



G00002

10 OCT 2002

1

"A FILE TRANSFER PROTOCOL"



166166

DESCRIPTION OF INVENTION :

THIS INVENTION relates to improvements in or relating to File Transfer Protocols, particularly those employed when data is to be transferred over an unreliable communications link.

For a mobile host to connect to a wired network (for example, the Internet) the mobile host must connect using either a wireless communications link or a telephone line and transmit a file in accordance with a file transfer protocol (FTP). Such connections are relatively expensive, unreliable and have limited bandwidths as compared to a normal local area network. It is not uncommon for such a link to fail during a file transfer. Besides going to the extra expense setting up a new connection and downloading the entire file again, mobile hosts typically also have resource constraints such as limited battery power. It is, therefore, important to minimise connection time for the mobile host. Unfortunately, many existing implementations of FTP do not work well under such unreliable link conditions. When an FTP transfer fails, the partial data transferred must be discarded and the whole transfer has to be re-started.

A solution to the above problem should take into consideration the existing FTP applications available, such as WS-FTP and the various web

browsers. Compatibility with these applications would be preferable. Besides using FTP for file transfer, the use of HTTP has also become very common due to the popularity of web browsers.

Attempts have been made to provide background file transfer. One such attempt is the "Background File Transfer Programme (BFTP)". The BFTP is an Internet background file transfer service that is built upon the third-party transfer model of FTP. It consists of two components: a user interface program to collect the parameters describing the required transfer; and a file transfer control daemon to carry them out. Both of the components are run on the same host although remote service can be achieved via a Telnet connection to the BFTP server-Telnet module.

There are also various FTP client applications, such as GetRight and AutoFTP that allow scheduled transfers. A user is allowed to specify the task to be performed at a later time and the FTP client is able to carry out repeated tries for transfers that have failed.

The FTP caters only for server-to-server file transfer, and not file download or upload to/from a mobile client. Furthermore, it is not intended specifically for a slow or unreliable environment.

For those FTP client applications that support scheduled transfers, the main drawback is that the mobile client has to maintain the connection during the transfer. This approach is expensive, and ideally the connection time should be minimised, as well as the bandwidth used by the mobile host in transferring a file.

It is an object of the present invention to provide a File Transfer Protocol that can reduce the time and cost expended on data transfer over an unreliable link.

Accordingly, the present invention provides a method of transferring a set of data between data storage media in a communication infrastructure comprising a data storage medium connected by a link to a computer which is connected to a network which is, in turn, connected to another data storage medium, the computer having means to receive, store and send data, the method comprising the steps of: if the link is temporarily severed so that only a portion of the set of data is transferred across the link, communicating between the computer and at least one of the data storage media to establish the extent to which the set of data has been transferred; and transferring the remainder of the set of data across the link when the link is re-established.

Advantageously, the set of data is to be transferred from the other data storage medium to the one data storage medium, the method comprising the further steps of: transferring the set of data to the one data storage medium and, if the link is severed during the transfer of data, continuing to transfer data through the network from the other data storage medium and storing this data in a memory of the computer.

Preferably, the method further comprising the step of, if the entirety of the remainder of the data to be transferred has been transferred to the memory of the computer when the link is re-established, transferring this data to the data storage medium across the link from the memory of the computer.

Conveniently, the method further comprising the steps of, if only a portion of the remainder of the data to be transferred has been transferred to the

memory of the computer when the link is re-established, transferring this portion of the data to the data storage medium from the memory of the computer across the link; and transferring the remaining portion of the remainder of the data to be transferred to the data storage medium through the computer from the other data storage medium.

Advantageously, the method further comprises the steps of: designating data to be transferred between the data storage media; and designating a time for the data to be transferred.

Preferably, the method further comprises the steps of, if the data is to be transferred from the data storage medium to the other data storage medium, transferring the designated data from the data storage medium across the link to the computer; and transferring the designated data across the network from the computer to the other data storage medium at the designated time.

Conveniently, the method further comprises the steps of, if the data is to be transferred from the other data storage medium to the data storage medium, transferring the data across the network from the other data storage medium to the computer; and transferring the data across the link from the computer to the data storage medium at the designated time.

Advantageously, the data is transferred from the data storage medium to the other data storage medium and includes instructions to transfer information stored on the other data storage medium to a further data storage medium.

Preferably, the method further comprises the steps of: associating a datastamp with the set of data to be transferred; and altering the datastamp associated with the set of data when the set of data is updated.

