

Fragmentación y Reensamblado en IP ICMP

Area de Ingeniería Telemática
<http://www.tlm.unavarra.es>

Arquitectura de Redes, Sistemas y Servicios
3º Ingeniería de Telecomunicación

Temario

1. Introducción
2. Arquitecturas, protocolos y estándares
3. Conmutación de paquetes
4. Conmutación de circuitos
5. Tecnologías
6. Control de acceso al medio en redes de área local
7. Servicios de Internet

Temario

1. Introducción
2. Arquitecturas, protocolos y estándares
- 3. Conmutación de paquetes**
 - Arquitectura de protocolos para LANs
 - Ethernet
 - Protocolos de Internet
 - Internetworking
 - Direccionamiento
 - **Fragmentación e ICMP.** IP en LAN
4. Conmutación de circuitos
5. Tecnologías
6. Control de acceso al medio en redes de área local
7. Servicios de Internet

Objetivo

- Completar los conceptos básicos sobre el nivel de red en Internet

Contenido

- Fragmentación y reensamblado
 - Necesidad
 - Implementación
 - Problemas
- ICMP
 - Características generales
 - Condiciones generales de envío
 - Mensajes
- Traceroute

Contenido

- **Fragmentación y reensamblado**
 - Necesidad
 - Implementación
 - Problemas
- ICMP
 - Características generales
 - Condiciones generales de envío
 - Mensajes
- Traceroute

Fragmentación y Reensamblado

Necesidad

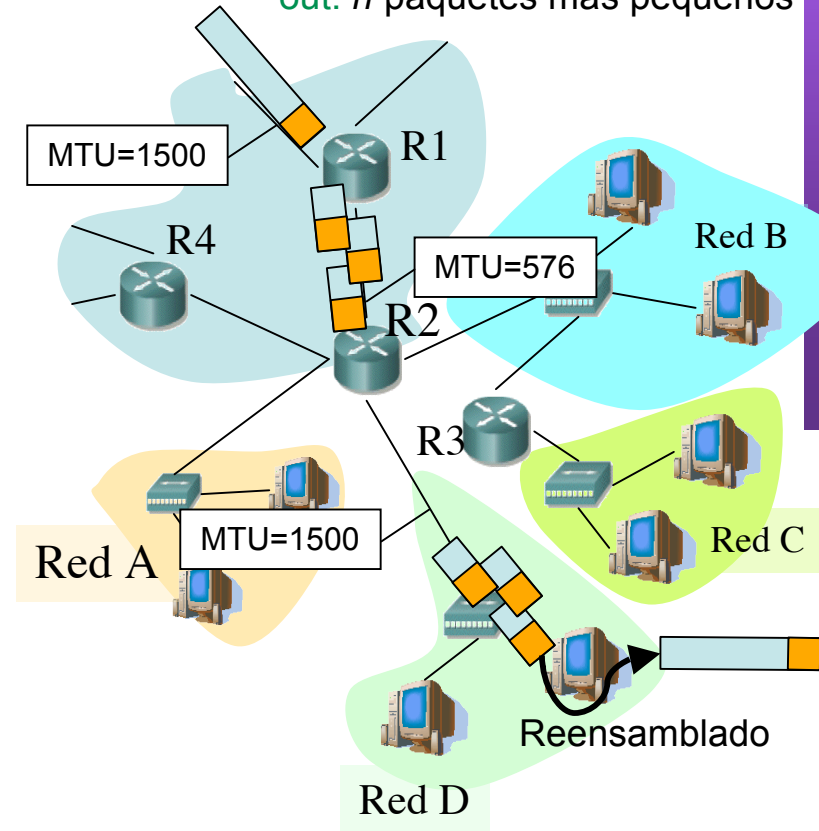
- El nivel de enlace impone unos límites al tamaño
- MTU = Maximum Transfer Unit
- Un datagrama IP es dividido dentro de la red (...)
- Un datagrama se convierte en varios paquetes
- Hosts y routers fragmentan
- *Los routers NO reensamblan (...)*
- Solo el host receptor final reensambla (...)

Red (RFC 1191)	MTU
16Mbps Token Ring	17914
IEEE 802.4	8166
FDDI	4352
Ethernet	1500
IEEE 802.3	1492
X.25	576

Fragmentación:

