

DISEÑO DE LA PRÁCTICA DE REDES DE COMUNICACIONES 2024/2025

Autores:

-Sergio Parra Martínez G1.2

-Carlos Pérez Pérez 23335807F G1.2

Profesor: Jose Rubén Titos Gil

Índice

1.Introducción.....	3
2.Formato de los mensajes del protocolo de comunicación con el Directorio.....	3
3.Formato de los mensajes del protocolo de transferencia de ficheros.....	3
4.Autómatas del protocolo.....	3
5.Ejemplo de intercambio de mensajes.....	3

1.Introducción.

En este documento se especifica el diseño de dos protocolos, uno que funciona para la comunicación con el directorio y otro para la transferencia de ficheros. Para ello se verá el diseño de los mensajes para dichos protocolos y los autómatas para los mismos.

Mejoras a hacer:

-Serve puerto efímero.(0.5)

-Filelist ampliado con servidores.(0.5)

2.Formato de los mensajes del protocolo de comunicación con el Directorio.

Para definir el protocolo con el cual nos comunicaremos con el Directorio, vamos a emplear mensajes textuales del tipo “**campo:valor**”. El valor que tome el campo “operation” indicará el mensaje y su formato, por lo que sabremos los campos que vienen a continuación.

Tipos y descripción de los mensajes.

Mensaje: **ping**

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el cliente de NanoFiles al directorio para ver si está activo.

Ejemplo:

```
operation: ping\n
protocolID: MORTADELA\n
\n
```

Mensaje: **ping_ok**

Descripción: Este mensaje lo envía el directorio al cliente de NanoFiles para decir que está activo.

Sentido de la comunicación : Directorio -> Cliente

Ejemplo:

```
operation: welcome\n\n
```

Mensaje: **ping_failed**

Descripción: Este mensaje lo envía el directorio al cliente de NanoFiles para decir que no está activo.

Sentido de la comunicación : Directorio -> Cliente

Ejemplo:

```
operation: denied\n\n
```

Mensaje: **filelist**

Sentido de la comunicación: Cliente → Directorio

Descripción: Este mensaje lo envía el cliente de NanoFiles al Directorio para solicitar la lista de ficheros posibles para descargar.

Ejemplo:

```
operation: filelist\n\n
```

Mensaje: **availableFiles**

Sentido de la comunicación: Directorio → Cliente

Descripción: Este mensaje envía al cliente la lista de ficheros disponibles.

Ejemplo:

```
operation: filelist\nfilename: document.pdf\nsharingserver: 192.168.1.1:10000\nharingsserver: 192.168.101.1:10000\nsize: 12345\nhash: abc123def\n\n
```

Mensaje: **serve**

Sentido de la comunicación: Cliente -> Directorio

Descripción: Este mensaje lo envía el cliente de NanoFiles al directorio para que lance un servidor de ficheros que escucha conexiones en el puerto deseado .

Ejemplo:

```
operation: serve\n\n
```

Mensaje: **serve_success**

Sentido de la comunicación: Directorio-> Cliente

Descripción: Este mensaje lo envía el directorio al cliente de NanoFiles para confirmar que lanza un servidor de ficheros que escucha conexiones en el puerto oportuno.

Ejemplo:

```
operation: success\n\n
```

Mensaje: **whohasit**

Sentido de la comunicación: Cliente→ Directorio

Descripción: Este mensaje envía al directorio una subcadena del nombre de un archivo.

Ejemplo:

```
operation: whohasit\n\nfilename: docu\n\n
```

Mensaje: **serverlist**

Sentido de la comunicación: Directorio→ Cliente

Descripción: Este mensaje envía al cliente la lista de servidores, con sus respectivas ip y número de puerto.

Ejemplo:

```
operation: serverlist\nserver: 192.168.1.1:10000\nserver: 192.168.101.1:10000\n\n
```

3.Formato de los mensajes del protocolo de transferencia de ficheros.

Formatos de los mensajes:

-Control:

<i>Opcode</i>
<i>1 byte</i>

-Operación:

<i>Opcode</i>	<i>Parámetro 1</i>	<i>Parámetro 2</i>
<i>1 byte</i>	<i>8 bytes</i>	<i>8 bytes</i>

-Tamaño Variable TLV:

<i>Opcode</i>	<i>Longitud</i>	<i>Valor</i>
<i>1 byte</i>	<i>0/1/2/4/8 bytes</i>	<i>n bytes</i>

Para definir el protocolo de comunicación con un servidor de ficheros, vamos a utilizar mensajes binarios multiformatos. El valor que tome el campo “opcode” (código de operación) indicará el tipo de mensaje y por tanto cuál es su formato, es decir, qué campos vienen a continuación.

