ICMP
Ping
Traceroute

Carlos Pérez Damián López Leandro Di Lorenzo

Grupo 5

Redes de Computadoras
Universidad Nacional de Quilmes

28 de Octubre, 2014

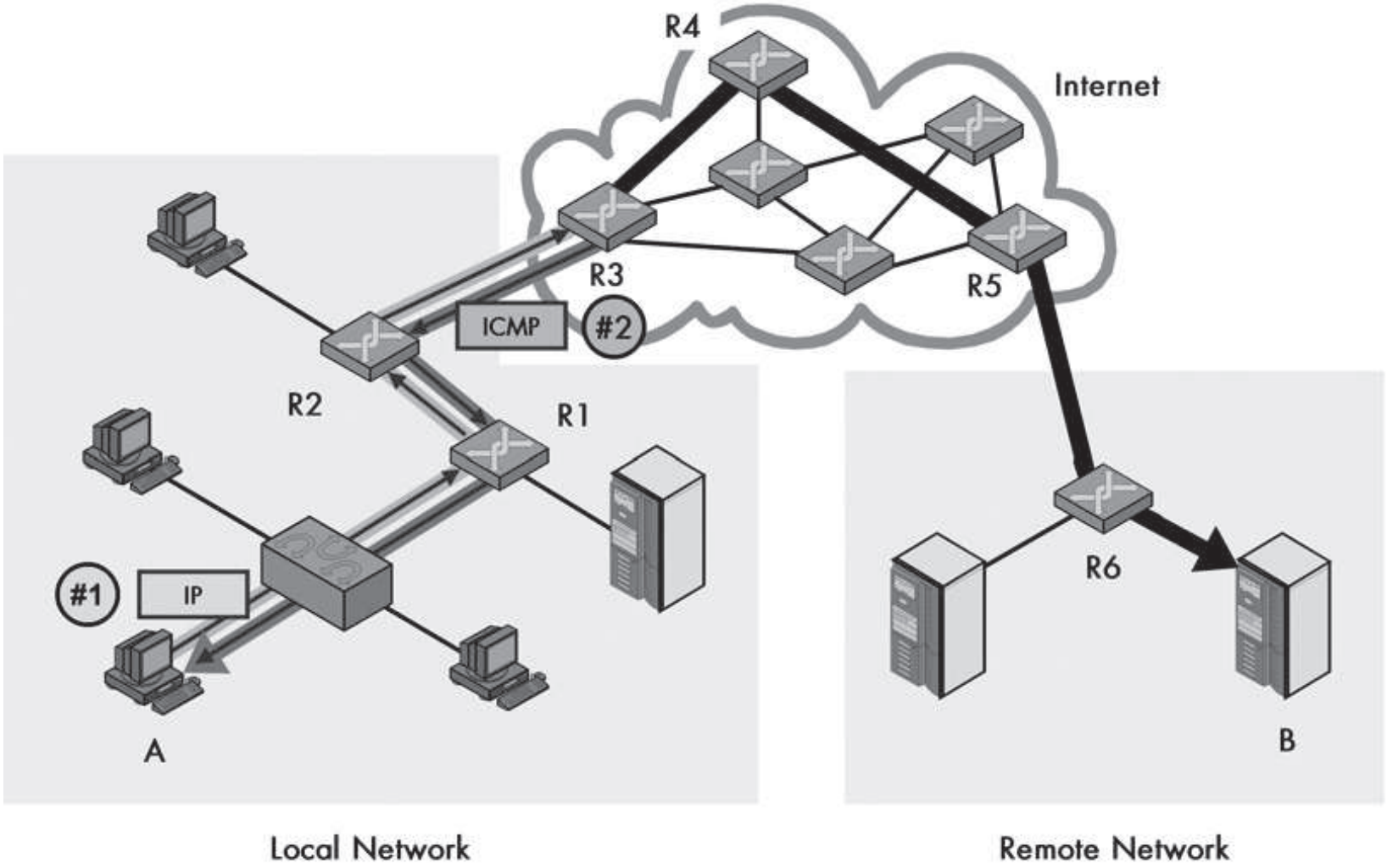
ICMP

Internet Control Message Protocol

Aspectos Técnicos

- ICMP colabora con IP respondiendole al host de origen cuando un datagrama suyo tiene un problema
- Si bien es un protocolo de capa 3, funciona utilizando los servicios de IP como si fuese un protocolo capa 4
- Está definido en el RFC 792
- Su función original era reportar condiciones de error en el envío de datagramas IP
- Actualmente
  - reportar un rango mayor de errores
  - genera intercambio de información de la red
  - cuenta con capacidad de testing y diagnóstico
- IPv4 $\rightarrow$ ICMPv4
- IPv6 $\rightarrow$ ICMPv6

Ejemplo de flujo ICMP



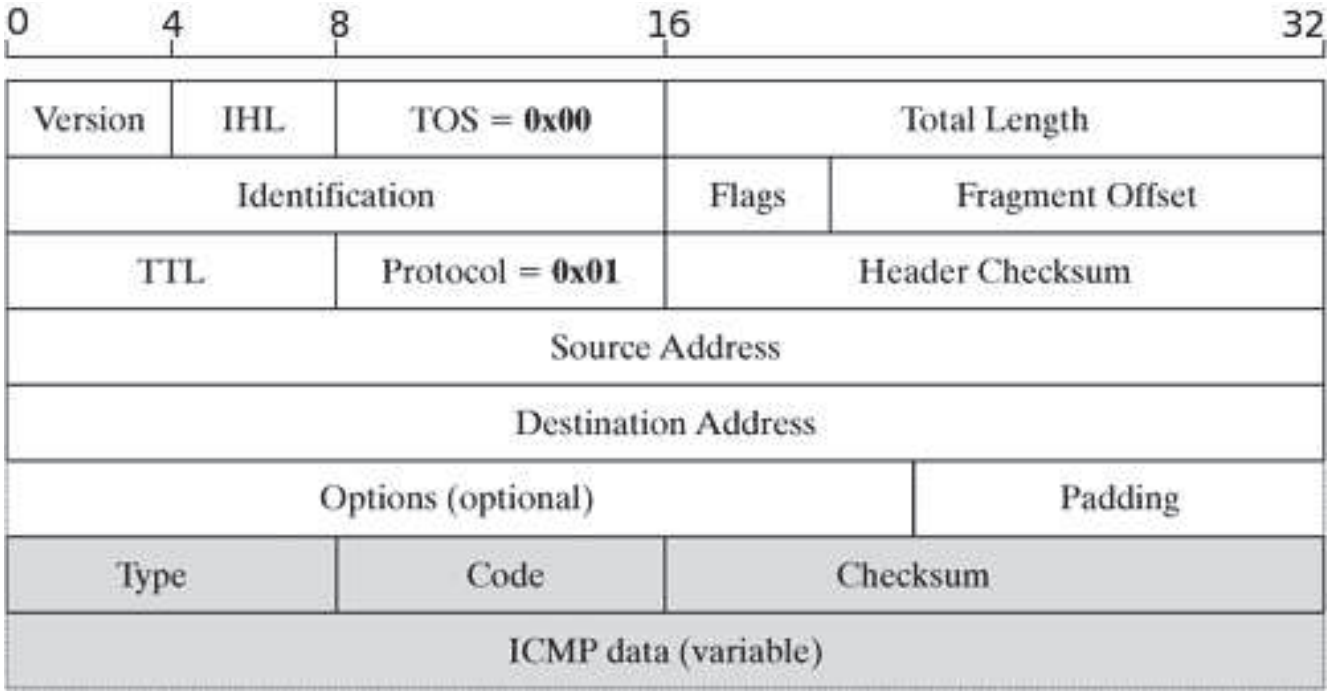
¿Por qué es necesario ICMP?

¿No puede encargarse IP de sus propios errores?

- Las principales características del envío de datagramas IP son
  - No orientado a conexión
  - Poco fiable
  - Sin reconocimiento
- IP pone los datagramas en la red y espera que lleguen
- Si tuviese que encargarse también del control de errores se volvería un protocolo muy complejo
- Entonces delega esa tarea a un asistente: ICMP
- Keep It Simple ...
  - IP empaqueta y manda
  - ICMP informa de los errores

Estructura de un segmento ICMP

La cabecera ICMPv4 comienza luego de la cabecera IPv4 y se identifica en el datagrama IP marcando `Protocolo=1` y `Type-of-Service=0`, con lo cual el datagrama será es tratado como uno normal sin prioridad.



Type (8 bits) Checksum ICMP (16 bits)
Code (8 bits) Data (variable)

Mensajes de Control

Originalmente se definieron 11, luego se incorporaron más.

Mensajes de Error

- Responden a un datagrama IP que tuvo un problema
- Proveen feedback al origen acerca de los datagramas que fallaron (dentro de lo posible)
- Son en respuesta a una acción que no puede cumplir IP

Mensajes de Información

- Pueden ser en respuesta a una petición ICMP
- Se utilizan para intercarbiar información, implementar funciones o hacer testing

Destination Unreachable Messages

¿Qué pasaría si no existiesen los mensajes de destino inalcanzable?

