**Práctica 4: El protocolo ICMP**

# Sesión L4

Lectura previa: Kurose 4.4.3 subapartado “Protocolo de mensajes de control ICMP”

En la sesión de laboratorio debes arrancar el ordenador en la partición “**Ubuntu**” e iniciar sesión con tu usuario y contraseña de upvnet.

# Introducción a ICMP

En esta práctica vamos a estudiar el protocolo ICMP (*Internet Control Message Protocol*) y algunas órdenes derivadas de él.

En Internet no disponemos de mecanismos hardware para comprobar la conectividad. Además, el protocolo IP no proporciona herramientas para la detección de fallos y problemas. Así es que se diseñó el protocolo ICMP para permitir a los hosts y routers enviar mensajes de control a otros hosts y routers. Está definido en el RFC 792.

ICMP nos permite saber, por ejemplo, por qué no se ha entregado un datagrama (no hay ruta, el destino no responde, se ha agotado su tiempo de vida, etc.). Informa de errores sólo al origen del datagrama. Además, no se encarga de corregir el problema, sólo de avisar.

Como los mensajes ICMP pueden generarse en el exterior de la red IP donde se generó el datagrama original necesitan viajar en el campo de datos de un datagrama IP, pero ICMP no se considera un protocolo de nivel superior a IP, sino de nivel de red.

Cabecera ICMP Mensaje ICMP

|  |  |
| --- | --- |
| Cabecera IP | Campo de datos del datagrama IP |

Campo de datos de la trama

Cabecera

de la trama

CRC

Cada mensaje ICMP tiene su propio formato, pero todos comienzan con los mismos campos:

* Tipo (8 bits): Identifica el tipo de mensaje
* Código (8 bits): Más información sobre el tipo de mensaje • *Checksum* (16 bits): Utiliza el mismo algoritmo que IP.

0 8 16 31

|  |  |  |
| --- | --- | --- |
| Tipo | Código | Checksum |
|  | Depende del tipo y el código | |

El tipo de mensaje determina su significado y su formato. Hay 15 tipos distintos. Entre los principales tenemos:

* Tipo = 0. Respuesta de eco.
* Tipo = 3. Destino inalcanzable.
* Tipo = 8. Petición de eco.
* Tipo = 11. Tiempo de vida excedido en datagrama (TTL=0).

Los mensajes de error contienen la cabecera IP y los ocho primeros bytes de datos del datagrama original. Hay que señalar que esa información contiene las direcciones IP fuente y destino, así como los puertos fuente y destino del datagrama que ha causado el error.

Para evitar problemas en la red, en particular *broadcast storms,* nunca se generan mensajes de error en respuesta a:

* Un mensaje de error ICMP.
* Un datagrama destinado a una dirección IP de difusión.
* Un fragmento que no sea el primero.
* Un datagrama cuya dirección origen no defina una conexión de red única (es decir, que la dirección origen no puede ser cero, la dirección de *loopback*, direcciones de difusión).

## Mensajes ICMP de eco

La respuesta a una petición de eco devuelve los mismos datos que se recibieron en la petición. Estos mensajes se utilizan para construir la herramienta *ping*, empleada por administradores y usuarios para detectar problemas en la red.

Permite:

* Comprobar si un destino está activo y si existe una ruta hasta él.
* Medir el tiempo de “ida y vuelta”.
* Estimar la fiabilidad de la ruta.
* Puede ser utilizado tanto por hosts como por routers.

## Mensajes ICMP de tiempo excedido

Este tipo de mensajes pueden ser enviados por routers y por hosts:

* Routers: cuando descartan un datagrama al llegar a cero su tiempo de vida.
* Hosts: al vencer un temporizador mientras esperan todos los fragmentos de un datagrama. El campo código explica cuál de los dos sucesos ha ocurrido.

En estos mensajes se apoya la orden *traceroute*, que se estudiará después.

## Mensajes de destino inalcanzable

Son enviados por un router o un host cuando no puede enviar o entregar un datagrama IP.

Se envían al emisor inicial del datagrama.

