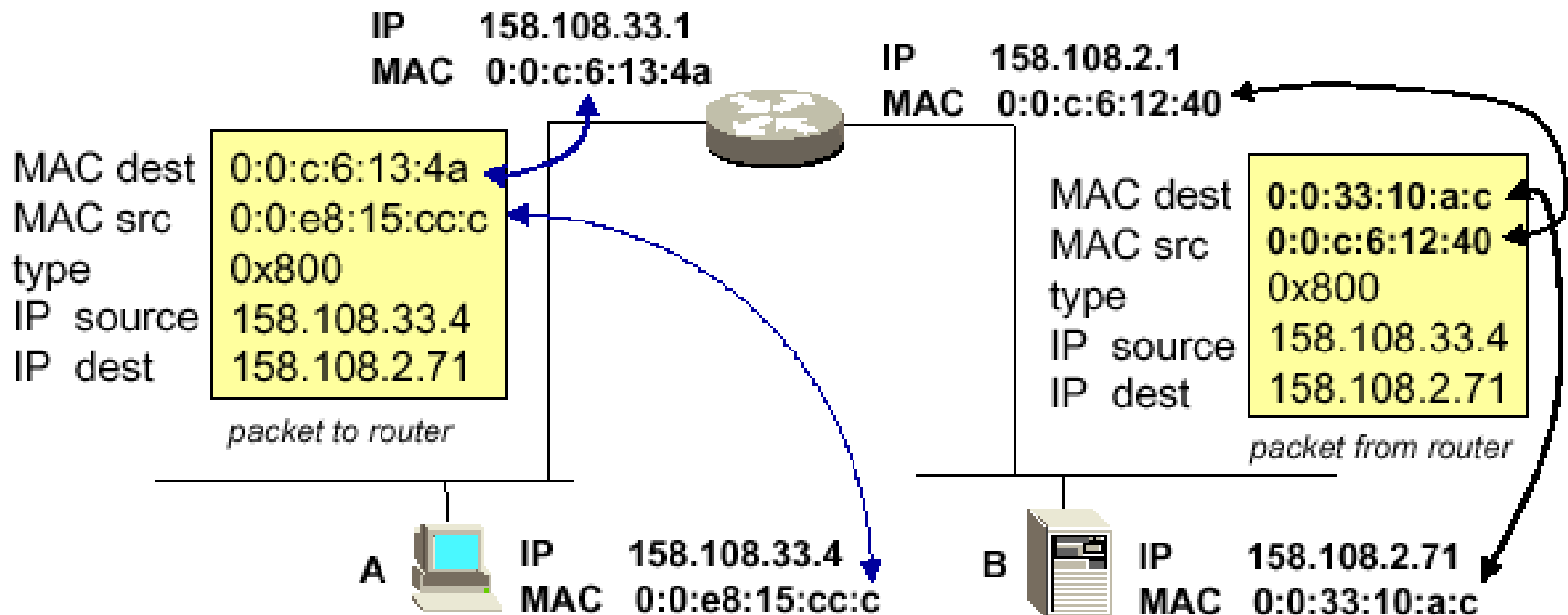
Capa de Red – Protocolo IP

- Basado en datagramas (no orientado a la conexión)
- Entrega basada en el “mejor esfuerzo” (servicio no confiable)
- Los datagramas se pueden perder
- Los datagramas son entregados fuera de orden
- Pueden entregarse duplicados
- Los datagramas se pueden demorar en la red
- Permite fragmentación (y ensamblado)
- Control de errores
 - Solo Checksum de la cabecera
 - Mensajes de error => ICMP
- End-to-end Argument
- Control de flujo => No hay
- Encaminamiento
 - Tablas de rutas en sistemas finales y sistemas intermedios
- Source routing y record routing

Protocolo IP

Encapsulamiento en PDU de Enlace

- IP will reframe the packet when A send data to B



Change MAC address, IP address be the same

Protocolo IP – Direccionamiento

- **Conceptos (Ideas)**
- **Nombre (Name)**
 - Especifica cómo se llama algo/alguien (Identificador).
 - Independiente del origen y destino.
- **Dirección (Address)**
 - Especifica dónde se puede ubicar algo/alguien.
 - Es independiente del origen.
- **Ruta (Route)**
 - Especifica cómo ubicar algo/alguien.
 - Es dependiente del origen y del destino.

Protocolo IP – Direccionamiento

- **Niveles de Direccionamiento**
- **Nombre de Host (FQDN)**
 - www.unlu.edu.ar
 - pollito.tyr.unlu.edu.ar
- **Dirección IP**
 - 170.120. 44.12
 - 168.10. 44.240
- **Dirección física (de NIC)**
 - MAC Ethernet: 00:C4:06:13:4C:33



Protocolo IP – Direccionamiento

- **Direcciones IP (Versión 4)**
- **Direcciones de 32 bits (4 bytes)**
 - notadas con decimales separados por puntos,
 - por ej. 198.107.2.25