- IP no es un protocolo confiable
- TCP usa IP tomando medidas para hacer confiable la comunicación
- Si un paquete no recibe confirmación, lo vuelve a enviar asumiendo que se perdió
- Si un paquete no recibe confirmación porque un router no puede hacerlo llegar a su destino, TCP lo vuelve a enviar asumiendo que se perdió
- y como se va a volver a perder, TCP lo va a volver a enviar
- hasta el fin de los tiempos

Destination Unreachable Messages

Tipos de errores

Existe muchas razones por las cuales un datagrama no pueda llegar a destino. Por eso se considera a `ICMP Destination Unreachable Messages` como una clase de mensajes de error. El receptor le informa al host de origen que el datagrama no se puede enviar al host y la razón por la cual no se envió.

Destination Unreachable Messages :: Error Codes

ICMPv4 Destination Unreachable Messages codes		
Code	**Nombre**	**Descripción**
0	Network Unreachable	No se puede alcanzar la red, en general por mal ruteo
1	Host Unreachable	No puede alcanzar el host, también por mal ruteo
2	Protocol Unreachable	No puede procesar el protocolo
3	Port Unreachable	Puerto inválido
4	Fragmentation Needed	Si un datagrama es más grande que un MTU, el router lo puede fragmentar para hacerlo pasar. Pero si está seteado DF (no fragmentar) el router tiene que tirar el paquete y mandarle este ICMP al host para que tome una decisión: fragmentar o buscar un camino más amplio (Path MTU Discovery)
5	Source Route Failed	Generado si una routa fue especificada por el datagrama pero el router no puede enviar por ahí

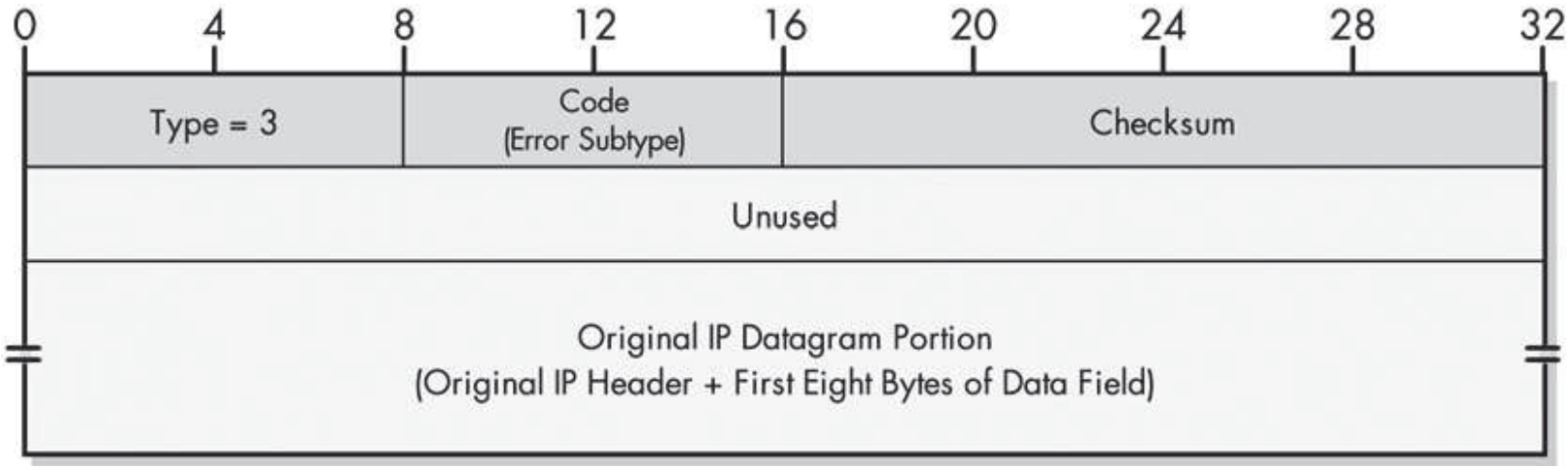
Destination Unreachable Messages :: Error Codes

ICMPv4 Destination Unreachable Messages codes		
Code	**Nombre**	**Descripción**
6	Destination Network Unknown	No se usa; se usa code 0 en su lugar
7	Destination Host Unknown	No se conoce el host de destino, en general lo usan los routers locales
8	Source Host Isolated	Obsoleto
9	Communication with Destination Network Is Administratively Prohibited	El dispositivo de origen no tiene permitido enviar a la red del dispositivo de destino
10	Communication with Destination Host Is Administratively Prohibited	El dispositivo de origen puede enviar a la red del dispositivo destino pero no puede enviarle a ese dispositivo en particular
11	Destination Network Unreachable for Type of Service	La red de destino no se puede alcanzar porque no provee los tipos de servicios requeridos por el datagrama

Destination Unreachable Messages :: Error Codes

ICMPv4 Destination Unreachable Messages codes		
Code	**Nombre**	**Descripción**
12	Destination Host Unreachable for Type Service	El host de destino no se puede alcanzar porque no provee los tipos de servicios requeridos por el datagrama
13	Communication Administratively Prohibited	El datagrama no se puede enviar porque los filtros bloquean el mensaje en base a su contenido
14	Host Precedence Violation	Lo envía el router del primer salto cuando el valor del tipo de servicio no es permitido
15	Precedence Cutoff in Effect	Lo envía un router cuando el valor de precedencia (prioridad) es menor que el mínimo permitido por la red

Destination Unreachable Messages :: Header



Original Datagram Portion $\rightarrow$ La cabecera IP completa y los primeros 8 bytes de datos para que el origen reconozca el datagrama

Source Quench Messages

Relentizar los mensajes entrantes

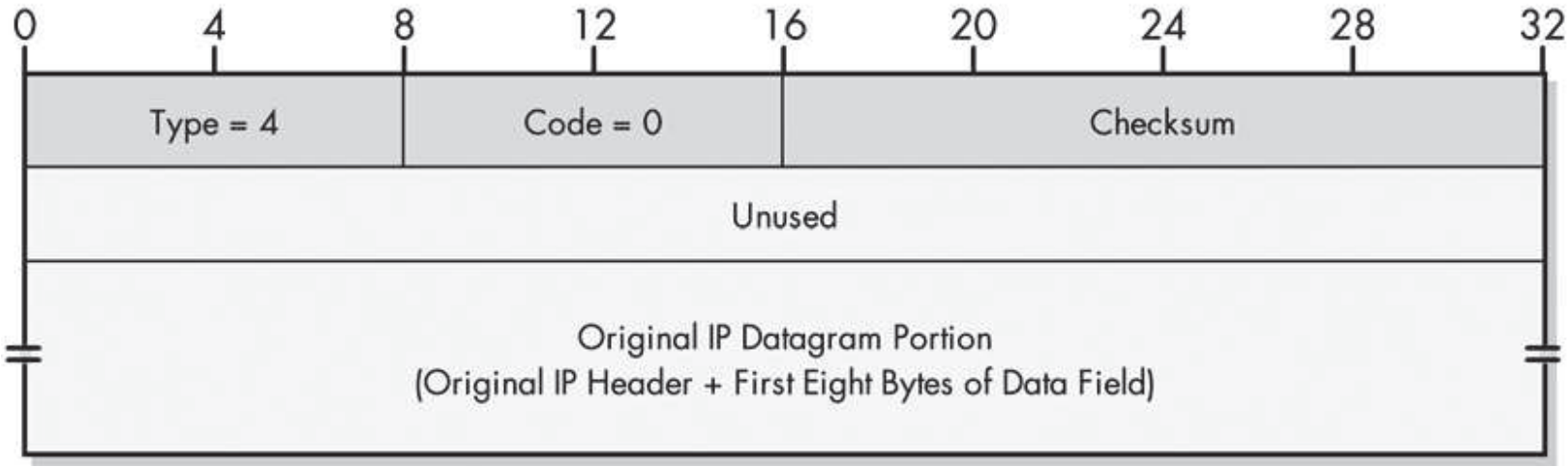
- Los dispositivos de destino reciben datagramas y los almacenan en el buffer para que las aplicaciones retiren sus datos
- Si la tasa de llegada de paquetes empieza a ser mayor que la tasa de lectura de las aplicaciones, el buffer eventualmente se va a quedar sin espacio
- Cuando el buffer se llena, los datagramas que no pueden ingresar se dropean
- Eventualmente la capa superior volverá a enviarlos y mientras el buffer siga lleno se seguirán perdiendo
- O puede enviar un mensaje a algunos orígenes que estén congestionando mucho la red para que disminuyan la tasa de envío

Source Quench Messages

Algunos ejemplos

- Un destino recibe muchos datagramas de muchos orígenes: Requests HTTP
- Dispositivo A y B intercambian información pero A envía mucho más rápido que B
- Un router recibe datagramas por un enlace de alta velocidad y necesita reenviarlos por un enlace de baja velocidad
- Una falla de hardware puede hacer que no se procesen determinados datagramas