-Tipos y descripción de los mensajes

Mensaje: **download** (opcode = 2)

Formato: TLV

Sentido de la comunicación: Cliente → Servidor de ficheros.

Descripción: El par cliente manda la petición para descargar un paquete al par servidor.

Ejemplo:

<i>Opcode (1 byte)</i>	<i>Longitud(2 bytes)</i>	<i>Valor (n bytes)</i>
<i>0x2</i>	<i>0x00 0x10</i>	<i>documento.pdf</i>

Mensaje: **file_not_found** (opcode = 1)

Formato: Control

Sentido de la comunicación: Servidor de ficheros → Cliente

Descripción: Este mensaje lo envía el par servidor de ficheros al par cliente (receptor) de fichero para indicar que no es posible encontrar el fichero con la información proporcionada en el mensaje de petición de descarga.

Ejemplo:

<i>Opcode (1 byte)</i>
<i>0x1</i>

Mensaje: **hash** (opcode = 3)

Formato: TLV

Sentido de la comunicación: Servidor de ficheros → Cliente

Descripción: El par del servidor manda el hash al cliente y este lo recibe.

Ejemplo:

<i>Opcode (1 byte)</i>	<i>Longitud(2 byte)</i>	<i>Valor (n bytes)</i>
<i>0x3</i>	<i>0x00 0x04</i>	<i>AB CD EF 12 AB CD EF 12</i>

Mensaje: **get_chunk** (opcode = 4)

Formato: Operación

Sentido de la comunicación: Cliente → Servidor de ficheros

Descripción: El cliente le dice al servidor de ficheros el chunk que quiere.

Ejemplo:

<i>Opcode (1 byte)</i>	<i>Parametro1(8 bytes)</i>	<i>Parametro2(4 bytes)</i>
<i>0x4</i>	<i>desplazamiento</i>	<i>tam chunk</i>

Mensaje: **chunk** (opcode = 5)

Formato: TLV

Sentido de la comunicación: Servidor de ficheros → Cliente.

Descripción: El servidor de ficheros le envía al cliente el chunk que este le ha solicitado.

Ejemplo:

<i>Opcode (1 byte)</i>
<i>0x5</i>

Mensaje: **all_downloaded (opcode = 6)**

Formato: Control

Sentido de la comunicación: Cliente → Servidor de ficheros.

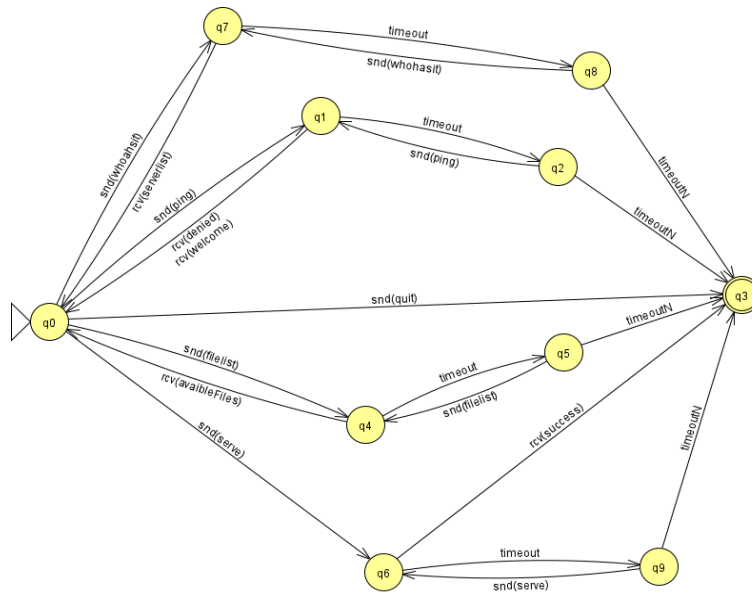
Descripción: El cliente no necesita más chunks de ese servidor, por tanto está completado.

