US 20060075467A1

(19) **United States**

(12) **Patent Application Publication**

Sanda et al.

(10) **Pub. No.: US 2006/0075467 A1**

(43) **Pub. Date: Apr. 6, 2006**

(54) **SYSTEMS AND METHODS FOR ENHANCED NETWORK ACCESS**

(76) Inventors: **Frank Seiji Sanda**, Tokyo (JP); **Naohisa Fukuda**, Tokyo (JP); **Edward W. Laves**, Golden, CO (US); **Robert L. Johnston**, Colorado Springs, CO (US); **Justin Owen Tidwell**, Aurora, CO (US); **Raymond T. Gurgone**, Woodstock, IL (US); **David S. Robins**, Buffalo Grove, IL (US); **Laura J. Worthington**, Centennial, CO (US); **Karlton Mark Zeitz**, Centennial, CO (US)

Correspondence Address:
KILPATRICK STOCKTON LLP
1001 WEST FOURTH STREET
WINSTON-SALEM, NC 27101

(21) Appl. No.: **11/167,747**

(22) Filed: **Jun. 27, 2005**

Related U.S. Application Data

(60) Provisional application No. 60/583,765, filed on Jun. 28, 2004. Provisional application No. 60/598,364, filed on Aug. 3, 2004. Provisional application No. 60/652,121, filed on Feb. 11, 2005. Provisional application No. 60/653,411, filed on Feb. 16, 2005.

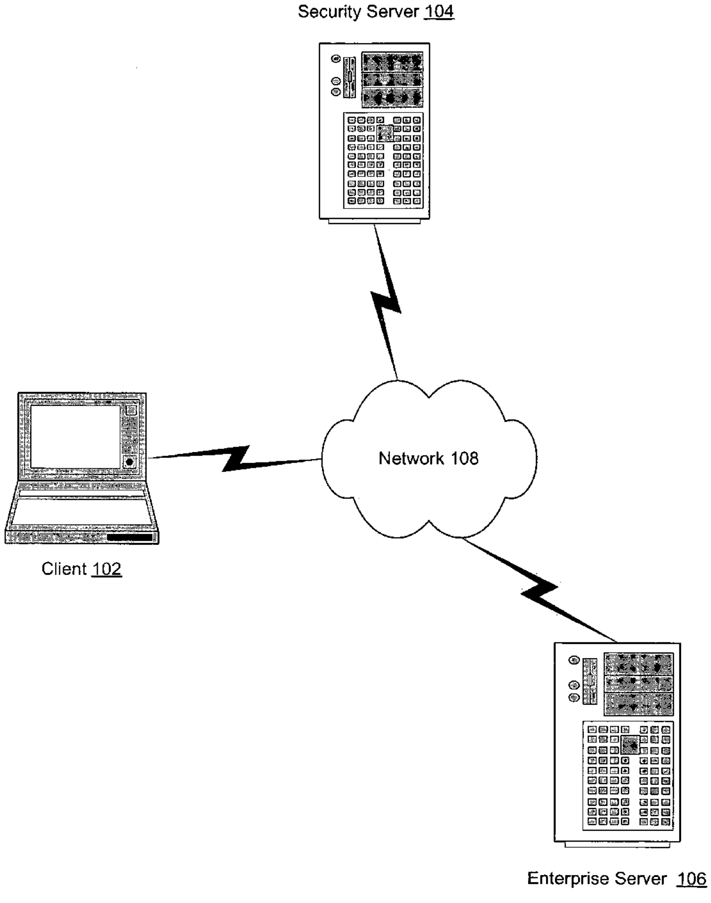
Publication Classification

(51) **Int. Cl.**
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **726/1**

(57) **ABSTRACT**

Systems and methods for enhanced network access are described. One aspect of one described embodiment includes receiving a request to connect to a network, the request associated with a user, determining a policy associated with the user, identifying at least one available network connection, determining at least one property of the at least one available network connection, evaluating the property based at least in part on the policy, and selecting the at least one available network connection based on the evaluation. In another embodiment, a computer-readable medium (such as, for example random access memory or a computer disk) includes code for carrying out such a method.