Source Quench Messages :: Header



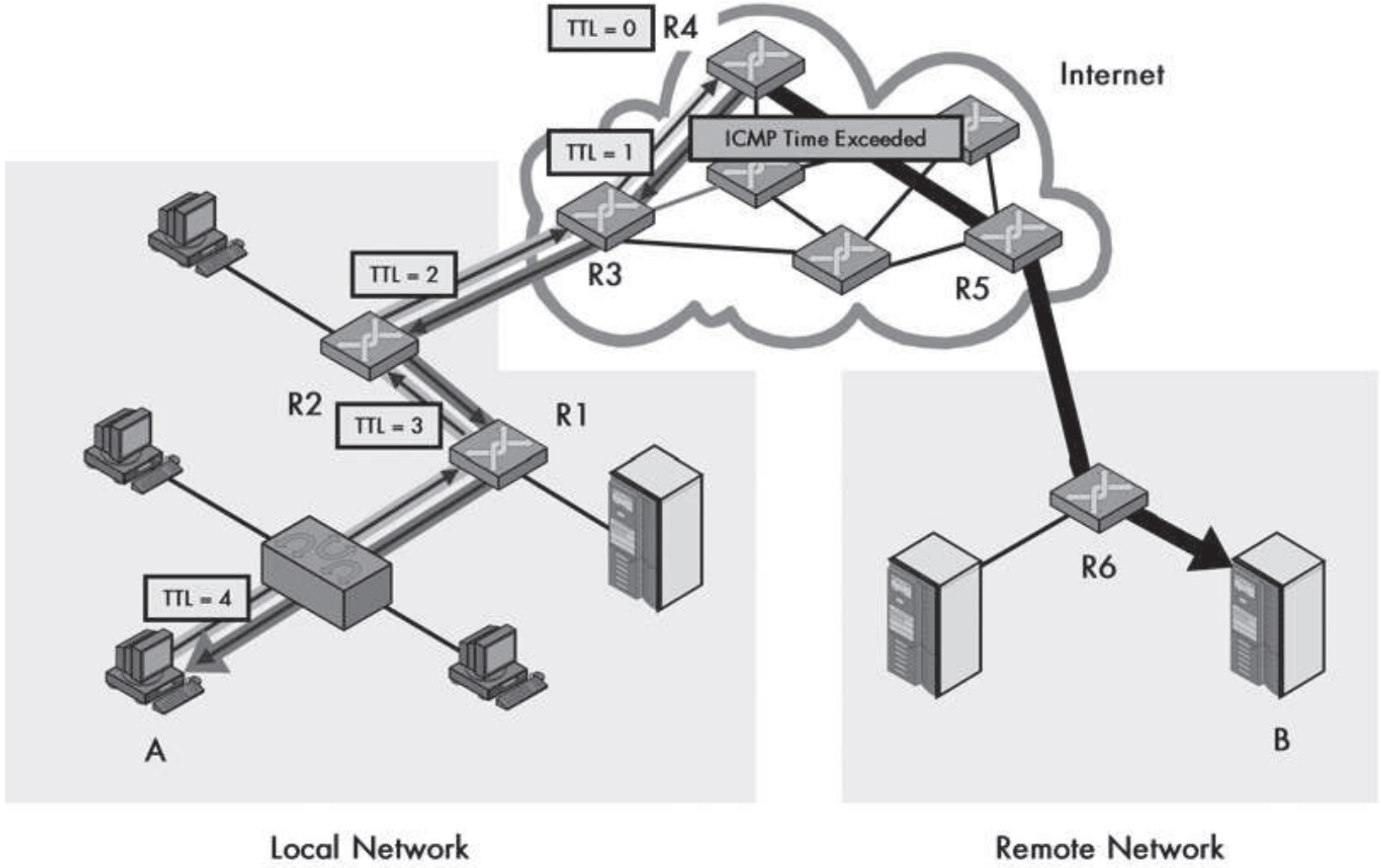
Original Datagram Portion $\rightarrow$ La cabecera IP completa y los primeros 8 bytes de datos para que el origen reconozca el datagrama

Time Exceeded Messages

Time Exceeded Messages

- Si una red está mal configurada podría pasar que determinados datagramas entren en un loop y nunca llegar a destino
- Para evitar eso se asigna a cada datagrama un tiempo de vida en el que puede circular en la red
- Pasado ese tiempo el datagrama se dropea
- El tiempo de vida (TTL) se mide en cantidad de saltos entre hosts
- cuando `TTL = 0` se dropea y se envía un mensaje ICMP de tiempo excedido
- El host de origen podría, por ejemplo, incrementar el TTL

Time Exceeded Messages



Time Exceeded Messages

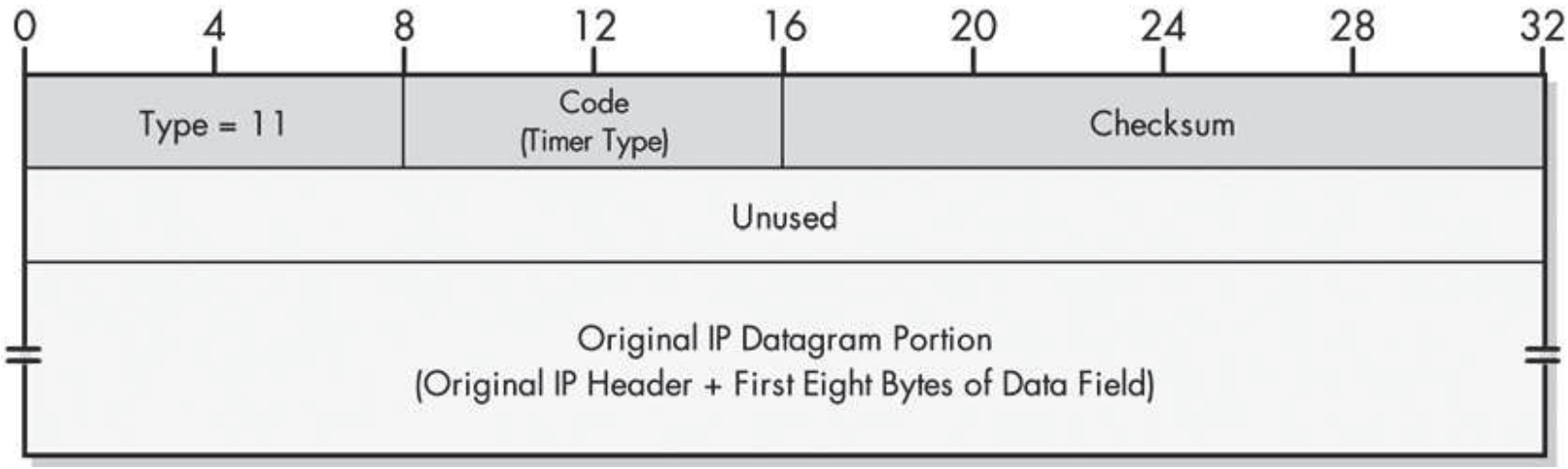
Aplicación estrella

Traceroute

Otro uso importante

Cuando un datagrama IP tiene que ser fragmentado, el destino tendrá que rearmar los fragmentos, que circulan por diferentes redes. Para evitar esperar eternamente todos los fragmentos, el host destino pone un timer cuando llega el primer fragmento. Si se vence el timer se descartan todos los fragmentos del datagrama y se envía una mensaje ICMP de tiempo expirado.

Time Exceeded Messages :: Header



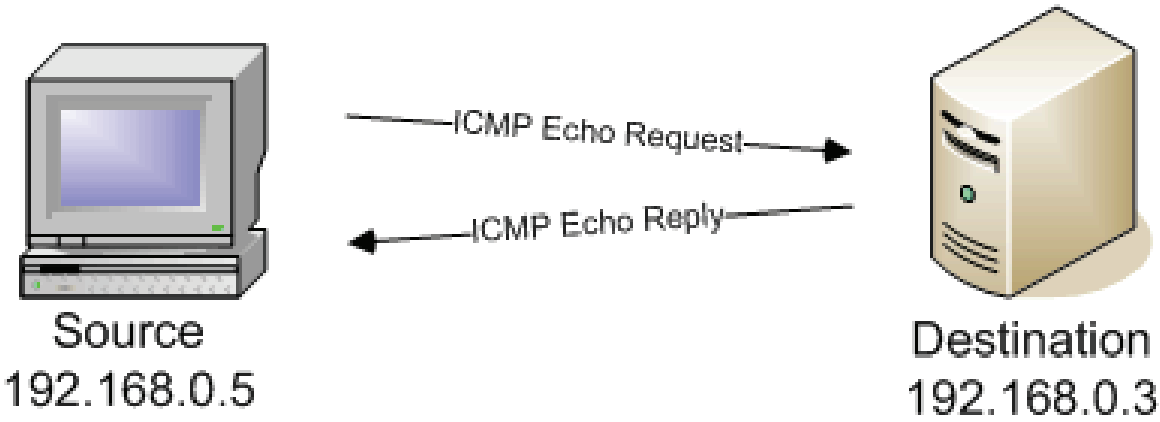
Code $= 0 \rightarrow$ Expiración del TTL de la cabecera IP