in: un datagrama grande

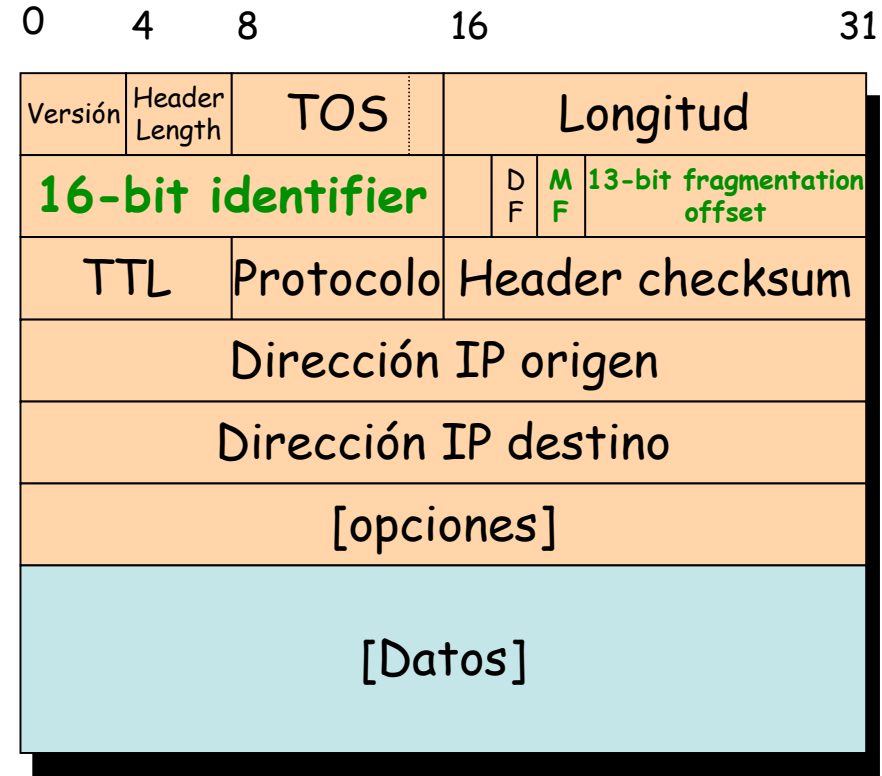
out: n paquetes más pequeños



Fragmentación y Reensamblado

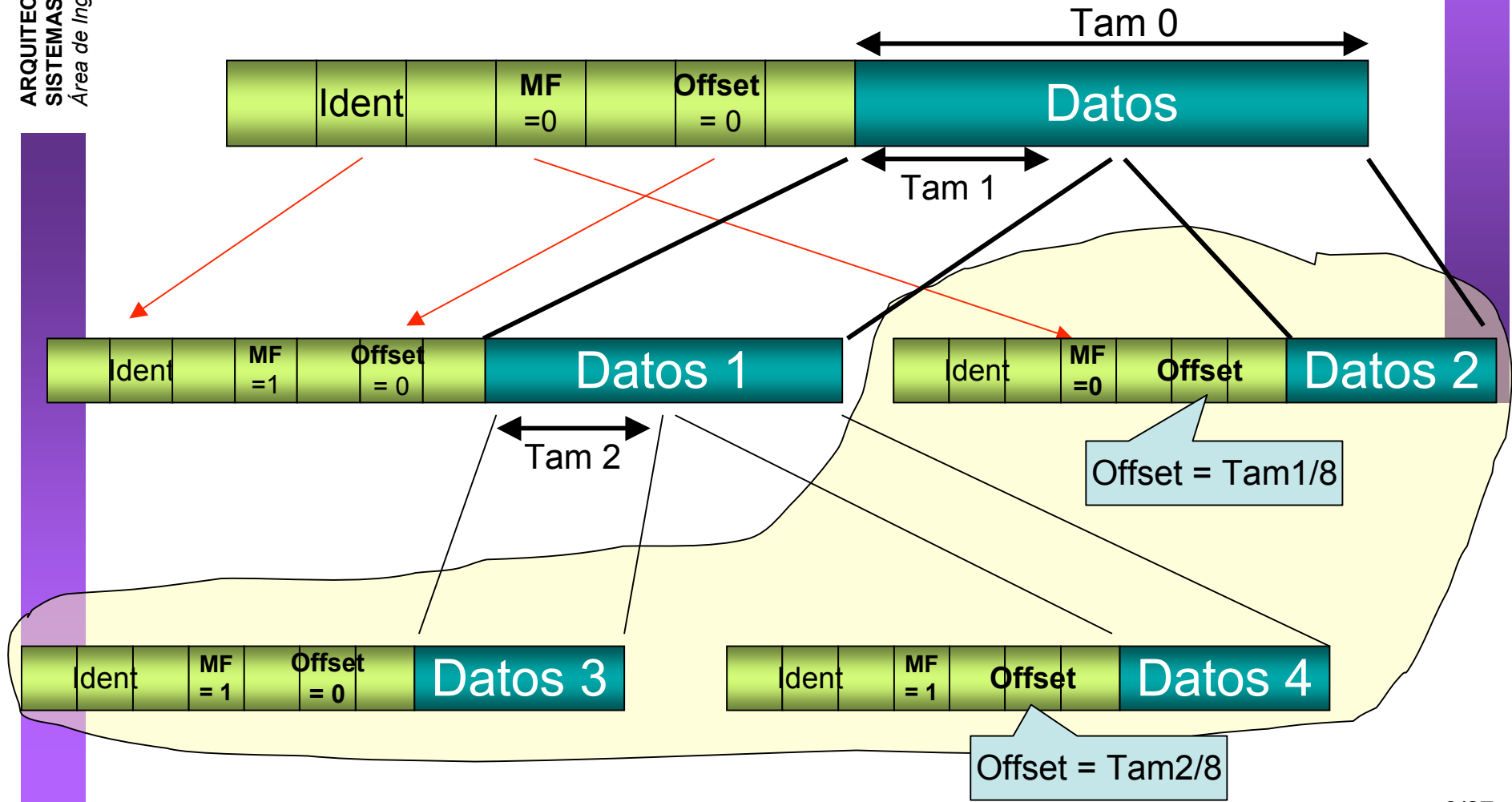
Codificación de la información

- Campos empleados:
 - Identificación
 - Bit MF
 - Fragment offset
- Fragmentos del datagrama:
 - Igual identificación, IP origen, IP destino y protocolo
- “Longitud” es la del paquete, no del datagrama
- Ante un primer fragmento ⇒ reservar zona de memoria donde reensamblar
- Debe reservar suficiente para reensamblar al menos datagramas de 576 Bytes