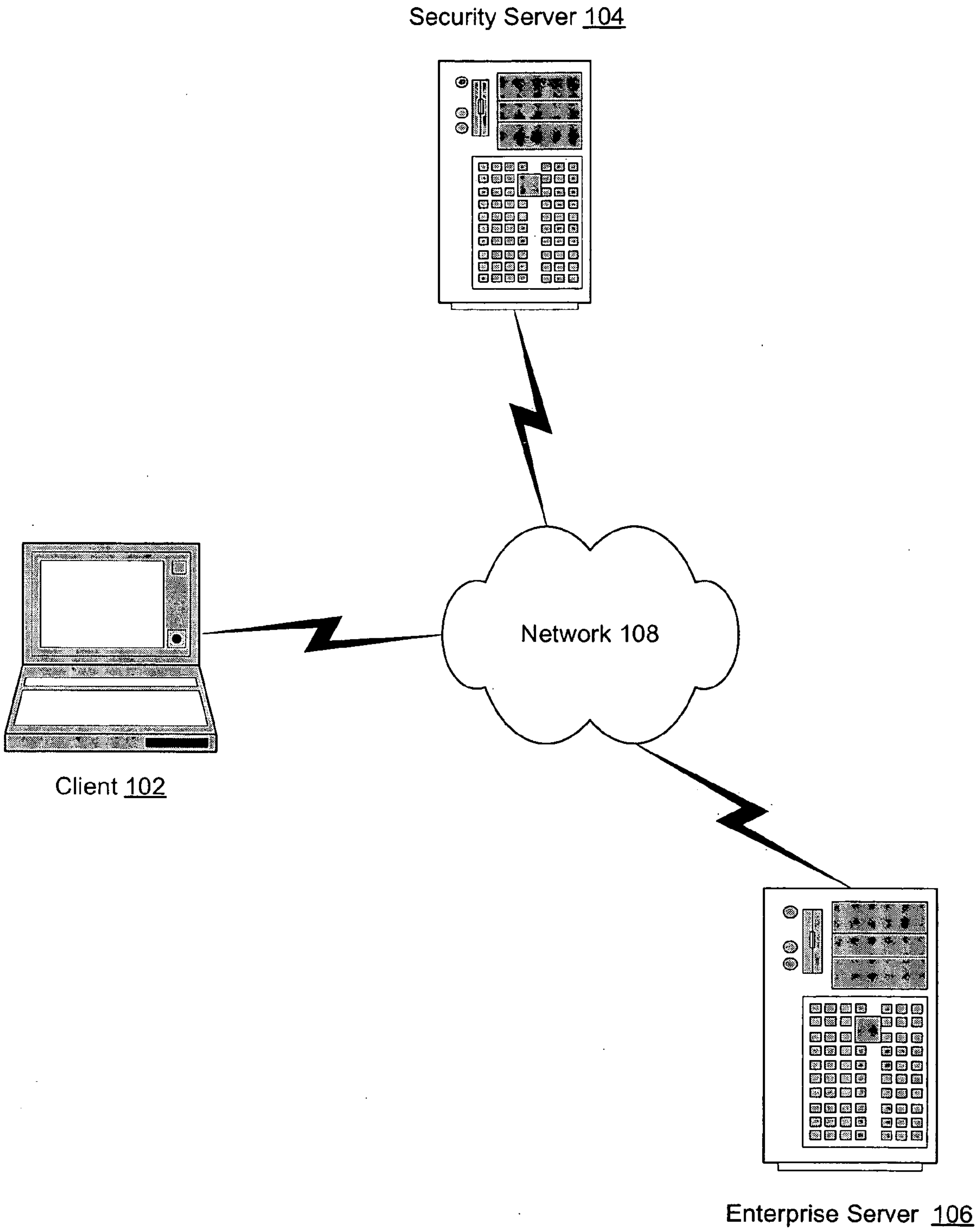

Security Server 104
Network 108
Client 102
Enterprise Server 106

Figure 1

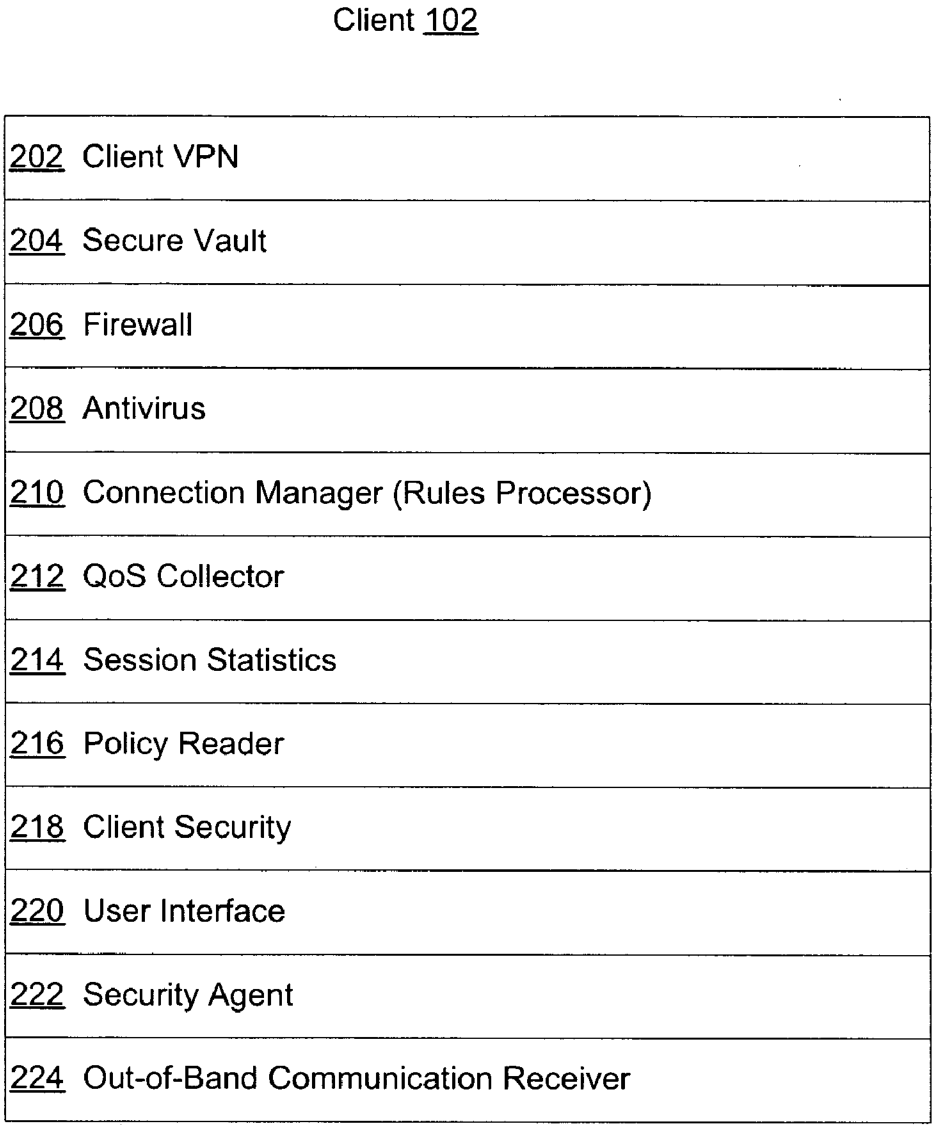

Client 102
202 Client VPN
204 Secure Vault
206 Firewall
208 Antivirus
210 Connection Manager (Rules Processor)
212 QoS Collector
214 Session Statistics
216 Policy Reader
218 Client Security
220 User Interface
222 Security Agent
224 Out-of-Band Communication Receiver

Figure 2

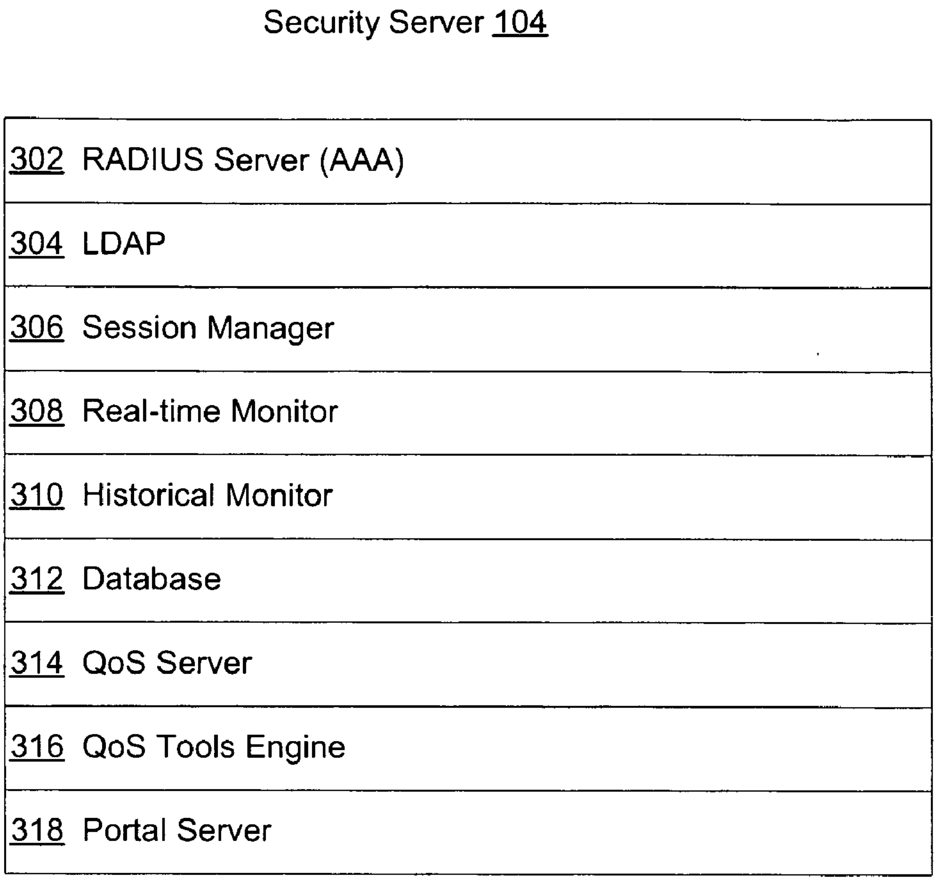

Security Server 104
302 RADIUS Server (AAA)
304 LDAP
306 Session Manager
308 Real-time Monitor
310 Historical Monitor
312 Database
314 QoS Server
316 QoS Tools Engine
318 Portal Server

Figure 3

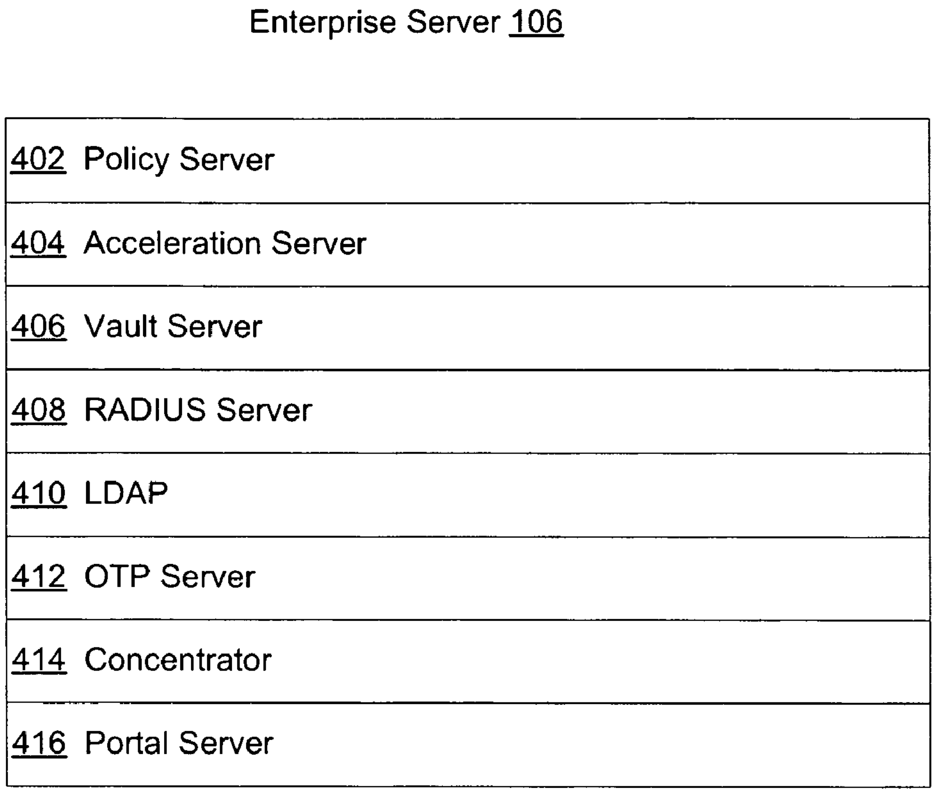

Enterprise Server 106
402 Policy Server
404 Acceleration Server
406 Vault Server
408 RADIUS Server
410 LDAP
412 OTP Server
414 Concentrator
416 Portal Server

Figure 4

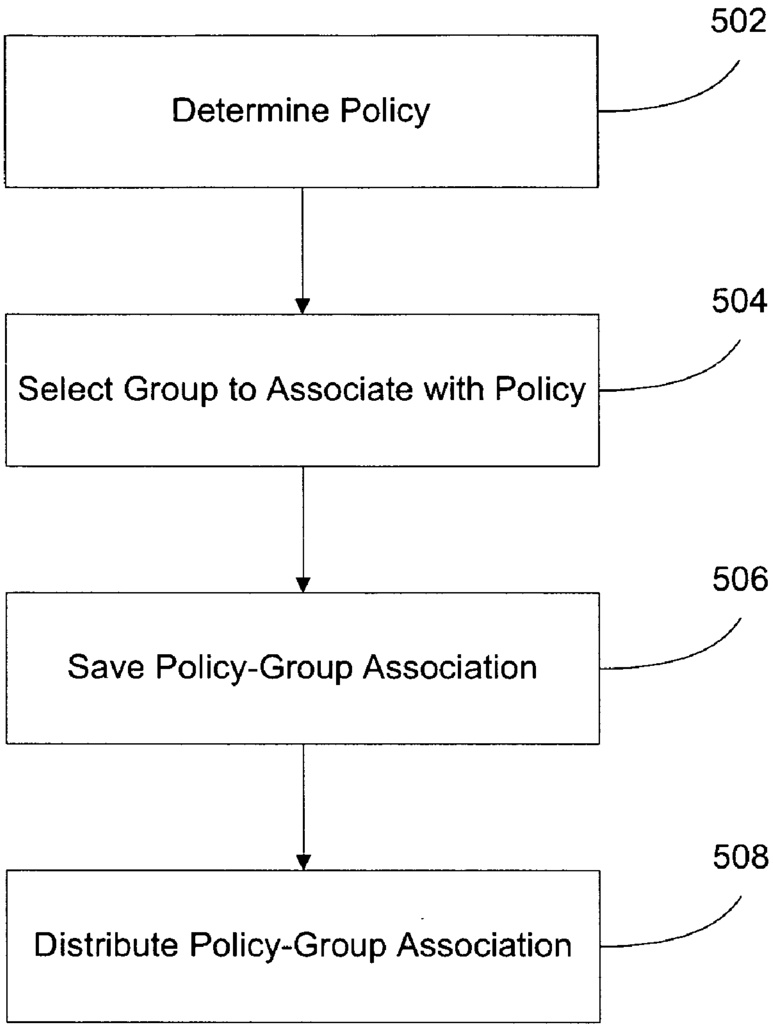

502
Determine Policy
504
Select Group to Associate with Policy
506
Save Policy-Group Association
508
Distribute Policy-Group Association