Code $= 1 \rightarrow$ Tiempo de espera de reensamblado excedido

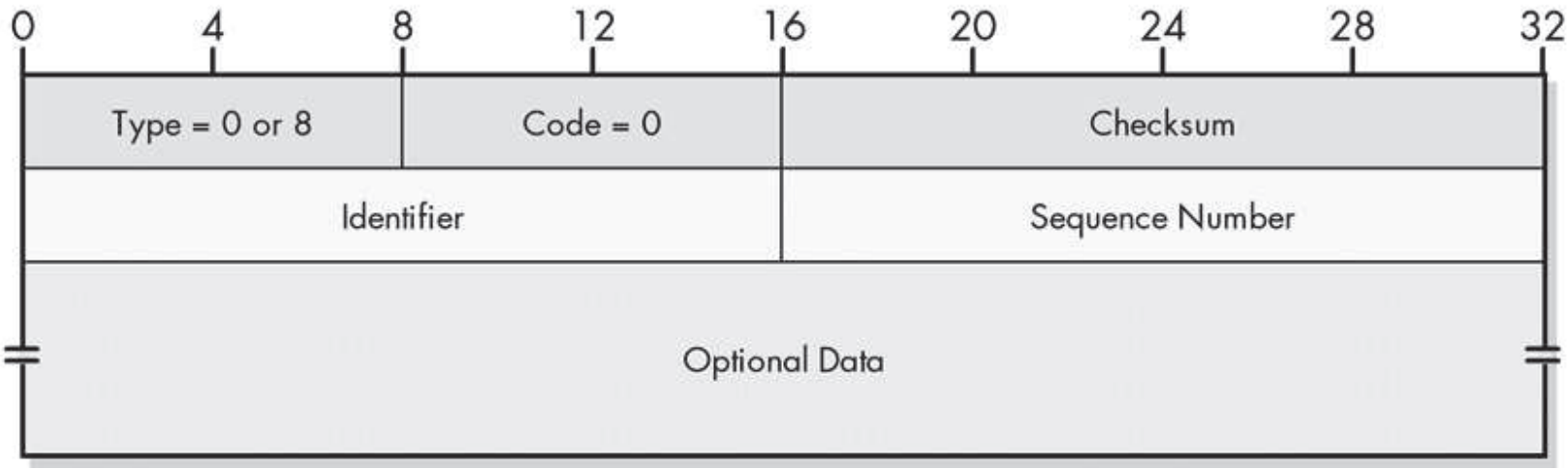
Echo (Request) & Echo Reply

La información primaria para cualquier host es saber si se puede comunicar con otros hosts.

1. *Source* le manda un mensaje de prueba a *Destination* y quiere saber si le llegó
2. *Destination* le responde que sí, que le llegó
3. *Source* ahora sabe que puede intercarbiar mensajes con *Destination*



Echo (Request) & Echo Reply :: Header



Type $= 0 \rightarrow$ Echo Reply

Type $= 8 \rightarrow$ Echo (Request)

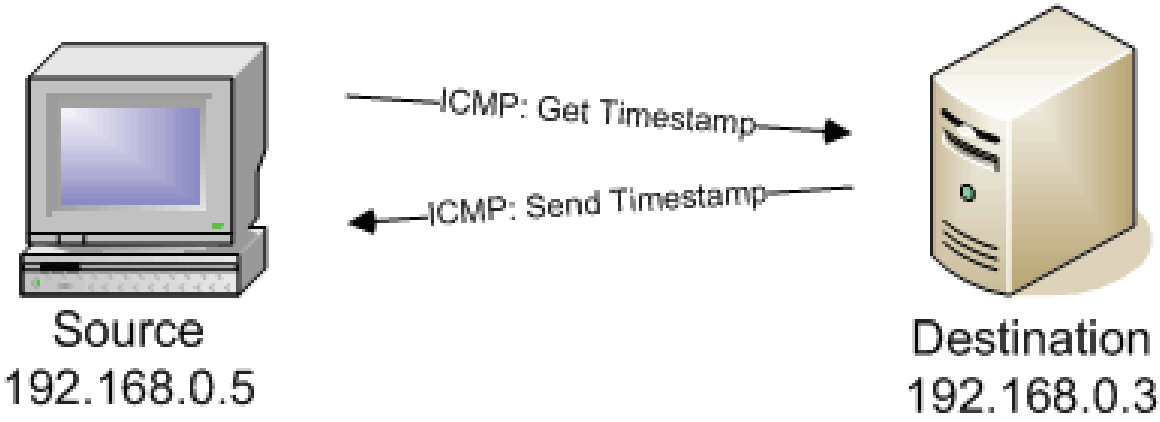
Aplicación estrella

Ping

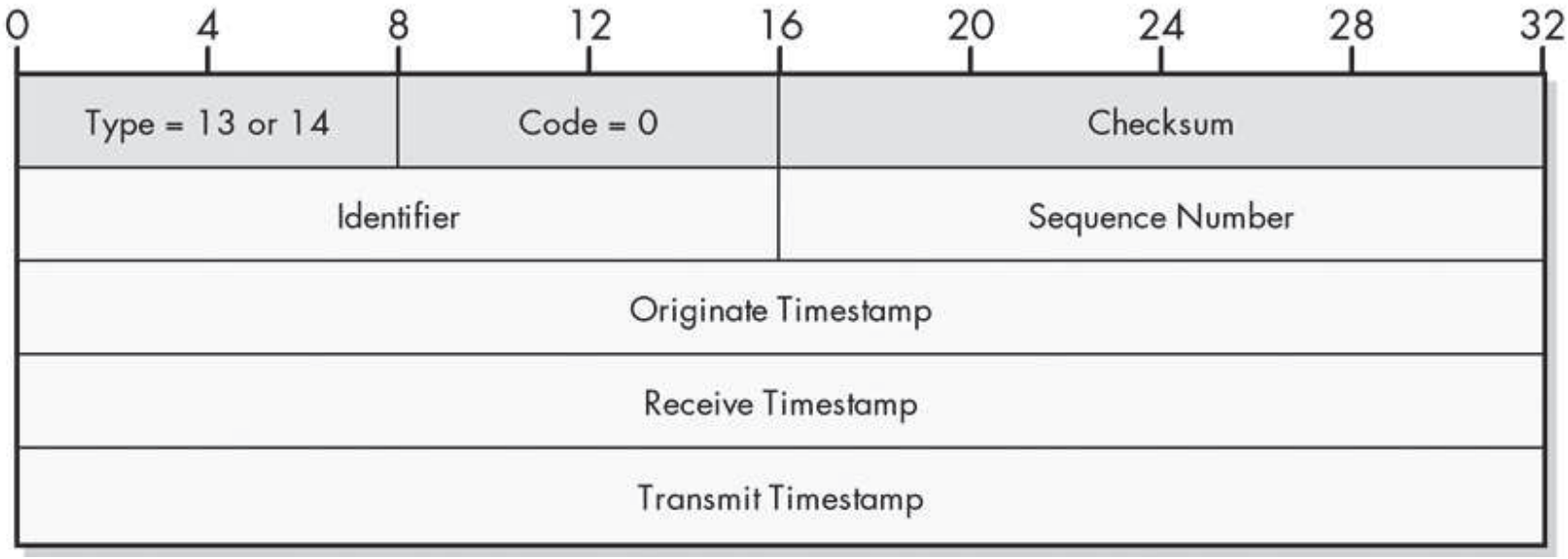
Timestamp (Request) and Timestamp Reply

Sincronizando relojes

1. *Source* manda a *Destination* un datagrama con su `marca de tiempo` justo antes de despacharlo
2. *Destination* pone su `marca de tiempo` justo al momento de recibirlo, lo procesa y le hace otra `marca de tiempo` justo antes de despacharlo de vuelta
3. *Source* sincroniza su reloj en base a las `marcas de tiempo`



Timestamp (Request) and Timestamp Reply :: Header



Type $= 13 \rightarrow$ Timestamp (Request)

Type $= 14 \rightarrow$ Timestamp Reply

Timestamps $\rightarrow$ Universal Time (UT)