198	107	2	25
-----	-----	---	----

8 bits . 8 bits . 8 bits . 8 bits

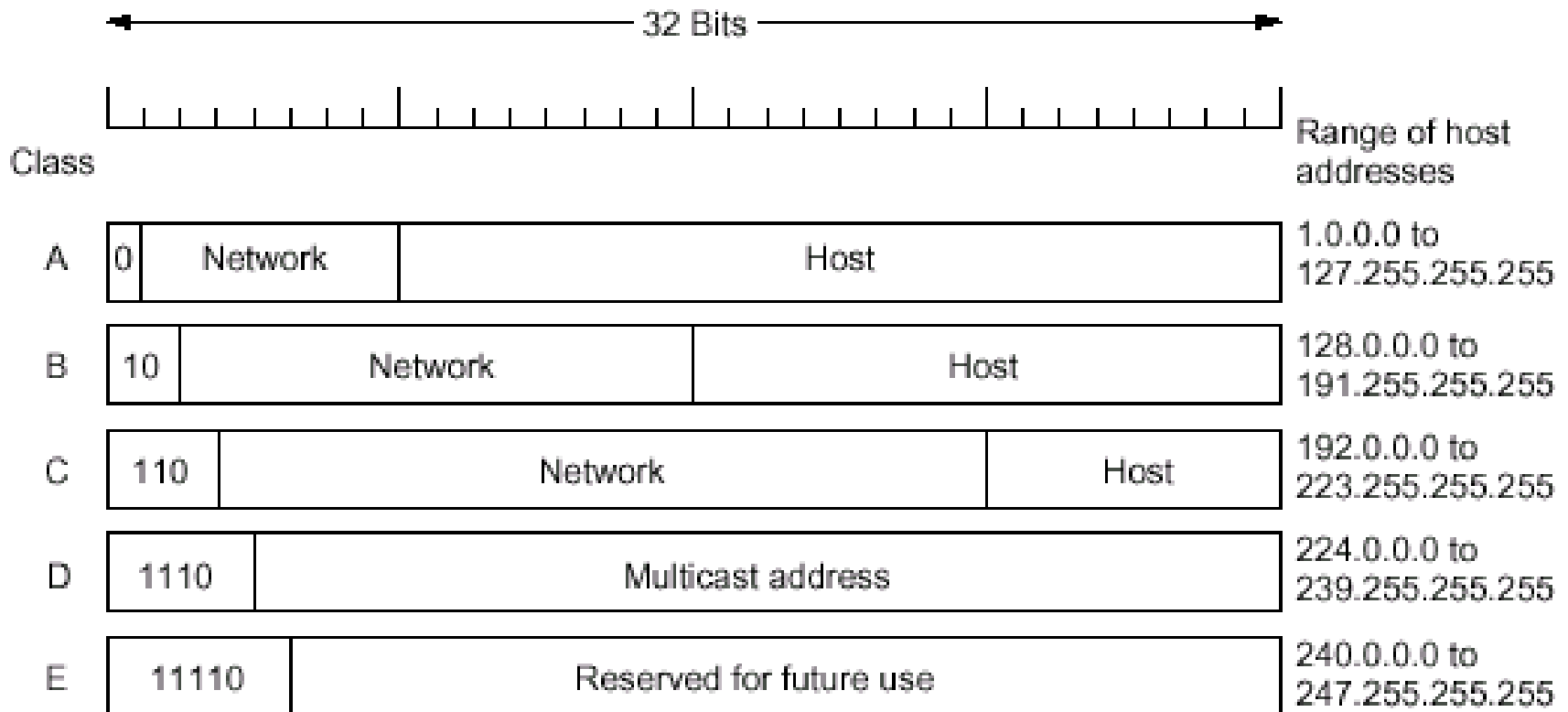
- **Dividida en dos partes**
 - **identificador** (dirección) de **red** (Información usada para ruteo)
 - **identificador** (dirección) de **host** (nodo específico dentro de una red)



← 32 bits →

Protocolo IP – Direccionamiento

■ Clases de Direcciones



Protocolo IP – Direccionamiento

■ Espacio de Direcciones

Clase	Bits Iniciales	Bits Red	Bits Hosts	Espacio de Dir.	
A	0	7	24	2^{24}	16777216
B	10	14	16	2^{16}	65536
C	110	21	8	2^8	256
D	1110	28	-		
E	11110	27	-		