Figure 5

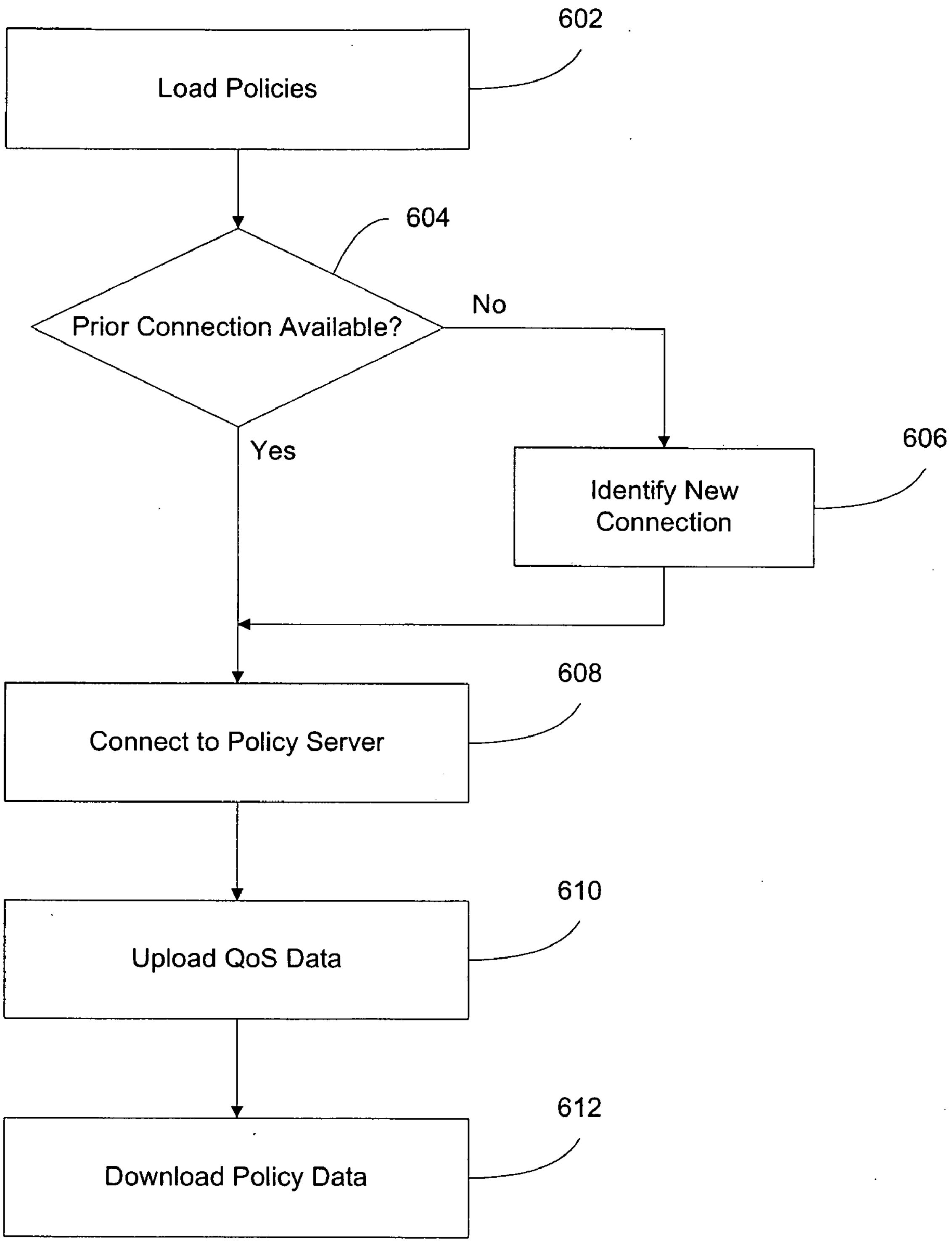

602
Load Policies
604
Prior Connection Available?
No
Yes
606
Identify New Connection
608
Connect to Policy Server
610
Upload QoS Data
612
Download Policy Data

Figure 6

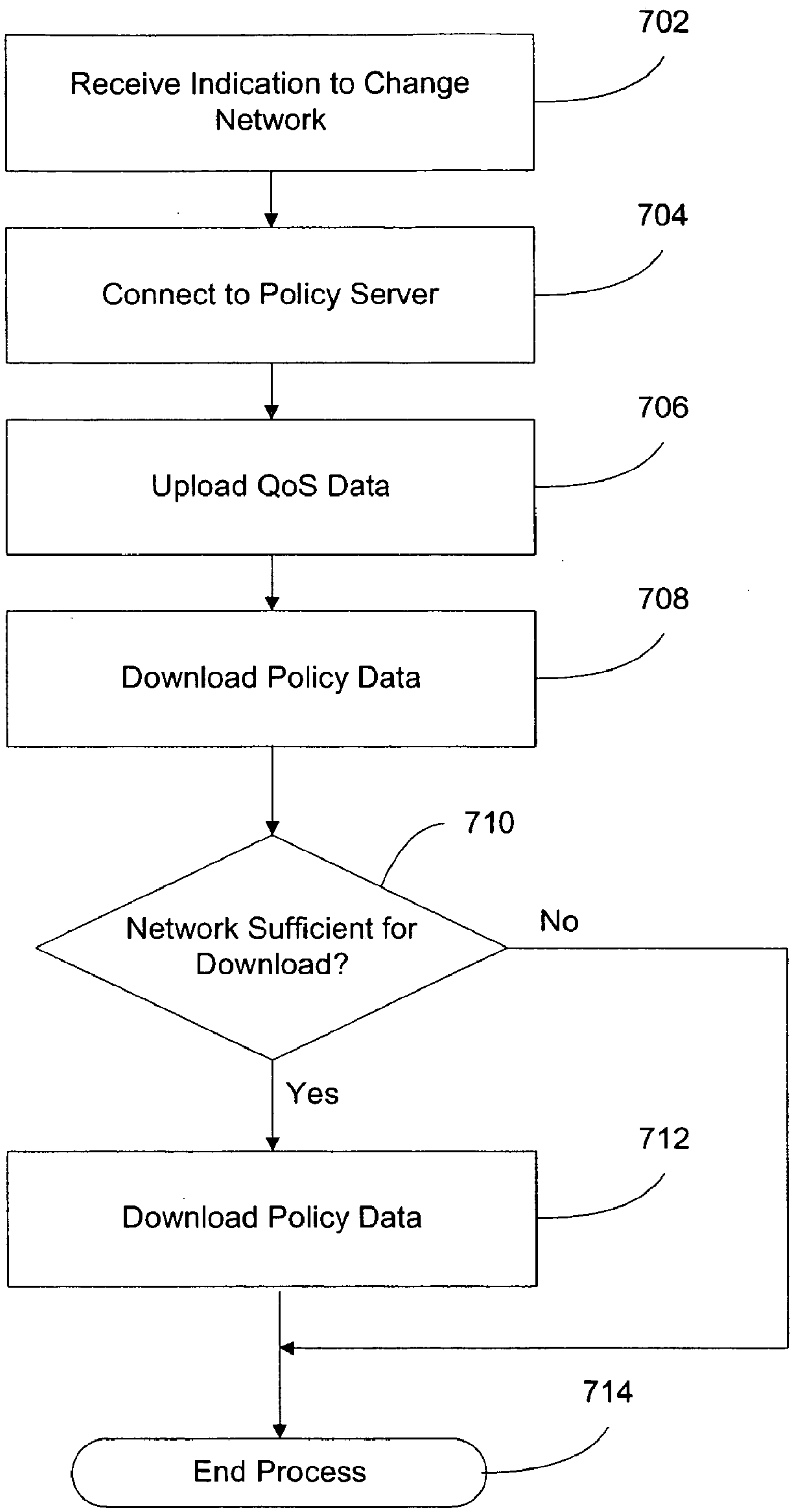

702
Receive Indication to Change Network
704
Connect to Policy Server
706
Upload QoS Data
708
Download Policy Data
710
Network Sufficient for Download?
No
Yes
712
Download Policy Data
714
End Process

Figure 7

SYSTEMS AND METHODS FOR ENHANCED NETWORK ACCESS

RELATED APPLICATIONS

[0001] This application claims priority to Application Ser. No. 60/583,765, filed on Jun. 28, 2004, titled "Controlling Use of a Mobile Work Station Based on Network Environment," Application Ser. No. 60/598,364, filed on Aug. 3, 2004, titled "Systems and Methods for Enhancing and Optimizing a User's Experience on an Electronic Device," Application Ser. No. 60/652,121, filed on Feb. 11, 2005, titled "Remote Access Services," and Application Ser. No. 60/653,411, filed on Feb. 16, 2005, titled "Creating an Environment for Secure Mobile Access Anywhere," the entirety of all of which are incorporated herein by reference.

