COMMUNICATIONS SYSTEM PROVIDING SHARED CLIENT-SERVER COMMUNICATIONS INTERFACE AND RELATED METHODS

Field of the Invention

[0001] The present invention relates to the field of communications systems, and, more particularly, to client-server communications and related methods.

Background of the Invention

[0002] One way in which applications communicate with one another is to use a client-server relationship. In such a relationship, one application functions as a client and provides an interface to the user. The other application is the server application, which resides on an application server and is responsible for the majority of computation and/or data processing.

[0003] This client-server relationship can be extended to World Wide Web applications where the client application (typically a Web browser) and the server component (a Web or application server on the Internet) will interact. One approach for Web-based client-server applications to communicate with one another is to use hypertext transfer protocol (HTTP) as a request-response protocol. Traditionally, HTTP is used on the World Wide Web for browser clients to

access and download content from Internet Web sites to users' computing environments (e.g., home, corporate network, etc.).

[0004] Many computing environments provide rich or sophisticated functionality to their users when the user is acting within the confines of his protected computing environment. For example, a corporate user may have access to proprietary corporate databases while using his desktop computer in his office. However, when a user is outside this environment (e.g., the user is on the road), he may still require access to such functionality.

[0005] Most computing environments allow connections originating within the environment to outside locations, but connections originating outside the environment are restricted from accessing the environment. This is typically accomplished through the use of a firewall, for example. Furthermore, some computing environments further restrict outbound network connections to access only HTTP services. This makes it difficult, if not impossible, for a roaming user to access important functionality or services from his protected computing environment.

[0006] The problem is perhaps most prevalent for home-based users. For example, it is difficult for users to connect from their personal computer at their home to their corporate servers at work. A dial-up or high-speed Web-based connection often requires client software on the home machine and/or a secure token for authentication. Furthermore, most corporations may not support corporate access using personal computers.

[0007] Various prior art approaches have been developed for allowing users to access information from outside a protected computing environment. By way of

example, Symmetry Pro from Infowave Software, Inc., is a software service that provides corporate users with wireless access to their corporate e-mail using a wireless handheld device. In particular, e-mail messages that arrive in a user's corporate inbox are encrypted and then delivered via the Symmetry Pro software service to the user's wireless handheld device.

[0008] Two other prior art approaches include Fire Extinguisher and Gnu HTTPTunnel. Both of these products attempt to encapsulate TCP traffic over an HTTP connection, acting as a generic bi-directional proxy. Yet, one significant drawback of such approaches is that they may not provide a desired level of authentication to protect secure communications in certain applications.

Summary of the Invention

[0009] In view of the foregoing background, it is therefore an object of the present invention to provide a communications system which provides enhanced client-server communication features and related methods.

[0010] This and other objects, features, and advantages in accordance with the present invention are provided by a communications system which may include a plurality of communications devices connected together in a network having a plurality of user accounts associated therewith. At least one of the communications devices may process requests using a hypertext transfer protocol (HTTP) client application associated therewith. Moreover, the system may also include an application server for accessing the user accounts via the HTTP client application, and an HTTP server for interfacing the HTTP client application with the application server. The HTTP server and the HTTP

client application may format requests to be communicated therebetween via the Internet in an HTTP format, and each may provide additional state information with the HTTP formatted requests recognizable by the other for authenticating the application server and the HTTP client application to one another. Furthermore, the HTTP client application may request a first universal resource locator (URL) from the HTTP server for accepting work requests from the application server, and request a second URL different from the first URL from the HTTP server for responding to work requests from the application server.

[0011] Accordingly, the communications system advantageously allows data or applications within a protected computing environment (e.g., a corporate network) to be securely accessed by users when outside of the environment. That is, the at least one communications device may be located within the protected environment (e.g., a user's desktop computer). Since the HTTP client application and HTTP server communicate using HTTP requests, the HTTP client application and HTTP server may advantageously communicate through a network port reserved for Internet traffic (i.e., HTTP formatted requests and responses). Thus, a user may access his account, which may include various types of applications or information (e.g., e-mail accounts, calendars, contacts, etc.) which may otherwise be blocked by a network firewall. Moreover, use of the first and second URLs allows the HTTP server to more readily distinguish and manage requests coming from or going to the HTTP client application.