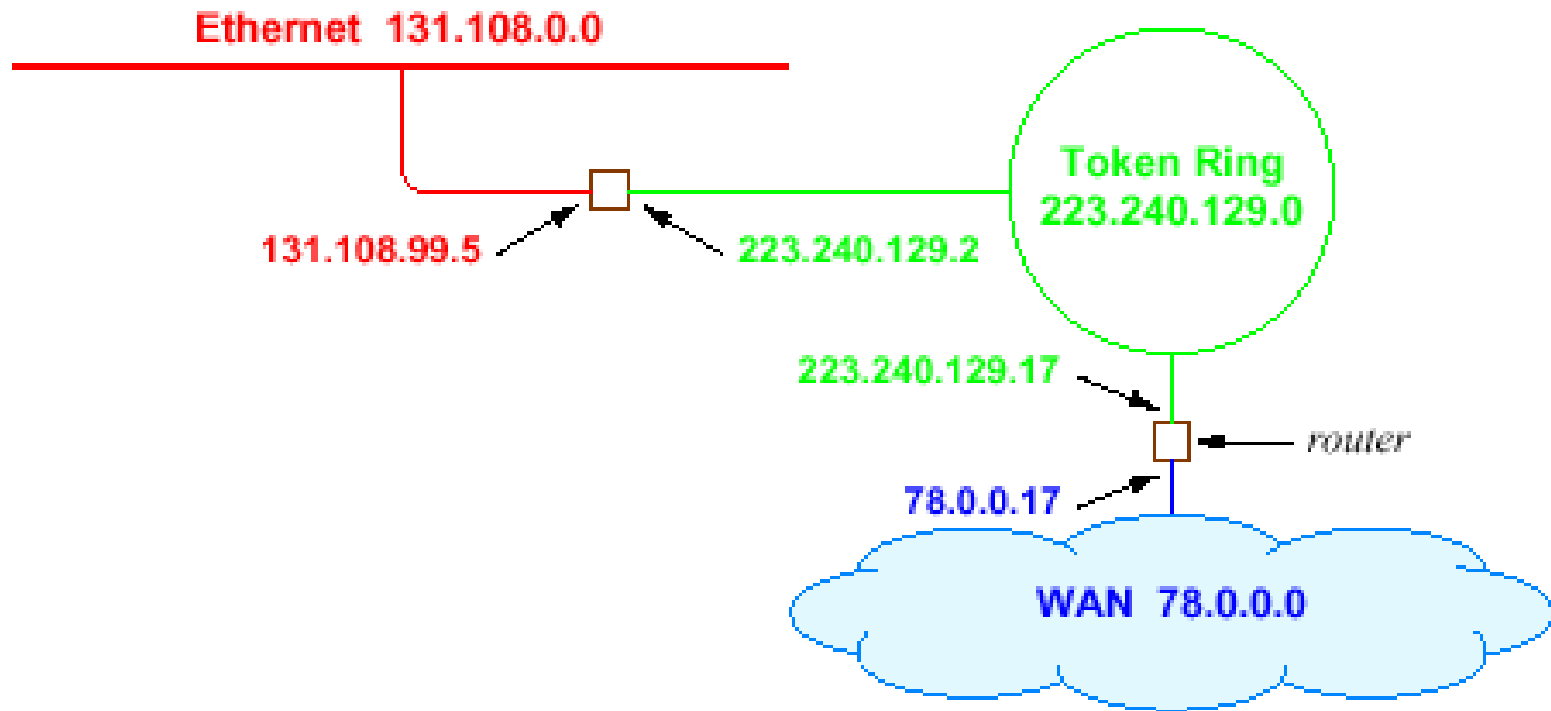
Protocolo IP – Direccionamiento

■ Direcciones Especiales

0 0		Este host de la red
0 0 ... 0 0	Host	Un host de esta red
1 1		Broadcast de red local
Network	1 1 1 1 ... 1 1 1 1	Broadcast red remota
127	(Anything)	Loopback (127.0.0.1)

Protocolo IP – Direccionamiento

■ Ejemplo

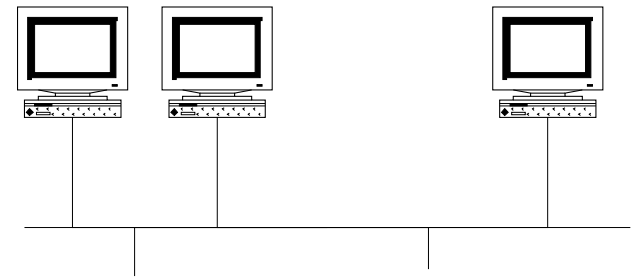


Protocolo IP – Direccionamiento

■ Algunos Problemas

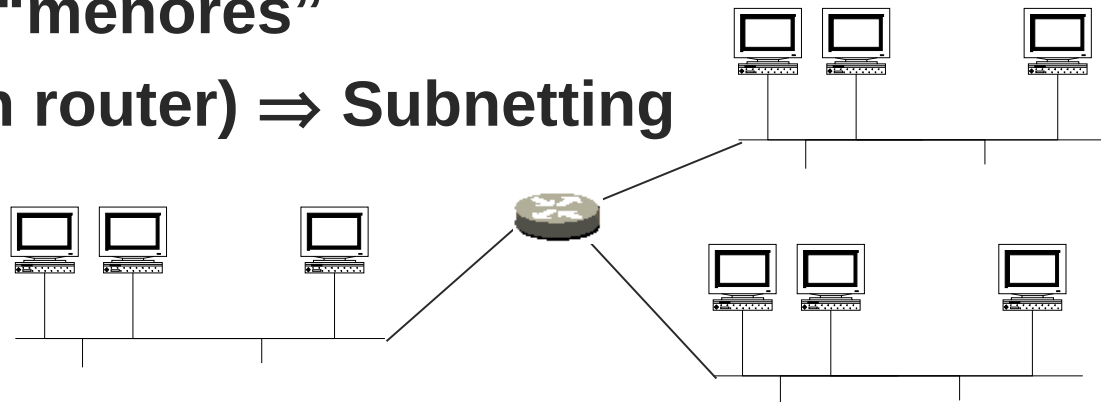
- Redes (por ej. clase B “plana”) con un número muy grande de hosts

- Administración?
- Performance?
- Espacio de direcciones limitado



■ División en redes “menores”

- (vinculadas por un router) ⇒ Subnetting



Protocolo IP – Direccionamiento

■ Subnetting

- Procedimiento descrito en el RFC 950
- Permite dividir una red clase A, B ó C en subredes

■ Beneficios

- Permite una mejor distribución de direcciones ante la creciente demanda
- Facilita el control del espacio de direcciones
- Permite ocultar la estructura interna de la red
- Reduce las tablas de rutas en routers

Protocolo IP – Subnetting

- **Máscara de Subred**
- **Números de 32 bits** (notados en decimal separados por puntos)
 - indican qué parte de la dirección completa corresponde a red y (subred) y qué parte a host.
 - **Regla:** “Los bits en 1 cubren la porción de la dirección correspondiente a red y subred”

		red	subred	host
Dirección IP	130.5.5.25	10000010.00000101	.00000101	.00011001
Máscara	255.255.255.0	11111111.11111111	.11111111	.00000000

