(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 July 2006 (27.07.2006)

PCT

(10) International Publication Number
WO 2006/077283 A1

(51) International Patent Classification:
H04L 12/56 (2006.01) *H04Q 7/22* (2006.01)
H04L 29/08 (2006.01)

(21) International Application Number:
PCT/FI2006/050032

(22) International Filing Date: 19 January 2006 (19.01.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
20050067 20 January 2005 (20.01.2005) FI

(71) Applicant *(for all designated States except US)*: **STREAM MOBILE OY** [FI/FI]; Tammasaarenlaituri 3, FI-00180 Helsinki (FI).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **HOUGHTON, Paul** [US/FI]; Helmitauluntie 13b, FI-02770 Espoo (FI). **REHTIJÄRVI, Pekka** [FI/FI]; Itälahdenkatu 13 A 32, FI-00210 Helsinki (FI).

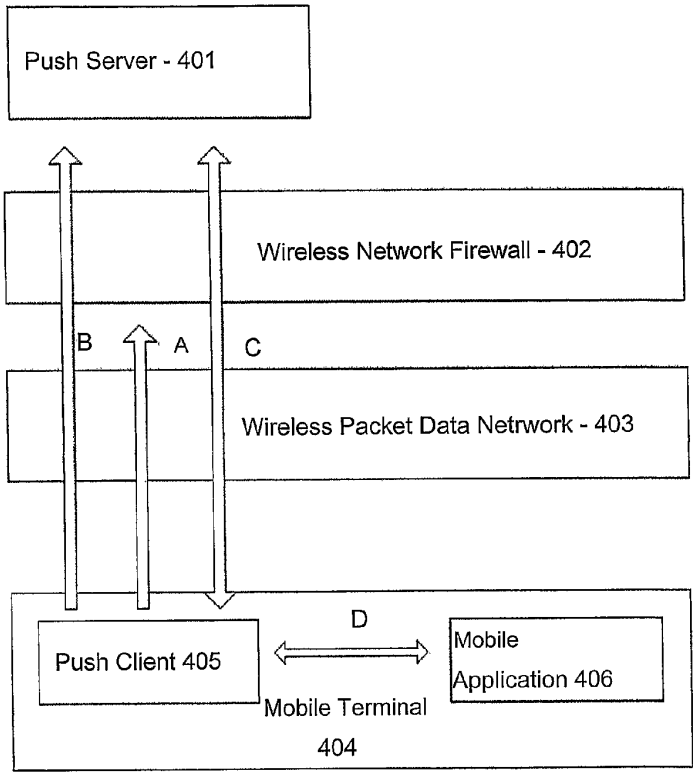
(74) Agent: **KOLSTER OY AB**; Iso Roobertinkatu 23, P.O. Box 148, FI-00121 Helsinki (FI).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: PUSH MESSAGING COMMANDED METHODS AND DEVICES



(57) Abstract: A server outside a wireless network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. The mobile terminal is provided with means, such as software, which initiates and maintains contact with the server using Internet technologies. This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time.

WO 2006/077283 A1

AMAZON-1007
IPR2026-00088, IPR2026-00106

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Title of the invention

Push Messaging Commanded Methods and Devices

Field of the invention

The invention relates to techniques for launching applications on a mobile terminal.

Description of the Related Art

The Internet is a collection of networks which exchange messages using IP protocol (Internet Protocol) and associated protocols which run on top of IP such as TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). Gateways between IP networks automatically route and transmit information over primarily the IP4 and IP6 protocols between two or more end points on any of the connected fixed and mobile networks. Each of these networks may also include or run on top of other network transport protocols such as ATM (Asynchronous Transfer Mode) or a wireless protocol which in turn carry the IP packets over that section of the Internet.

Currently, the most common method of starting applications on a mobile terminal it to manually launch them by clicking on an icon or selecting a menu item. If the application, dynamically loaded program module or user interface feature is already launched, these applications must also be manually commanded by the user to change state to display newly available information provided by a telecommunications network. Some applications are also launched automatically when the terminal is started. This automatic start includes both applications that were installed on the phone before delivery to the customer including the basic terminal operating system software and programs installed by service providers and users and which by various means ask the terminal software to start them automatically after the terminal is powered up.