[0012] More particularly, the additional state information may be a global unique identifier (GUID) associated with the HTTP client application. Additionally, the HTTP client application and the HTTP server further provide sequencing information with the HTTP formatted requests. The sequencing information advantageously allows a given response to be matched with a respective request. Furthermore, the HTTP client application and the HTTP server may format the additional state information as HTTP headers for respective HTTP formatted requests.

[0013] A method aspect of the invention is for accessing a plurality of user accounts associated with a plurality of communications devices using an application server. The communications devices may be connected together in a network, and at least one of the communications devices may process requests using an HTTP client application associated therewith. The method may include interfacing the application server with the HTTP client application using an HTTP server. Moreover, the HTTP server and the HTTP client application may format requests to be communicated therebetween in an HTTP format via the Internet, and each may provide additional state information with the HTTP formatted requests recognizable by the other for authenticating the application server and the HTTP client application to one another. The method may further include using the HTTP client application to request a first universal resource locator (URL) from the HTTP server for accepting work requests from the application server. Additionally, a second URL different from the first URL is requested from the HTTP server for responding to work requests from the application server.

Brief Description of the Drawings

[0014] FIG. 1 is a schematic block diagram of a communications system in accordance with the present invention.

[0015] FIG. 2 is a flow diagram illustrating a client-server communications method in accordance with the present invention.

Detailed Description of the Preferred Embodiments

[0016] The present invention will now be described more fully hereinafter with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art. Like numbers refer to like elements throughout.

[0017] Generally speaking, the present invention allows an HTTP client to act in a server capacity while still following accepted HTTP client behavior. The invention thus advantageously allows a client application in a user's protected computing environment (e.g., a corporate network) to establish a secure connection with an Internet service and then respond to requests from an authenticated user (e.g., the user's home computer or wireless communications device).

[0018] Referring initially to FIG. 1, a Web-based client-server communications system **100** is first described. The system **100** illustratively includes an HTTP client or client application **104**, located in a protected computing environment **106**. By way of example, the protected computing environment may be a corporate

network **107** having a plurality of communications devices **108a-108n** (e.g., personal computers (PCs)) connected thereto, and a firewall **112** for limiting external access to the network, as will be appreciated by those skilled in the art. It should be noted that while the firewall **112** and network **107** are shown as separate elements for clarity of illustration, the various firewall and network routing functions performed thereby may be implemented in one or more network servers or other devices, as will be appreciated by those skilled in the art.

[0019] The HTTP client application **104** communicates bi-directionally with an HTTP server **102**, which in the present example is outside the protected computing environment **106**, via the Internet **109**, for example. The HTTP server **102** illustratively communicates with an application server **101** to retrieve or process any application-related data. In one exemplary embodiment, the HTTP server **102** may belong to a service provider that interfaces users with their respective communications devices **108a-108n** within the protected computing environment **106**. Accordingly, the application server **101** may be for performing e-mail delivery or aggregation services using the HTTP server **102** to provide an interface to a user's communications device **108** within the protected computing environment **106**, as will be described further below. Of course, other types of data may be accessed as well, as will be appreciated by those skilled in the art. A user could then access the e-mail (or other) data collected by the application server **101** via a home computer, wireless communications device (e.g., a personal data assistant (PDA)), etc., as will also be appreciated by those skilled in the art.

[0020] In accordance with the invention, the HTTP server **102** and the HTTP client application **104** preferably follow accepted HTTP server-client behaviors and/or relationships. This allows the two to communicate using a dedicated network port reserved for Internet (i.e., HTTP) traffic (typically port 80), without being blocked by the firewall **112**. Yet, the HTTP server **102** and the HTTP client application **104** are also both able to insert additional state information into requests and responses, and recognize state information inserted by the other.

[0021] In the illustrated embodiment, the client application **104** is an "intelligent" application that is running on a computer in the user's protected computing environment **106**. The HTTP client application **104** establishes an outbound network connection to the designated HTTP server **102**, and requests a specific uniform resource locator (URL) therefrom. In addition, the HTTP client application **104** provides additional HTTP headers, such as data specifying a globally unique identifier (GUID) to the HTTP server **102**, for example. This establishes a semi-permanent connection that is available for the HTTP server **102** to use for accessing the HTTP client application **104** without being blocked by the firewall **112**.