- Una máscara de subred 255.255.255.0 para una dirección clase B la divide en 254 subredes 130.5.1.x a 130.5.244.x

Protocolo IP – Subnetting

■ Ejemplos

■ Para una dirección clase B y máscara:

255.255.0.0 (00000000. 00000000)	⇒ 0 subredes con 65534 hosts
255.255.192.0 (11 000000. 00000000)	⇒ 2 subredes con 16382 hosts
255.255.255.0 (11111111 . 00000000)	⇒ 254 subredes con 254 hosts
255.255.255.252 (11111111.11111100)	⇒ 16382 subredes con 2 hosts

■ Para una dirección clase C y máscara:

255.255.255.0 (00000000)	⇒ 0 subredes con 254 hosts
255.255.255.224 (111 00000)	⇒ 6 subredes con 30 hosts
255.255.255.240 (1111 0000)	⇒ 14 subredes con 14 hosts

Protocolo IP – Subnetting

■ Ejemplos de Interpretación

Dirección IP

158.108.2.71

Máscara

255.255.255.0

⇒ Host 71 de la subred
158.108.2.0

130.122.34.3

255.255.255.192

⇒ Host 3 de la subred
130.122.34.0

130.122.34.132

255.255.255.192

⇒ Host 4 de la subred
130.122.34.128

15.20.15.2

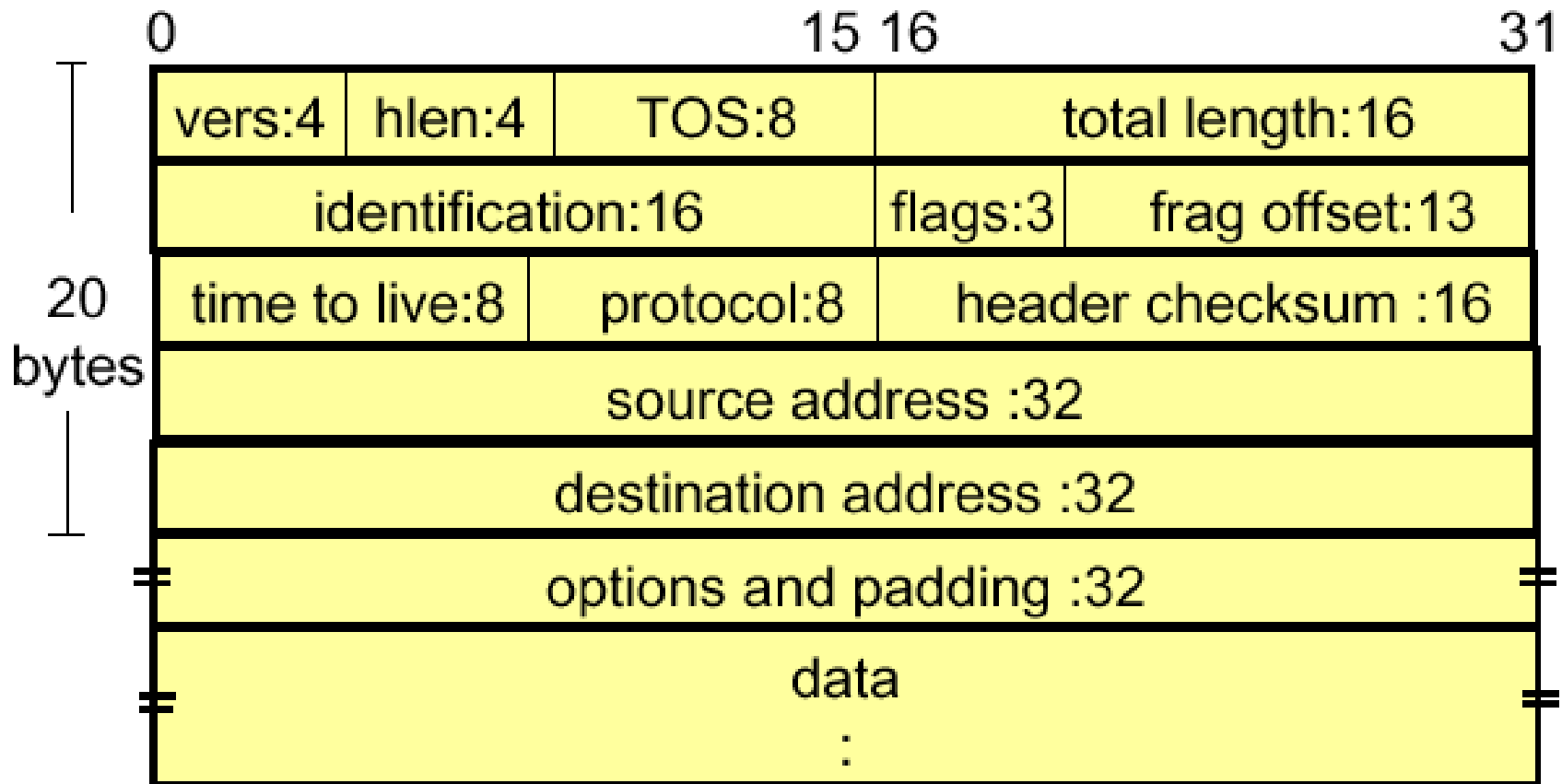
255.255.0.0

⇒ Host 15.2 de la subred
15.20.0.0

Protocolo IP – Subnetting

- **Tipos de Subnetting**
- **Estático**
 - Todas las subredes generadas utilizan la misma máscara de subred
 - Simple de implementar y mantener
 - No aprovecha el espacio de direcciones de la manera más óptima
- **De Máscara Variable (Variable Length Subnet Mask)**
 - Cada subred puede utilizar una máscara diferente
 - Un poco más complejo de implementar y mantener
 - Aprovecha el espacio de direcciones de la manera más óptima