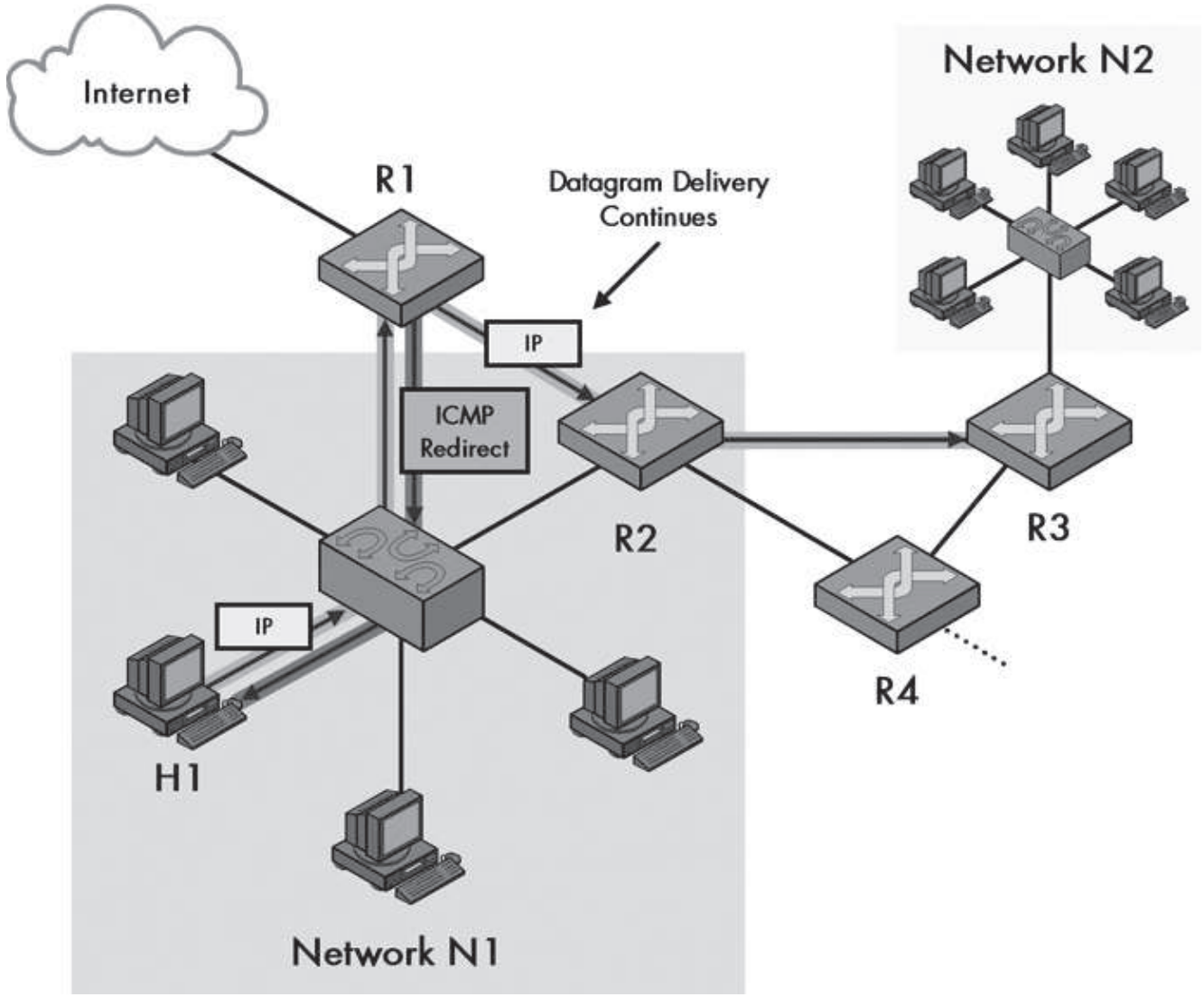
(aka GMT Greenwich Mean Time)

Timestamp (Request) and Timestamp Reply

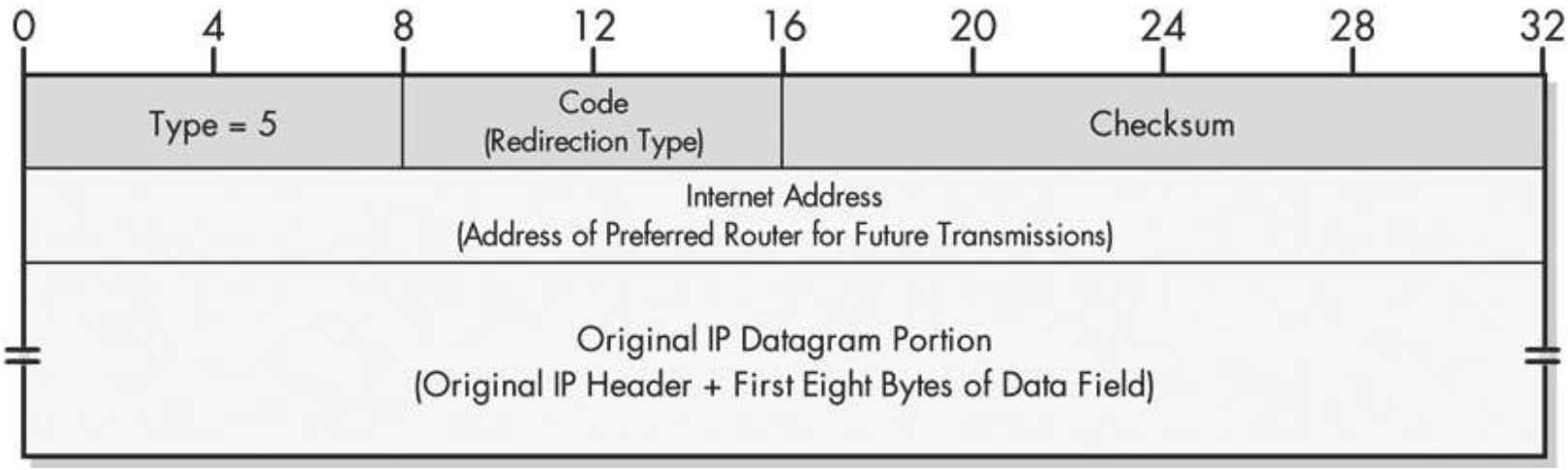
Problemas

En la práctica no funciona tan bien, sobre todo en redes grandes y congestionadas. Los host modernos a veces utilizan métodos más sofisticados para sincronizar relojes, como Network Time Protocol (NTP)

Redirect


Internet
R1
Datagram Delivery
Continues
Network N2
IP
ICMP
Redirect
R2
R3
IP
R4
H1
Network N1

Redirect :: Header



Internet Address → IP del router al que tiene que mandar los futuros datagramas si manda a esa IP destino

Code=0 → Redirigir datagramas a la red, no sólo a este router (obsol.)

Code=1 → Redirigir sólo los datagramas correspondientes al host destino

Code=2 → Redirigir datagramas que tengan este ToS (obsol.)

Code=3 → Redirigir sólo los datagramas correspondientes al host destino y que tengan este ToS

Mensajes de error :: Sumario

Type (8 bit) $\rightarrow$ 256 tipos de mensajes
Code (8 bit) $\rightarrow$ 256 códigos por tipo (65536 combinaciones)

ICMPv4 Error Messages			
Tipo	**Nombre**	**Descripción**	**RFC**
3	Destination Unreachable	El paquete no puede ser entregado a nivel red, host, protocolo, puerto, etc.	792
4	Source Quench	Informa al host de origen que debe reducir la tasa de envío	792
5	Redirect	Permite al router informar al host que hay una ruta mejor	792
11	Time Exceeded	Se usa para informar que el datagrama fue descartado por TTL = 0	792
12	Parameter Problem	Error de puntero (0), Falta opción requerida (1), Mala longitud (3)	792

Mensajes de información :: Sumario

ICMPv4 Information Messages			
Tipo	**Nombre**	**Descripción**	**RFC**
0	Echo Reply	Enviado en respuesta a un Requerimiento de Eco (ping). Se usa para testear conectividad.	792
8	Echo (Request)	Enviado a un dispositivo para testear conectividad (ping)	792
9	Router Advertisement	Usado por los router para promocionar sus capacidades y rutas.	1256
10	Router Solicitation	Lo utilizan los hosts para solicitar un router para enviar una Publicidad de Router.	1256
13	Timestamp (Request)	Lo mandan los dispositivos para sincronización	792
14	Timestamp Reply	Respuesta al timestamp request	792
15	Information Request	Obsoleto. Se usaba para pedir información a otro dispositivo	792

Mensajes de información :: Sumario (cont.)

ICMPv4 Information Messages			
Tipo	**Nombre**	**Descripción**	**RFC**
16	Information Reply	Obsoleto. Se usaba para proveer información en respuesta al pedido	792
17	Address Mask Request	Pide la máscara de subred	950
18	Address Mask Reply	Responde con la máscara de subred	950
30	Traceroute	Es un protocolo de traceroute más eficiente que utilizar TTLs	1393

Referencias

 RFC 792 *Internet Control Message Protocol*

 RFC 950 *Internet Standard Subnetting Procedure*

 RFC 1256 *ICMP Router Discovery Messages*

 RFC 1393 *Traceroute Using an IP Option*

 RFC 1716 *Towards Requirements for IP Router*

 RFC 1812 *Requirements for IP Version 4 Routers*

 The TCP/IP Guide

Internet Control Message Protocol (ICMP/ICMPv4 & ICMPv6)

http://www.tcpipguide.com/free/t_InternetControlMessageProtocolICMPICMPv4andICMPv6.htm

PING

Ping es una herramienta de diagnóstico para verificar la conectividad entre dos computadoras en una red. Envía paquetes ICMP con Respuesta de Eco a una dirección IP remota y observa las respuestas ICMP.

- Como programa, ping es una utilidad diagnóstica

- En redes de computadoras comprueba el estado de la conexión del host local con uno o varios equipos remotos de una red TCP/IP por medio del envío de paquetes ICMP de solicitud y de respuesta.

- Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.

- Ejecutando Ping de solicitud, el Host local envía un mensaje ICMP, incrustado en un paquete IP. El mensaje ICMP de solicitud incluye, además del tipo de mensaje y el código del mismo, un número identificador y una secuencia de números, de 32 bits, que deberán coincidir con el mensaje ICMP de respuesta; además de un espacio opcional para datos.

Detalles técnicos

La utilidad Ping trabaja en la capa de red del protocolo TCP/IP y es un tipo de mensaje de control del protocolo ICMP, subprotocolo de IP. El funcionamiento de Ping y del protocolo ICMP, en general, están definidos en la RFC 792.

El protocolo IP encapsula el mensaje ICMP dentro de un paquete y lo envía. Suele llamarse *Paquete ICMP*. En el paquete pueden distinguirse dos conjuntos de datos: La Cabecera IP, que contiene los datos estándar de la Capa de red, y el subpaquete ICMP, que contiene los datos de control. En la Cabecera IP se especifican los valores *protocolo* como 1 y *tipo de servicio* como 0 de forma obligatoria. En el subpaquete ICMP se especifican los valores *tipo de mensaje ICMP* a 8 (petición) ó 0 (respuesta) y *code* a 0 (en ambos casos).