El campo código contiene un entero con información adicional. Los más importantes son:

* Código = 0. Red inalcanzable.
* Código = 1. Host inalcanzable.
* Código = 2. Protocolo inalcanzable.
* Código = 3. Puerto inalcanzable. Se genera usualmente cuando se recibe un datagrama UDP destinado a un puerto UDP que está cerrado en el destino.
* Código = 4. Se requiere fragmentación, pero bit DF activado.
* Código = 6. Red destino desconocida.
* Código = 7. Host destino desconocido.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ejercicio 1:**  El computador B ha recibido los datagramas IP mostrados en la tabla, que han sido enviados por el computador A. Durante la recepción de los datagramas los únicos puertos abiertos en B eran los puertos **TCP** 22 y 30.000.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | N.º | Identificador | MF | OFFSET | Long. Total | Protocolo | Tipo (si ICMP) / Puerto si UDP o TCP | | 1 | 1340 | 1 | 185 | 1500 | ICMP | 8 | | 2 | 1341 | 0 | 0 | 877 | UDP | 8.000 | | 3 | 1342 | 1 | 0 | 1500 | TCP | 22 | | 4 | 1340 | 0 | 370 | 78 | ICMP | 8 | | 5 | 1342 | 0 | 185 | 1340 | TCP | 22 |   ◦ ¿Qué datos recibirá el nivel de transporte? Justifica la respuesta. El 1342.   * **El ID 1340** pertenece al protocolo ICMP -> nivel de red. * El ID 1341 utiliza el puerto 8.000, **que no está abierto**. * **En el ID 1342 se envían todos los fragmentos** (último fragmento tiene el bit **MF (*More Fragments*) a 0**), **el puerto (22) está abierto** y **el protocolo TCP pertenece al nivel de transporte.**   ◦ ¿Se generarán mensajes ICMP? Justifica la respuesta. En caso afirmativo indica qué datagrama(s) lo(s) generará(n). Sí. Los mensajes ICMP se generan cuando hay algún tipo de error, avisa de problemas, etc. Es decir, es una forma de verificar que los paquetes se han entregado correctamente.   * Por tanto, **en el ID 1340 se generará un código de error de tipo 0** (la red es inalcanzable, **ya que falta el primer fragmento**, ya que debería empezar con el OFFSET (desplazamiento) a 0. * **En el ID 1341 generará un código de error de tipo 3** (puerto inalcanzable de tipo UDP), **ya que el puerto 8.000 se encuentra cerrado.** * **En el ID 1342 no se generará ningún código de error.** |

# Análisis de la cabecera IP

**Ejercicio 2:**

Inicia el analizador de protocolos *Wireshark* (desde el botón de la barra superior). Si te solicita contraseña teclea la de tu usuario de UPVNet. Captura los paquetes que se generan al cargar en el navegador la página [www.uv.es.](http://www.uv.es/) Utiliza un filtro de captura para eliminar el resto del tráfico. Recuerda que los protocolos de aplicación se filtran indicando el puerto del servidor (port 80 para HTTP y 443 para HTTPS). No olvides comprobar que la interfaz de captura es la adecuada (debe coincidir con la que tiene asociada la dirección IP pública). Detén la captura, analiza los primeros 4 paquetes generados, y responde a las siguientes cuestiones referidas a la cabecera IP de dichos paquetes:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | ***Identificador***  campo **Identification** en IPv4 | ***TTL***  Campo ***Time To Live (IPv4)*** | ***Dirección IP fte.*** | ***Dir. IP destino*** |  |  |
| **Paquete 1** | 0xd906 (55558) | 128 | 192.168.1.134 | 147.156.200.249 |  |
| **Paquete 2** | 0x0000 (0) -> indica que no está fragmentado. | 53 | 147.156.200.249 | 192.168.1.134 |  |
| **Paquete 3** | 0xd907 (55559) | 128 | 192.168.1.134 | 147.156.200.249 |  |
| **Paquete 4** | 0xd909 (55560) | 128 | 192.168.1.134 | 147.156.200.249 |  |
| Con respecto al campo TTL (*Time To Live*) de la cabecera IP de los paquetes capturados:   * ¿Tiene siempre el mismo valor? No, cada ordenador tiene un TTL distinto. Mi ordenador tiene por defecto TTL de 128, pero el servidor los tiene de 53 (todos sus paquetes tienen TTL de 53). * En general, todos los paquetes que envía un ordenador, ¿tienen siempre el mismo TTL inicial? Sí. * ¿Cuál sería el valor inicial del TTL en el paquete 2 (el primero que ha enviado el servidor)? El valor inicial sería 64.   Observa cómo varía el campo identificador en el cliente y en servidor. Describe lo que observas. Anota el valor del campo protocolo. En este caso, ¿a qué protocolo se refiere? Los paquetes del cliente van continuos (se van sumando uno en cada siguiente) y el primero del servidor es 0x0000, que indica que no está fragmentado. **Utilizan el protocolo TCP.** | | | | |  |

# La orden ping

Mediante la orden ***ping*** (quese ejecuta desde un terminal) se obtiene una estimación del tiempo de ida y vuelta de un paquete (RTT), desde la estación origen a una estación destino que se especifica. Para ello se almacena el instante de tiempo en el que se envía el paquete y cuando llega la respuesta al valor almacenado se le resta del tiempo actual. El funcionamiento de la orden ***ping*** se basa en el uso de mensajes ICMP de tipo 0 (*Echo reply*) y 8 (*Echo request*).