Conveniently, the method further comprises the steps of: when the link is re-established, comparing the datastamp associated with the portion of the set of data that has already been transferred across the link with the datastamp associated with the remainder of the set of data; and discarding the portion of the set of data that has already been transferred if the comparison of the datastamp associated with the portion of the set of data that has already been transferred with the datastamp associated with the remainder of the set of data indicates that the set of data at the one data storage medium has been updated since the portion of the set of data that has already been transferred was transferred across the link.

Advantageously, the datastamp incorporates time information.

Preferably, the datastamp incorporates information concerning the version of the set of data.

Another aspect of the present invention provides a computer program product having means, when the computer program is loaded on a computer, to transfer a set of data between data storage media in a communication infrastructure comprising a data storage medium connected by a link to a computer which is connected to a network which is, in turn, connected to another data storage medium, the computer having means to receive, store and send data, the method comprising the steps of: if the link is temporarily severed so that only a portion of the set of data is transferred across the link, communicating between the computer and at least one of the data storage media to establish the extent to which the set of data has been transferred; and transferring the remainder of the set of data across the link when the link is re-established.

A further aspect of the present invention provides a protocol for implementing the above method. In order that the present invention may be more readily understood, embodiments thereof will now be described, by way of example, with reference to the accompanying drawings in which:

Figure 1 is a schematic diagram of the system architecture of a system on which the method of the present invention may be carried out;

Figure 2 is a flow chart of a request from a client for the download of a file after a network disconnection between the client and the mobile gateway;

Figure 3 is a further scenario of the client requesting the download of a file after a network disconnection between the client and the mobile gateway;

Figure 4 is a flow chart of a scheduled FTP download session;

Figure 5 is a flow chart of a scheduled FTP upload session;

Figure 6 is a flow chart of a scheduled FTP server-server session; and

Figure 7 is a flow chart of a scheduled HTTP GET session;

Turning to Figure 1, the File Transfer Protocol system consists of three major components, namely the client proxy component 1, the mobile gateway component 1 and the mobile FTP protocol 3. The client proxy component 1 resides at the mobile client 4 and the mobile gateway component 1 resides at a computer 5 connected to the Internet 6. The physical link between the mobile client 4 and the computer 5 hosting the mobile gateway

component 1 can be either wired or wireless, for example a modem dial-up line or a GSM wireless link.

To employ a file transfer protocol embodying the present invention, the client proxy component 1 of the protocol must be installed at the mobile client 4, for example in the memory of a laptop computer. The client proxy component 1 consists of the following modules: a client proxy server (CPS) 7; a protocol application programming interface (API) 8 and a cache module 9. The CPS 7 is needed to support a File Transfer Protocol application and a web browser. A File Transfer Protocol application incorporating the present invention or the CPS 7 can call the application programming interface to perform the following functions:

1. User authentication.
2. Indication/cancellation of FTP transfer request.
3. Query of FTP transfer status.
4. FTP retrievable/store.
5. FTP cache management.

The mobile gateway component 1 of the File Transfer Protocol system embodying the present invention consists of the following modules: mobile gateway proxy server (MGPS) 10; the cache module 11 and the job scheduler module 12.

The communication protocol between the client proxy component 1 and the mobile gateway component 1 of the File Transfer Protocol system embodying the present invention is called the mobile FTP protocol 13.

The File Transfer Protocol embodying the present invention supports two modes of file transfer, namely an immediate and a scheduled mode. For each mode, a file download or upload to or from the mobile client, or a server-to-server transfer could be performed.

IMMEDIATE MODE

This behaves like a normal File Transfer Protocol to the user, although the File Transfer Protocol of the present invention splits the FTP session into two sessions: one between the mobile client 4 and a mobile gateway 14 (on which the mobile gateway component 1 resides); and the other between the mobile gateway 14 and a target server 15. Caches are maintained at both the client proxy component 1 and the mobile gateway component 1 during the FTP session.