Remote launch to a known state as specified in a network command of programs on mobile terminals is supported in various telecommunications standards. Voice Over IP (VoIP) and other multimedia sessions can be started remotely using Session Initiation Protocol (SIP). Wireless Application Protocol Push (WAP Push) is another standard for launching programs on mobile terminals based on remote events. Java Mobile Information Device Profile (MIDP) and various native mobile terminal operating systems also support remote application launch based on receipt of a text message (Short Message Service,

SMS) or incoming Internet (IP) packet of information to a designated communications port number. However it is important to note from an economics and practical service deployment standpoint that each of these technologies either requires payment to the mobile network operator for sending an SMS or privileged access to the mobile phone from inside the operator's firewall in order for IP data to actually reach the phone.

Mobile phones and similar mobile communication devices where originally not permanently connected to the Internet. Instead, for the transmission and display of visual information they have relied on different telecommunications standards and proprietary communication systems such as SMS (Short Message Service, also known as text messaging) and MMS (Multimedia Message Service), which were built into mobile terminals.

Various techniques have been proposed for remotely commanding applications on a mobile terminal using push technologies:

US20020183051 relates to remote application management of a wireless device.

US20030208602 relates to pushing data in an internet protocol network environment.

US20010020255 relates to remote control and interaction with a mobile run time environment component.

US6687743 relates to client server communications for a mobile computing device.

US6779019 relates pushing information from a host to a mobile data communications device.

US6701378 relates pushing information from a host to a mobile data communications device.

US6463464 relates pushing information from a host to a mobile data communications device.

US6463463 relates pushing information from a host to a mobile data communications device.

US6389457 relates pushing information from a host to a mobile data communication device.

US2004122907 relates to secure interaction between a mobile client device and an enterprise application in a communication system.

WO0195695 discloses a system for wireless push and pull based services.

US2002187775 discloses a WAP service personalization, management and billing object oriented platform.

US2003106022 relates to outputting dynamic local content on mo-bile terminals.

US2004185834 discloses a method for enabling IP push capability to wireless devices on a wireless network.

US2002174184 shows a mailbox access mechanism using SMS push and WAP technologies on mobile terminals.

WO02077842 shows a push method for delivering data to mobile device.

Wireless Application Protocol Forum, Ltd., "Wireless Application Environment Specification, Version 2.0", Version 7-Feb-2002, 2002. www.wapforum.org.

Sun Microsystems, Ltd., "The MIDP 2.0 Push Registry", Network and Timer MIDlet activation. developers.sun.com/techtopics/mobility/midp/articles.

Nokia, Ltd., "SIP Session Initiation Protocol", Forum Nokia 2005. www.forum.nokia.com/main/0,6566,1_44,00.html.

Nokia, Ltd., "SIP Frequently Asked Questions", Forum Nokia 2004. SIP_FAQ_v1_0_en.

RFC 3261, "SIP: Session Initiation Protocol", 2002. www.faqs.org/rfcs/rfc3261.html.

Push technology is characterized by the server proactively sending documents or information to a client or user. E-mail (Electronic mail) is actually a special kind of push technology that has been prevalent for a long time. The first and most common general category of mobile content push technologies is SMS (Short Message Service) based technologies. SMS and MMS (Multimedia Messaging Service) are the most common forms of "push" messaging in the mobile industry. These store and forward messaging systems run by mobile telecommunications operators push all or part of the data to the mobile terminal over closed telecommunications signaling networks such as SS7 (Signaling System Seven). These signaling networks are closed; they are not part of the Internet. Each operator controls and encrypts this traffic, charging high prices for their use and limiting their use and usefulness.