Otras utilidades de la orden ***ping*** son:

* Averiguar si un destino está operativo, conectado a la red y sus protocolos TCP/IP en funcionamiento.
* Conocer la fiabilidad de la ruta entre origen y destino (calculando el porcentaje de paquetes que obtienen respuesta).

Ejemplo:

user@rdc14:~$ ping www.uji.es

PING www.uji.es (84.124.83.62) 56(84) bytes of data.

64 bates from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp\_seq=1 ttl=112 time=22.1 ms

64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp\_seq=2 ttl=112 time=21.6 ms

64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp\_seq=3 ttl=112 time=21.6 ms

64 bytes from 84.124.83.62.static.user.ono.com (84.124.83.62): icmp\_seq=4 ttl=112 time=22.0 ms

--- www.uji.es ping statistics --- 4 packets transmitted, 4 received, 0% packet loss, time 3004ms rtt min/avg/max/mdev = 21.646/21.872/22.139/0.217 ms

La orden ***ping*** admite una serie de opciones, algunas de las más útiles se muestran a continuación. Para ver una información más completa puede consultarse el man de la orden.

**ping [-b][-c count] [-s packetsize] [-t ttl] destino**

Opciones:

**-b**  Permite hacer broadcast a una dirección de difusión **-c cantidad** Cantidad de solicitudes de eco a enviar.

1. **tamaño** Número de bytes de datos.
2. **ttl** Tiempo de vida.

Para interrumpir la ejecución del programa ping: hay que presionar Ctrl-C.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ejercicio 3:**  Haz un *ping -c 3* a las direcciones siguientes: zoltar.redes.upv.es (servidor dentro del Laboratorio de Redes), www.upv.es (servidor web de la UPV), www.rediris.es (servidor web de RedIris situado en Madrid), **www.uq.edu.au** (servidor web de la Universidad de Queensland en Australia), www.berkeley.edu (servidor web de la Universidad de California en Berkeley). La opción **-c 3** configura la orden ***ping*** para que realice únicamente tres intentos. Anota los resultados en la tabla siguiente:   |  |  |  |  | | --- | --- | --- | --- | |  | **Tiempo de ida y vuelta (m** | | **s)** | | **Mínimo** | **Medio** | **Máximo** | | **zoltar.redes.upv.es** | 24ms | 29ms | 37ms | | **www.upv.es** | 24ms | 24ms | 25ms | | **www.rediris.es** | 33ms | 33ms | 33ms | | **www.berkeley.edu** | 192ms | 192ms | 193ms |   Analiza a qué se debe la disparidad de los resultados entre los distintos destinos.  Zoltar es una subred de la UPV, así que, haciéndolo desde casa, el acceso a la UPV debe ser más rápido que el de Zoltar. Para llegar a Rediris, hay que atravesar más routers, por lo que es más lento, y para llegar a una Universidad de otro país, pues aún más. |

Los resultados que se obtienen mediante la orden ***ping*** son, a veces, difíciles de interpretar. El usuario obtiene poca información de por qué el tiempo de ida y vuelta es mayor en unos destinos que en otros. Incluso cuando no hay respuesta al *ping*, no es posible conocer cuál es el problema: el destino solicitado está fuera de servicio, no existe una ruta desde el origen al destino o la saturación de la red es tan alta que no se obtiene respuesta del destino en un tiempo razonable. También, en ocasiones por motivos de seguridad y para evitar dar información sobre los ordenadores conectados a la red, los administradores de las redes filtran los mensajes de *ping* en los cortafuegos o desactivan el servicio en los propios ordenadores. A pesar de lo dicho, es una de las herramientas que más utilizan los administradores y usuarios de equipos conectados en red.