[0022] More specifically, the application(s) running on the application server **101** is now able to access the HTTP client **104** from outside the protected computing environment **106** by making a request to the HTTP server **102**. When an the application server **100** makes an indirect request of the HTTP client application **104** via the HTTP server **102**, the HTTP server **102** in turn formats that request into a valid HTTP request. This request is then encapsulated into an HTTP response to

the HTTP client application **104**. The response includes a header section, which includes both data required by the HTTP specification as well as additional state and sequencing information injected by the HTTP server **102**, and a body section, which includes a full HTTP request.

[0023] When the HTTP client application **104** receives the response, it is then able to access the response body, which includes an HTTP request, which further includes both a header and body section. The HTTP client application **104** is then able to act on the request and gather the appropriate results based thereon. The results of the requests are then communicated back through the HTTP server **102** to the application server **101** by contacting the HTTP server and making a request of another URL different than the first URL noted above. This HTTP request encapsulates an HTTP response, where the request headers include required data as well as enough state information to allow the HTTP server **102** to associate the encapsulated response with a previous request. The request body includes a full HTTP response.

[0024] In accordance with one particularly advantageous aspect of the invention, the communications device **108a** may function as a shared interface allowing the application server **101** to also access user accounts associated with the communications devices **108b-108n**. That is, since the communications devices **108a-108n** are connected in a network configuration (such as a local area network (LAN) or wide area network (WAN), for example), these devices may potentially access user account information stored on the network **107** (e.g., on a network server), and/or on one another, as well as other network data, as will be appreciated by those skilled in the art. By way of

example, the user accounts may be e-mail accounts, but numerous other types of information such as address/contact data, calendar data, etc., may also be accessed in this manner. As such, even though the HTTP client application **104** is only installed on the communications device **108a**, it may advantageously provide a "gateway" for the application server **101** to access user accounts associated with other communications devices **108b-108n**, as will be appreciated by those skilled in the art. Of course, it will also be appreciated that a separate HTTP client application **104** could be installed on one or more of the other communications devices **108b-108n**, if desired.

[0025] Turning additionally to FIG. 2, a flow diagram illustrating the decision path to connect the HTTP client application **104** to the HTTP server **102** is now described. Before the illustrated process flow begins (Block **200**), the HTTP client application **104** is installed on the communications device **108a** in the protected computing environment **106**. It should be noted that in some embodiments the HTTP client application **104** may instead be installed on a network server, for example, and provided the shared or common access functionality for multiple communications devices as described above. The software could advantageously be downloaded from the service provider hosting the HTTP server **102** and application server **101**, for example. For purposes of the present example, it will be assumed that the HTTP client application **104** is installed on a user's desktop PC in the protected computing environment **106** (i.e., on his desktop PC at work).

[0026] Upon installation, the HTTP client application **104** is assigned a GUID, which is saved in a knowledge base (not shown) accessible by the HTTP

server **102** and/or application server **101**. The HTTP client application **104** supplies this GUID in all communications with the HTTP server **102**. The decision flow begins with the user running a session of the HTTP client application **104** on the computing device **108a** in the protected computing environment **106**, at Block **201**. For example, the user may run the HTTP client application **104** upon leaving the office for the evening or for an extended period. The HTTP client application **104** opens a connection to the HTTP server **102**, at Block **202**, and identifies itself uniquely by supplying the GUID, at Block **206**. The HTTP client application **104** then requests a first dedicated URL to indicate that it is ready to accept work requests coming from the HTTP server **102**.

[0027] The HTTP server **102** then performs authentication to ensure a successful connection, at Block **208**. If the authentication succeeds, the HTTP server **102** then waits for a response, at Block **212**. If the authentication fails, a failure message is provided (Block **210**), and the HTTP server **102** loops back to the original starting point (Block **200**). The HTTP server **102** does not proceed until a successful authentication is registered.

[0028] As noted above, once a successful authentication is accepted, the HTTP server **102** waits for a response, at Block **212**, and then determines whether there is a timeout, at Block **214**. If there is a timeout, the HTTP server **102** then determines whether the HTTP reply is received, at Block **218**. If there is no timeout, the connection is closed (Block **216**), and the system loops back to the step illustrated at Block **202**.