Protocolo IP – Datagrama IPv4



Protocolo IP – Datagrama IPv4

- **Hlen**: Longitud de la cabecera, palabras de 32 bits (hlen = 5)
- **TOS**: Tipo de Servicio
- **Total Length**: Longitud total del datagrama (max. 64 kb, incluyendo header)
- **Identification**: ID del datagrama (usado en fragmentación/ensambado). OJO!, No es un número de secuencia.
- **Flags**: Banderas (usadas en fragmentación/ensambado)
- **Frag.Offset**: Desplazamiento del fragmento (fragment/ensamblado)
- **Time to Live**: Cantidad máxima de saltos
- **Protocol**: Código de protocolo de nivel superior (TCP=6)
- **Header checksum**: Checksum de la cabecera
- **Options**: Opciones (Source routing, record route)

0	1	2	3	4	5	6	7
Prec.		D	T	R	0	0	
bits		if 0			if 1		
0-2		Precedence					
3		Normal delay		low delay			
4		Normal throughput		High throughput			
5		Normal Reliability		High reliability			
6-7		Reserved					

Protocolo IP

Fragmentación y Ensamblado

- Cada red define su MTU (Ethernet: 1500 bytes, FDDI: 4500 bytes)
- Estrategia => Fragmentar cuando sea necesario (Datagrama > MTU)
- Los fragmentos son datagramas completos
- Fragmentan los sistemas intermedios
- Ensambla el sistema final
- Re-fragmentación es posible
- Banderas

R

Reservada

DF

0 – Puedo fragmentar

1 – No puedo fragmentar

MF

0 – Último fragmento

1 – Hay más fragmentos

- Offset: Distancia del fragmento desde el comienzo del datagrama original, medida en unidades de octetos.

Protocolo IP

■ Fragmentación y Ensamblado

- **Ejemplo:** Datagrama de 1600 bytes (20 bytes header + 1400 bytes data). ID: 327

3 fragmentos

- Fragmento #1: 20 bytes header + 600 bytes data. ID: 327, Offset =0, MF=1
- Fragmento #2: 20 bytes header + 600 bytes data. ID: 327, Offset =75 (600/8), MF=1
- Fragmento #3: 20 bytes header + 600 bytes data. ID: 327, Offset =150 (1200/8), MF=0

Problemas

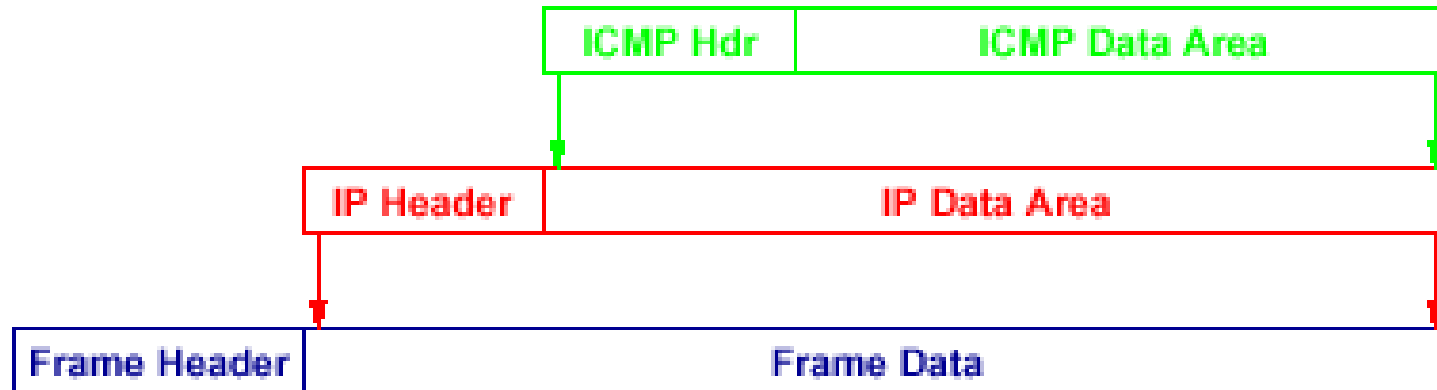
- El reensamblado es un proceso relativamente costoso
- El destino no sabe cuántos fragmentos forman el datagrama original (buffer por si es de tamaño máximo)
- Si se pierde un fragmento no es posible recuperarse del error (Se genera un mensaje ICMP)

Protocolo ICMP

- IP es un protocolo de “Mejor esfuerzo”, por lo tanto no se puede recuperar de situaciones de error.
- Puede detectar:
 - Bit corruptos
 - Direcciones ilegales
 - Loops en rutas
 - Pérdida de fragmentos
- Se auxilia del protocolo ICMP para reportar errores (Por ejemplo: ttl excedido, destino no alcanzable) e información de la red (ping, traceroute) al origen de un mensaje IP (sin embargo, NUNCA se generan mensajes ICMP por errores producidos por otros mensajes)

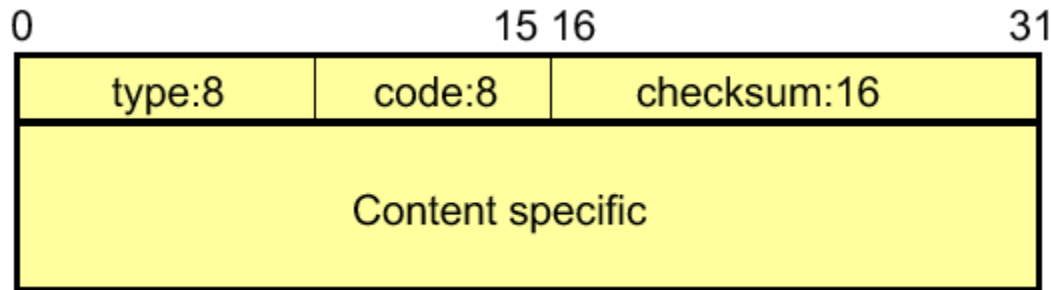
Protocolo ICMP

- Un mensaje ICMP se encapsula dentro de un datagrama IP.