FIG 1 shows a generic implementation known in the art for pushing launch and operational commands to applications on mobile terminals. Email, MMS, WAP (Wireless Application Protocol) pages, and custom application

data are delivered through existing telecommunications infrastructure to a mobile terminal using SMS as the push messenger medium. First, push server software 101 delivers the notification to an SMS Gateway 102 , as illustrated by the arrow A. The SMS Gateway puts the actual SMS push message in either readable text or compressed binary format into the telecommunications signaling network 103, usually SS7, which delivers it in a store and forward fashion often involving several intermediate steps to the mobile terminal 104, as illustrated by the arrow B. The mobile terminal standard software 105 decodes header information in the SMS and based on the fields in this binary SMS header not visible to end users, delivers this as a normal SMS, an MMS, WAP pages or a custom application command action (C) to the indicated mobile application 106, launching the application if necessary . These custom application launches by SMS require the application to first register itself with the mobile terminal in order to receive SMS push message command events.

Besides a high price per message, SMS messages cost different amounts depending on where the sender and receiver are located. In some countries the sender pays for the message, in other countries the receiver pays. Computers called Short Message Service Center (SMS-C) nodes and owned and managed by each mobile operator deliver SMS messages. In many cases, if the sender and receiver are on different continents the operators do not even offer message delivery. Two operators must negotiate, sign and implement a legal contract known as an interconnect agreement before messages flow between SMS-Cs and thus between mobile terminals on different operator networks. Since not all operators have such contracts with all other operators, this limited connectivity also means that commanding mobile applications based on any SMS technologies can not be delivered on a global basis. For example an SMS-based remote application command server run from a European telecommunications operator will face the practical limits of incomplete operator interconnect agreements when pushing application commands to their North American and Asian customers. Also, because of the billing models in different countries are also variable, high costs and delays when roaming, it is difficult to develop reliable global application command using SMS-technologies. For example, such SMS delivery arrangements to all operators within one country or in multiple countries usually rely on expensive arrangements with middlemen or brokers that route computer-originated mes-

sages to phones in many locations. Even the best of these such middlemen still do not offer global coverage to all countries.

A newer telephone company standards-based method to delivery content and command applications which have not been explicitly requested by the user of a mobile terminal is WAP Push. WAP Push technology relies on an SMS to start the message delivery process. Both SMS and WAP Push using SMS can be used to command applications on the phone. Java applications on mobile terminals can also register themselves to be launched upon receipt of an SMS and then interpret the contents of the SMS as a type of launch command or launch content. But in all these cases operators continue to control the push medium and charge money for each push event. This operator control of content on mobile terminals is known in the industry as the "walled garden" approach.

SMS messaging standards include the ability to send binary messages to an application installed on the mobile terminal. Parameters in the message header and applications in the mobile terminal are matched by the mobile terminal at delivery time and the selected application receives the pushed SMS contents, starting the application if it is not already running. This is a standard software feature installed on the phone at time of manufacture as part of the mobile terminal's operating system.

The cost of using such an remote command capability is quite high-the cost of an SMS each time the program is commanded, thereby limiting the broad utility of this feature for all but a telecommunications operator supplying the service only to their own customers. These standard SMS-based telecommunications technologies for push application remote command are all thus limited by economics in both the frequency with which they can be used and the breadth of market which they can reach. Also due to the low speed of SMS delivery as a store-and-forward messaging system, it can take a long time for the application to launch to a known state based on a remote command thus making this unsuitable for use in any real time or time critical situations.

A further limitation of existing SMS remote command technologies is the inability to prioritize, select and deliver messages in different ways to a user based on their current situation, location and individual preferences. More important application command events may be buried behind several less important events causing critical items to be missed or confused. A user can not select with these SMS-technologies not to be notified, or to be notified in a

more polite manner, or to go ahead and start to a known state or change the visible operating state of the application without politely asking in some circumstances. And a user can not specify that they would like to only receive or receive a request for permission to execute a certain category of application command events during a given period of time.