Paquete ICMP

	Bit 0 - 7	Bit 8 - 15	Bit 16 - 23	Bit 24 - 31
Encabezado IP (20 bytes)	Versión/IHL	Tipo de servicio	Longitud	
	Identificación		*flags* y *offset*	
	Tiempo de vida (TTL)	Protocolo	Checksum	
	Dirección IP origen			
	Dirección IP destino			
ICMP Carga (8 + bytes)	Tipo de mensaje	Code	Checksum	
	Identificador + Secuencia numérica			
	Datos (*opcional*)			

El total de la cabecera IP no podrá superar los 160 bits (20 bytes), tras la cual se situará el mensaje ICMP, con un tamaño estándar de 64 bits (8 bytes)

Composición de un paquete ICMP

- A partir del cuadro anterior podemos desglosar los siguientes valores propios de un paquete ICMP, en la carga ICMP:
- Cabecera IP:
  - Protocolo cambia a 1 y Tipo de servicio a 0, según IANA en la especificación de Números de protocolo IP. 8 bits.
  - Variables estándar del protocolo IP.
- Carga ICMP:
  - Tipo de mensaje y código ICMP. Especificado por IANA en ICMP Parameters. Cada uno 8 bits, ambos 2 bytes.
  - Checksum, calculado en base a la carga ICMP y excluyendo la cabecera IP. 16 Bits.
  - Identificador y Secuencia numérica. Cada uno 16 bits, ambos 32 bits.
  - Datos. Contenido opcional y tamaño arbitrario.

Variables ICMP en petición

- Una petición Ping (*echo request*) no es más que un mensaje ICMP enviado a un host determinado con expectativas de recibir de él una *respuesta Ping*. Las variables propias de la carga ICMP en petición son:
- Tipo de mensaje, definido obligatoriamente como 8.
- Código ICMP, definido obligatoriamente como 0.
- Identificador
- Secuencia numérica
- Datos: Variables y opcionales
- El identificador y la secuencia numérica pueden coincidir en la respuesta. Los datos de la petición deben obligatoriamente coincidir con los de la respuesta.

Variables ICMP en respuesta

- Una respuesta (*echo reply*) no es más que un mensaje ICMP enviado a un host determinado como respuesta a una *petición PING*.Las variables propias de la carga ICMP en respuesta son:
- Tipo de mensaje, definido obligatoriamente como 0.
- Código ICMP, definido obligatoriamente como 0.
- Identificador
- Secuencia numérica
- Datos: Variables y opcionales
- El identificador y la secuencia numérica pueden coincidir con la del pedido. Los datos de la petición deben obligatoriamente coincidir con los del pedido.

Su uso en consolas de comandos

- Todos los sistemas operativos y plataformas incorporan la posibilidad de ejecutar esta utilidad mediante la utilización de comandos.

Sistemas Windows

parametros

- Aplicable todo o en parte en Windows XP, Windows Server 2003, Windows Vista, Windows 7, Windows Server 2008 y derivados. Archivo **ping.exe** en la carpeta del sistema*system32*, invocable desde el símbolo del sistema mediante el comando *ping*, con los siguientes parámetros:
- **/t**:Hace el ping al host hasta que se detiene.
- **/a:**: Resuelve la dirección como nombre de host.
- **/l**: Especifica el tamaño del paquete ICMP en bytes, con un máximo de 65527 bytes.
- **/f**: Especifica que los paquetes ICMP no deben fragmentarse.
- **/i**: Especifica el TTL (tiempo de vida) de los paquetes enviados ICMP, con un valor estándar en equipos con Windows XP (host), esto es típicamente de 128 y un máximo de 255.

Sintaxis

- La sintaxis utilizada para el comando Ping es la misma que para el resto de comandos en Windows.
  ping <*ip*> /*parámetro* valor /*parametro2* valor …
  Donde ip es una variable obligatoria y que es sustituida por la dirección IP o la dirección DNS del host

Petición a un dominio

- ping nombredeldominio.tld /l 64 /i 250

Petición a un dominio

- ping nombredeldominio.tld /l 64 /i 250
  En el ejemplo se observa la utilización de una dirección DNS o nombre de dominio en lugar de una dirección IP. Se añaden los parámetros *l* e *i*, que determinan el tamaño del paquete a 64 bytes y el Tiempo de vida (TTL) a 250 milisegundos.

Petición a una dirección IP

- ping 192.168.0.1 /i 147 /a

Petición a una dirección IP

- ping 192.168.0.1 /i 147 /a
  En el ejemplo se utiliza una dirección IP local.
  Se especifica el Tiempo de vida (TTL) a
  147 milisegundos y se exige que se resuelva
  como nombre de host.

Sistemas GNU/Linux

Parámetros

- Aplicable a todas las distribuciones Linux (Debian, Knoppix, Red Hat Linux y derivadas).
- *"-i:"* Espera x segundos entre el envío de cada paquete ICMP. El tiempo estándar es 1 segundo. También sirve para, en el caso de que el host origen tenga más de un interfaz, identificar por que interfaz se realizará el ping.
- *"-c número:"* Especifica el número de pings a hacer, por defecto es infinito, o hasta que se detenga al programa, Esta opción permite una vez que se haya pasado el número de pings especificados, se detenga.
- *"-s:"* Especifica el tamaño de la porción de datos del paquete ICMP. El tamaño estándar es 56 bytes de datos (+ 20 bytes fijos de la cabecera IP + 8 bytes de la cabecera ICMP, en total 84 bytes).
- *"-l preload:"* Especifica que los paquetes ICMP deben ser enviados lo más rápido posible.
- *"-t:"* Especifica el tiempo de vida (TTL) de los paquetes a enviar. El tiempo de vida estándar variará según la versión de sistema operativo, siendo el máximo en todos los casos de 255.
- *"-n:"* Especifica que no habrá salida a nombre de host DNS, solo numérica (dirección IP).

Sintaxis

- La sintaxis utilizada para el comando Ping es la misma que para el resto de comandos en Linux.
  ping <ip> -*parámetro* valor -*parametro2* valor
  ...
  Donde ip es una variable obligatoria y que es sustituida por la dirección IP o la dirección DNS del host.

Petición a un dominio

- ping nombredeldominio.tld -i 200 -t 15

Petición a un dominio

- ping nombredeldominio.tld -i 200 -t 15
  En el ejemplo se observa la utilización de una dirección DNS o nombre de dominio en lugar de una dirección IP. Se añaden los parámetros i y t, que determinan el tiempo de espera para el envío de cada paquete (200 segundos) y el tiempo de vida (TTL) del mismo (15 equipos).

Petición a una dirección IP

- ping 192.168.0.1 -l preload

Petición a una dirección IP

- ping 192.168.0.1 -l preload
  En el ejemplo se utiliza una dirección IP local.
  Se exige que los paquetes se envíen lo más rápido posible

PING de la muerte

Decidir si se debe permitir el acceso a paquetes ICMP a traves de un firewall es una desición dificil de tomar. Ciertamente hay muy buenos usos para ICMP, pero tambien existen ataques basados en ICMP ,por ejemplo:

Un **ping de la muerte** es un tipo de ataque enviado a una computadora que consiste en mandar numerosos paquetes ICMP muy grandes (mayores a 65.535 bytes) con el fin de colapsar el sistema atacado.

Es un tipo de ataque a computadoras que implica enviar un ping deformado a una computadora. Un ping normalmente tiene un tamaño de 64 bytes; algunos sistemas operativos no podían manejar pings mayores al máximo de un paquete IP común, que es de 65.535 bytes. Enviando pings de este tamaño era posible hacer que esas computadoras dejaran de funcionar.

PING de la muerte

```
Haciendo ping a 10.22.3.71 con 6000 bytes de datos:
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
Respuesta desde 10.22.3.71: bytes=6000 tiempo<1m TTL=128
```

Verificación del funcionamiento de una red

- El comando ping, a pesar de su sencillez, es una eficaz ayuda para la verificación de redes durante su configuración y para la detección de fallos en la misma.
  Asumamos como ejemplo que hemos configurado una red con una dirección IP privada 192.168.1.0
  La misma está conectada a Internet a través de una puerta de enlace con dirección IP 192.168.1.1
  La verificación la haremos desde una PC a la cual le asignamos manualmente la dirección IP 192.168.1.100, estando conectada en el mismo tramo físico otra PC con la dirección IP 192.168.1.101.

Comandos MS-DOS de Red

hostname: Muestra el nombre de la computadora que estamos utilizando.

ipconfig: Muestra y permite renovar la configuración de todos los interfaces de red.
ipconfig/all: Muestra la configuración de las conexiones de red.

net: Permite administrar usuarios, carpetas compartidas, servicios, etc.

- **net view:** muestra las computadoras conectadas a la red.
- **net share:** muestra los recursos compartidos del equipo, para la red.
- **net user:** muestra las cuentas de usuario existentes en el equipo.
- **net localgroup:** muestra los grupos de usuarios existentes en el equipo.

Netsh es una utilidad de línea de comandos que ofrece varias opciones para la configuración de una red.