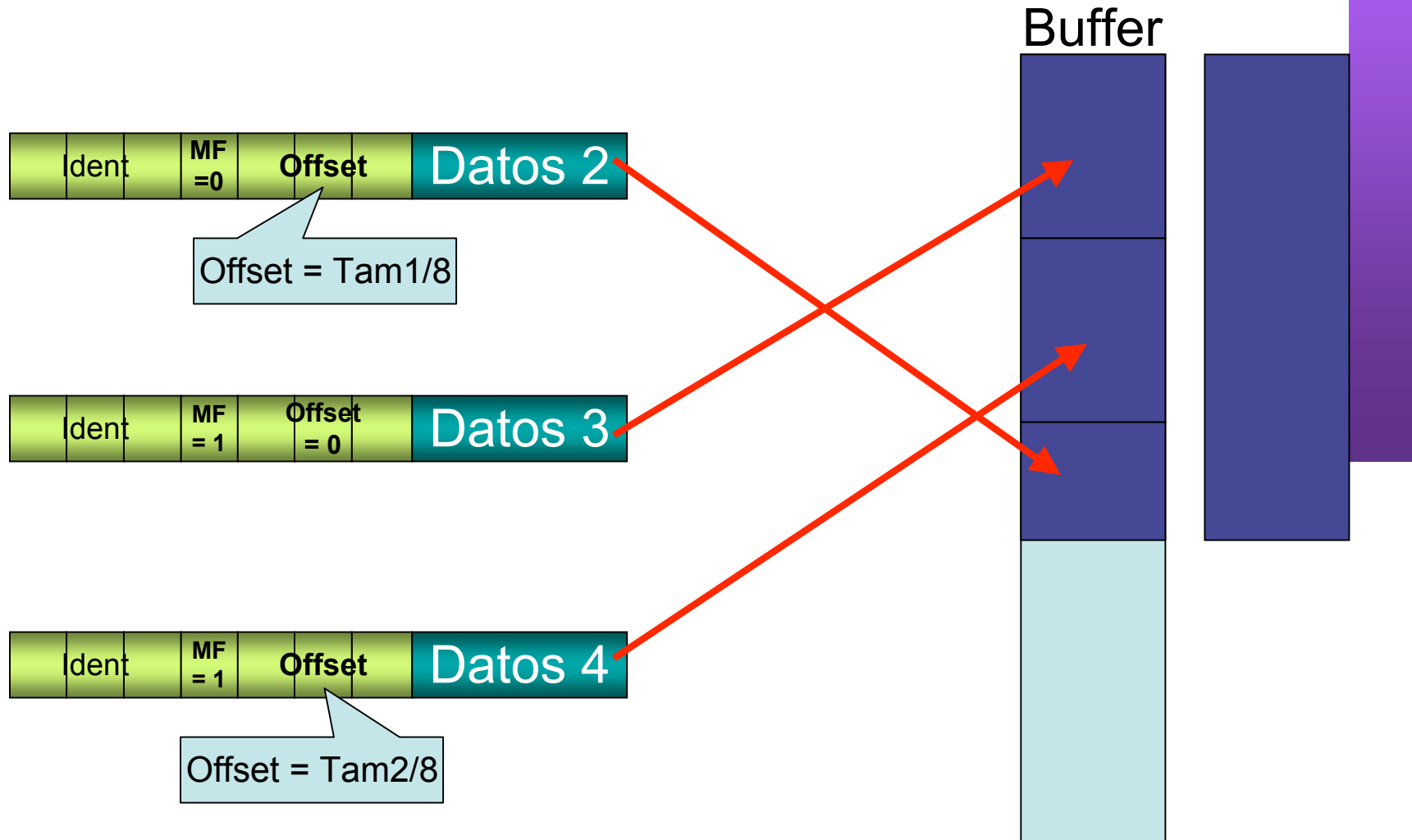
Fragmentación

Implementación



Reensamblado

Implementación



Situaciones de “error”

- Bit DF:
 - En la cabecera IP
 - $DF==1 \Rightarrow$ routers no pueden fragmentar el paquete
 - $(Tam > MTU) \&\& (DF==1) \Rightarrow$ lo descarta y devuelve al host origen un paquete indicando el error (ICMP)
- Reensamblado:
 - Inicia un *timer* con el primer fragmento que recibe
 - Si caduca el *timer* sin tener todos los fragmentos descarta todo lo recibido y devuelve al origen un paquete indicando el error (ICMP)

Problemas de la fragmentación

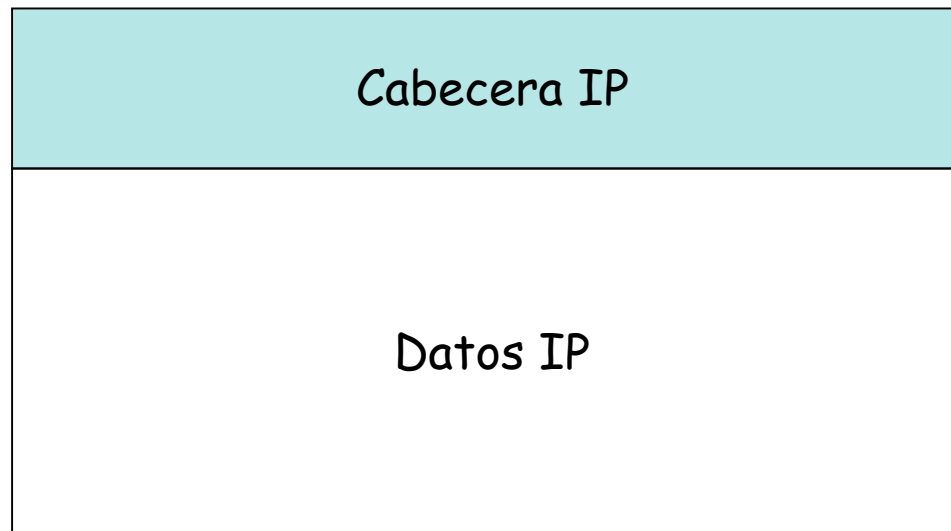
- Menor cociente Datos/Cabeceras
- Añade más carga a los routers (IPv6 la elimina)
- Si se pierde un fragmento:
 - El receptor no puede recomponer el datagrama
 - Tira todos los fragmentos recibidos
- Hasta que no se reciba todo el datagrama no se pueden pasar los datos al nivel de transporte (mayor retardo)