A second general category of mobile push technologies is IP-based technologies. These include IP connections initiated from the server and IP connections initiated from the client. Server-initiated push connections are the most commonly described. These solutions require IP communication with software installed on the terminal and a server which has been notified of the IP address of this terminal. The server then, when a push event is desired, initiates a connection to the terminal and delivers information to it which software in the mobile terminal can then act upon, most commonly to notify the client of an incoming email message. For reasons described below, these push technologies are not generally used on mobile terminals but are more commonly used between one server and another as in the SMTP mail transport protocol when delivering email. Similar methods are also used for launching, commanding and communicating with applications on servers and in some cases on mobile terminals.

There are, however, difficulties with commanding applications on mobile terminals based on receipts of an IP packet when there is not an already established IP connection between a mobile application program module or feature and the command push server. In practice, mobile terminals are on sections of the Internet which are most commonly behind a firewall or gateway server device creating a controlled mobile network. This difference from the above-mentioned server-to-server push technologies is non-obvious when reading standards and results in many server-to-client and server-to-mobile-terminal push technologies which in practice do not work because of firewalls protecting the mobile terminal. Thus firewall and mobile gateway devices at the connection between the security-controlled mobile network and the more public Internet frequently restrict computing devices such as servers on the public Internet from directly contacting mobile terminals. Additionally, IP connection and security techniques, such as network address translation, firewalls and gateways, may hide the actual address of the mobile terminal further complicating direct connection from an Internet computing device. Further, the mobile

terminal may be unavailable at any given time making such messages undeliverable.

The public telecommunications operator, company, organization, or private individual setting up a mobile Internet space with one or more users uses the above-mentioned firewalls, security devices and gateways to protect the mobile terminals from hostile or unwanted contact common on the public Internet. They may also do so to reduce background traffic, protect their control over this space and thereby limit outside or undesired service providers through technical connection policies set in the firewall or gateway device. The result is, if a fixed or mobile computing device from outside the controlled mobile network tries to contact a mobile terminal inside the controlled mobile network, the connection is not permitted unless the organization or individual controlling the firewall or gateway server device makes an exception to their security policy and explicitly allows the connection. Instead, in standards-based mobile Internet telecommunications, an intermediary computer with privileged access from inside the firewall to the mobile terminals is used. These server computers such as a telecommunications operator SIP Server or WAP Gateway is used to control and deliver messages and store messages when the mobile terminal is unavailable and deliver them on the senders behalf, possibly later then the terminal becomes available, or to forward them to a different gateway device, for example if the user is roaming to a different controlled mobile network.

The most common solutions to accessing mobile terminals subject to this mobile network restriction are to either make an agreement with the public or private network operator removing the security restriction on a specific outside server contacting mobile terminals, or to move the server inside the controlled network. In the common case of a public mobile network operator, the state of the art and most common solution is to sell the server and associated software to the mobile operator so that they can provide the service to their customers. Both of these solutions require time, effort and often large financial commitments. And they must be done for each and every restricted mobile network and operator in a given market to achieve universal access to a service requiring server-initiated IP push. Achieving such universal penetrations on all operators in all markets is not practical. However for those operators with whom such a relationship has been established, mobile application launch and command with standard IP technologies is thus possible.

Client-initiated mobile IP connections are less common in mobile terminals. These are also not part of state of the art mobile application command telecommunications standards. Client-initiated IP connections involve installing more complex software on each mobile terminal and having these devices initiate contact to a server on the public Internet. These connections generally operate in a polling or "content pull" manner, connecting intermittently to the server to ask if any new information is available. Mobile email is a common example of this approach for content download. While the result appears to be push to the end user when they are notified several minutes later of the incoming email, they were not notified immediately by the server when the message is available as in true push technologies. The network settings of the mobile terminal may also need to be adjusted to permit this since Internet access through default gateway devices may not permit this type of connection.

FIG 2 shows a generic implementation known in the art for pushing launch and operational commands to applications on mobile terminals using Internet technologies. Symbian, Linux, Microsoft and other mobile IP terminal operating system software all include application launch capabilities based on this socket-based triggering mechanism. Each mobile application 205 has a fixed IP port number such as TCP/IP or UDP/IP port numbers. When a remote program such as a push server 202 sitting inside the operator's firewall 201 and with permission to contact ports on the mobile terminal 203 sends an IP packet to the terminal (A), the IP terminal software 204 routes the packet with it's commands or application-specific data to the mobile application 205 (B), launching the application if it is not already running. Java applications sitting on top of the lower levels of IP terminal software can be similarly launched based on Java MIDP 2.0 and later "push registry" settings.