Netstat :(*network statistics*) es una herramienta de línea de comandos que muestra un listado de las conexiones activas de una computadora, tanto entrantes como salientes.

Nslookup: funciona tanto en Windows como en UNIX para obtener la dirección IP

Referencias

- «Operation of the ping Utility». «http://www.tcpipguide.com/free/t_TCPIPCommunicationVerificationUtilitypingping6-2.htm».
- «Operation of the ping Utility». «The ping utility is implemented using ICMP Echo (Request) and Echo Reply messages».
- «Operation of the ping Utility».
- «Packet Internet Groper».
- Michael John Muuss (Mike Muuss). «The Story of the PING Program». United States Army Research Laboratory. Consultado el 21 de marzo de 2012. «From my point of view PING is not an acronym standing for Packet InterNet Grouper, it's a sonar analogy.».
- «Communication Verification Utility». «and is present in just about every TCP/IP implementation.».
- RFC 792
- «INTERNET CONTROL MESSAGE PROTOCOL - DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION». «Type of Service - 0 and Protocol - 1 (ICMP).».
- «INTERNET CONTROL MESSAGE PROTOCOL - Source Quench Message».
- «ICMPv4 Echo and Echo Reply Message Format».
- «ICMPv4 Echo and Echo Reply Message Format».
- . «For Echo messages the value is 8».

- . «for Echo and Echo Reply messages; set to 0.».
- «ICMPv4 Echo and Echo Reply Message Format».
- . «for Echo Reply messages the value is 0.».
- . «for Echo and Echo Reply messages; set to 0.».
- «Ping». «Specifies that reverse name resolution is performed on the destination IP address.».
- «Ping». «Specifies that ping continue sending Echo Request messages to the destination until interrupted.».
- «Ping». «Specifies the length, in bytes, of the Data field in the Echo Request messages sent.».
- «Ping». «Specifies that Echo Request messages are sent with the Don't Fragment flag in the IP header set to 1.».
- «Ping». «Specifies the value of the TTL field in the IP header for Echo Request messages sent. The default is the default TTL value for the host. For Windows XP hosts, this is typically 128. The maximum TTL is 255.».
- «Comando Ping». Consultado el 8 de septiembre de 2010. «Si se especifica preload, ping envía tantos paquetes tan rápido como le sea posible antes de volver a su comportamiento normal.».
- «Default Time To Live (TTL) values» (en inglés). Consultado el 8 de septiembre de 2010.
- «ping(8) - Linux man page» (en inglés). Consultado el 8 de septiembre de 2010.

Traceroute

Traceroute

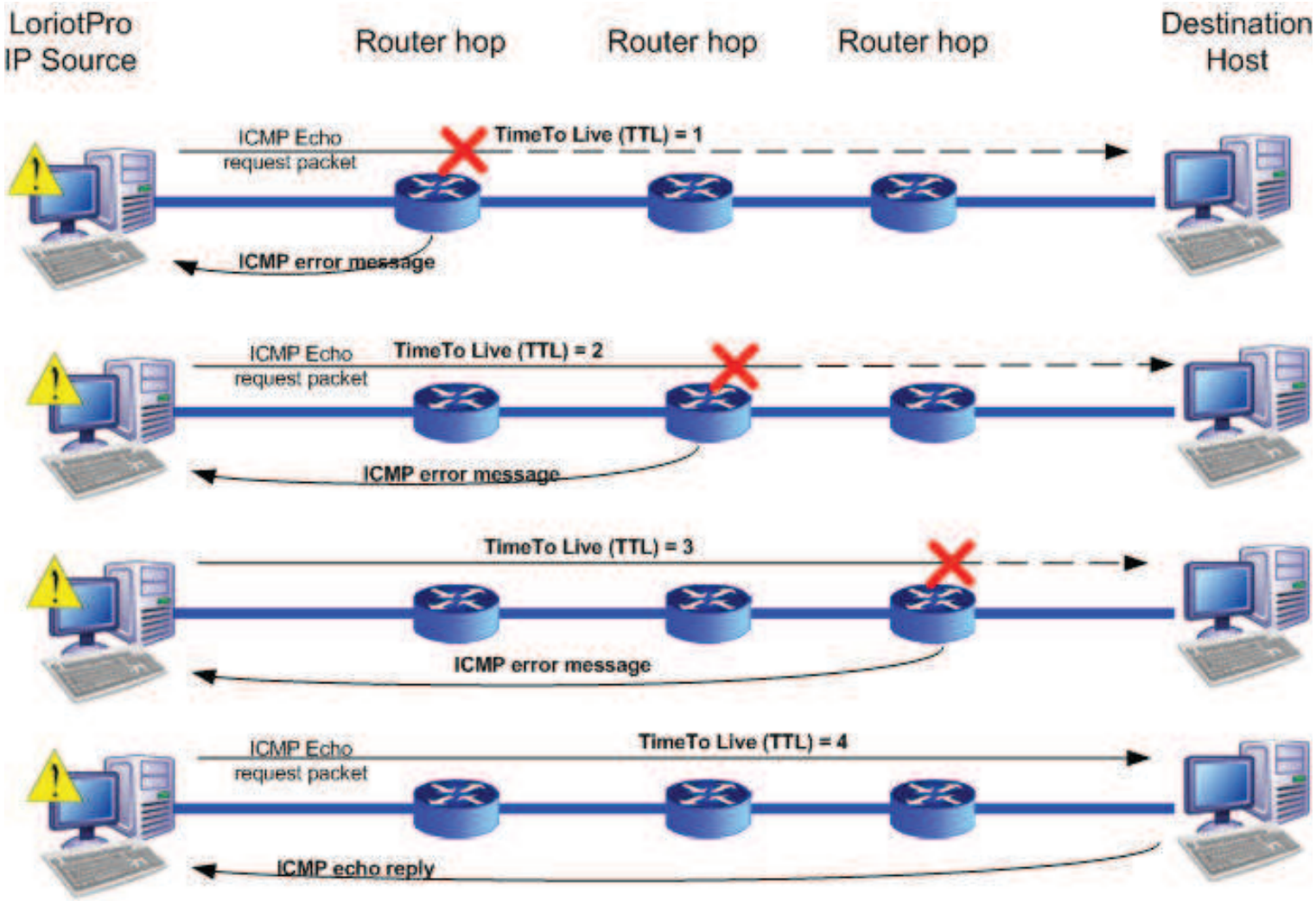
El programa original de traceroute fue escrito por Van Jacobson en 1987, usando un soporte de Kernel ICMP codificado por Mike Muuss, el creador de Ping.

Está definida por la RFC 1393

Las primeras versiones usaban ICMP paquetes, pero debido a restricciones de fabricantes con respecto a la respuesta a paquetes ICMP (IP spec. RFC 791) , se resolvió utilizar UDP.

Cabecera IP

0-3	4-7	8-15	16-18	19-31
Versión	Tamaño Cabecera	Tipo de Servicio	Longitud Total	
Identificador			Flags	Posición de Fragmento
Tiempo de Vida		Protocolo	Suma de Control de Cabecera	
Dirección IP de Origen				
Dirección IP de Destino				
Opciones				Relleno


LoriotPro
IP Source
Router hop
Router hop
Router hop
Destination
Host
ICMP Echo
request packet
TimeTo Live (TTL) = 1
ICMP error message
ICMP Echo
request packet
TimeTo Live (TTL) = 2
ICMP error message
TimeTo Live (TTL) = 3
ICMP error message
ICMP Echo
request packet
TimeTo Live (TTL) = 4
ICMP echo reply
LUTEUS Copyrights 2008

Traceroute por ICMP

Envía paquetes ICMP echo request para obtener del destino un paquete echo reply, al igual que ping.

Es el método utilizado por defecto en tracert de Windows.

También puede utilizarse en Linux mediante la extensión icmp -I

Suele ser bloqueado por la mayoría de los firewalls.

Traceroute por UDP

Envía paquetes UDP a un puerto no convencional, empezando por el 33434 e incrementando En 1 en cada envío.

Como dicho puerto no suele estar en uso, el destino Contesta con un mensaje ICMP unreach port.

Es el método utilizado por defecto en traceroute de Linux y puede ser utilizado por cualquier usuario

Traceroute por TCP

Si hay filtros o firewalls en el camino, tanto paquetes UDP a puertos en desuso o paquetes ICMP de echo request van a ser bloqueados.

Este método envía un paquete TCP syn como si iniciara una conexion normal al puerto 80 del destino.

Si el destino tiene el puerto cerrado responde con un 1 en el flag reset lo que culmina la conexión.

Si el puerto está abierto, responde normalmente con un TCP syn+ack.