Contenido

- Fragmentación y reensamblado
 - Necesidad
 - Implementación
 - Problemas
- **ICMP**
 - **Características generales**
 - **Condiciones generales de envío**
 - **Mensajes**
- Traceroute

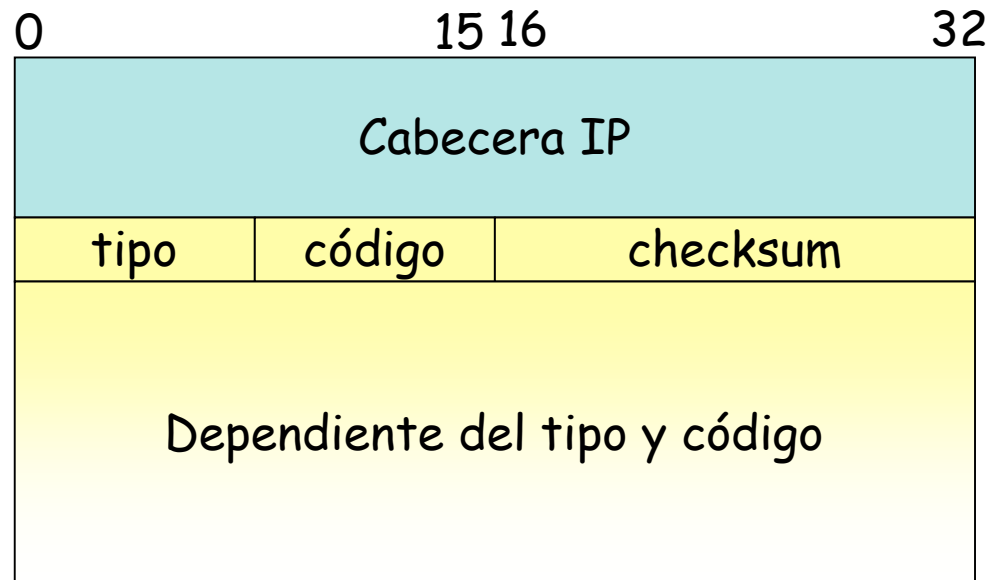
Características generales

- *Internet Control Message Protocol* (RFC 792)
- Para comunicar mensajes de error y otra información del nivel de red
- Mensajes transportados dentro de datagramas IP
- El destino es la dirección del paquete IP que generó el error
- Parte del nivel IP
- Estructura general del mensaje (...):