Protocolo ICMP

■ Estructura de Datos



- type: Tipo de mensaje
- code: código de mensaje (información con más detalle)
- content: contenido específico
- Ejemplos:
 - type = 8, echo request y type = 0, echo reply (Utilizados en ping)
 - type = 3, destination unreachable (red o un host no se puede alcanzar)
 - type = 5, route change request (ICMP Redirect)
 - type = 11, Time Exceeded (TTL llegó a cero)

Protocolo ICMP

■ Ping

- Herramienta de depuración
- Información de RTT (Variación de la latencia entre origen y destino)
- Destino alcanzable (ruteable)
- La pila de protocolos es funcional
- Pérdida de paquetes
- Salida

```
tolosoft@s10:~$ ping www.google.com
PING www.google.akadns.net (64.233.171.104): 56 data bytes
64 bytes from 64.233.171.104: icmp_seq=0 ttl=236 time=205.0 ms
64 bytes from 64.233.171.104: icmp_seq=1 ttl=237 time=174.8 ms
64 bytes from 64.233.171.104: icmp_seq=2 ttl=238 time=201.7 ms
64 bytes from 64.233.171.104: icmp_seq=3 ttl=237 time=203.7 ms
64 bytes from 64.233.171.104: icmp_seq=4 ttl=239 time=196.7 ms
--- www.google.akadns.net ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 174.8/196.3/205.0 ms
```


Protocolo ICMP

■ Traceroute

- Determina la ruta activa a un destino. ¿Cómo? Utilizando mensajes ICMP.
- Se envía un mensaje a un puerto UDP no utilizado en destino con TTL = 1
- El primer router en la ruta decrementa el TTL, descarta el mensaje y retorna un mensaje ICMP Time Exceeded (con la info del router que lo descartó)
- Luego, se incrementa el TTL y se retransmite (un salto más)
- TTL++ hasta que el mensaje llega a destino
- El destino retorna un mensaje ICMP Service Unavailable
- Fin de la traza (para mejor comprensión, realice y analice una captura)

Protocolo ICMP

■ Traceroute

- Salida

```
tolosoft@s10:~$ traceroute www.google.com
```

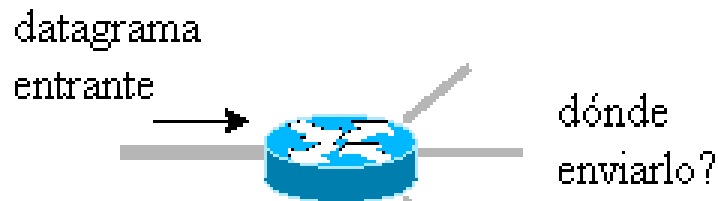
```
Traza a la dirección www.google.akadns.net [64.233.161.104] máx de 30 saltos:
```

```
 1  <10 ms  <10 ms  <10 ms  s10.ch.unlu.edu.ar [192.168.1.1]
 2      *    10 ms   10 ms   10.10.10.1
 3   10 ms   10 ms   10 ms   int-200-68-249-129.bellsouth.net.ar [200.68.249.129]
 4  421 ms   60 ms   *       172.30.249.221
 5   30 ms   30 ms   10 ms   172.30.67.253
 6  260 ms   171 ms  170 ms  at-5-0-0-3596.ipa1.Atlanta1.Level3.net [166.90.192.9]
 7  160 ms   170 ms  161 ms  ge-5-0-8.hsa2.Atlanta1.Level3.net [209.244.24.102]
 8  161 ms   160 ms  160 ms  ge-6-0-1.bbr1.Atlanta1.Level3.net [64.159.1.253]
 9  180 ms   *       *       as-2-0.bbr1.Washington1.Level3.net [64.159.1.2]
10  200 ms   231 ms  190 ms  ge-7-2.ipcolo2.Washington1.Level3.net [4.68.121.140]
11  170 ms   370 ms  171 ms  unknown.Level3.net [166.90.148.174]
12  170 ms   201 ms  170 ms  216.239.47.158
13  200 ms   180 ms  271 ms  216.239.49.214
14  281 ms   180 ms  180 ms  64.233.161.104
```

Protocolo IP - Encaminamiento

¿Qué es el encaminamiento o ruteo?