Ejemplo:

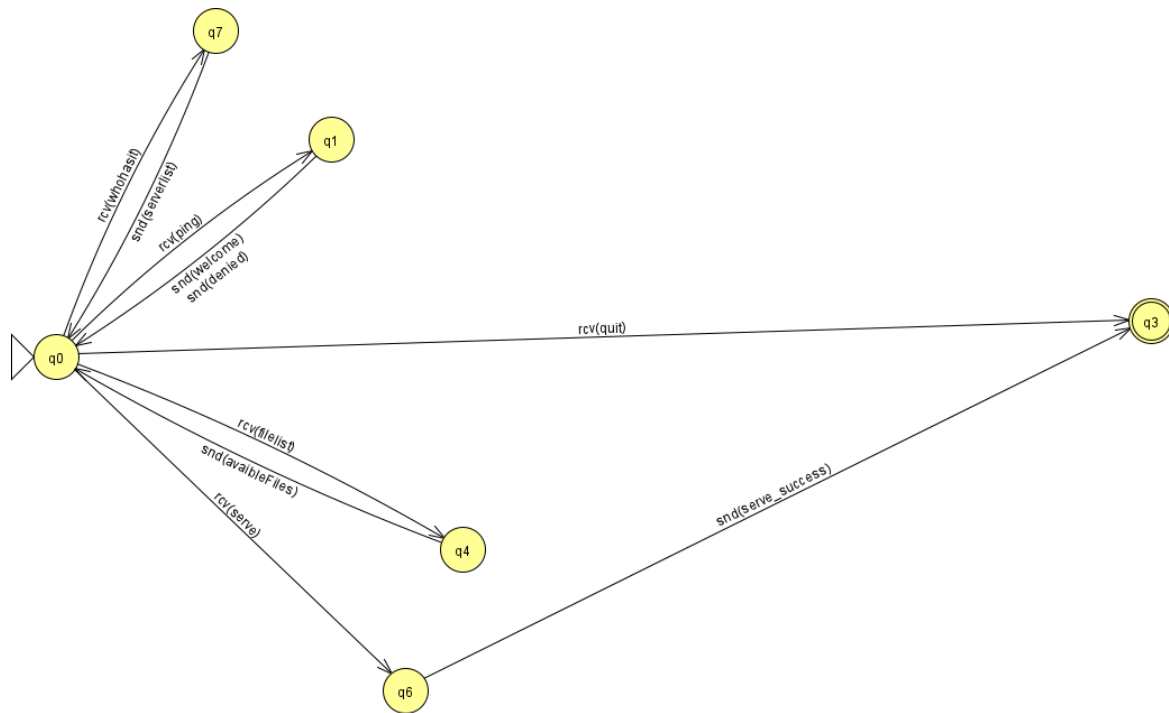
<i>Opcode (1 byte)</i>
<i>0x6</i>

4. Autómatas del protocolo.

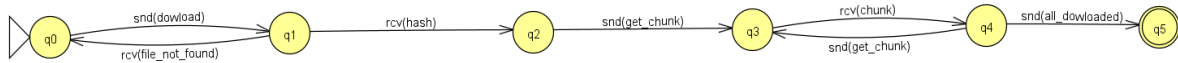
-Autómata rol cliente de directorio:



-Autómata rol servidor de directorio:



-Autómata cliente de ficheros:



-Autómata servidor de ficheros:



5. Ejemplo de intercambio de mensajes.

Operación **ping**, ejemplo de intercambio de mensajes:

-Cliente:(Selecciona localhost con y envía la operación ping y el protocolID=MORTADELA)

```
*WARNING: No files found in folder nf-shared
NanoFiles shell
For help, type 'help'
Do you want to use 'localhost' as location of the directory server? (y/n)
Using directory location: localhost
(nanoFiles@nf-shared) ping
* Checking if the directory at localhost is available...
operation:welcome

* Directory is active and uses compatible protocol MORTADELA
(nanoFiles@nf-shared)
```

-Servidor:(recibe la operación ping y el protocolID, lo compara y envía welcome, en caso de ser correcto imprime bienvenido)

```
Directory [Java Application] C:\Users\carlo\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Probability of corruption for received datagrams: 0.7
Directory starting...
Directory ignored datagram from /127.0.0.1:53728
Data received: operation:ping
protocolid:MORTADELA

Bienvenido maquina
```