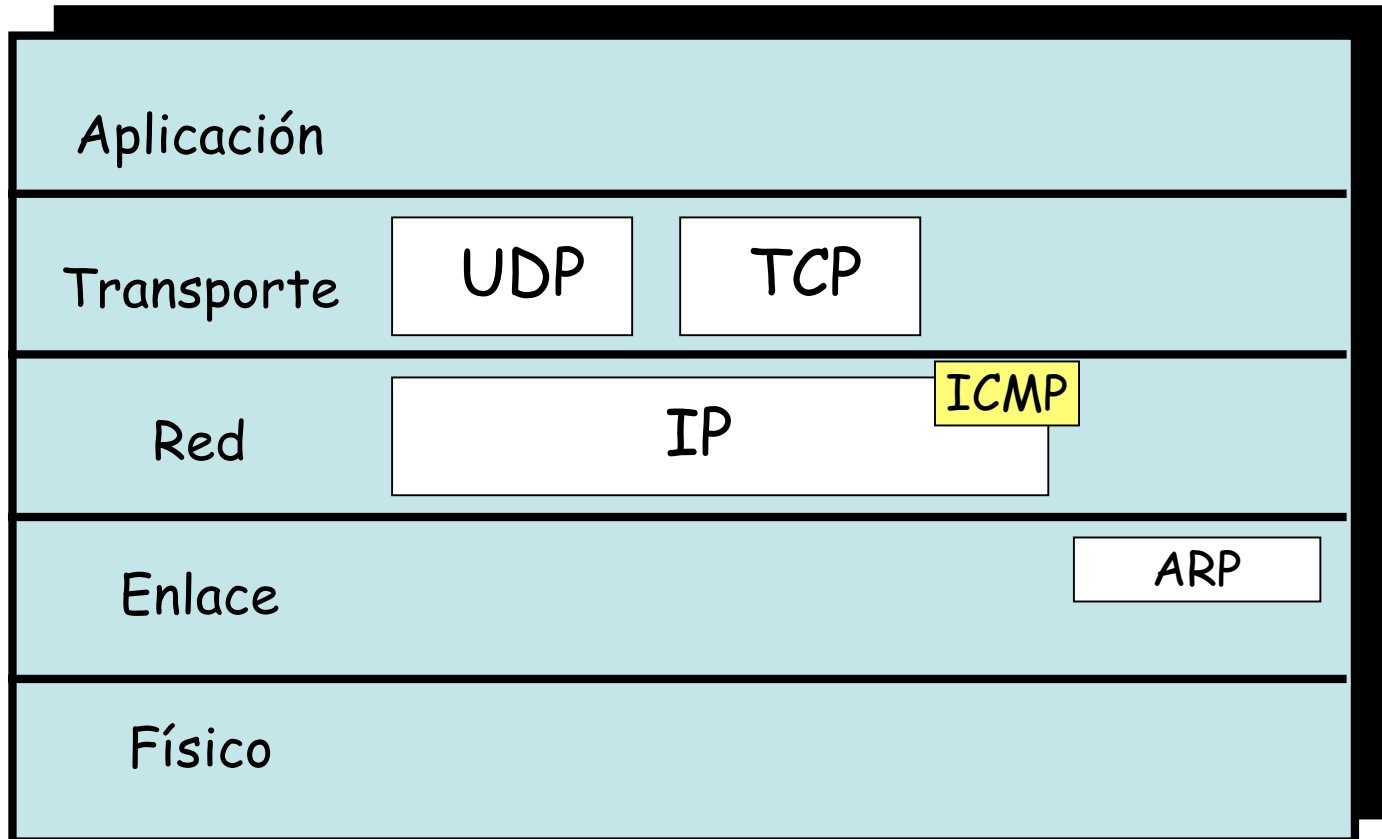


Características generales

- *Internet Control Message Protocol* (RFC 792)
- Para comunicar mensajes de error y otra información del nivel de red
- Mensajes transportados dentro de datagramas IP
- El destino es la dirección del paquete IP que generó el error
- Parte del nivel IP
- Estructura general del mensaje (...):



¿Dónde encaja ICMP en la pila TCP/IP?



Clases de mensajes ICMP

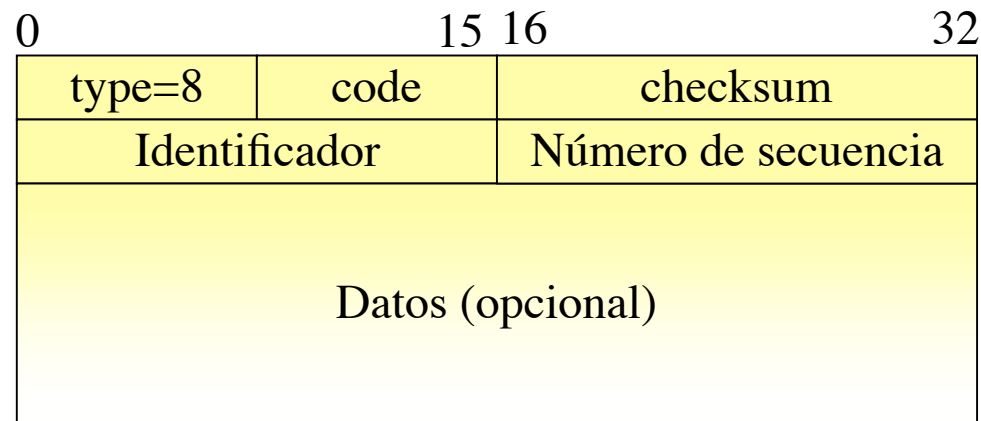
- Mensajes de Error:
 - Destino inalcanzable
 - *Redirect*
 - Tiempo excedido
 - *Source Quench*
 - Problema de parámetros
- Mensajes de pregunta (*query*):
 - Echo
 - *Router Advertisement*
 - *Timestamp*
 - Información
 - *Address Mask*

Condiciones generales de envío

- Para evitar tormentas de errores
- Nunca se generan ICMPs *de error* en respuesta a:
 - Un ICMP de error
 - Un datagrama destinado a una IP de broadcast o multicast
 - Un broadcast (o multicast) a nivel de enlace
 - Un fragmento que no sea el primero
 - Un datagrama cuya IP origen no sea *single-host*:
loopback, broadcast, multicast

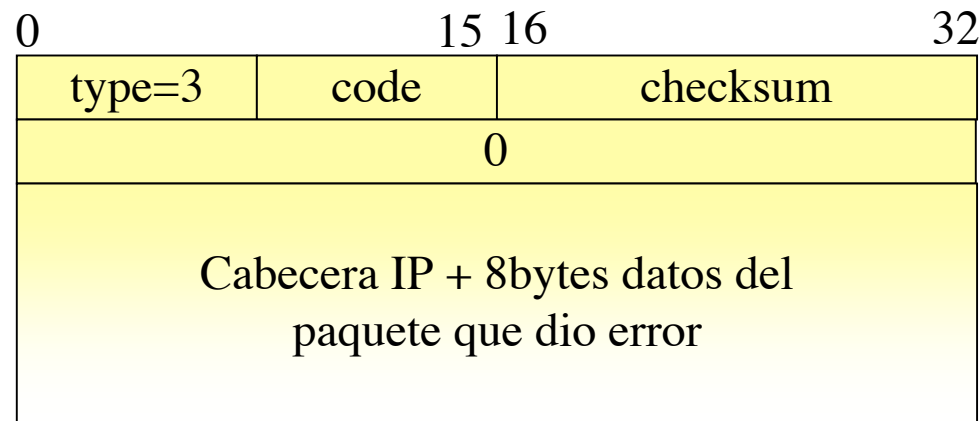
Mensajes ICMP

- Echo request/reply (*query*) (PING)
 - tipo = 8 (request) o 0 (reply), código = 0
 - Servidor debe hacer *echo* del paquete (incluidos los datos)
 - Obligatorio de implementar (generalmente en el kernel)



Mensajes ICMP

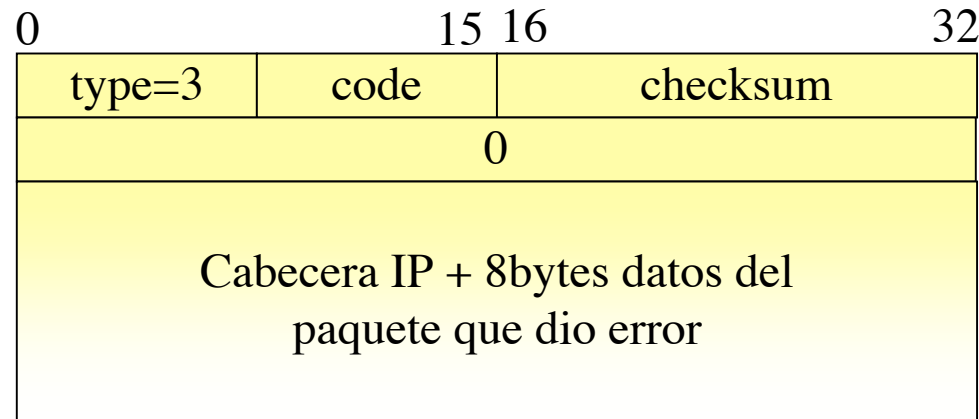
- Destino inalcanzable (*error*)
 - tipo = 3
 - Si según la tabla de rutas no se puede llegar al destino, host/router debe enviarlo (...)