- El ruteo es un proceso de elección de un camino por el cual enviar un datagrama a partir de información contenida en una tabla (tabla de rutas) y de verificar el prefijo de red de la dirección IP del destinatario. El ruteo ocurre en capa 3 y lo realiza el protocolo IP.
- Todos los dispositivos IP realizan ruteo pero:
 - Un host (o multihomed host) solo puede rutear datagramas propios
 - Un ruteador puede rutear datagramas propios y de terceros (históricamente llamados gateways)



Concepto fundamental: *“La dirección IP destino en un datagrama siempre se refiere al último destino (final). Cuando un dispositivo entrega un datagrama a otro (próximo salto), esta dirección no forma parte del destinatario en el datagrama”*

Protocolo IP - Encaminamiento

■ Tablas de Rutas

- Contienen información acerca de próximos saltos en una ruta (es decir, a dónde enviar un datagrama)

```
s10:~# route
```

```
Kernel IP routing table
```

<u>Destination</u>	<u>Gateway</u>	<u>Genmask</u>	<u>Flags</u>	<u>Metric</u>	<u>Ref</u>	<u>Use</u>	<u>Iface</u>
192.168.3.0	*	255.255.255.0	U	0	0	0	eth2
192.168.2.0	*	255.255.255.0	U	0	0	0	eth1
192.168.1.0	*	255.255.255.0	U	0	0	0	eth1
10.10.10.0	*	255.255.255.0	U	0	0	0	eth0
default	10.10.10.1	0.0.0.0	UG	0	0	0	eth0

Protocolo IP - Encaminamiento

■ Las tablas de rutas se completan:

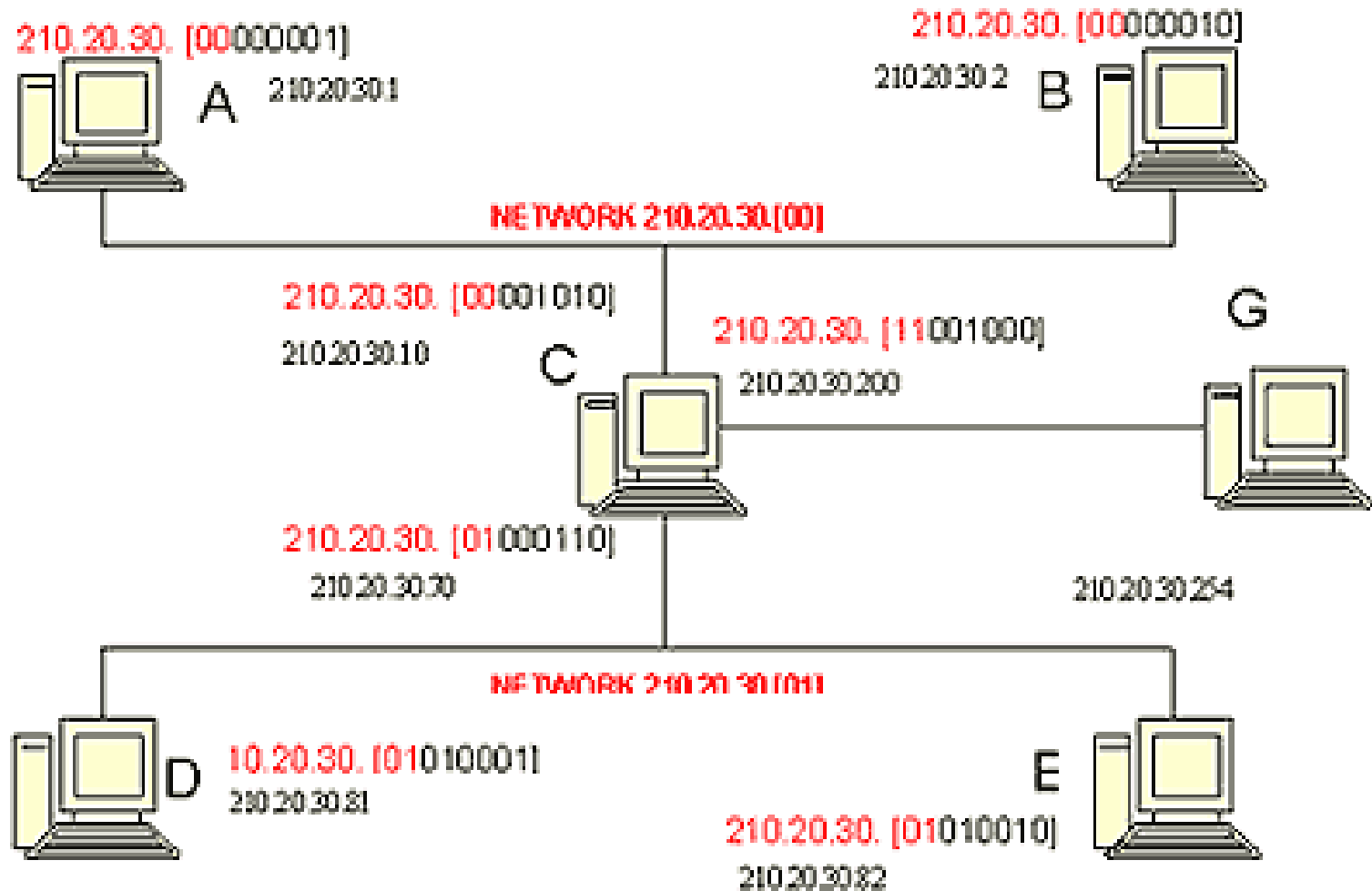
- A mano – rutas estáticas
- `route add -net 200.10.15.0 netmask 255.255.255.0 gw 170.210.96.99 eth2`
- Protocolos de ruteo – rutas dinámicas (RIP, OSPF)
- Por mensajes ICMP redirect

■ IP busca tablas de rutas por:

- la dirección del host destino
- la dirección de la red destino
- una entrada por defecto (default gateway)

(si existe más de una ruta para el mismo destino, se evalúa la métrica para comparar las rutas. Pueden expresar distancia, throughput, retardo, error rate, costo)

