Arquitectura de Redes y Servicios

Práctica tema 8: Uso de ICMP para PING

Dr. Diego R. Llanos
Departamento de Informática
Universidad de Valladolid
Versión 1.3

4 de diciembre de 2017

1. El protocolo ICMP

El protocolo ICMP (*Internet Control Message Protocol*) es uno de los protocolos Internet. Se define en el documento RFC 792. Este protocolo se utiliza para realizar diagnósticos sobre el correcto funcionamiento de la red y de los nodos conectados a ella, así como para enviar respuestas de error cuando una petición no puede completarse.

Los detalles de uso del protocolo ICMP se han explicado en clase. Aquí veremos el uso de los datagramas ICMP para realizar un *ping*.

2. Uso de ICMP para hacer un *ping*

En esta práctica utilizaremos el protocolo ICMP para implementar un servicio *ping* básico. El formato del mensaje ICMP utilizado para el *ping* puede verse en la figura 1. Los campos son los siguientes:

- Cabecera ICMP, formada por:
  - El tipo del mensaje (1 byte). Para el *ping*, la solicitud lleva tipo 8.
  - El código del mensaje (1 byte). Para el *ping*, la solicitud lleva código 0.
  - El checksum de todo el datagrama ICMP. Recordar que el checksum hay que ponerlo inicialmente a cero, y cuando hayamos construido todo le datagrama hay que calcularlo y guardarlo allí.
- Dos bytes que identifican al proceso que ha realizado el *ping*. Lo habitual es guardar allí el resultado de invocar la funcion `getpid()`.
- Un número de secuencia, también de dos bytes. En nuestro ejemplo, basta con inicializarlo a 0.
- El *payload*, que en nuestro caso serán 64 bytes, y que puede inicializarse con una cadena arbitraria.

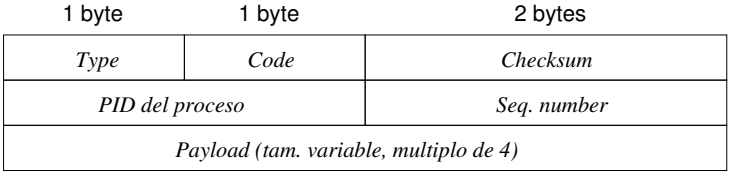

Figura 1: Formato del datagrama ICMP para la realización de un *ping*.

Si todo ha salido bien, la máquina a la que hemos enviado el *ping* nos responderá con un datagrama IP que contendrá, tras la cabecera IP, un datagrama ICMP. Ese datagrama ICMP de respuesta tendrá el campo *Type* a 0 y el campo *Code* a 0. En caso de error, el tipo de error viene indicado en esos dos campos. La lista de los códigos de error posibles pueden obtenerse en la página en inglés sobre ICMP de la Wikipedia.

3. Práctica a realizar

La práctica consiste en desarrollar un cliente de tipo ping, llamado miping. El programa se invocará desde la línea de comandos, con el siguiente formato:

```
miping direccion-ip [-v]
```

Los parámetros son los siguientes:

- La dirección IP de la máquina a la que deseamos hacer un *ping*.
- Un parámetro opcional `-v`, que hará que el cliente desarrollado informe de cada uno de los pasos realizados (cada uno de estos pasos irá precedido por los caracteres `->`). En este caso, además de mostrar los detalles de la operación, el programa deberá mostrar por pantalla el valor del parámetro TTL de la cabecera IP del datagrama recibido.