Mensajes ICMP

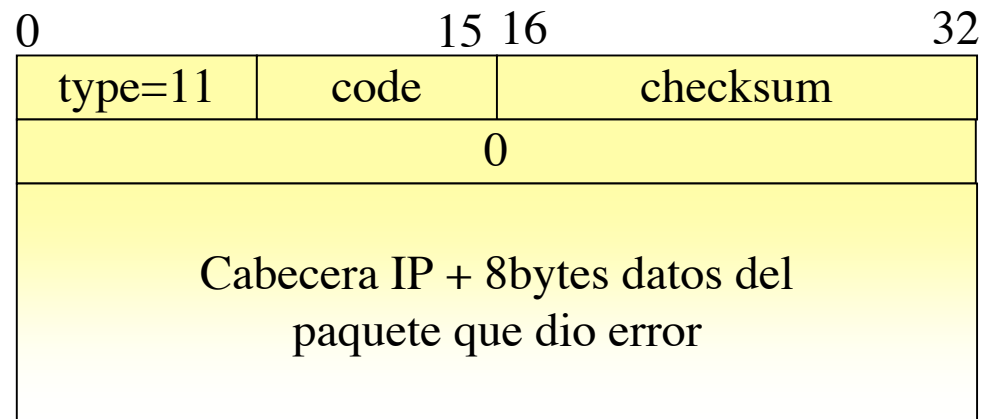
(Destino inalcanzable)

- Código:
 - 0 = Red destino inalcanzable
 - 1 = Host destino inalcanzable
 - 2 = Protocolo destino inalcanzable
 - 3 = Puerto destino inalcanzable
 - 4 = Fragmentación necesaria y DF activo
 - 5 = Source route failed



Mensajes ICMP

- Tiempo excedido (*error*)
 - tipo = 11
 - código = 0 (TTL=0 en tránsito), 1 (timeout durante reensamblado, necesita primer paquete)

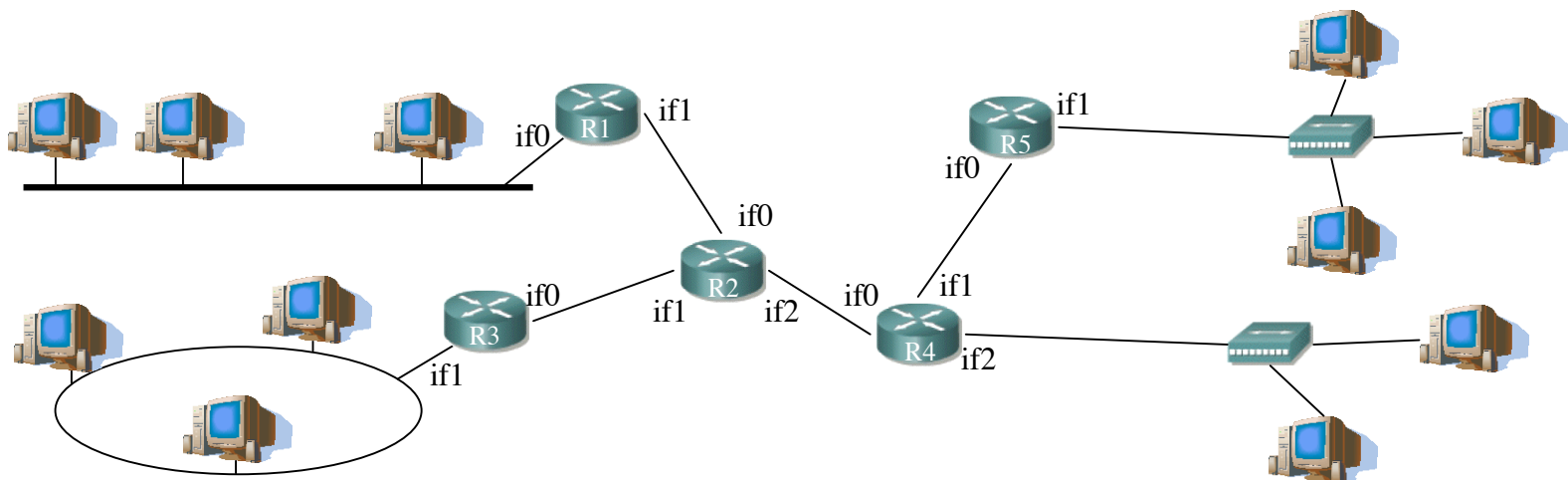


Contenido

- Fragmentación y reensamblado
 - Necesidad
 - Implementación
 - Problemas
- ICMP
 - Características generales
 - Condiciones generales de envío
 - Mensajes
- **Traceroute**

Traceroute

- Permite averiguar *el camino* entre dos hosts
- Suponiendo que el camino se mantiene entre diferentes paquetes
- Requiere que el destino final soporte UDP
- Requiere que se generen ciertos mensajes ICMP
- *Implemented by **Van Jacobson** from a suggestion by Steve Deering. Debugged by a cast of thousands with particularly cogent suggestions or fixes from C. Philip Wood, Tim Seaver, and Ken Adelman.*