FIG 3 shows the same steps A and B shown in FIG 2, however the push server 301 may be located outside the firewall 302 because the wireless network firewall has been configured by the mobile network operator to allow communications from that particular push server. This allows IP traffic to flow from the push server 301 to the IP terminal 303 and be internally routed 304 to the mobile application 305.

The important point to note in the standard methods shown in both FIG 2 and FIG 3 is that the permission of the wireless network operator is required to either locate a push server within their firewall or open a firewall hole to a given push server. This permission is in practice expensive for an arbitrary

push server to achieve with a given operator. It is virtually impossible to achieve with all mobile network operators in all markets. A viable market alternative to the state of the art methods shown in FIGURES 1, 2 and 3 for the mobile terminal and push server to initiate and maintain contact for push application launch and configuration commands without explicit arrangements with each mobile network operator is thus needed and described in the invention below.

Once such an IP connection is established between the server and one or more mobile terminals, the connection can be used to transfer messages to the mobile terminal. The most interesting such messages for the present invention are event trigger messages to launch or command a mobile application. Other interesting uses include the sending of data files, new applications and applications updates to replace or improve installed applications. These messages, while simply inexpensive IP traffic can if sufficient in number result in significant costs or the user perception of their incurring significant data traffic costs. For this reason it is also desirable to route such information including application updates, bug fixes, additional features, additional application data including multimedia, vector graphics, text and executable code through the lowest cost available network connection. Further it is desirable but not generally possible to automatically do so only during those times that a lower cost network connection is available as through the presence of a broadband wireless access device.

Numerous mobile applications such as games and productivity software are now included with new mobile terminals. They can be used to perform an endless variety of tasks, some of which involve interaction with elements outside the mobile terminal, for example on desktop computers, servers and other mobile terminals which are also connected to the Internet. The process of a user manually performing actions such as starting the mobile application to become aware of the state of remote computers and terminals is referred to as pull because a user action triggers or "pulls" the information for display to the phone.

Multimedia audio and video presentation applications, Voice over IP (VoIP) communications, multimedia interaction applications, games including games which involve other player and automated server actions such as game events and challenges from other mobile terminals, mobile business process applications, real time back office connectivity applications and corporate mes-

saging programs are examples of areas where mobile terminal applications benefit from the ability of server placed arbitrarily on the public Internet having the ability to initiate the display of or remotely configure the view presented by an already running program on a mobile terminal at any time. Due to technical limitations of the terminal and user preferences, it is generally not possible for a mobile terminal to support multiple such large or resource-intensive applications running continuously. Therefore there is a need for a more general solution to automatically install, update, provide data to, launch and configure these applications from the background on mobile terminals on any mobile network.

Commanding an application on the mobile terminal, launching an application with parameters specified in the launch message so as to view specific information after the program starts, changing the current view of an application which is already running, pushing or pulling new information into an application so as to change what is presented to the user are all referred to herein as changing the state of the application. This application state change process generally requires personal initiative and several manual and cumbersome steps and decisions on the part of the user: opening the application or changing the view of the application, selecting and then examining the content to see if any of the changes are of interest. Because of the limited size, network bandwidth and user interface of mobile terminals, performing these steps is significantly slower on a mobile terminal than performing the same actions on a larger computer. The process of checking if an application should be launched now because new and interesting Internet information might be available is therefore less interesting to mobile terminal users than on a larger terminal because of the limited capabilities of the mobile terminal to rapidly display many alternative content formats. This frustrating or slow user interface with for example small buttons, touch screen or relatively slow interface response on the mobile terminal is one of the reasons there is commercial need for eliminating the user steps and push commanding the mobile application remotely from a server only when new and interesting information is available.