A continuación aparecen dos ejemplos de uso:

```
diego@lab5v00:$ ./miping 74.125.230.32 -v
-> Generando cabecera ICMP.
-> Type: 8
-> Code: 0
-> Identifier (pid): 1000.
-> Seq. number: 0
-> Cadena a enviar: Este es el payload.
-> Checksum: 0x73df.
-> Tamaño total del paquete ICMP: 72.
Paquete ICMP enviado a 74.125.230.32
Respuesta recibida desde 74.125.230.32
-> Tamaño de la respuesta: 92
-> Cadena recibida: Este es el payload.
-> Identifier (pid): 1000.
-> TTL: 56.
Descripción de la respuesta: respuesta correcta (type 0, code 0).

diego@lab5v00:$ ./miping 74.125.230.32
Paquete ICMP enviado a 74.125.230.32
Respuesta recibida desde 74.125.230.32
Descripción de la respuesta: respuesta correcta (type 0, code 0).
```

Para simplificar el desarrollo, en el entorno virtual hay un fichero llamado `ip-icmp-ping.h` con tres estructuras de datos: la cabecera IP, la cabecera ICMP, y el cuerpo del datagrama ICMP necesario para realizar el *ping* y para almacenar la respuesta. Para usar este fichero, basta con copiarlo en nuestro directorio local y añadir la línea `include "ip-icmp-ping.h"` al principio de nuestro programa.

Como puede verse, no sólo hay que mostrar el contenido de los campos *Type* y *Code* de la respuesta, sino también indicar si todo ha ido bien o si se ha producido un error. Si el datagrama de respuesta indica un error, el error deberá describirse por pantalla. La lista de los errores y su descripción puede descargarse de la página sobre ICMP de la Wikipedia (en inglés). Por ejemplo, si intentamos acceder a una dirección que está protegida por un cortafuegos, la respuesta del programa debería ser:

```
...
Descripción de la respuesta: Destination Unreachable: Communication
administratively prohibited (type 3, code 13).
```

Dos cuestiones importantes. La primera: tal como se ha explicado en clase, los datagramas ICMP no se comportan como los datagramas UDP. Cuando queremos enviar un datagrama ICMP, deberemos construir el datagrama ICMP a mano y luego usar sendto() para enviarlo a destino. La función sendto() se encargará de ponerle la cabecera IP y de enviarlo. Sin embargo, cuando recibimos la respuesta con recvfrom(), esta función **no** le quita la cabecera IP antes de entregarnos la respuesta, sino que el buffer de recepción contendrá primero los 20 bytes de la cabecera IP, y luego todo el datagrama ICMP. Gracias a esta funcionalidad, podemos fisgar en la cabecera IP, por ejemplo para ver cómo se ha decrementado el contador TTL.

La segunda cuestión es que los mensajes ICMP no funcionan correctamente si los enviamos a la propia dirección IP, ni tampoco a la de *loopback*. Si lo intentamos, el sistema nos devolverá nuestro propio mensaje, no la respuesta que estamos esperando.

Como en otras ocasiones, es imprescindible que el alumno comprenda perfectamente el significado de cada línea de código desarrollada.

4. Entrega y evaluación de esta práctica

1. Esta práctica deberá realizarse en la máquina virtual instalada en la primera práctica.
2. El fichero de código fuente deberá comenzar con un comentario indicando el nombre de su autor, con el siguiente formato:

   ```
   // Practica tema 8, Apellido1 Apellido2 Nombre
   ```

3. El cliente deberá estar adecuadamente comentado. De lo contrario, se restarán dos puntos de la nota final.
4. El cliente deberá compilar sin advertencias (opción -Wall del compilador gcc). De lo contrario, se restarán tres puntos de la nota final.
5. Cuando esté finalizada, se subirá el fichero fuente al entorno virtual, con el nombre siguiente: `miping-`*`apellido1-apellido2.c`* (no comprimir el fichero). Un fallo en las condiciones de entrega supondrá un punto menos en la nota final.
6. El plazo de entrega finalizará el **jueves 21 de diciembre de 2017 a las 23:55**.
7. El profesor evaluará las prácticas, contactando individualmente con los alumnos para resolver dudas e incidencias.
8. Esta práctica supondrá un 30 % de la calificación de prácticas de la asignatura.
9. Se utilizará un sistema automático de detección de copias. En caso de copia, todos los alumnos involucrados deberán presentar todas las prácticas, incluida la práctica copiada, y se corregirán sobre 7, no sobre 10. En el caso de que la calificación media ponderada de todas las prácticas sea menor que cinco, dichos alumnos habrán suspendido la convocatoria ordinaria de la asignatura.