Traceroute

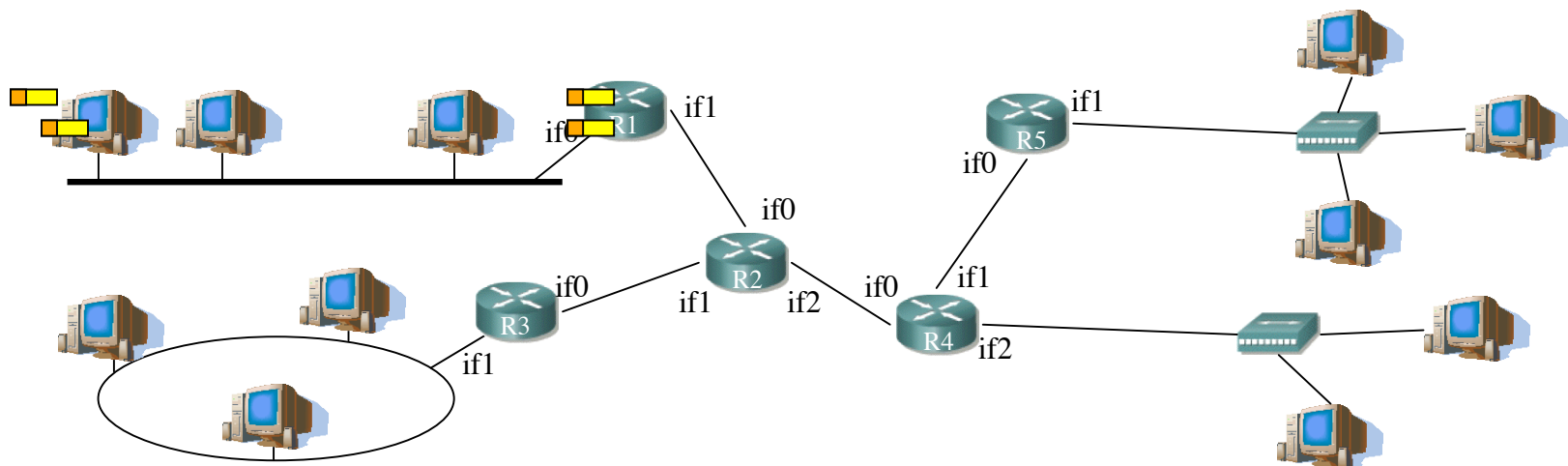
- El host inicial envía un datagrama UDP (...)

- Dirigido al host final
- Con TTL = 1

- El primer router decremента el TTL a 0 (...)

- Tira el paquete
- Devuelve al origen un ICMP de Error *Tiempo excedido en tránsito*
- Este es un paquete IP con dirección origen la del interfaz de R1 en la red del host (... ..)

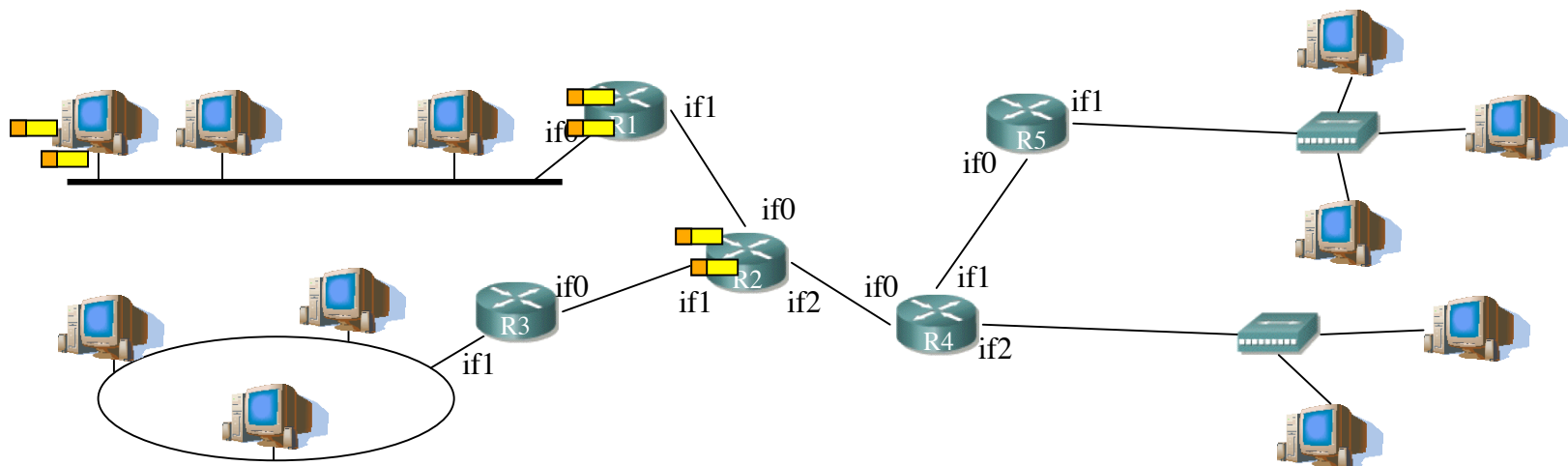
<u>TTL</u>	<u>IP</u>
1	IPR1 _{if0}



Traceroute

- El host inicial envía un datagrama UDP (...)
 - Dirigido al host final
 - Con TTL = 2
- El primer router decrementa el TTL a 1 y lo reenvía (...)
- El segundo router decrementa el TTL a 0 (...)
 - Tira el paquete
 - Devuelve al origen un ICMP de Error *Tiempo excedido en tránsito*
 - Este es un paquete IP con dirección origen la del interfaz de R2 en dirección hacia el host origen (...)

