Application Layer Protocol

This is the Application Layer Protocol used for an FTP server that was made for a College Computer Communications Class.

Table Of Contents

Table Of Contents **1**
Overview **2**
- Control Channel 2
- Data Channel 2

FTP Datagram **2**
Valid Command Names **2**
Use Cases **3**
- Establishing connection 3
  - Primary Flow 4
  - Notes: 4
- Disconnecting 4
  - Primary Flow 4
  - Notes: 4
- Downloading a file 4
  - Primary Flow 4
  - Notes 5
- Uploading a file 5
  - Primary Flow 5
  - Notes 5
- Deleting a file 5
  - Primary Flow 6
  - Notes 6
- Getting a list of all files on FTP server 6
  - Primary Flow 6
  - Notes 6
- Errors 7
  - Invalid Packet 7

Overview

The application uses two channels: a control channel, and a data channel

Control Channel

This is established and open for as long as a client is connected to the server. It is used to pass commands

Data Channel

This is used to pass the data for files. It is open and closed each time it is used.

FTP Datagram

Figure 1
FTP Datagram

Packet Number (2 bytes)	Command Name (6 bytes)	DataSize (4 bytes)
Data		

Packet Number: A counting up number for the given packet, number are only counted by sender (*Note this is still implemented, though it ended up not being used*)
Command Name: Uses the Alias of the command to send (see Valid Command Names)
Data Size: the number of bytes in the total package, header data included
Data: the actual data to send with the packet

Valid Command Names

Figure 2
Valid FTP Command Names

Full Name	Alias	Data	Expected Use
Connect	Con	PortNumber to send data to	Establish a connection
Connection Acknowledge	ConAck	PortNumber to send data to	Acknowledge a connection

Disconnect	DisCon		Disconnecting a channel
Get	Get	FileName	Download a file
Put	Put	FileName	Upload a file
Delete	Del	FileName	Delete a file
List Request	LSReq		Ask for a list of all file names
Acknowledge	Ack	Packet Number of packet to acknowledge	Acknowledge a previous command
Invalid Packet	InvPac	Packet Number of Invalid packet	Packet was received ok, but something in the data was invalid. I.e. a get command was sent for a filename that doesn't exist
File Manifest	FMan	Number of expected packets	[UNUSED] Information about the incoming file
File Packet	File	FileData	Used when sending part of a file
File Status	FStat	Either good or bad	[DEPRECIATED] Used to give the status of a file when all packets are sent, did the receiver assemble everything correctly?

Use Cases

Establishing connection

When a connection is established between two sockets

It is initially used for setting up the connection between the Control sockets
But it can also be later reused on the Data Channel to set up those sockets

Primary Flow

Control Channel
Client: 1 Con (DataPortNumber)
Server: 1 ConAck (DataPortNumber)
Client: 2 Ack (1)

Notes:

- The DataPortNumbers are saved by both sides to be used later for sending files

Disconnecting

When a connection is being disconnected between two sockets
It is mainly used for tearing down the control channel, since the data channel is torn down after every use.

Primary Flow

Control Channel
Client: 1 DisCon

Notes:

- When the control channel is torn down the server will shut itself off

Downloading a file

When the client wants to download a file

Primary Flow

Control Channel
Client: 1 Get (FileName)
Server: 1 Ack (1)

Data Channel

Client: 1 Con (DataPortNumber)
Server: 1 ConAck (DataPortNumber)
Client: 2 Ack (1)
Server: 2 FMan (Number of expected packets for FileName)
Client: 3 Ack(2)
Server: 3 File (FileData)

Notes

- If the FileName the client is trying to download does not exist on the FTP server the server will respond with InvPac(1) rather than Ack(1)
- The Data Channel is automatically closed down by both sides when done

Uploading a file

When the client wants to upload a file

Primary Flow

Control Channel
Client: 1 Put (FileName)
Server: 1 Ack(1)

Data Channel
Client: 1 Con (DataPortNumber)
Server: 1 ConAck (DataPortNumber)
Client: 2 FMan (Number of expected packets for FileName)
Server: 2 Ack (2)
Client: 3 File (FileData)

Notes

- The Data Channel is automatically closed down by both sides when done

Deleting a file

When the client wants to delete a file on the FTP server.

Primary Flow

Control Channel
Client: 1 Del (FileName)
Server: 1 Ack (1)

Notes

- If the FileName the client is trying to delete does not exist on the FTP server the server will respond with InvPac(1) rather than Ack(1)

Getting a list of all files on FTP server

When the client wants to get a list of all the files on the FTP server.

It uses almost the same process as downloading any other file.
However this file is just a list of all the file names on the FTP server.

Primary Flow

Control Channel
Client: 1 LSReq
Server: 1 Ack(1)

Data Channel
Client: 1 Con (DataPortNumber)
Server: 1 ConAck (DataPortNumber)
Client: 2 Ack (1)
Server: 2 FMan (Number of expected packets for FileName)
Client: 3 Ack(2)
Server: 3 File (FileData)

Notes

- The expected file will contain all the file names with a return character between them (“\n”)
- The Data Channel is automatically closed down by both sides when done

Errors

Invalid Packet

If InvPac is received the current process is stopped and an error message is displayed.
This will most often happen if the client is trying to download or delete a file that does not exist on the server