Práctica

7

GESTIÓN Y UTILIZACIÓN DE REDES LOCALES

Curso 2001/2002

TCP/IP: protocolo ICMP

Introducción

El **protocolo IP** tiene como cometido el transporte de datagramas desde un host origen a otro cualquiera en Internet. El servicio que realiza IP no garantiza una mínima calidad de servicio, pudiendo ocurrir la pérdida de datagramas, su entrega desordenada, errores en los mensajes que transporta, etc.

Cuando un datagrama viaja por la red tendrá que atravesar una serie de routers, los cuales procesan el datagrama para encaminarlo adecuadamente hacia su destino. Si un router no puede encaminar ese datagrama, o bien detecta alguna condición especial en la que se ve incapacitado para hacerlo (congestión de red, líneas fuera de servicio, etc.), entonces ese datagrama se pierde.

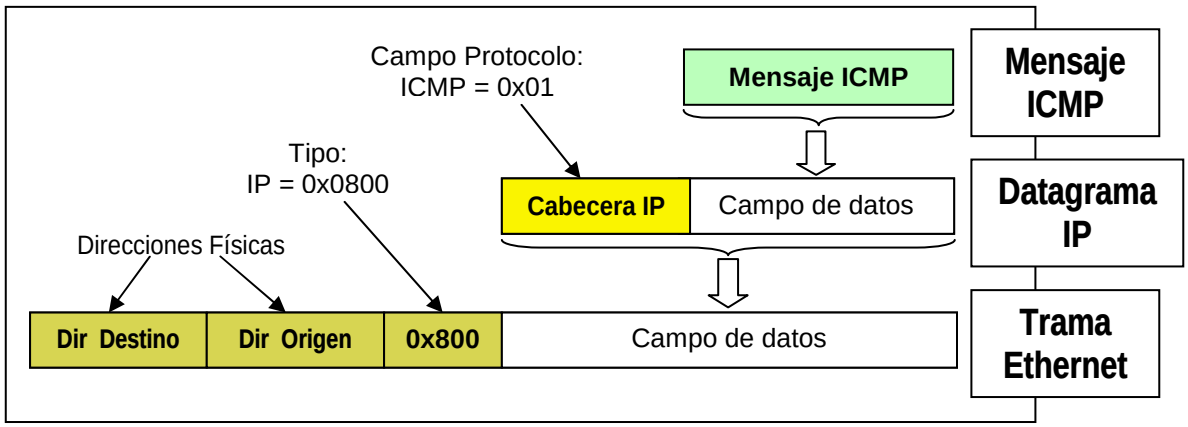
Éstas y otras circunstancias en el tratamiento de los datagramas en su viaje hacia el destino hacen necesaria la creación de un mecanismo que, al menos, informe de estas situaciones al host origen, para que éste sea consciente de los problemas que ha sufrido el datagrama que ha enviado y, si procede, tome las acciones oportunas. De aquí nace el protocolo ICMP.

El **protocolo ICMP** (*Internet Control Message Protocol*) es un mecanismo que informa de la aparición de errores en la manipulación de los datagramas. Siempre que un router detecte un error o excepción en un datagrama, utiliza el protocolo ICMP para informar al host origen de la circunstancia. **ICMP no realiza ninguna acción para corregir el error que se haya producido, solamente se encarga de comunicarlo al host origen para que éste realice las acciones oportunas para corregir el error**. Inicialmente, ICMP fue diseñado como un protocolo para los routers, sin embargo los hosts también lo pueden utilizar.

Aunque este protocolo fue diseñado para detectar las incidencias que se producen en el transporte de un datagrama hacia el host destino, no todas ellas pueden ser detectadas. Entre estas causas se encuentra la pérdida de un datagrama que lleva un mensaje ICMP. En este punto podríamos pensar que para solucionar este problema, esta pérdida podría ser notificada con otro mensaje ICMP. Más que solucionar el problema, lo estaríamos agravando cuando la razón de esa pérdida sea una congestión en la red. Por eso, NO SE PERMITE notificar mediante mensajes ICMP la pérdida de datagramas que lleven un mensaje ICMP. De hecho, la pérdida de un datagrama que transporta un mensaje ICMP no se notifica. Otra norma general que impone este protocolo es que las notificaciones de error se hacen SOLAMENTE al host origen.

Formato de los mensajes ICMP

Los mensajes ICMP van encapsulados en datagramas IP, como muestra la figura siguiente:



Aunque veamos que ICMP va encapsulado en un datagrama de IP, eso no quiere decir que ICMP sea un protocolo de nivel superior. Se debe considerar como parte de IP, como si

fuese una herramienta auxiliar que dispone IP para poder detectar errores en el transporte de los datagramas a sus destinos.