FIELD OF THE INVENTION

[0002] The present invention relates generally to computer networking and, more particularly to systems and methods for enhanced network access.

BACKGROUND

[0003] As the workforce becomes more mobile, enterprises often must provide a means for their users to connect to the enterprise network remotely. Enterprises and their users have much greater flexibility in selecting methods of connecting to the enterprise network as well as other resources, such as the Internet. With this added flexibility comes a concomitant increase in complexity and risk. Thus, although remote access may be necessary, enterprises may resist providing their users with remote access.

[0004] Each remote method for connecting to an enterprise network opens a potential security hole that might be exploited. For instance, listeners on a network, such as rogue access points, may be able to determine a user's username/password combination for accessing the network.

[0005] Also, each connection type may be purchased from a different network provider. The enterprise must reconcile charges from each of the providers for each of the users accessing the network remotely.

[0006] Further, every conventional connection product connecting to the enterprise network provides a unique interface. And although each interface may be relatively straightforward, complexity arises from the enterprise and its users having to deal with multiple interfaces for each of the various network connections the user wishes to make.

SUMMARY

[0007] Embodiments of the present invention provide systems and methods for enhanced network access. One aspect of one embodiment of the present invention comprises receiving a request to connect to a network, the request associated with a user, determining a policy associated with the user, identifying at least one available network connection, determining at least one property of the at least one available network connection, evaluating the property based at least in part on the policy, and selecting the at least one available network connection based on the evaluation. In another embodiment, a computer-readable medium (such as, for example random access memory or a computer disk) comprises code for carrying out such a method.

[0008] This illustrative embodiment is mentioned not to limit or define the invention, but to provide one example to aid understanding thereof. Illustrative embodiments are discussed in the Detailed Description, and further description of the invention is provided there. Advantages offered by the various embodiments of the present invention may be further understood by examining this specification.

FIGURES

[0009] These and other features, aspects, and advantages of the present invention are better understood when the following Detailed Description is read with reference to the accompanying drawings, wherein:

[0010] **FIG. 1** is a block diagram showing an illustrative environment for implementation of one embodiment of the present invention;

[0011] **FIG. 2** is a block diagram illustrating the modules present on a client device **102** in one embodiment of the present invention;

[0012] **FIG. 3** is a block diagram illustrating the modules present on a security server **104** in one embodiment of the present invention;

[0013] **FIG. 4** is a block diagram illustrating the modules present on an enterprise server **106** in one embodiment of the present invention;

[0014] **FIG. 5** is a flowchart illustrating a process for creating a policy for a group in one embodiment of the present invention;

[0015] **FIG. 6** is a flowchart illustrating a process for updating policy information on a client in one embodiment of the present invention; and

[0016] **FIG. 7** is a flowchart illustrating a method for changing a network connection while downloading data in one embodiment of the present invention.

DETAILED DESCRIPTION

[0017] Embodiments of the present invention provide systems and methods for enhanced network access. There are multiple embodiments of the present invention. By way of introduction and example, one illustrative embodiment of the present invention provides a method for a client device to seamlessly switch from a first network connection to a second.

[0018] As a mobile client device moves from a first location to a second location, the device is provided with an indication that the second network connection is available. The device determines a set of properties regarding the second connection, such as speed, reliability, and cost for the connection. The client device then evaluates these properties based on a set of policies, which are specified by the enterprise with which the user is associated, so that at any one time, the user is connected to the "best" network from the point of view of the enterprise. A rules engine automatically determines which of the two connections is most suitable based on the policies and the properties of the connection.

[0019] For instance, the second connection may be faster and cheaper than the first. However, the first connection is more reliable. The enterprise policies weigh speed and cost more highly than reliability. Thus, the client device automatically switches from the first connection to the second connection without any user intervention.

[0020] This introduction is given to introduce the reader to the general subject matter of the application. By no means is the invention limited to such subject matter. Illustrative embodiments are described below.

System Architecture

[0021] Various systems in accordance with the present invention may be constructed. Referring now to the drawings in which like numerals indicate like elements throughout the several figures, **FIG. 1** is a block diagram showing an illustrative environment for implementation of one embodiment of the present invention. The system shown in **FIG. 1** includes a client **102**. The client is in communication with a security server **104**.

[0022] Communication with the security server **104** occurs via a network **108**. The network **108** may comprise a public or private network and may include the Internet. The network may also comprise a plurality of networks, including, for example, dedicated phone lines between the various components. In one embodiment, the client **102** communicates with the security server **104** via a virtual private network ("VPN") established over the Internet.

[0023] The security server **104** is also in communication with an enterprise server **106** via a network. The network **108** may comprise various elements, both wired and wireless. In one embodiment, the communication between the security server **104** and enterprise server **106** occurs over a static VPN established over dedicated communication lines.

[0024] In one embodiment, a user connects a client device **102** to the network **108** using a network access user interface. The network access user interface is always on and only allows the user to connect to the network **108** via the interface. The network access user interface automatically causes the client **102** to connect to the security server **104** through the network **108**. The security server **104** provides value added services to the client **102** and to one or more enterprises. Access to other services, such as the Internet, may be provided via the security server **104**.

[0025] Although **FIG. 1** includes only a single client **102**, security server **104**, and enterprise server **106**, an embodiment of the present invention will typically include a plurality of clients **102** and may include a plurality of security servers **104** and enterprise servers **106**.

[0026] **FIGS. 2 through 4** are block diagrams illustrating components on the client **102**, security server **104**, and enterprise server **106**. Each of the components shown may be a third-party application, a custom application, or a combination of both. Each of the components may also be implemented in hardware, software, or a combination of hardware and software.

Client Devices

[0027] **FIG. 2** is a block diagram illustrating the modules present on a client device **102** in one embodiment of the present invention. Examples of client device **102** are personal computers, digital assistants, personal digital assistants, cellular phones, mobile phones, smart phones, pagers, digital tablets, laptop computers, Internet appliances, and other processor-based devices. In general, a client device **102** may be any suitable type of processor-based platform that is connected to the network **108**, and that interacts with one or more application programs. The client device **102** can contain a processor coupled to a computer-readable medium, such as RAM. Client device **102** may operate on any operating system, such as Microsoft® Windows® or Linux. The client device **102** is, for example, a laptop computer executing a network access user interface.

[0028] The modules shown in **FIG. 2** represent functionality of the client **102**. The modules may be implemented as one or more computer programs that include one or more modules. For instance, in one embodiment, all the modules shown in **FIG. 2** are contained within a single network access application. Also, the functionality shown on the client **102** may be implemented on a server in other embodiments of the present invention. Likewise, functionality shown in **FIGS. 3 and 4** as being on a server may be implemented on the client **102** in some embodiments of the present invention.

[0029] The client **102** shown in **FIG. 2** comprises a VPN client **202**. The VPN client **202** allows the client **102** to connect to the enterprise server **106**. In one embodiment of the present invention, the VPN client **202** is used to determine whether or not the VPN client **202** is active and whether or not the VPN client **202** is connected to a VPN server. For instance, an embodiment of the present invention may determine whether or not to connect to a particular service based on whether or not the VPN client **202** is enabled.

[0030] In another embodiment of the present invention, the VPN client **202** is used for four purposes: (1) to manage policy files, which include information, such as a gateway Internet Protocol (IP) address, secrecy and authentication level, and hash; (2) automatically connecting a VPN; (3) automatically disconnecting the VPN; and (4) monitoring the status of the VPN. Each of these four purposes may be affected by other modules, including, for example, the connection manager **210**.

[0031] The client **102** also comprises a secure vault **204**. The secure vault **204** protects content on the client **102**. In one embodiment, the secure vault **204** is responsible for storing encrypted content on the client **102** and allowing access to the encrypted content based on a set of permissions or policies. In such an embodiment, a content creator can provide access via a viewer to secured content and allow a recipient of the content read-only access or allow the recipient to perform other tasks, such as modifying the content and forwarding it to other users. In another embodiment, the secure vault **204** allows the user to create and distribute secure content to other clients **102**, the content creator can decide to send a document to several users and allow two of the users full access and one of the users read-only access.

[0032] The client **102** shown in **FIG. 2** also comprises a firewall **206**. The firewall **206** allows port blocking via predefined policies. For instance, in one embodiment, an information technology ("IT") manager specifies port blocking based on two zones, a safe zone and a dangerous zone. The IT manager specifies one of these two zones for each of the network interface devices installed on the client **102**. The IT manager is then able to set port-blocking rules by zone on the firewall **206**.

[0033] For example, the IT manager may classify a Wireless Fidelity ("Wi-Fi") network interface as dangerous since it has traditionally been considered fairly unsafe. And the IT manager may apply more restrictive port-blocking rules to the dangerous zone than to the safe zone and network interface devices, such as those used to connect to a wired Local Area Network ("LAN") or a Personal Handyphone System ("PHS") cellular connection. The PHS standard is a TDD-TDMA based microcellular wireless communications technology and has been traditionally considered relatively safer than Wi-Fi connections. The PHS cellular connection may also be referred to as a wireless wide area network ("WWAN") as opposed to a dial-up connection providing access to a wide area network ("WAN").

[0034] In various other embodiments, the port-blocking rules of the firewall **206** may be based on time of day, client IP address, terminating IP address, terminating and originating port, protocol, and other variables. In one embodiment, the port-blocking rules are based on policy data associated with individual users logged into the client **102**.

[0035] In one embodiment, the port-blocking rules of the firewall **206** include a blacklist. The blacklist allows an IT manager to prevent an application from executing on the client **102**. For instance, an IT manager may blacklist a DVD player so that a user is unable to view DVD's on the client **102**. The firewall **206** may provide a message to the user informing the user that an application is unavailable.