[0029] If the HTTP reply is not received, the process also loops back to the step illustrated at Block **202**. If a reply is received, the HTTP server **102** unpacks the embedded HTTP request, at Block **220**, and processes the request, at Block **222**. The application server **100** ensures that the request is coming from a valid client application by retrieving the appropriate GUID from the knowledge base. The application server **101** then makes a request to HTTP server **102**, including the GUID. The HTTP server **102** turns the application request into a valid HTTP request, and forwards that request to the HTTP client application which has the identical GUID.

[0030] The HTTP client application **104** then performs the requested work, gathers the results, and creates an HTTP response, at Block **224**. The HTTP client application **104** contacts the HTTP server **102**, requests a second URL different from the first to indicate that it wishes to send back results rather than seeking work, and encapsulates the results as a valid HTTP response within the body of an HTTP request.

[0031] The HTTP client application **104** then determines whether an HTTP connection is open, at Block **226**. If it is open, the HTTP client application **104** sends a request for the second URL, at Block **232**. However, if the HTTP connection is not opened, the HTTP client application **104** opens another HTTP connection (Block **228**), authenticates the information (Block **230**), and then requests the revised URL (Block **232**).

[0032] After the HTTP client application **104** requests the revised URL, the HTTP client application sends the HTTP response as part of the HTTP request body, at Block **234**. The HTTP client application **104** then determines whether the HTTP connection is still

open, at Block **236**. If it is opened, the HTTP client application **104** loops back to the step illustrated at Block **204** to request the URL. If the connection is not open, the HTTP client application **104** loops back to the step illustrated at Block **202** to open the HTTP connection, and the process repeats itself.

[0033] Many modifications and other embodiments of the invention will come to the mind of one skilled in the art having the benefit of the teachings presented in the foregoing descriptions and the associated drawings. Therefore, it is understood that the invention is not to be limited to the specific embodiments disclosed, and that modifications and embodiments are intended to be included within the scope of the appended claims.

THAT WHICH IS CLAIMED IS:

1. A communications system comprising:

a plurality of communications devices connected together in a network and having a plurality of user accounts associated therewith, at least one of said communications devices processing requests using a hypertext transfer protocol (HTTP) client application associated therewith;

an application server for accessing the user accounts via said HTTP client application; and

an HTTP server for interfacing said HTTP client application with said application server;

wherein said HTTP server and said HTTP client application are for formatting requests to be communicated therebetween in an HTTP format via the Internet, and each providing additional state information with the HTTP formatted requests recognizable by the other for authenticating the application server and said HTTP client application to one another;

wherein said HTTP client application is for requesting a first universal resource locator (URL) from said HTTP server for accepting work requests from said application server, and requesting a second URL different from the first URL from said HTTP server for responding to work requests from said application server.

2. The communications system of Claim 1 wherein the additional state information comprises a global unique identifier (GUID) associated with said HTTP client application.

3. The communications system of Claim 1 wherein the user accounts comprise email accounts.

4. The communications system of Claim 1 wherein said HTTP client application and said HTTP server are for further providing sequencing information with the HTTP formatted requests.

5. The communications system of Claim 1 wherein said HTTP client application and said HTTP server are for formatting the additional state information as HTTP headers for respective HTTP formatted requests.

6. The communications system of Claim 1 wherein said at least one communications device is within a protected computing environment.

7. A method for accessing a plurality of user accounts associated with a plurality of communications device using an application server, the communications devices being connected together in a network, and at least one of the communications devices processing requests using a hypertext transfer protocol (HTTP) client application associated therewith, the method comprising:

interfacing the application server with the HTTP client application using an HTTP server, the HTTP server and the HTTP client application formatting requests to be communicated therebetween in an HTTP format via the Internet, and each providing additional state information with the HTTP formatted requests

recognizable by the other for authenticating the application server and the HTTP client application to one another; and

using the HTTP client application to request a first universal resource locator (URL) from the HTTP server for accepting work requests from the application server, and to request a second URL different from the first URL from the HTTP server for responding to work requests from the application server.

8. The method of Claim 7 wherein the additional state information comprises a global unique identifier (GUID) associated with the HTTP client application.

9. The method of Claim 7 wherein the user accounts comprise email accounts.

10. The method of Claim 7 wherein the HTTP client application and the HTTP server further provide sequencing information with the HTTP formatted requests.

11. The method of Claim 7 wherein the HTTP client application and the HTTP server format the additional state information as HTTP headers for respective HTTP formatted requests.

12. The method of Claim 7 wherein the at least one communications device is within a protected computing environment.