The link between the mobile client 4 and the mobile gateway 14 may be disconnected in the midst of a file download/upload. Such disconnection may be due to handoff, network congestion or sudden power failure. Figure 2 shows the sequence of events when such a disconnection occurs during a file download from the mobile gateway 14 to the mobile client 4. When there is a network disconnection between the mobile client 4 and the mobile gateway 14 during the file transfer, the MGPS 10 in the mobile gateway component 1 of the File Transfer Protocol system continues and completes the FTP transfer with the target server 15. When the link between the mobile client 4 and the mobile gateway 14 is re-established and the mobile client attempts to download the same file again, the CPS 7 verifies the partial file in its own cache by checking the modification time and size with the actual target file at the target server 15. Assuming that 50% of the file has been transferred successfully to the mobile client 4 during the previous attempt (before the link was

disconnected), the mobile client 4 will get the remaining 50% from the cache of the mobile gateway 14 if the whole file has already been downloaded to the mobile gateway 14. If the file transfer between the target server 15 and the mobile gateway 14 is still ongoing (as is shown in Figure 3), the mobile client 4 downloads whatever is available from the cache of the mobile gateway 14 and gets the remainder of the file from the target server 15 via the mobile gateway 14.

In certain circumstances, a mobile link may be disconnected for a sufficiently long time that, when the link is re-established, the file that was being transferred across the link has been updated at its source. It would not be appropriate, if this is the case, to continue with the transfer where it was left off, as some or all of the information that was transferred before the failure of the mobile link may be obsolete. To accommodate this possibility, the data to be transferred across the mobile link includes a datastamp, which records the time of creation of the file, the time of the transfer and/or the version of the file that is being transferred. Using this system, when a mobile link is re-established after failing during a data transfer, the datastamp of the new incoming data is compared with that of the data which has already been transferred. If this comparison of the datastamps reveals that the partially transferred data is or may be obsolete, then the partially transferred file is discarded, and the transfer of the entirety of the new, updated version of the file begins. The datastamp may consist of time information regarding the time of the start of the initial transfer of data, or may relate to the version of the data. the datastamp may comprise a hybrid of time information and version information. If time information is used, information regarding whether or not the file has been updated since the time recorded in the datastamp would need to be gathered in order to decide whether or not the data that has already been transferred must be discarded. In the case of version information being used, all that is required is a comparison of the versions of the data - if the new incoming data is of a

more recent version than that which has already been transferred, then the data that has already been transferred must be discarded.

No special benefits are derived from using the File Transfer Protocol of the present invention for file uploading or activating a server-to-server file transfer in immediate mode.

SCHEDULED MODE

In this mode, the mobile gateway 14 is instructed by the CPS 7 or mobile client 4 to carry out the FTP transfer in the background. A new protocol, known as the Schedule File Transfer Protocol (SFTP) is defined for this type of scheduled transfer. There are four types of schedule transfer:

I. FTP download; II. FTP upload; III. FTP server-server transfer; and IV. HTTP GET download.

I. Schedule of FTP download - for the case of a scheduled FTP download, this is the normal FTP download from the target server 15 to the mobile client 4, except that in this case, the mobile gateway 14 functions as a store and forward gateway. The file is first downloaded to the mobile gateway 14. The user can then check the status of the file transfer at any time, and fetch the files from the mobile gateway 14 at a later time when the download is completed. Figure 4 shows the message sequence chart 4A Scheduled FTP Download Session.

II. Scheduled FTP upload - scheduled upload from the mobile client 4 is useful when a user decides that a particular file upload should take place at some later time, maybe when the outgoing link is less busy. In this case, the mobile gateway 14 acts as a buffer store for the file to be sent to the target

server 15. After getting the file from the mobile client 4 the mobile gateway 14 will forward the file to the target server 15 at the scheduled time. Figure 5 shows the message sequence chart for a scheduled FTP upload session.

III. Scheduled FTP server-server transfer - the scheduled server-server transfer is based on the third party server-server FTP model. A user may wish to transfer a file from a server to another server, neither of which is a local host. The mobile gateway 14 will perform the role as a co-ordinating body to control the file transfer between the two remote servers. The mobile client 4 does not need to stay connected throughout the whole file transfer session. For example, a user from outside may initiate a file transfer from a remote server to his desktop PC at night. The user only needs to log into the mobile gateway 14 submit the request and then log off. The mobile gateway 14 will co-ordinate the file transfer between the FTP server and the desktop PC. When the user returns to his desktop PC the next morning, the file may already be transferred there and ready for use. Figure 6 shows the message sequence chart for such a scheduled FTP server-to-server session.

IV. Scheduled HTTP GET - scheduled download to the mobile client 4 can be used for HTTP GET transfer as well. This is very similar to the "scheduled FTP download" except that not all the various FTP settings are applicable here and the download is done using the HTTP protocol GET method. Similarly, the mobile gateway 14 functions as a store and forward gateway. The file is stored in the mobile gateway 14 and the mobile client 4 can fetch the files from the mobile gateway 14 at a later time. Figure 7 shows the message sequence chart for a scheduled HTTP GET session.