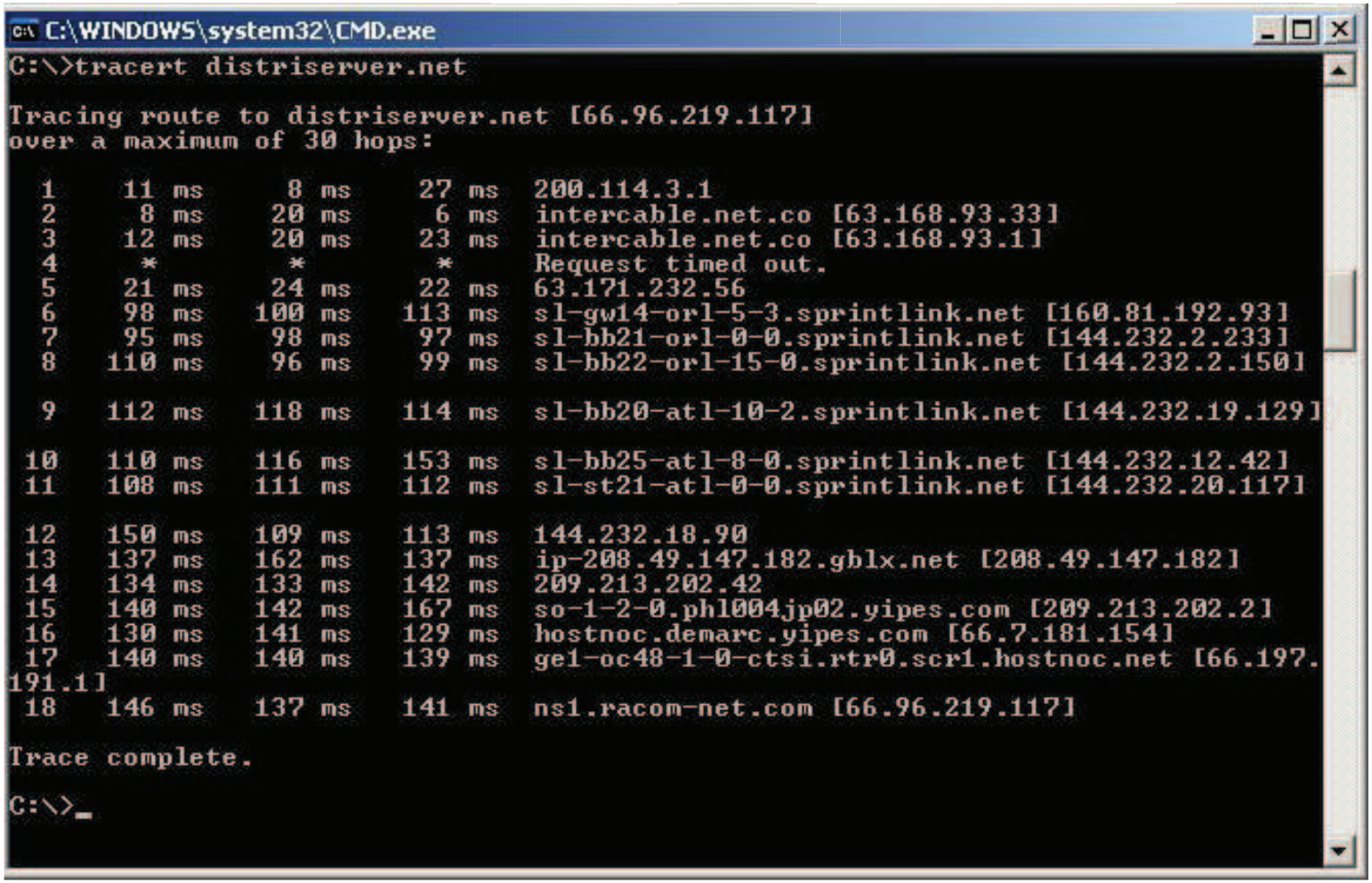
En este caso el origen en vez de enviar un ack, envía un TCP reset para cerrar la conexion.

Tracert (Windows) Sintaxis

tracert [**-d**] [**-h** *MaximumHops*]
[**-j** *HostList*] [**-w** *Timeout*] [*TargetName*]

[-d]	Evita que tracert resuelva los IP de los routers y puntos intermedios con sus nombres. Esto hace que la resolución sea más rápida
[-h]	Define la cantidad máxima de saltos a efectuar hasta llegar a destino. Por defecto son 30.
[-j]	Permite especificar una lista de puntos intermedios a recorrer hasta el destino. Se colocan las ip con un máximo de 9.
[-w]	Especifica el tiempo de espera del mensaje de ICMP antes de resolver la ip y pasar al siguiente paso. Por defecto es 4s.

Ejemplo Tracert (Windows)



Ejemplo Traceroute (Linux)

traceroute [opciones] **host** [**longitud_paquete**]

traceroute www.google.es

```
traceroute to www.google.es (209.85.146.106), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  3.079 ms  4.305 ms  4.345 ms
 2  10.169.64.1 (10.169.64.1)  12.455 ms  15.702 ms  20.226 ms
 3  10.175.240.153 (10.175.240.153)  21.071 ms  21.447 ms  21.436 ms
 4  10.254.3.33 (10.254.3.33)  29.421 ms  29.965 ms  30.027 ms
 5  10.254.1.153 (10.254.1.153)  35.352 ms 10.254.1.229 (10.254.1.229)  61.411 ms
 6  10.254.3.222 (10.254.3.222)  39.790 ms 10.254.3.226 (10.254.3.226)  47.875 ms
 7  216.239.43.233 (216.239.43.233)  50.272 ms  62.717 ms  62.748 ms
 8  209.85.249.31 (209.85.249.31)  54.304 ms  54.335 ms  53.935 ms
 9  216.239.46.85 (216.239.46.85)  56.403 ms  70.881 ms 216.239.46.73 (216.239.46.
10  bru01s01-in-f106.1e100.net (209.85.146.106)  63.395 ms  63.001 ms  63.353 ms
```

Ejemplo Traceroute (Linux)

traceroute –m 10 www.google.es 30

```
traceroute to www.google.es (209.85.229.147), 10 hops max, 30 byte packets
 1  192.168.1.1 (192.168.1.1)  1.172 ms  1.599 ms  2.000 ms
 2  10.169.64.1 (10.169.64.1)  10.804 ms  12.149 ms  15.708 ms
 3  10.175.240.121 (10.175.240.121)  16.948 ms  17.277 ms  21.097 ms
 4  10.175.242.9 (10.175.242.9)  36.586 ms  36.959 ms  36.941 ms
 5  10.254.3.37 (10.254.3.37)  56.454 ms  56.768 ms  56.750 ms
 6  10.254.1.153 (10.254.1.153)  38.950 ms 10.254.2.226 (10.254.2.226)  33.961 ms
 7  10.254.3.222 (10.254.3.222)  27.469 ms  31.312 ms  31.187 ms
 8  209.85.252.83 (209.85.252.83)  52.867 ms 216.239.49.45 (216.239.49.45)  58.824
 9  209.85.243.85 (209.85.243.85)  61.550 ms  61.536 ms 209.85.243.77 (209.85.243.
10  ww-in-f147.1e100.net (209.85.229.147)  59.687 ms  59.338 ms  61.033 ms
```

Ejemplo Traceroute (Linux)

traceroute –r www.google.es

```
traceroute to www.google.es (209.85.229.147), 30 hops max, 60 byte packets
connect: La red es inaccesible
```

traceroute –r 192.168.1.1

```
traceroute to 192.168.1.1 (192.168.1.1), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.968 ms  2.487 ms  3.111 ms
```

traceroute –p 80 192.168.1.1

```
traceroute to 192.168.1.1 (192.168.1.1), 30 hops max, 60 byte packets
 1  192.168.1.1 (192.168.1.1)  2.681 ms  2.302 ms  2.705 ms
```

Ejemplo Traceroute (Linux)

traceroute –w 0 www.google.es

```
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Ejemplo Traceroute (Linux)

traceroute –n www.google.es

```
traceroute to www.google.es (209.85.229.147), 30 hops max, 60 byte packets
 1  192.168.1.1  1.529 ms  1.821 ms  2.208 ms
 2  10.169.64.1  11.939 ms  12.223 ms  16.056 ms
 3  10.175.240.121  17.059 ms  17.520 ms  21.281 ms
 4  10.175.242.9  36.325 ms  36.647 ms  36.700 ms
 5  10.254.3.37  29.431 ms  29.554 ms  30.165 ms
 6  10.254.1.153  41.295 ms 10.254.2.226  39.704 ms 10.254.1.153  39.732 ms
 7  10.254.3.222  61.906 ms  101.642 ms  101.477 ms
 8  209.85.252.83  55.175 ms 216.239.49.45  107.603 ms  58.978 ms
 9  209.85.243.77  63.623 ms 209.85.243.81  55.872 ms 209.85.243.73  55.697 ms
10  209.85.229.147  53.641 ms  52.128 ms  51.432 ms
```

Ejemplo Traceroute (Linux)

traceroute –v

```
Modern traceroute for Linux, version 2.0.13, Nov 23 2009
Copyright (c) 2008  Dmitry Butskoy,   License: GPL v2 or any later
```

TRACERT.ORG ®

traceroute.org

you get signal

WebSitePulse™
take IT easy™

Referencias

•http://en.wikipedia.org/wiki/Traceroute
•http://security.stackexchange.com/questions/39178/how-does-traceroute-over-tcp-work-what-are-the-risks-and-how-can-it-be-mitig
•http://wikis.uca.es/wikiunix/index.php/Traceroute
•http://www.cisco.com/c/en/us/support/docs/ios-nx-os-software/ios-software-releases-121-mainline/12778-ping-traceroute.html
•http://www.inetdaemon.com/tutorials/troubleshooting/tools/traceroute/definition.shtml
•http://tools.ietf.org/html/rfc1393
•http://linux.die.net/man/8/traceroute