[0036] In another embodiment, the firewall **206** implements a white list. The white list is somewhat more restrictive than the blacklist described above. The white list allows only specified applications to execute. For example, an IT manager may allow only MS Word, Excel, PowerPoint, and Outlook to execute. No other applications will be permitted to execute. The firewall **206** may be a custom firewall or a third-party firewall integrated into an embodiment of the present invention.

[0037] The embodiment shown in **FIG. 2** also includes an antivirus module **208**. The antivirus module **208** shown determines whether policy files, virus dictionary, or other virus-related resources are out of date and provides the client **102** with a mechanism for updating the files or data. The antivirus module **208** may restrict access to various connections, applications, and other functionality when the policy files are out of date. For instance, the antivirus module **208** may restrict the client **102** to connecting to a single gateway through which the policy files are available. In one embodiment, the antivirus module **208** comprises a third-party antivirus product that is integrated with the other modules on the client **102**.

[0038] The client **102** also comprises a connection manager **210**, which includes a rules processor. In one embodiment, the connection manager **210** assigns a priority number to every connection, e.g., one to one hundred, and selects the connection with the highest number to connect to.

[0039] The connection manager **210** may provide a connection to a variety of networks, including, for example, dial-up, LAN, digital subscriber line ("DSL"), cable modem, Wi-Fi, wireless local area network ("WLAN"), PHS, and satellite.

[0040] In one embodiment, the connection manager **210** differentiates between public and private connections. A public connection is a connection provided by a service provider who has a relationship with the administrator of the security server **104**, which allows the security server **104** to authenticate the connection. For instance, the security server **104** administrator may have a business arrangement with a hotspot provider. In order to connect, the client **102** connects to a local access point and the authentication of the user occurs automatically at the security server **104**. In contrast, a private connection requires that all aspects of the authentication mechanism for a connection are managed in the absence of the security server **104**, although the connection manager may provide certain facilities to allow for automated authentication where possible.

[0041] In one embodiment, the connection manager **210** makes connections available or unavailable to the client **102** based on policies present on the client **102**. The connection manager **210** may also download changes to policy data and transmit quality of service ("QoS") and other data to the security server **104** or the enterprise server **106**.

[0042] In one embodiment, the connection manager **210** determines the type of connections that are available based on signals provided by hardware associated with the client **102**. For example, when the client **102** passes near a hotspot, a Wi-Fi card in the client **102** senses the hotspot and sends a signal to the connection manager **210**. For instance, the Wi-Fi card may sense a broadcast service set identifier ("SSID"). Once the signal exceeds a threshold, the connection manager **210** provides a signal to a user of the client **102** that the network is available or may automatically connect to the hotspot. Alternatively, the Wi-Fi card may poll for a non-broadcast SSID. The connection manager **210** may provide a single connection to the client **102** at one time or may provide multiple connections to the client **102**.

[0043] The client **102** shown in **FIG. 2** also comprises a QoS collector **212**. The QoS collector **212** collects data values, including, for example, the number of bytes sent and received, the average transfer rate, the average signal strength at connection, termination cause, failed connections, and a network identifier. In another embodiment, the QoS collector **212** collects data during the session to determine when a connection provides inconsistent performance.

[0044] In one embodiment, the QoS collector **212** collects data regarding a connection during a session but does not send the data for a session until the next session. Thus, if a session is terminated abnormally, the QoS data will still be collected and transferred successfully. In another embodiment, the QoS collector **212** transfers data only when a particular type of connection is detected, such as a high-speed or low cost connection.

[0045] The client **102** also comprises a session statistics module **214**. The session statistics module stores data representing user characteristics. For instance, the session statistic module **214** may store a list of the applications a user generally accesses, how often the user is connected, the typical CPU and memory utilization measure, keyboard sequences, and other characteristics of a user. If a particular user deviates from the expected characteristics by greater than a threshold, such as N standard deviations, and the significance of the statistic is more than a specified amount, the session statistics module **214** can identify the current user as a potential unauthorized user.

[0046] The session statistics module **214** may perform other tasks as well. For instance, in one embodiment, the session statistics module **214** pre-loads applications based on a user's general usage patterns.

[**0047**] The client **102** shown in **FIG. 2** also comprises a policy reader **216**. In one embodiment, a company's policies are housed on the enterprise server **106**. For instance, individual groups and users within an enterprise are identified and associated with policies, such as what types of connections they are able to access and what a user's VPN profile is. The user may also be able to specify a VPN policy on the client **102**. In such an embodiment, the policy reader **216** downloads the policy rules from the enterprise server **106** and accesses local user policies and reconciles any conflicts between the two.

[**0048**] For example, an IT manager may establish a VPN profile to be used by a user when connecting to a Wi-Fi network. However, the user may wish to create a secondary VPN profile to be used if the first VPN becomes unavailable. The policy reader **216** loads both local and enterprise VPN profiles, resolving any conflict between the two VPN profiles.

[**0049**] In one embodiment, the policy reader **216** accesses data at an enterprise, department, and user level. In such an embodiment, some of the policy rules may be stored in a lightweight directory access protocol ("LDAP") server on the client **102**, security server **104**, or enterprise server **106**. In another embodiment, the policy reader **216** receives only changes to policy data and does not typically download all of the policy data at once. Policies downloaded by the policy reader **216** may be provided to the rules processor of the connection manager **210**.

[**0050**] The client **102** may also comprises a client security module **216**. In one embodiment, the client security module **216** implements a client asset protection process. When the client security module **216** receives a signal indicating that the client asset protection process is to be executed, the client security module **216** may, for example, disable devices and interfaces on the client device **102** and may, in some embodiments, encrypt the hard drive of the client device **102** so that the files stored on the drive are not easily accessible.

[**0051**] The client **102** may also comprise a user interface **220**. The user interface **220** may control the underlying operating environment or the user's view of the underlying environment. For example, in one embodiment, the user interface **220** supplants the Microsoft® Windows operating system interface from the user's perspective. In other words, the user is unable to access many of the standard Windows features. Such a user interface may be implemented to limit the applications and configuration setting a user is able to access. In some embodiments, such as a personal digital assistant ("PDA"), no user interface is provided by an embodiment of the present invention; the standard PDA user interface is utilized.

[**0052**] The user interface **220** provides the user with an easy-to-use mechanism for accessing network connections. In one embodiment, when the user interface **220** is visible, it provides a very easy-to-use format that displays network connection types and provides other functionality to the user. For example, during complex operations, such as connecting to a new network type, the user can simply select a single button within the user interface **220** and the client **102** will properly disconnect from the previous network, acquire the new network, perform all authentication and policy-based requirements, and then allow the user to continue using an application on the new network. This simple, easy-to-use user interface **220**, the complexity of which may be hidden and completely automatic, allows a less-technical user to successfully operate the client **102**. All network connection, authentication, secure sign on, VPN parameters, and other aspects of the connection are managed by the user interface **220**.

[**0053**] The client **102** shown in **FIG. 2** also comprises a security agent **222**. In some embodiments, the security agent **222** is also referred to as a "bomb." In one embodiment, an IT manager indicates that the security agent **222** should be activated when the client **102** next connects to the enterprise server **106**. The IT manager may do so because the client **102** has been reported stolen. Subsequently, the client **102** connects to the enterprise server **106**, either directly or indirectly and receives the message to initiate the security agent **222**.

[**0054**] In one embodiment, when the security agent **222** activates, it stops all applications from being able to run and encrypts the data on the hard drive of the client **102**. For instance, the security agent **222** may implement a white list as (described above and then implement a secure vault for all data on the client **102**. The connection manager **210** may also be configured so that no connections are possible.

[**0055**] In one such embodiment, since the data is merely encrypted by security agent **222**, rather than erased, the data may be recovered if the client **102** is subsequently recovered. For instance, the enterprise may retain the key needed for decrypting the local drive. The client **102** is returned to the enterprise, which then decrypts the drive. In another embodiment, the data on the local drive of the client is rendered inaccessible by, for example, writing over the data multiple times.

[**0056**] The client **102** shown in **FIG. 2** also comprises an out-of-band communication receiver **224**. The out-of-band communication receiver **224** allows the client to receive communications other than through a network-based connection. The connection manager **210** may manage the out-of-band communication. For instance, the command to activate the security agent **222** may be transferred via a short messaging service ("SMS") communication received by the out-of-band communication receiver **224**.

Security Server

[**0057**] **FIG. 3** is a block diagram illustrating the modules present on a security server **104** in one embodiment of the present invention. The security server **104** shown in **FIG. 3** comprises a remote authentication dial-in user service ("RADIUS") server **302**, which may also be referred to as an AAA (authentication, authorization, and accounting) server. RADIUS is the standard by which applications and devices communicate with an AAA server.

[**0058**] The RADIUS server **302** provides authentication services on the security server **104**. In some embodiments of the present invention, the RADIUS server **302** proxies to a RADIUS server on the enterprise server **106**. In one embodiment, the RADIUS server **302** provides mutual authentication for the client **102** using Extensible Authentication

Protocol Transport Layer Security ("EAP-TLS"). Although EAP-TLS itself is strictly an **802**.lx authentication protocol, designed primarily for WiFi connections, the underlying TLS authentication protocol may be deployed in both wired and wireless networks. EAP-TLS performs mutual secured sockets layer ("SSL") authentication. This requires both the client device **102** and the RADIUS server **302** to have a certificate. In mutual authentication, each side may prove its identity to the other using its certificate and its private key.