Cada tipo de mensaje ICMP tiene su propio formato, aunque todos ellos comienzan con tres campos comunes. El resto de campos puede variar en función del tipo de mensaje. Los campos comunes son el campo **tipo**, el campo **código** y el campo **checksum**.

El campo **TIPO** identifica el tipo de mensaje ICMP. En la siguiente tabla se muestran algunos de los tipos de mensajes que contempla este protocolo. El campo **CÓDIGO** se usa para dar más información acerca del tipo de mensaje ICMP. Este campo no siempre es necesario. El último campo contendrá el checksum del mensaje ICMP.

Campo tipo	*Tipo de mensaje ICMP*
0	*Contestación de eco*
3	*Destino inalcanzable*
4	*Paquete de restricción*
8	*Petición de eco*
11	*Tiempo de vida agotado*

La descripción de cada uno de los tipos de mensajes ICMP de la tabla anterior es la siguiente:

- **Petición de eco** y **Contestación de eco**: Son dos mensajes que se usan conjuntamente para determinar la alcanzabilidad de un host o un router. Normalmente son utilizados por los hosts, de forma que estos puedan extraer información acerca del estado del host remoto, el retardo que introduce la red en la entrega de los mensajes y el porcentaje de mensajes perdidos.
- **Destino inalcanzable**: Se trata de un mensaje que es generado por un router cuando no puede encaminar un datagrama. Existen diferentes causas que provocan la emisión de este mensaje y que están codificadas en el campo **código** del mensaje ICMP. El mensaje es dirigido al host que ha enviado el datagrama y en su interior se especifican los primeros 64 bits del datagrama que lo ha causado (el que no se puede entregar al destino).
- **Paquete de restricción**: Es un mensaje que utilizan los routers para frenar el ritmo de inyección de mensajes en la red de un determinado host. Esta situación se produce cuando un router se ve sobrecargado con la recepción de datagramas (una posible situación de congestión) teniendo que descartar algunos por falta de *buffers*. Cuando se produce esta situación, el router envía un mensaje de este tipo al host origen del datagrama descartado, diciéndole que baje el ritmo de inyección de datagramas ya que en ese momento hay una situación temporal de congestión.
- **Tiempo de vida agotado**: Cuando un router encamina un datagrama, una de sus tareas es decrementar en una unidad el campo "**Tiempo de vida**" de la cabecera del mismo. Si tras la operación el campo vale "0", el router debe descartar el datagrama y enviar un mensaje ICMP de este tipo hacia el host origen.

El control ICMP

En esta práctica vamos a utilizar parte del protocolo ICMP, más concretamente los mensajes de tipo "**Petición de eco**" y "**Contestación de eco**". Para ello utilizaremos un control específico de ICMP que proporciona el software **IPWorks**. Este software añade diversos controles al entorno de programación de Visual Basic. Estos controles están

relacionados con los diferentes protocolos que conforman la familia de protocolos **TCP/IP**. Antes de poder utilizar estos controles debemos instalar el paquete **IPWorks**. Para ello basta con ejecutar el programa "**`Setup`**" que se encuentra en el servidor herodes, en el directorio **`\\herodes\practicas\Gyurl\Practica7`**. Durante el proceso de instalación, elegid siempre la opción por defecto que os sugiera el programa. Una vez instalado el paquete IPWorks, basta con incluir en la barra de herramientas de Visual Basic el control que deseemos utilizar. En esta práctica debemos elegir el control de ICMP, llamado "**IPWorks! V4 ICMPPort Control**". Para agregarlo hay que ir al menú **Proyecto** y elegir el submenú **Componentes**. De las propiedades de este control, las más significativas son las siguientes:

- **Active**: Activa el control. Debemos poner esta propiedad a "True" al comienzo de nuestro programa. Lo mas cómodo es ponerlo en el evento "load" del formulario de nuestra aplicación. Ejemplo: **`ICMPPort1.Active = True`**
- **MessageType**: Identifica el tipo de mensaje ICMP que se quiere enviar (ver la tabla anterior). Ejemplo: **`ICMPPort1.MessageType = 8`** (* Petición de eco *)
- **MessageSubType**: Algunos mensajes ICMP llevan un campo adicional al campo tipo. Este es el campo que hemos llamado código en el formato de los mensajes ICMP. Ejemplo: **`ICMPPort1.MessageSubType = 0`**
- **RemoteHost**: Guarda la dirección IP del host remoto al que queremos enviar un mensaje ICMP. Si le pasamos el nombre del host automáticamente se realiza una consulta al servidor de nombres para averiguar la dirección IP. Ejemplo: **`ICMPPort1.RemoteHost = "www.upv.es"`**
- **TimeToLive**: tiempo de vida del paquete IP que lleva el mensaje ICMP que vamos a enviar. Si se le asigna un cero, se toma el valor por defecto que tenga el control. Si es diferente de cero, entonces especifica el número de routers que puede atravesar el mensaje antes de ser eliminado. Ejemplo: **`ICMPPort1.TimeToLive = 30`**
- **DataToSend**: Se utiliza para enviar un mensaje ICMP de tipo **`MessageType`** (**`MessageSubType`**) al host **`RemoteHost`** llevando en el campo de datos el string que se le pasa. Es una propiedad de sólo escritura, de forma que cuando le asignamos un string se provoca el envío del mensaje ICMP. Ejemplo: **`ICMPPort1.DataToSend = "Mensaje de prueba"`**. **ADVERTENCIA: Para que el control funcione adecuadamente el campo de datos no puede estar vacío (""), sino que el mensaje debe contener al menos cuatro caracteres. Se trata de un problema en la implementación del control y no de un requisito del protocolo ICMP**.

Para recibir mensajes ICMP disponemos del evento **DataIn**, cuya descripción es:

```
Sub ICMPPort1_DataIn (MessageType As Integer, MessageSubType As Integer,
MessageData As String, Checksum As Integer, SourceAddress As String)
```

Cada vez que llega un mensaje ICMP se dispara este evento, informando a nuestra aplicación del tipo y código del mensaje recibido así como el campo de datos, el **`cheksum`** del mensaje ICMP y la **`dirección IP`** del host que lo envía. Por tanto, este evento es indispensable para la recepción de mensajes ICMP.

Realización de la práctica

En esta práctica se pretende realizar un programa que permita comprobar la alcanzabilidad de un host (ping) y también nos proporcione la secuencia de routers entre nuestro host y otro cualquiera de Internet (tracert o traceroute). Tanto el comando "**ping**" como el

comando "**traceroute**" ya los hemos utilizado previamente en diversas prácticas. Nosotros pretendemos desarrollar una versión **MUY SIMPLIFICADA** de los mismos.

Para determinar la alcanzabilidad de un host remoto, utilizaremos mensajes ICMP de tipo "**eco**". El mecanismo es sencillo: cuando un host envía una "**petición de eco**" hacia un host remoto, éste debe responder con un mensaje ICMP de tipo "**contestación de eco**", devolviendo en el campo de datos la misma información que portaba el primer mensaje (petición de eco) en su campo de datos. La máquina destino puede estar en otra red distante, por lo que puede que el mensaje ICMP tenga que atravesar varios routers. Por esta razón, si no obtenemos respuesta cuando le enviamos el mensaje de "**Petición de eco**" puede ser debido a varias causas: pérdida del mensaje de peticion de eco (mal encaminamiento de un router intermedio, un error de checksum, etc.) o bien el host remoto está fuera de servicio. Puede también ocurrir que la máquina destino esté muy cargada y se demore en mandar el mensaje de respuesta "**Contestación de eco**".

Por otra parte, para descubrir la secuencia de routers que hay entre nuestro host y otro cualquiera utilizaremos también los mensajes de tipo "**eco**", pero jugando con el campo "**Tiempo de vida"** de los datagramas que los transportan. Así enviaremos mensajes "**Petición de eco**" comenzando con un tiempo de vida igual a uno para descubrir el primer router (su dirección IP), ya que contestará con un mensaje de tipo "**Tiempo de vida agotado**". Con un valor en el campo de tiempo de vida igual a dos, descubriremos el segundo router, quien nos devolverá dicho mensaje y así sucesivamente hasta que lleguemos al host remoto. En este caso el host remoto nos devolverá un mensaje de tipo "**contestación de eco**".

Para desarrollar el programa solicitado y comprobar el funcionamiento del control se recomienda hacer un formulario sencillo, como el mostrado en la figura siguiente. Como se puede observar, dicho formulario consta de una **TextBox**, en la que introducimos el nombre de la máquina que queremos alcanzar, o bien su dirección IP. Como ejemplo se ha propuesto la dirección del servidor web de nuestra universidad. Para poder ejecutar tanto la orden ping como tracert, incorporamos dos botones a nuestro formulario. La respuesta que obtengamos será diferente en función de la orden ejecutada. En el caso de la orden tracert deberemos mostrar, en la **ListBox** del formulario, la secuencia de routers (sus direcciones IP) por las que va atravesando el datagrama. En el caso de la orden ping, basta con mostrar el contenido del campo de datos del datagrama que hayamos recibido.