

[0086] According to a preferred embodiment of the invention, a special version of the socket library functions is provided as a shared library. The wrapper process refers to this library. The special version of the socket library writes data sent to or received from the socket into a buffer. Once the “USER” command is received, the FTP-wrapper process creates the FTP process. From this point on, the FTP server communicates directly with the client without the intermediation of the FTP-wrapper, i.e., the original functions are called, rather than the functions of the special version of the library. This results in a slight overhead.

An FTP Session

[0087] FIG. 1 schematically illustrates a typical FTP session, according to one embodiment of the invention.

[0088] At step 1, a client connects to an FTP port of a hosting computer. The Inetd receives the request, opens a port (i.e., communication channel with the client).

[0089] At step 2, carried out once the communication channel with the client has been established, any received FTP command except the USER command is stored in a buffer rather than processed (but a suitable reply/acknowledgment is sent to the client, so that the client will not terminate the communication), until a USER command arrives.

[0090] At step 3, the “USER xxx%aaa.com” command is received. Only the command that is relevant to the FTP process is stored in said buffer, i.e.—“USER xxx”.

[0091] At step 4, an FTP process for the aaa.com domain is created.

[0092] At step 5, if the wrapper is a process (rather than a function), the execution of the wrapper is terminated. It should be noted that from steps 1 to 5 the session has been carried out between the client and the Inetd or by the wrapper, and not between the client and the FTP server, as in the prior art.

[0093] At step 6, the FTP process receives the commands stored in said buffer. Actually, the FTP server communicates with the wrapper (or with the information that the wrapper retained in the buffer) instead of with the client, but the FTP server is not “aware” of this fact. From this point on, the client communicates with the FTP server, as in the prior art.

[0094] At step 7, the PASS command is sent by the client to the FTP server in order to be validated.

[0095] At step 8, after the password has been validated by the FTP server, the client can send file-related commands such as download, upload, delete, and so forth, to be performed by the FTP server. It should be noted that the verification of the password is handled by the FTP server, and not by the wrapper. Therefore, if the user prefers to install special authentication modules for its FTP server, or to perform special actions on a failed login—these operations will be performed anyway.

Applying the Invention to Other Web Services

[0096] With a Web mail service, one can view and respond to his new e-mails from any computer with an Internet connection. The e-mail messages remain on the mail servers until the user checks his mail from his home computer again.

[0097] With POP (Post Office Protocol), mail is delivered to a server accessible through the Web, and a personal computer user periodically connects to the server and downloads all of the pending mail to the “client” machine. Thereafter, all mail processing is local to the client’s machine.

[0098] SMTP (Simple Mail Transfer Protocol) is the standard e-mail protocol on the Internet. SMTP servers route SMTP messages throughout the Internet to a mail server, such as POP3 or IMAP4, which provides a message store for incoming mail.

[0099] Like the FTP protocol, Web mail protocols restrict the provision of services to a user by authenticating the user, which typically is carried out by the user name and a password. Since a user is usually associated with an account on the hosting computer, the provision of such services can be implemented in the same way as FTP.

[0100] HTTP (HyperText Transport Protocol) is the communications protocol used to connect to servers on the World Wide Web. Its primary function is to establish a connection with a Web server and transmit HTML pages to the client browser. Addresses of Web sites begin with an “http://” prefix.

[0101] For HTTP, there is a well-known solution called “Virtual hosting”. Using this technique, the HTTP server can detect the domain that the user wishes to access (from a special line in the command), and therefore to access a different directory tree for each domain. However, this solution implies sharing the same Web server between the domains. Hence, this solution does not suite a Web hosting company, which has an interest to limit the resources that each domain receives, and probably charges each account according to its actual use.

[0102] According to one embodiment of the invention, a plurality of requests from a Web server can be executed simultaneously, such that each domain (or a group of domains) receives its own request. However, since only one Web server can use a certain port on a computer at a given moment, only one Web server can be active at that moment.

[0103] This limitation can be overcome by piping the data from the original socket to an “internal” socket, which actually can be accessed by a plurality of Web servers at a given moment.

[0104] Hence, the invention may be applied to HTTP services as well as to FTP and mail. By implementing the virtual hosting methods used in the prior art, i.e. the virtual hosting ability that is provided by a Web server, all the domains are served by a single Web server. However, according to the invention, several Web servers can run simultaneously, each one serving some of the domains. Therefore, a WHP can achieve better performance of the hosted Web sites.

[0105] It should be noted that receiving the connection, reading the requested domain and handing the socket with the information to the relevant Web server, is transparent to the server.

[0106] The above examples and description have of course been provided only for the purpose of illustration, and the invention can be implemented to any Web service including MIRC, Telnet, SSH, Rtelnet, and Shell.