[**0059**] The security server shown in **FIG. 3** also comprises an LDAP server **304**. The LDAP server **304** uses the LDAP protocol, which provides a mechanism for locating users, organizations, and other resources on the network. In one embodiment of the present invention, the LDAP server **304** provides access control at the network layer to various components that an enterprise customer may or may not purchase. For example, a customer may choose to implement a secure vault as described in relation to **FIG. 1**. In such a case, the customer or users or groups associated with the customer are also associated with the firewall module. The LDAP entry is then used to determine that the firewall is to be enabled on a client.

[**0060**] In some embodiments, the LDAP server **304** is implemented as a list of user identifiers not using the LDAP protocol. In another embodiment, data in the LDAP server **304** is propagated from data present in the enterprise server **106**.

[**0061**] The security server **104** shown in **FIG. 3** also comprises a session manager **306**. The session manager **306** controls sessions, including sessions between the client **102** and enterprise server **106**. In some embodiments, the session manager **306** also determines how to route data requests. For instance, the session manager **306** may determine that a particular data request should be routed to the Internet rather than to the enterprise server **106**. This may be referred to as "splitting the pipe" and provides a mechanism to replace "split tunneling" (a traditional configuration option with most standard VPN clients) at the client device by the more secure split of traffic not intended for the enterprise at the security server, allowing monitoring of all traffic without the enterprise incurring the expense of the extra bandwidth required.

[**0062**] In some embodiments, the client **102** and enterprise server **106** establish a VPN for communication. In such an embodiment, the session manager **306** may be unable to route requests to any location other than the enterprise—the packets are encrypted and thus, cannot be separately evaluated.

[**0063**] In one embodiment, the session manager **306** performs automated authentication of a client device **102** or user. For example, if the session manager **306** determines that a client **102** is approaching a Wi-Fi hotspot, the session manager **306** is able to pre-populate the hotspot with the certificate that the hotspot requires to authenticate the user. In this manner, the authentication appears very fast to the user. The session manager **306** may also control the manner in which data is queued for download to the client device **102**.

[**0064**] In one such embodiment, the session manager **306** provides two modes for data queuing. In a first mode, the session manager **306** determines that the network down time will be brief, e.g., the user is moving through a tunnel, which interferes with network access. In such a case, the session manager queues a minimal amount of data. In a second mode, the session manager **306** determines that the network down time will be of a longer duration, e.g., the user is boarding a plane from New York to Tokyo. In such a case, the session manager **306** may queue a larger amount of data. In one such embodiment, the session manager **306** determines the mode by querying the user for the downtime interval. When the user reconnects to the security server **104**, the session manager **306** determines the best manner of downloading the queued data and begins the download.

[**0065**] In one embodiment, the session manager **306** comprises a packet shaper (not shown). The packet shaper provides various functional capabilities to the session manager **306**. For example, in one embodiment, the packet shaper provides a mechanism for prioritizing packets sent between the enterprise server **106** and the client **102**. In one embodiment, the packet shaper utilizes Multiprotocol Label Switching ("MPLS"). MPLS allows a specific path to be specified for a given sequence of packets. MPLS allows most packets to be forwarded at the switching (layer 2) level rather than at the (routing) layer 3 level. MPLS provides a means for providing QoS for data transmissions, particularly as networks begin to carry more varied traffic.

[**0066**] The session manager **306** may also provide session persistence capabilities. For instance, in one embodiment, when a user drops a connection or moves from one provider network coverage area to another, the connection manager **306** persists a virtual connection as the first connection is terminated and the second is initiated.

[**0067**] The session manager **306** may include a server-side rules engine. The server-side rules engine may use historical information, such as the session statistics described above, for statistical attack determination. For instance, session manager **306** may access a stored statistic regarding a client device **102** and based on monitoring of the current statistics for the client device **102** determine that an unauthorized user is using the client device **102**.

[**0068**] The security server **104** shown in **FIG. 3** also comprises a real-time monitor **308**. The real-time monitor **308** monitors the status of communications, such as which clients and users are logged on, the amount of data being transferred, ongoing QoS measures, ports in use, and other information.

[**0069**] When the real-time monitor **308** detects a problem, it may issue an alert to network support. In one embodiment, data from the real-time monitor **308** is provided to users via a portal available on the security server **308**. In another embodiment, the real-time portal **308** transfers information to the enterprise server **106**, from which users access the data.

[**0070**] The embodiment shown in **FIG. 3** also comprises a historical monitor **310**. The historical monitor **310** provides information similar to the real-time monitor **310**. However, the underlying data is historical in nature. For instance, in one embodiment, the historical monitor **310** provides audit information for making intelligent business decisions and for dealing with regulatory compliance issues.

[**0071**] The information available via the historical monitor **310** may include, for example, historical QoS data, registration compliance data, and metrics consistency data. The historical data monitor **310** may be used to determine that certain clients are not performing optimally by comparing metrics of various clients over time. For instance, by evaluating information available via the historical data monitor **310**, a support person may be able to determine that a radio tuner on a specific client device **102** is failing. If the user of one client device **102** is complaining about the availability of service, but other users are able to successfully access service, then the client device's radio may be the problem.

[0072] The historical data monitor **310** may also be used to reconcile information captured on the security server **104** regarding connections and data provided by telecommunication carriers. The data may be used to determine when certain resources need to be increased and when a certain carrier is not performing adequately.

[0073] The security server also comprises a database **312**. In embodiments of the present invention, the database **312** may be any type of database, including, for example, MySQL, Oracle, or Microsoft SQL Server relational databases. Also, although the database **312** is shown as a single database in **FIG. 2**, the database **312** may actually comprise multiple databases, multiple schemas within one or more databases, and multiples tables within one or more schemas. The database **312** may also be present on one or more other machines, e.g., database servers.

[0074] In one embodiment of the present invention, the database **312** stores customer information regarding enterprises served by the security server **104**, such as a list of valid users, a list of valid cellular cards, the relationships between the individual users and groups within the enterprise, and other customer information.

[0075] For example, in one embodiment, the database **312** stores an association between users and cellular data cards. The enterprise may allocate a single user to a specific data card. Alternatively, the enterprise may associate a group of users with a group of cellular data cards. Other types of data may also be stored in the database **312**, such as billing data.

[0076] The security server **104** shown in **FIG. 3** also comprises a QoS server **314**. The QoS server **314** uploads information from the QoS collector **212** on the client device **102** and stores the QoS data. The QoS server **314** can collect data from multiple clients and store it in the database **312**.

[0077] The security server also comprises a QoS tools engine **316**. The QoS tools engine **316** displays data made available by the QoS server **314** and other processes, such as the real-time monitor **308**.

[0078] In one embodiment, the QoS tools engine **316** provides an aggregation of QoS data in a spreadsheet. In another embodiment, the QoS tools engine **316** provides data using map views, pie charts, and graphs. The QoS tools engine **316** may also provide the capability for setting QoS-based alarms and may provide data to users via a portal.

[0079] In the embodiment shown in **FIG. 3**, the security server **104** also comprises a portal server **318**. The portal server **318** may be, for example, a web server. Any standard web server application may be utilized, including Microsoft® Internet Information Server ("IIS") or Apache.

[0080] Although the security server **104** shown in **FIGS. 1 and 3** is illustrated as a single server, it may comprise multiple servers. For example, in one embodiment of the present invention, the security server **104** comprises multiple regional servers.

[0081] Also, the description above suggests that data is provided to and queried from the security server **104** by the client **102**, i.e., the client pulls the data. However, in some embodiments, the client **102** also comprises a listener (not shown) so that the security server **104** can push data to the client **102**.

Enterprise Server

[0082] **FIG. 4** is a block diagram illustrating the modules present on an enterprise server **106** in one embodiment of the present invention. The enterprise server **106** may also be referred to herein as a customer server and may comprise one or more servers for one or more enterprises linked to one or more security servers **104**.

[0083] The enterprise server **106** shown in **FIG. 4** comprises a policy server **402**. The policy server **402** provides a means for managing the policy rules, including, for example, available VPN profiles, available transports (e.g. WiFi, LAN, PHS, Dialup), firewall rules, such as blacklists and white lists, connection rules, and antivirus rules. The policy server **402** may include other rules as well, such as the level of data throttling to perform for each client or group of clients. Data throttling limits the data transfer rate to a particular client **102** so that connection resources can be optimized.

[0084] The policies may be managed at one or more levels. For example, an IT manager may wish to create a VPN profile for the enterprise as a whole, but a different VPN profile for an engineering group since the engineering group needs access to various unique applications.

[0085] The policy server **402** may also provide a mechanism for configuring the location of various servers that the client **102** will utilize. For instance, the policy server **402** may allow an IT manager to specify the IP address of an acceleration server **404** or a vault server **406**

[0086] In one embodiment, the policy server also allows the IT manager to specify which users receive updates for various components on the client **102**. The policy server **402** may also allow the IT manager to perform connection configuration. For instance, the IT manager may use the policy server to specify phone numbers for PHS connections, Wi-Fi SSID's for private connections, and other connection configuration information.

[0087] The enterprise server **106** shown in **FIG. 4** also comprises an acceleration server **404**. The acceleration server **404** performs processes to improve the performance of data transfer. For instance, the acceleration server **404** may automatically compress images that are to be transferred to a client **102**.

[0088] In one embodiment, the acceleration server **404** communicates with the policy server **402**. An IT manager sets acceleration rules using the policy server **402**, and the acceleration server **404** uses these rules to determine what level of acceleration to use for a particular communication. In one embodiment, the IT manager sets a default level of acceleration for all communication and a specific level of acceleration for one group of users. The specific level of acceleration may be referred to as an override.

