

Arquitectura de Redes y Servicios

Práctica tema 7: Cliente TFTP

Dr. Jorge Fernández, Dr. Diego R. Llanos
Departamento de Informática
Universidad de Valladolid
Versión 1.5

14 de noviembre de 2019

1. El protocolo TFTP

El servicio TFTP permite el envío y recepción de ficheros a través de Internet. Se trata de un servicio estándar, y como tal está asociado a un puerto bien conocido.

El protocolo TFTP (Internet Trivial File Transfer Protocol) se definió en 1981 en el RFC 783, y fue revisado en 1992 en el RFC 1350. Se trata de un sistema sencillo para transferir archivos entre dos equipos. Se implementa con datagramas UDP.

TFTP es mucho más sencillo que el servicio FTP, y tiene muchas menos funcionalidades (no muestra el directorio remoto, no hace gestión de usuarios, etcétera). La principal utilidad de TFTP es el envío de un kernel de arranque a máquinas que no dispongan de unidades de almacenamiento. Estas máquinas tienen un cliente TFTP en ROM y lo utilizan para recibir el kernel de un servidor remoto.

El protocolo TFTP define cinco paquetes de datos diferentes. Su formato aparece en la figura 1.

1.1. Inicio del diálogo entre cliente y servidor

El diálogo comienza con el envío, por parte del cliente, de un datagrama con una solicitud de lectura (RRQ) o de escritura (WRQ) de un archivo. Como puede verse en la figura 1, ambas solicitudes constan de un código de operación, una cadena indicando el nombre del archivo a leer o a escribir y acabada en 0, y una cadena con el modo de transmisión, que puede ser *netascii* u *octet*. El modo de transmisión *netascii* garantiza un envío y recepción correcto de ficheros de texto, realizando las conversiones entre los diferentes estándares existentes para, por ejemplo, marcar el final de línea (CR o bien CR+LF). El modo *octet*, en cambio, realiza una transmisión literal del fichero. El modo *octet* será el único modo que emplearemos en nuestra práctica.

1.2. Lectura de un fichero

- Si el cliente solicita la lectura de un fichero, el servidor comprobará que existe un fichero con ese nombre, lo abrirá e irá leyendo bloques y enviándoselos al cliente. Vamos a suponer que los nombres de los ficheros pueden tener como máximo 100 caracteres.
- El envío se realiza a través de paquetes de datos de hasta 512 bytes. Los paquetes de datos llevan un número de orden de bloque: el primer paquete enviado por el servidor lleva el número 1.
- Cada vez que el cliente recibe un paquete de datos con número de bloque x , debe responder al servidor con un paquete de confirmación (ACK) indicando el número de bloque x . Cuando el servidor reciba el ACK del bloque x , enviará el bloque $x + 1$, y así sucesivamente.
- El protocolo define que el último bloque de datos tendrá un tamaño estrictamente menor de 512 bytes.

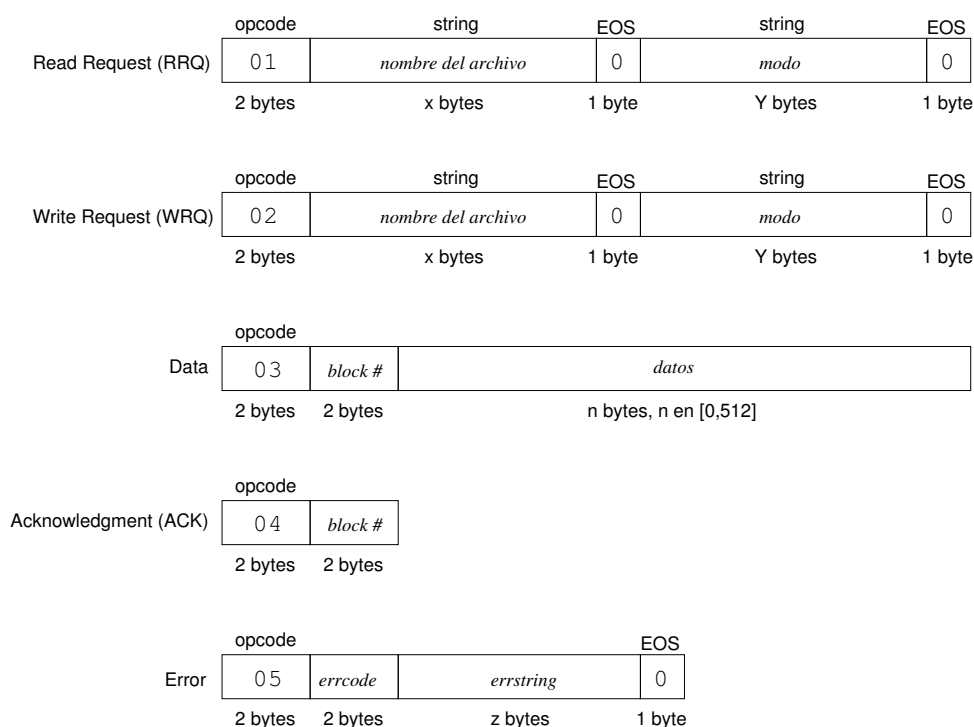


Figura 1: Formato de los paquetes intercambiados en TFTP.