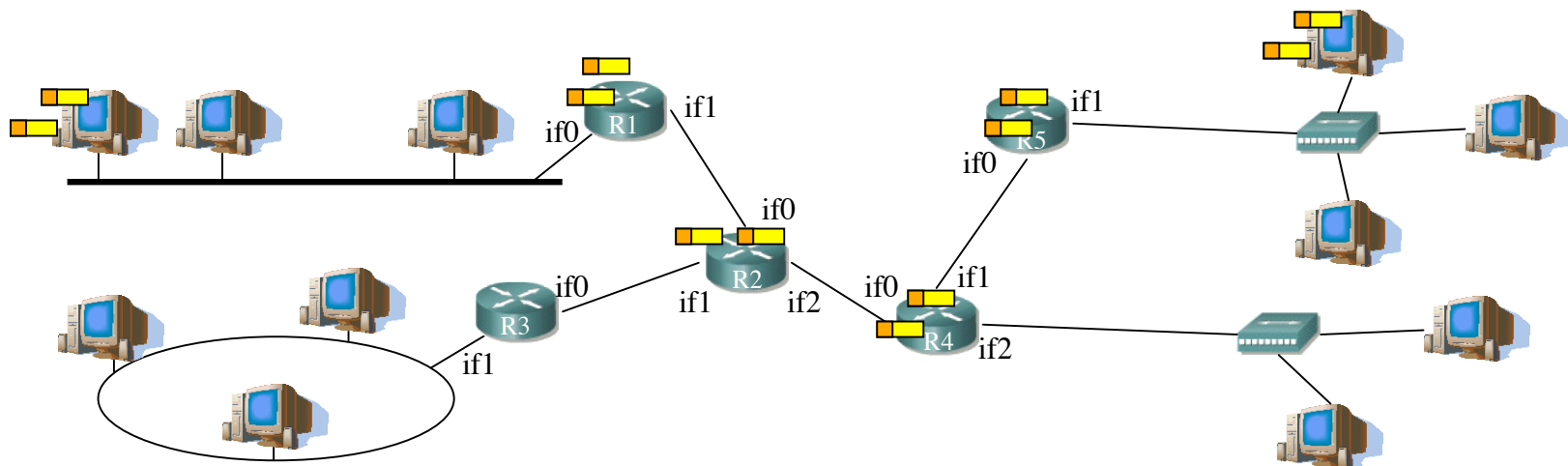
TTL	IP
1	IPR1 _{if0}
2	IPR2 _{if0}



Traceroute

- Idem con TTL=3 y TTL=4 (...)
- Con TTL suficientemente grande el paquete llega hasta el destino final (... ..)
- En el destino no hay aplicación esperando paquetes UDP en ese puerto:
 - Lo tira
 - Devuelve al origen un ICMP de Error *Puerto destino inalcanzable* (... ..)

<u>TTL</u>	<u>IP</u>
1	IPR1 _{if0}
2	IPR2 _{if0}
3	IPR4 _{if0}
4	IPR5 _{if0}
5	IPhost



Traceroute (Ejemplo)

```
daniel% traceroute www.berkeley.edu
traceroute to arachne.berkeley.edu (169.229.131.109), 30 hops max, 40 byte packets
 1 arce-un.red.unavarra.es (130.206.160.1) 1.691 ms 0.438 ms 0.417 ms
 2 ss-in (130.206.158.25) 1.015 ms 3.091 ms 0.658 ms
 3 unavarra-router.red.unavarra.es (130.206.158.1) 1.587 ms 1.87 ms 1.506 ms
 4 fe0-1-2.eb-pamplona0.red.rediris.es (130.206.209.13) 1.49 ms 1.741 ms 1.25 ms
 5 nav.so2-3-0.eb-bilbao0.red.rediris.es (130.206.240.61) 5.279 ms 4.402 ms 4.398 ms
 6 pav.so2-0-0.eb-iris2.red.rediris.es (130.206.240.29) 50.039 ms 16.511 ms 16.35 ms
 7 so0-0-0.eb-iris4.red.rediris.es (130.206.240.2) 16.341 ms 17.982 ms 16.405 ms
 8 rediris.es1.es.geant.net (62.40.103.61) 118.998 ms 16.741 ms 16.755 ms
 9 es.it1.it.geant.net (62.40.96.186) 96.679 ms 39.288 ms 39.513 ms
10 it.de2.de.geant.net (62.40.96.61) 91.118 ms 48.088 ms 49.83 ms
11 abilene-gw.de2.de.geant.net (62.40.103.254) 141.935 ms 141.812 ms 141.716 ms
12 atlang-washng.abilene.ucaid.edu (198.32.8.65) 157.505 ms 157.692 ms 164.648 ms
13 hstng-atlang.abilene.ucaid.edu (198.32.8.33) 177.182 ms 177.144 ms 177.201 ms
14 losang-hstng.abilene.ucaid.edu (198.32.8.21) 199.049 ms 198.489 ms *
15 hpr-lax-gsr1--abilene-la-10ge.cenic.net (137.164.25.2) 199.004 ms 198.621 ms 284.873 ms
16 svl-hpr--lax-hpr-10ge.cenic.net (137.164.25.13) 215.55 ms 218.166 ms 206.364 ms
17 hpr-ucb-ge--svl-hpr.cenic.net (137.164.27.134) 210.841 ms 207.409 ms 207.479 ms
18 vlan187.inr-201-eva.berkeley.edu (128.32.0.33) 283.445 ms 207.842 ms 207.318 ms
19 g5-1.inr-210-srb.berkeley.edu (128.32.255.65) 211.052 ms 207.341 ms 207.408 ms
20 arachne.berkeley.edu (169.229.131.109) 207.431 ms 207.451 ms 207.4 ms
```

Probadlo y ved los paquetes con tcpdump o ethereal