|  |
| --- |
| **Ejercicio 4:**  Antes de iniciar la captura lee el ejercicio hasta llegar a las cuestiones.  Aplica un filtro de captura (no de visualización) que capture únicamente los paquetes ICMP generados tras la ejecución de la orden ping -c 3 zoltar.redes.upv.es. Ejecuta la orden dos veces.  Detén la captura cuando terminen los seis intentos y observa cuántos mensajes ICMP se producen, prestando especial atención a los **campos de la cabecera ICMP:** **tipo**, **código**, y **bytes de datos**. Observa la diferencia entre los mensajes ICMP de petición y de respuesta de eco. Asimismo, analiza las **cabeceras IP** de cada uno de ellos, y en concreto los campos longitud **de la cabecera**, **longitud total** y **bytes de datos**. Compara el valor del campo protocolo con el que observaste en el ejercicio 1.  Respecto a los mensajes ICMP:  Los mensajes ICMP son mensajes de error, así que, para llevarlo a cabo desde casa, no debo estar conectado a la red de la UPV.   * ¿Por qué los mensajes ICMP no llevan números de puerto fuente y destino? No tiene puerto porque ICMP es un protocolo de nivel de red, y no llega al nivel de aplicación (que son los que utilizan los puertos) * ¿Para qué se utiliza el **número de secuencia** de la cabera ICMP? El número de secuencia es un número generado por el remitente para relacionar el Echo del ICMP con su correspondiente respuesta Echo. * ¿Para qué se utiliza el campo **identificador** de la cabecera ICMP? Para relacionar todos los echo de un mismo ping. Exactamente, se utiliza para identificar por qué interfaz se realizará el ping. |

# La orden traceroute

La orden ***traceroute* (**que se ejecuta desde un terminal) permite conocer el camino (secuencia de routers) que debe atravesar un paquete para llegar desde la estación origen a la estación destino. El funcionamiento se basa en gestionar adecuadamente un parámetro de la cabecera de los datagramas IP (el campo TTL: tiempo de vida) y en la información que aportan los mensajes ICMP que generan los routers cuando les llega un datagrama cuyo tiempo de vida se ha agotado.

Por cada nuevo router atravesado por el datagrama se dice que hay un *salto* en la ruta. Podemos decir, que el programa ***traceroute*** calcula y describe el número de saltos de una ruta.

Generalmente, el campo TTL tiene 8 bits que el emisor inicializa a algún valor. El valor recomendado actualmente en el RFC de números asignados (RFC 1700) es de 64. Cada router que atraviesa el datagrama debe reducir el TTL en una unidad. Cuando un router recibe un datagrama IP con TTL igual a uno y decrementa este valor obtiene un cero. Consecuentemente, el router descarta el datagrama y envía un mensaje ICMP de tipo 11 (*tiempo excedido)* al origen que generó el datagrama. La clave para el funcionamiento del programa ***traceroute*** es que este mensaje ICMP contiene la dirección IP del router que lo ha enviado.

En el caso del *traceroute*, el primer datagrama IP se envía al ordenador destino con TTL igual a 1. Si el destino no está en la misma red que el host origen, el primer router con el que se encuentre este datagrama decrementará el TTL y al obtener un cero lo descartará, enviando un mensaje ICMP de “tiempo excedido” (*Time exceed*) al origen. Así se identifica el primer router en el camino. A continuación, se envía un datagrama con TTL igual a 2 para encontrar la dirección del segundo router, y así sucesivamente.

Cuando el datagrama alcance un valor de TTL suficiente para llegar a su destino, necesitaremos que el destino envíe un mensaje que nos permita detener el proceso. Para ello ***traceroute*** utiliza dos opciones distintas:

* Enviar mensajes ICMP de eco (es la que se usa en Microsoft Windows). La respuesta al alcanzar el destino será un mensaje de respuesta de eco.
* Enviar mensajes UDP a un puerto arbitrariamente grande (en principio es el 33434) y muy probablemente cerrado (es la opción que se usa en Linux/Unix). El sistema responderá con un mensaje ICMP de puerto inalcanzable si el puerto está cerrado en el destino, pero si estuviera abierto no se recibiría respuesta y no se detectaría que ya se ha alcanzado el destino.

Por defecto, para averiguar cada nuevo salto se envían tres datagramas y para cada uno de ellos se calcula el valor del tiempo de ida y vuelta. Si en un tiempo máximo (configurable) no hay respuesta se indica en la salida mediante un asterisco.

Algunas puntualizaciones:

* No hay ninguna garantía de que la ruta que se ha utilizado una vez vaya a ser utilizada la siguiente.
* No hay ninguna garantía de que el camino seguido por el paquete de vuelta sea el mismo que ha seguido el paquete de ida. Esto implica que a partir del tiempo de ida y vuelta que ofrece ***traceroute*** puede no ser directo estimar el tiempo de ida o de vuelta por separado (si el tiempo que tarda el paquete en ir desde el origen hasta el *router* es de 1 segundo y el tiempo que tarda el paquete de vuelta es de 3 segundos, el valor que nos proporcionará ***traceroute*** será de 4 segundos).
* La dirección IP que se devuelve en el mensaje ICMP es la dirección de la interfaz entrante del router (aquella por la que se recibió el paquete).