- Ante cualquier problema, el servidor enviará un código de error, utilizando un paquete con opcode 5. El servidor puede, opcionalmente, añadir una descripción del error en dicho paquete. Los códigos de error son los que pueden verse en la figura 2.

1.3. Escritura de un fichero

- Si el cliente solicita la escritura de un fichero, el servidor comprobará que dicha escritura está permitida y enviará al cliente un ACK con número de bloque 0. A partir de ese momento, el cliente deberá enviar paquetes de datos de hasta 512 bytes, comenzando con el número de bloque 1.
- Cada vez que el servidor reciba un paquete de datos con número de bloque x , escribirá el bloque en el disco y responderá al cliente con un paquete de confirmación (ACK) indicando el número de bloque x . Cuando el cliente reciba el ACK del bloque x , enviará el bloque $x + 1$, y así sucesivamente.

errcode	Descripción
0	No definido. Comprobar <i>errstring</i> .
1	Fichero no encontrado.
2	Violación de acceso.
3	Espacio de almacenamiento lleno.
4	Operación TFTP ilegal.
5	Identificador de transferencia desconocido.
6	El fichero ya existe.
7	Usuario desconocido.

Figura 2: Códigos de error devueltos por el servidor TFTP.

- El protocolo define que el último bloque de datos enviado por el cliente tendrá un tamaño estrictamente menor de 512 bytes. En ese momento el servidor almacenará el bloque correspondiente en el disco y cerrará el fichero, enviando el último ACK.
- **Nota 1:** Para almacenar el bloque recibido en el disco debe usarse la función `fwrite()`, que permite guardar un número determinado de bytes en el disco. No debe usarse `fprintf()`, ya que esta función sirve para almacenar cadenas, y puede que el bloque recibido contenga ceros que provoquen que se almacenen menos bytes de los deseados (¿por qué?).
- **Nota 2:** La única manera de probar si nuestra implementación de la escritura funciona correctamente es (a) hacer una copia de un fichero cualquiera, (b) enviar esa copia, (c) borrar la copia local, (d) traerse el fichero de nuevo, y (e) haciendo un `diff` para comparar la copia que acaba de llegar y el fichero original. Esto hay que hacerlo, naturalmente, con ficheros de diferentes tamaños.

2. Práctica a realizar

La práctica consiste en desarrollar un cliente UDP para el servicio TFTP. La práctica deberá desarrollarse en lenguaje C en la máquina virtual. El cliente desarrollado se invocará a través de la siguiente línea de comandos:

```
tftp-client ip-servidor {-r|-w} archivo [-v]
```

Los parámetros son los siguientes:

- La dirección IP del servidor TFTP.
- Un `-r` indicando si se desea leer un archivo del servidor, o un `-w` si se desea escribirlo en él.
- El nombre del archivo a transferir.
- Un parámetro opcional `-v`, que hará que el cliente desarrollado informe de cada uno de los pasos necesarios para enviar o recibir el fichero, incluyendo el envío y recepción de paquetes con el servidor.

A continuación aparece un ejemplo:

```
diego@lab5v00:$ ./tftp-client 127.0.0.1 -r datos-justos.dat -v
Enviada solicitud de lectura de datos-justos.dat a servidor tftp en 127.0.0.1.
Recibido bloque del servidor tftp.
Es el primer bloque (numero de bloque 1).
Enviamos el ACK del bloque 1.
Recibido bloque del servidor tftp.
Es el bloque con codigo 2.
Enviamos el ACK del bloque 2.
El bloque 2 era el ultimo: cerramos el fichero.
diego@lab5v00:$
```

En el servidor situado en 10.0.25.250 (visible sólo desde las máquinas virtuales) hay tres ficheros que pueden utilizarse para probar el cliente desarrollado:

- `datos-cortos.dat`, de 100 bytes.
- `datos-justos.dat`, de 512 bytes.
- `datos-largos.dat`, de 12 345 678 bytes.

Dado que la red que conecta las máquinas virtuales no pierde datagramas, no se implementarán mecanismos de recuperación ante errores en la transmisión, aunque sí hay que comprobar que los paquetes se reciben en el orden correcto.

Es imprescindible que el alumno comprenda perfectamente el significado de cada línea de código desarrollada. Los únicos recursos necesarios para realizar la práctica son (a) los apuntes desarrollados por el profesor y (b) las páginas `man`.

3. Condiciones de entrega y evaluación

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

```
// Practica tema 7, 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 en Slackware Linux (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: `tftp-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 **domingo 1 de diciembre de 2019 a las 23:55**.
7. En la sesión del **lunes 2 de diciembre de 2019** se presentará la siguiente práctica, por lo que la asistencia a esa sesión es obligatoria para todos los alumnos.
8. Esta práctica supondrá un 35 % 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 figurarán como suspensos en la convocatoria ordinaria, debiendo presentar todas las prácticas por e-mail al profesor para poder presentarse a la convocatoria extraordinaria. Todas las prácticas se corregirán sobre 7.