Protocolo IP - Encaminamiento



Protocolo IP - Encaminamiento

■ Encaminamiento

- Todos los dispositivos IP rutean (Host < > Router)

■ Entrega directa

- El destinatario del mensaje se encuentra dentro de la misma red que el emisor, entonces se debe enviar el datagrama (en una PDU de enlace) directamente al destinatario. En el ejemplo anterior, si el equipo A (210.20.30.1) quiere enviar un mensaje al equipo B (210.20.30.2), realiza Entrega Directa

■ Entrega indirecta

- El destinatario del mensaje no se encuentra dentro de la misma red que el emisor, entonces se debe enviar el datagrama (en una PDU de enlace) al gateway. En el ejemplo anterior, si el equipo A (210.20.30.1) quiere enviar un mensaje al equipo E (210.20.30.82), realiza Entrega Indirecta a través de C

Algoritmos de Ruteo

■ Objetivos de Diseño

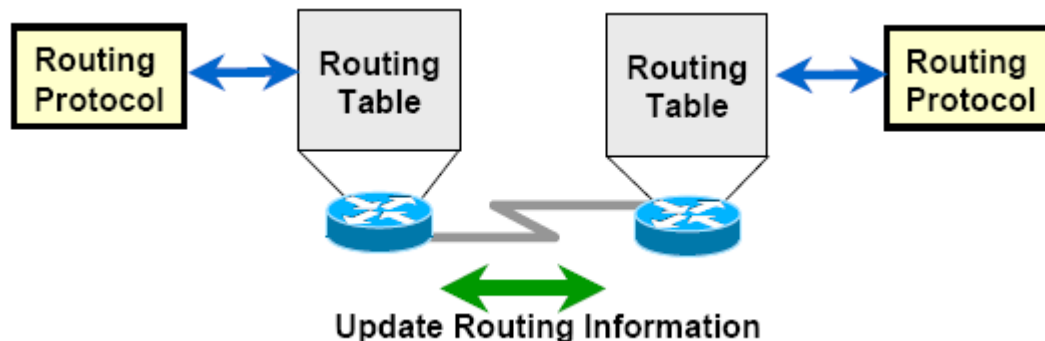
- **Optimalidad:** Calcular la mejor ruta.
- **Simplicidad (bajo overhead):** Eficiencia.
- **Estabilidad/Robustez:** Operar correctamente bajo circunstancias no usuales.
- **Convergencia Rápida:** Responder rápidamente a los cambios en la topología de la red.
- **Flexibilidad:** Adaptarse a diferentes tipos de redes.

Protocolos de Ruteo

“Es un protocolo para intercambiar información acerca del estado y la topología de la red. Se implementa en ruteadores”

■ Funciones:

- Crear entradas en las tablas de rutas
- Mantener actualizada la tabla de rutas
- Calcular la mejor opción para el “próximo salto” (GW). ¿Cómo? Mediante métricas usadas para “comparar” rutas (distancia, throughput, error rate, retardo, costo).



Algoritmo de Ruteo

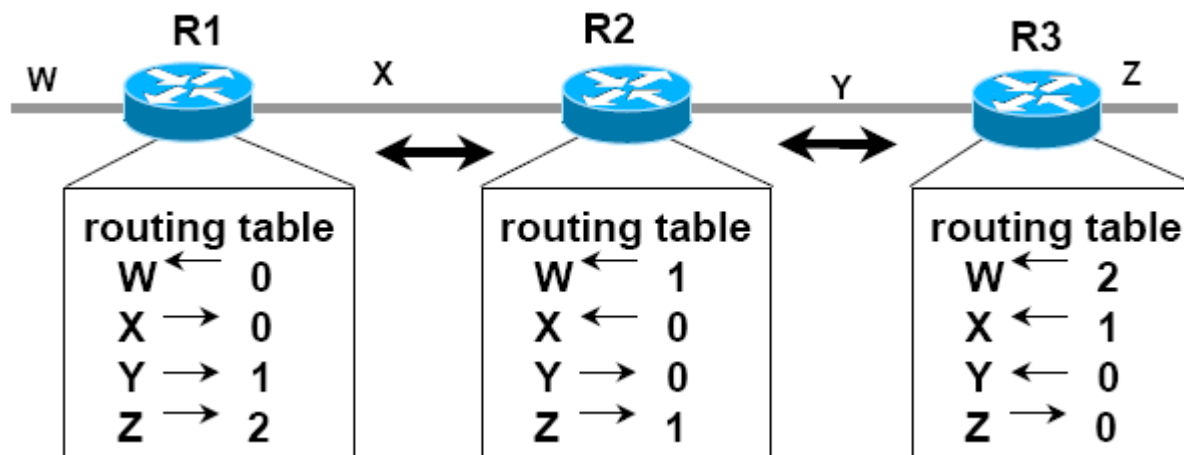
Algoritmo de Ruteo: Vector de Distancias

La distancia es la métrica y el vector contiene los destinos.

Se envía (inundación) la información de tablas de rutas a los vecinos (Algoritmo de Bellmann-Ford).

Ejemplo: RIP (Routing Information Protocol)

Problemas: Convergencia lenta, cuenta a infinito



Algoritmo de Ruteo

Algoritmo de Ruteo: Estado del Enlace

La métrica es el costo.

Se intercambia la información del estado de los enlaces a todos los demás nodos

Cada router calcula todos los caminos más cortos a todos los destinos (Shortest Path First). (Algoritmo de Dijkstra).

Ejemplo: OSPF (Open Shortest Path First)

Convergencia rápida