When in immediate mode, no special notification mechanism is needed as the user is on-line all the time. For the scheduled FTP transfer mode, the

user will be given a complete status of the transfer at the end of the scheduled transfer, by e-mail on whether the transfer is completed successfully, or whether the transfer fails after repeated tries. The e-mail will be sent when the background file transfer is completed or when the transfer is aborted after repeated unsuccessful attempts. Alternatively, the user can also log into the mobile gateway 14 to check the status of the request. For the case of FTP or HTTP GET downloads to the mobile client 4, the user can then retrieve the file from the mobile gateway 14.

When a mobile client 4 moves into a foreign network, a mechanism is needed for the mobile gateway 14 in that foreign network to make known its presence to the visiting mobile client 4. It is assumed that the mobile client 4 is able to maintain the Internet connection through schemes such as mobile IP while roaming the foreign network.

The discovery of a new mobile gateway 14 when a mobile client 4 enters a foreign network is accomplished through a pair of ICMP messages: an MGPS 10 advertisement message; and an MGPS 10 solicitation message. The mobile gateway periodically broadcasts ICMP packets to advertise its availability to mobile clients 4. Furthermore, if the mobile gateway 14 receives an MGPS 10 solicitation message, the mobile gateway 14 will reply to this with a MGPS 10 advertisement message.

CLAIMS:

1. A method of transferring a set of data between data storage media in a communication infrastructure comprising a data storage medium connected by a link to a computer which is connected to a network which is, in turn, connected to another data storage medium, the computer having means to receive, store and send data, the method comprising the steps of: if the link is temporarily severed so that only a portion of the set of data is transferred across the link, communicating between the computer and at least one of the data storage media to establish the extent to which the set of data has been transferred; and transferring the remainder of the set of data across the link when the link is re-established, wherein the set of data is to be transferred from the other data storage medium to the one data storage medium and the method comprises the further steps of: transferring the set of data to the one data storage medium and, if the link is severed during the transfer of data, continuing to transfer data through the network from the other data storage medium and storing this data in a memory of the computer.
2. A method according to Claim 1, the method further comprising the step of, if the entirety of the remainder of the data to be transferred has been transferred to the memory of the computer when the link is re-established, transferring this data to the data storage medium across the link from the memory of the computer.
3. A method according to Claim 1 or 2, the method further comprising the steps of, if only a portion of the remainder of the data to be transferred has been transferred to the memory of the computer when the link is re-established, transferring this portion of the data to the data storage medium from the memory of the computer across the link; and transferring the remaining portion of the remainder of the data to be transferred to the data storage medium through the computer from the other data storage medium.
4. A method according to any preceding claim further comprising the steps of: designating data to be transferred between the data storage media; and designating a time for the data to be transferred.

5. A method according to Claim 4, the method further comprising the steps of, if the data is to be transferred from the data storage medium to the other data storage medium, transferring the designated data from the data storage medium across the link to the computer; and transferring the designated data across the network from the computer to the other data storage medium at the designated time.
6. A method according to Claim 4 further comprising the steps of, if the data is to be transferred from the other data storage medium to the data storage medium, transferring the data across the network from the other data storage medium to the computer; and transferring the data across the link from the computer to the data storage medium at the designated time.
7. A method according to any one of Claims 4 to 6, wherein the data is transferred from the data storage medium to the other data storage medium and includes instructions to transfer information stored on the other data storage medium to a further data storage medium.
8. A method according to any preceding claim, further comprising the steps of: associating a datastamp with the set of data to be transferred; and altering the datastamp associated with the set of data when the set of data is updated.
9. A method according to Claim 8, further comprising the steps of: when the link is re-established, comparing the datastamp associated with the portion of the set of data that has already been transferred across the link with the datastamp associated with the remainder of the set of data; and discarding the portion of the set of data that has already been transferred if the comparison of the datastamp associated with the portion of the set of data that has already been transferred with the datastamp associated with the remainder of the set of data indicates that the set of data at the one data storage medium has been updated since the portion of the set of data that has already been transferred was transferred across the link.
10. A method according to Claim 8 or 9, wherein the datastamp incorporates time information.

11. A method according to Claim 8 or 9, wherein the datastamp incorporates information concerning the version of the set of data.
12. A protocol for implementing the method of any preceding claim.