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ejercicio 5:**  Ejecuta la orden ***traceroute*** para los siguientes destinos y anota el número de saltos.   |  |  | | --- | --- | |  | **Saltos** | | www.uv.es | 10 | | www.ua.es | 14… | | www.usc.edu |  |   Observa que, si se alcanza el destino, la última línea mostrada corresponde a dicho destino (en nuestro caso un servidor web) y no a un router.  Analiza cuáles pueden ser las causas de la respuesta obtenida al ejecutar la orden *traceroute* [www.ua.es](http://www.ua.es)[.](http://www.ua.es/) Ya que parece que no se ha alcanzado el destino (la última línea corresponde al servidor de Rediris y, a partir de ahí, todos son tiempos de espera agotados para las solicitudes), podría ser debido a que el puerto echo esté cerrado para que no se pueda averiguar la estructura de la red.  En el *traceroute* a [www.usc.edu,](http://www.usc.edu/) aparecen diferencias importantes en el retardo de los enlaces que se observa, ¿cuál crees que es el motivo? Que accede a routers fuera del país (es una Universidad de California). |

**Ejercicio 6:**

Desde el navegador accede a la página **http://www.telstra.net/cgi-bin/trace**. En esta página puedes indicar una dirección IP destino. El servidor de Telstra que está en Melbourne hará un traceroute desde su máquina al destino que le hayas indicado. En nuestro caso le vamos a poner la dirección IP de nuestra máquina de trabajo en el Laboratorio de Redes. Así obtendremos todos los routers por los que pasa un datagrama desde el servidor de Telstra hasta llegar a nuestra red.

A continuación, realiza un traceroute al servidor [**www.telstra.net,**](http://www.telstra.net/) para obtener la ruta inversa Compara ambos resultados.

¿Se sigue el mismo recorrido desde la UPV a [www.telstra.net](http://www.telstra.net/) y viceversa? No debería, ya que la ida y la vuelta no tienen por qué seguir el mismo camino.

Observarás que algunos routers tienen nombres similares en los dos casos, pero con direcciones IP distintas, ¿a qué crees que es debido? A que se entra a los mismos routers, pero desde interfaces distintas.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Ejercicio 7:** Captura los paquetes IP derivados de la ejecución de la orden traceroute –n www.upv.es. Para capturar también los paquetes enviados por tu ordenador tendrás que modificar el filtro para que incluya paquetes UDP. Puedes emplear el filtro de captura “icmp or (udp and host 158.42.180.X)”, donde la X representa el valor en decimal del último octeto de la dirección IP de tu ordenador.  Para los paquetes que envía tu ordenador:   * En la cabecera IP: ¿cómo se modifica el TTL? Cuando se envía un paquete al primer router, siempre empieza con TTL = 1 (como tracert envía 3 paquetes seguidos, habrán 3 con TTL = 1). Al llegar al router, se resta el TTL, y al router que le ha llegado, lo descarta y envía una respuesta ICMP con su IP al ordenador origen. A continuación, el ordenador origen envía un paquete con TTL = 2, que cuando le llegará al siguiente router lo descartará, y enviará su respuesta ICMP con su IP al ordenador origen, y así sucesivamente hasta llegar al ordenador destino. * En la cabecera UDP: ¿cómo se modifica el puerto destino? En Linux, se envían segmentos UDP a un puerto poco probable, y responde con un mensaje ICMP de *puerto inalcanzable (3 – 3).* En Windows, se envían pings, por lo que responde con un mensaje ICMP de *respuesta de eco (0 – 0).*   Para los paquetes de respuesta completa la tabla siguiente (es suficiente un par de paquetes de cada uno de los tipos ICMP recibidos):   |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | **Cabecera ICMP** | | **Datos ICMP**  (Información sobre el error) | | | Tipo |  | Código | TTL  (cabecera IP) | Protocolo (cabecera UDP) | | Type: 11 (Time-to-live exceeded) |  | Code: 0 (Time to live exceeded in transit) | Time to live: 1 | Code: 0 -> red inalcanzable. |   Basándote en la información anterior, relaciona las respuestas recibidas con los paquetes que envió tu ordenador. Observa que las respuestas a los paquetes enviados pueden recibirse desordenadas, ya que vienen de distintos dispositivos.  Indica por qué se envía información sobre las cabeceras IP e ICMP en los paquetes ICMP de error. |