[0089] The enterprise server **106** also comprises a vault server **406**. The vault server comprises two components, an automatic component and an administration component. In one embodiment, the automatic component integrates with an enterprise's mail server (not shown) and performs operations on emails to and from the mail server. For instance, the vault server **406** may quarantine an email, automatically encrypt the email before it is sent, add a legal disclaimer to an email, or perform other functions on the email.

[0090] In one embodiment, the automatic component of the vault server **406** searches an email based on words or based on the domain or specific address to which the email is addressed or from which the email originated. Using this information, the user can perform functions on the email, such as those described above.

[0091] The administration component of the vault server **406** allows a user to terminate access to secure content, either by a specific user or by all users. It also logs activity. Using one embodiment of the vault server **406**, a user can indicate that a set of users whose employment has been terminated will no longer have access to any secure content. In an alternative embodiment of the vault server **406**, a user can indicate that a given element of secure content, say a price list, is now out of date, and so that piece of secure content will no longer be viewable by any user. When each user accesses the secure content, the vault server **406** logs the event. So for each secure content element, the vault server **406** creates a log of all activity on the secure content.

[0092] In one embodiment, the vault server **406** also compresses data. For instance, one embodiment utilizes standard PKZIP compression to compress all content. In another embodiment, an IT manager may identify three types of images and specify a different level of compression for each type of image based on the level of resolution necessary for each type of image.

[0093] The enterprise server **108** also comprises a RADIUS server **408** and LDAP server **410**, which are similar to those described above in relation to the security server **104**. The RADIUS server **302** on the security server **104** may proxy to the RADIUS server **408** on the enterprise server **106**. Similarly, data in the LDAP server **410** may be propagated to the LDAP server **204** on the security server **104**.

[0094] The enterprise server **106** also comprises a one-time password ("OTP") server **412**. The OTP server **412** provides a mechanism for authentication. For instance, in one embodiment of the present invention, the enterprise server **106** uses the OTP server **412** to perform a mutual authentication process.

[0095] The enterprise server **106** also comprises a concentrator **414**. The concentrator **414** provides remote access capability to the client **102**. For instance, the concentrator **414** may serve as a means for terminating a VPN between the client **102** and enterprise server **106**.

[0096] The enterprise server **104** shown in **FIG. 4** also comprises a portal server **416**. The portal server **416** may comprise a standard web server, such as IIS or Apache. The portal server **416** may provide one or more portals. For example, in one embodiment, the portal server **416** provides two portals, portal one and portal two.

[0097] Portal one provides a configuration interface for managing the various elements shown in **FIGS. 2 and 3**, including, for example, the policy server **402** and LDAP server **410**. Portal two provides an interface for accessing data, such as QoS data and session data.

[0098] For instance, a user may use historical QoS data on portal two to determine how a particular provider is performing in terms of throughput, user connections, and other QoS metrics. Portal two may also provide real-time information, such as how many users are currently connected.

[0099] For instance, in one embodiment, an IT manager determines that twenty users have been rejected by a carrier in the last three minutes due to authentication failure and five users with the same user identifier are currently logged on to five different devices. The IT manager uses this information to detect a potential security problem. Portal two may also be used to set alerts as described above.

[0100] It should be noted that the present invention may comprise systems having a different architecture than that which is shown in **FIGS. 1 through 4**. For example, in some systems according to the present invention, the security server **104** and enterprise server **106** may comprise a plurality of security and enterprise servers. The system **100** shown in **FIGS. 1 through 4** is merely illustrative, and is used to help explain the illustrative systems and processes discussed below.

Illustrative Methods of Enhanced Network Access

[0101] The following illustrative embodiments utilize a central policy server **402** on an enterprise server **106**. The client device **102** downloads policies from the policy server **402** and the connection manager **210** utilizes the policies to make connections. In other embodiments, policy files are created and distributed to the client device **102** in other ways. For example, an email attachment or disk may be distributed to each client device **102**. Each time an update is necessary, a new disk or email is distributed.

[0102] In embodiments of the present invention, policies are created and distributed to client devices **102**. The client devices **102** utilize the policies for making connections. **FIG. 5** is a flowchart illustrating a process for creating a policy for a group in one embodiment of the present invention.

[0103] In the embodiment shown in **FIG. 5**, policy administrator first determines a policy **502**. The policy may be based on a variety of factors, such as, for example, regulatory issues, the type of data and/or applications to be used, the level of control desired by the enterprise, the physical environment in which a client device or group of client devices will operate, the experience and technical knowledge of the user group associated with the policy, and the job to be performed by the group or groups associated with the policy. The policy may include various elements, including, for example a list of allowed connections or applications and a list of disallowed connections or applications, acceleration preferences, and a VPN profile. The policy may also include connection preferences of the enterprise or of the users. For instance, an enterprise may decide not to use certain WiFi hotspots. The policy may vary by time of day.

[0104] The policies may be based on a number of factors, including, for example, an enterprise's need to minimize overall transport cost when billed on a usage basis, an enterprise's wish to minimize perceived security exposure based on assumed insecurity on some transports and specific connections, and an enterprise's wish to ensure the highest speed and most reliable usage experience for their users. The policies may be based on third party parameters as well. For instance, the policy may be based on the enterprise's security provider's desire to minimize its transport costs overall.

[0105] The policy administrator next selects a group to be associated with the policy **504**. The administrator may manually enter group names and user identifiers. Alternatively, the administrator may select groups from a central directory, such as the LDAP **410** or a Microsoft Active Directory Server. Each policy may be associated with, for example, an enterprise, a group within the enterprise or across enterprises, or with an individual.

[0106] Once the administrator has created a policy and associated the policy with a group, the administrator saves the policy-group association **506**. The policy-group association may be stored in a database (not shown) in communication with the policy server **402**. Alternatively, the policy-group association may be stored in a file, for example, in an XML format in a file.

[0107] The administrator then causes the policy and policy-group association to be distributed to client devices **508**. For instance, the policy and policy-group association may reside in the database, which is in communication with the policy server **402**, so that when a client device attempts to download a policy, the policy-group association is used to determine which policy or policies to download. The policy and policy-group association may be distributed via a network, such as network **108**, or by media, such as CD-ROM. In one embodiment, only changes to the policies are downloaded. In other embodiments, all policies are downloaded each time a download occurs.

[0108] In some embodiments, once the policies are downloaded to the client device, the user may make changes to them. For instance, the user may set up an alternative VPN profile on the client if the VPN associated with the VPN profile downloaded from the policy server **402** is temporarily unusable.

[0109] **FIG. 6** is a flowchart illustrating a process for updating policy information on a client in one embodiment of the present invention. In the embodiment shown, the policy reader **216** on the client device **102** loads policies **602**. For instance, the policies may exist in an XML file, which the policy reader **216** opens and reads.

[0110] The connection manager **210** then determines what the most recently used connection was and whether the most recently used connection is available **604**. For example, a user may shut down a client device **102** while the client device **102** is connected to a WiFi hotspot. When the user starts the client device **102**, the client device will attempt to connect to the WiFi hotspot.

[0111] If the client device **102** is not able to connect to the most recently used connection, the client device **102** attempts to identify a new connection **606**. For example, the client device **102** may have moved out of range of the WiFi hotspot to which the client was connected. The client device **102** identifies all currently-available connections. The client device **102** also identifies one or more policies associated with the user. The client device **102** then identifies properties of the network connection or connections and compares the network properties to the policy (rule). A network property may be, for example, quality of service measures, such as security, reliability, and speed may be utilized. The client device **102** may also use cost or a combination of cost and a plurality of other properties in making the determination. In one embodiment, the client device **102** applies a normalization algorithm to the plurality of properties to come up with a single number for each network connection. The client device **102** then compares the single numbers to determine to which network to connect.

[0112] Using either the most recently used connection or the newly-identified connection, the client device connects to the policy server (**402**) **608**. For instance, the client device **102** may connect to the most recently used WiFi hotspot and then establish a connection with the policy server **402** and the QoS server **310** over the Internet.

[0113] Once the client device **102** has established a connection with the network, the client device uploads QoS data to the QoS server (**310**) **610**. In one embodiment, QoS data from the previous session is uploaded at the start of the current session so that interruptions in service, such as a lost connection, can be accurately tracked.

[0114] The client device next downloads the latest policy data from the policy server (**402**) **612**. The policy data may comprise only changes since the last connection. For example, the client device **102** may store a last download date and only download policies from the policy server **402** that have been created or changed since the last update date. The client device may download other information as well. For instance, the administrator may determine that a particular client device **102** has been stolen and set an indicator to cause the client device to encrypt data on its hard drive. When the client device **102** connects, it downloads the indicator.

[0115] The process may be transparent to the user. In one embodiment, the download process runs as a service. Each time the client device **102** starts up, the process executes. The process may also include having the client device **102** connect to a VPN automatically so that the user can access enterprise applications.

[0116] **FIG. 7** is a flowchart illustrating a method for changing a network connection while downloading data in one embodiment of the present invention. In the embodiment shown, the connection manager **210** receives an indication to change network connections **702**. The indication may be due to an identification of a newly available network, a manual selection by a user, or some other indication. For instance, a hardware device, such as a cellular data card, may indicate that a new connection is available. The user receives the indication and decides to change to the new network and clicks a button on a user interface, indicating the desire to change.

[0117] In one embodiment, the connection manager **210** identifies a new network. The connection manager then compares a first property of the currently connected (existing) network to a first property of the second new network. The properties may signify the same or similar information about the two networks, e.g., the type of network. Based on policies, the connection manager **210** determines to which network to connect.

