

# Appendix C. FTP: Problems in Grid Environment

## C.1 Problems with Partial Files

Typical FTP clients present problems when transferring files in a grid environment. The typical FTP clients do not transfer files such that a server can tell the difference between an end-of-file and a partial transfer.

Problem 1: Since a server cannot distinguish between the end-of-file and a dropped/aborted transfer, partial files may end up in the underlying mass storage system.

Problem 2: There is a server acknowledge back to the client at the end of a transfer, but it is not a handshake. Therefore, the server doesn't actually know whether the client is there or not.

The **size** command gets around these problems, but is usually not required for transferring files. Imposing use of this command would be incompatible with numerous ftp clients. Similarly, the ftp block mode contains enough information to determine if a complete file has been sent or if a transfer has been aborted. This mode is defined in the standard, but is not in widespread use.

For either solution (size or block), normal ftp installations would be expected (although not required) to keep the partial files so that the user can complete the transfer. For a grid connected to a storage system, these partial files are an added nuisance, although not a severe one (the system could just throw them away).

## C.2 ACL Timeouts

Many networks have ACL limitations imposed on them. A common ACL is the "reflexive" ACL, for which a remote client allows a connection only if the local side is the initiator. These reflexive ACLs have timeouts usually in the minutes, and almost always less than 30 minutes. We must walk the line between lengthening the time window enough to decrease the number of timeouts but not so much as to increase the security risks.

Transfers that take a long time to complete (slow link, big files) are the common victims. These transfers involve plenty of activity on the FTP data socket, but no traffic on the FTP control socket. The ACL on the FTP control

times out and the server cannot send the client its acknowledgement. Moreover, the client can't send more commands without stopping and restarting (because the control port connection is broken).

### **C.3 Third Party Passive Transfers Impossible**

Third party passive transfers are not possible. Typically, passive transfers are used in the cases of firewalls and reflexive ACLs. The FTP passive protocol requires that control and data ports be negotiated before the filename is sent. Since data could be in many places (e.g., in different storage pools), the server has no way of negotiating ports for an unknown, and more than likely different, computer. The only way around this problem is to introduce an “adaptor tunnel”, whereby the file is first transferred from the storage pool to a known computer and then to the client's computer. This gets around the port negotiation, but introduces a scalability problem. There is a bottleneck introduced in the adaptor.