GWD-R
GridFTP WG
http://www-isd.fnal.gov/gridftp-wg/

Igor Mandrichenko, FNAL (editor)

July 25, 2003
Revised July 29, 2003

GridFTP GET/PUT Proposal

Igor Mandrichenko
July 28, 2003

Background

In passive mode, FTP server [1] is required to produce data socket address *before* it receives the information about the file to be transferred. This makes it very difficult to implement distributed FTP server where the data is stored on multiple computers. In this case, the decision what computer should be used for particular transfer must be based on file identification (path) and perhaps some other information, and therefore such server can not provide data socket address in response to PASV, because at that point file path is unknown.

Several solutions for this problem have been discussed. One of them introduces new protocol command (PRET, pre-transfer). This command sends all necessary information before PASV. Then file path sent with subsequent STOR/RETR command must be the same as provided with PRET:

```
Client              Server
PRET /path/file.dat
                    2xx OK
PASV
                    2xx Entering passive mode (12,24,35,42,53,61)
STOR /path/file.dat
                    1xx Data connection initiated
                    2xx Transfer complete
```

GET and PUT Commands

This proposal introduces 2 new commands, GET and PUT. They combine functionality of PRET, PORT/PASV, and then STOR and RETR respectively. For example:

```
Client              Server
GET file=/path/file.dat;mode=active;port=34,23,45,12,48,14;…
                    1xx Data connection established
                    2xx Transfer complete
```

This single command performs file retrieval in active mode equivalent to the following standard FTP sequence:

```
Client              Server
PORT 34,23,45,12,48,14
                    2xx OK
RETR /path/file.dat
                    1xx Data connection established
                    2xx Transfer complete
```

Here is example of standard passive store FTP sequence:

```
Client              Server
PASV
                    2xx OK (13,67,102,34,23,65)
STOR /path/file.dat
```

ivm@fnal.gov

```
                    1xx Data connection established
                    2xx Transfer complete
```

As you can see, the server has to answer with socket address before it receives file name. Using proposed PUT command eliminates this drawback:

```
Client              Server
PUT file=/path/file.dat;mode=passive;…
                    1xx Wait
                    1xx Wait
                    1xx PORT=13,67,102,34,23,65
                    1xx Transferring
                    2xx Transfer complete
```

In this example, server receives file name as one of parameters of PUT command and replies with socket address later, when it is ready using one of 1xx responses.

Along with file name, mode specification and socket address, some other, even perhaps site-specific information can be sent as parameters of GET/PUT command. This allows future expansion of the commnd functionality if necessary.

Also, sending all information necessary to perform the transfer in a single command simplifies protocol state machine and eliminates the need for the server to carry transaction state information between individual commands.

As you can see in the previous example, server does not have to send data socket address right away. It may send several 1xx replies before sending data socket address to keep the connection alive or sending some other information back to the client while making the data socket available for server connection. This may be a solution for the problem of firewalls disconnecting idle data sockets before the transfer begins.