[0118] In one embodiment, the connection manger **210** rules engine makes connection decisions based on six core pieces of data for connection that is physically available (the correct device is installed and operating, and a signal is available):

[0119] (1) Is the connection allowed for this user;

[0120] (2) How secure is the connection deemed to be;

[0121] (3) How reliable is the connection deemed to be;

[0122] (4) How fast is the connection deemed to be;

[0123] (5) How expensive is the connection relative to others for the enterprise; and

[0124] (6) How expensive is the connection relative to others for a service provider, such as a network security service provider?

[0125] Item (1) in this list is specified based on whether or not the connection is available to the enterprise in general and whether the enterprise has made the connection available to the user (or, more precisely in some implementations, not barred the user from the connection). Item (2) is based on enterprise preference indication. It could also be based on attack detection algorithms automatically applied, e.g., if relatively more attacks on a specific type of connection or specific location are detected, then relatively more attacks are occurring.

[0126] Item (3) is based on connection statistics. In one such embodiment, the enterprises has the option to indicate perceived relative reliability measures. Item (4) is also based on connection statistics. Item (5) is based on the pricing plan that the enterprise has entered into with the provider. And item (6) is based on carrier pricing arrangements and usage assumptions for various connections.

[0127] One embodiment of the present invention takes each of these six items and uses a normalization algorithm to work these elements (with their relative strengths) into a "weighting" within a range. Then, the rules engine on the client device **102** simply selects the connection with the highest weighting.

[0128] In some cases, despite the rules based analysis, an enterprise may not wish for a user to use a specific connection for a given, short period of time. In one embodiment, the system allows the enterprise to specifically exclude a connection for a short time.

[0129] Before disconnecting, the connection manager **210** sends a signal to the session persistence server **316** to suspend any currently active data transfer **704**. By suspending data transfer, the connection manager **210** helps to eliminate the potential for losing data.

[0130] The connection manager **210** then disconnects the client device **102** from the network (**108**) **706**. The process for disconnecting may differ between various networks. During the period when the client device **102** is disconnected, the session persistence server **316** caches data. After a period of time, which may be very brief, the connection **210** manager attempts to reconnect to a network **708**. The network **108** may be the most recently used network or may be a newly identified network. The connection may be dropped for a period of time. In one embodiment, a user may specify the duration that the user expects to be disconnected when the disconnect occurs. The persistence server **316** uses this information to determine how much data to cache during the period of disconnection.

[0131] The connection manager **210** or persistence server **316** then determines whether the download that was occurring before the disconnect should continue **710**. For instance, the connection manager **210** may determine that the new connection is too slow to support the data download. The connection manager **210** may also look to the policies to determine what rules apply to the connection.

[0132] If the download, should continue, the connection manager **210** resumes the data transfer **712**. The data transfer may then complete or may be subject to subsequent disconnects. If the new network is not suitable to support the download, or if the download is complete, the process ends **714**.

[0133] For example, in one embodiment, a salesperson needs to download a large document containing a price list. The salesperson's computer is currently connected via a wide area network connection, which is relatively slow. The salesperson enters a coffee shop that the salesperson knows has a high speed WiFi connection.

[0134] When the client device **102** indicates that the WiFi connection is available, the user indicates that the client device **102** should change networks. The client device seamlessly connects to the WiFi network and, based on rules established in the policies, begins downloading the document. If the user must leave the coffee shop before the download is complete, the connection manager **210** can signal the session persistence server **316** to pause the download until the user reenters the coffee shop or connects to another high-speed network.

[0135] In one embodiment, session persistence operates on the following process: when a disconnection event occurs, the connection manager **216** buffers application data coming to the client device **102**, making applications "believe" that they are still connected. At the same time, session persistence server **216** buffers information on the server side, making the server **104**"believe" that it is still connected. Once a network connection has been reconnected, the connection manager **210** and session persistence server **316** empty the buffers that have built up on both sides.

[0136] One such embodiment implements a kernel mode driver at the NDIS layer in the Microsoft Protocol stack (roughly equivalent to layer 3 in the OSI model). This kernel mode driver is implemented as an "Intermediate Driver" on the Microsoft W2K/WXP operating systems. The driver acts as a single "virtual device" through which all network communications goes. This single device routes this traffic to the appropriate physical device (directed through a virtual device associated with a third-party VPN when appropriate), depending on the current physical connection.

[0137] In order to make application layer components believe that a network is still up and running, no "disconnect" signal is transmitted to the application layer components when a network interruption occurs. In this way, application layer components treat the connection as if it is simply slow.

[0138] The primary interface component between the client **102** and server **104** in such an embodiment is an indication that a client will be disconnecting for a long period of time, but wishes to persist the session over this extended time. In this case, the client **102** provides the user a means to enter the time that the system will be unconnected (say for the duration of a domestic flight), and the client notifies the server of the expected length of the disconnection event.

[0139] On the server **104**, the session persistence server **316** functions as a proxy for connections from clients to network resources. Be those resources at the enterprise data center or public resources. The server implementation includes a similar intermediate driver architecture to that on the client, combined with application layer components to manage caching locations and recording for billing purposes. There may be cases where enterprises wish caching to occur at the enterprise. In this case, the system allows for the cache to be on the other side of a static tunnel to the enterprise.

General

[0140] The foregoing description of the embodiments of the invention has been presented only for the purpose of illustration and description and is not intended to be exhaustive or to limit the invention to the precise forms disclosed. Numerous modifications and adaptations thereof will be apparent to those skilled in the art without departing from the spirit and scope of the present invention.

That which is claimed:

1. A method comprising:

receiving a request to connect to a network, the request associated with a user;

determining a policy associated with the user;

identifying at least one available network connection;

determining at least one property of the at least one available network connection;

evaluating the property based at least in part on the policy; and

selecting the at least one available network connection based on the evaluation.

2. The method of claim 1, wherein the policy comprises a list of allowed connections.

3. The method of claim 1, wherein the policy comprises a list of disallowed connections.

4. The method of claim 1, wherein the property comprises a property selected from the group consisting of security, reliability, speed, and cost.

5. The method of claim 1, wherein the at least one property comprises a plurality of properties and further comprising applying a normalization algorithm to the plurality of properties.

6. The method of claim 1, wherein determining the policy comprises retrieving the policy from a central policy repository.

7. The method of claim 1, wherein the at least one available network connection comprises a public connection.

8. The method of claim 1, wherein the at least one available network connection comprises a private connection.

9. The method of claim 1, further comprising receiving a quality of service datum associated with the at least one available network connection.

10. The method of claim 1, wherein the at least one available network connection comprises a plurality of available network connections.

11. A method comprising:

receiving an indication that a new network connection is available for a client device associated with a user;

determining at least one first property associated with the new network connection;

determining at least one second property associated with an existing network connection;

determining a policy associated with the user;

evaluating the at least one first property based at least in part on the policy;

evaluating the at least one second property based at least in part on the policy;

disconnecting from the existing network connection; and

connecting to the new network connection.

12. The method of claim 11, further comprising signaling that the new network connection is available.

13. The method of claim 11, further comprising before disconnecting from the existing network connection, opening a data cache for buffering data during a disconnection.

14. The method of claim 13, further comprising after connecting to the new network connection:

receiving data from the data cache; and

closing the data cache.

15. The method of claim 13, further comprising receiving a measure for determining the size of the data cache.

16. The method of claim 15, wherein the measure comprises an estimated disconnection duration.

17. The method of claim 11, wherein the indication that a new network connection is available comprises a signal from a hardware device.

18. The method of claim 17, wherein the hardware device comprises an adapter selected from the group consisting of a WiFi adapter, a LAN adapter, a cellular adapter (WWAN), and a dial-up (WAN) adapter.

19. A computer-readable medium on which is encoded program code, the program code comprising:

program code for receiving a request to connect to a network, the request associated with a user;

program code for determining a policy associated with the user;

program code for identifying at least one available network connection;

program code for determining at least one property of the at least one available network connection;

program code for evaluating the property based at least in part on the policy; and

program code for selecting the at least one available network connection based on the evaluation.

20. The computer-readable medium of claim 19, further comprising program code for receiving a quality of service datum associated with the at least one available network connection.

21. A computer-readable medium on which is encoded program code, the program code comprising:

program code for receiving an indication that a new network connection is available;

determining at least one first property associated with the new network connection;

determining at least one second property associated with an existing network connection;

determining a policy associated with a user;

evaluating the at least one first property based at least in part on the policy;

evaluating the at least one second property based at least in part on the policy;

disconnecting from the existing network connection; and

connecting to the new network connection.

22. The computer-readable medium of claim 21, further comprising signaling that the new network connection is available.

23. The computer-readable medium of claim 21, further comprising before disconnecting from the existing network connection, opening a data cache for buffering data during a disconnection.

24. The computer-readable medium of claim 23, further comprising after connecting to the new network connection:

receiving data from the data cache; and

closing the data cache.

25. The computer-readable medium of claim 23, further comprising receiving a measure for determining the size of the data cache.

26. A system comprising:

a policy reader operable to determine a policy associated with a user;

a connection manager operable to:

receive a request to connect to a network, the request associated with a user;

identify at least one available network connection;

determine at least one property of the at least one available network connection;

evaluate the property based at least in part on the policy; and

select the at least one available network connection based on the evaluation.

27. The system of claim 26, further comprising a quality of service collector operable to receive a quality of service datum associated with the at least one available network connection.

28. A system comprising:

a policy reader operable to determine a policy associated with a user;

a connection manager operable to:

receive an indication that a new network connection is available;

determine at least one first property associated with the new network connection;

determine at least one second property associated with an existing network connection;

evaluate the at least one first property based at least in part on the policy;

evaluate the at least one second property based at least in part on the policy;

disconnect from the existing network connection; and

connect to the new network connection.

* * * * *