Given the shortcomings in the prior art, there is a need for a method whereby a computer on the Internet can rapidly and at reasonable cost be permitted to at any time initiate control of one or more applications on an individual's mobile terminal to deliver through that mobile application real time in-

formation, interactivity and rich media content, regardless of which mobile operator is being used and where the mobile user happens to be at that moment.

Brief description of the invention

An object of the present invention is a new technique for providing a push-type command of applications on a mobile terminal by a remote server computer. The object is achieved by the invention is characterized by what is stated in the attached independent claims. The preferred embodiments of the invention are disclosed in the dependent claims.

In a preferred embodiment of the present invention is a system and method for a server placed freely on a connected network outside the wireless network firewall or other corresponding wireless network security entity, to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. The mobile terminal is provided with means, such as software, which initiates and maintains contact with the server using Internet technologies, thereby avoiding cost, latency, contractual and firewall penetration issues of the server contacting the mobile terminals by telecommunications messaging or Internet technologies. This mobile-initiated permanent IP connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time. If the server would attempt to use IP to contact the mobile device directly, the attempt would fail due to the firewall(s).

In an embodiment of the invention, the mobile terminal maintains contact with the server by periodically testing the connection with a suitable keepalive message, a "ping", which resets timers in intervening operator stateful firewalls and/or NAT (network address translation) firewalls. The mobile terminal may also automatically re-establish the connection when the connection gets broken and adjust its connection settings based on past experience with failure to connect and such broken connections in a given mobile network.

In an embodiment of the invention, the push messages may include, for example, a message stimulating, triggering or commanding the launch, display or shift to foreground of a user interface feature of the terminal such as an application which is part of the basic terminal software or and an additional installed application beyond the base software supplied with the mobile termi-

nal. In response to receiving such a message, a mobile terminal may spontaneously open the application, dynamically loaded module or user interface feature of the terminal thereof to a designed application view or location, present new information of the availability of new information, a new feature or operating state. If such application, module or user interface features is already operating in the mobile terminal, the action of such a message would instead stimulate the application or user interface feature to present itself to the foreground of the user interface and change state as specified in the application-specific message received from the server. In further embodiments of the invention, push messages may include, for example, messages triggering the mobile terminal to play a sound notifying the recipient of the application command or state change, or display notification providing event information from which the user may select or decline to display the application.

Note that in some cases the packaging of mobile applications involves combining multiple applications delivered in a single bundle or application suite preinstalled or later installed to the mobile terminal. Commanding the display of an additional, more resource-intensive user interface feature such a rich media, or highly interactivity modes within the same bundle of applications or application features as for example by explicitly or implicitly loading or calling an additional library such as a user-interface related code or other embodiment of an application on a mobile terminal within the same application based on push event messages recently received from an IP connection is simply an alternative packaging of this same invention. For brevity herein, this type of commanding of a new user interface modes or view to appear within the same application suite is considered and discussed as the commanding of a new application since the packaging of rich user interface features and program modules within a bundle or separately is a relatively minor detail in the embodiment of the invention.

The cost for a user's mobile terminal to maintain a connection to the push server and receive such messages over mobile Internet connections is minimal, thereby solving the problem of some mobile information services not being economic to offer. The advantages are: all the convenience of standard push services such as SMS-based application launch and command together with the mobile network-independent and economic benefits of placing the server outside of any single operator's network and of the service provider not having to pay an operator or operator middleman per command event.

IP push messages utilizing the principles of the invention are delivered rapidly, bypassing the public and often crowded SMS, MMS, WAP Push and email store-and-forward mechanism within an operator and between operators. This guarantee of very rapid event push provides usability benefits to users which may be then designed into the mobile applications. For example the ability to expect prompt delivery allows modification to interactive mobile application system design. The mobile process surrounding the application can be changed such that a user on one mobile terminal may take actions which result in the initiation of a push event to a user on a second terminal and then wait for a short time for the second user to respond after the launching or commanding to return to be the focus of user interface attention their copy of the same, similar or complimentary mobile application function involving Internet communication directly or mediated by a server between the two mobile applications.

The advantages of performing push service application command in accordance with the present invention include the ability to run such a service from outside the mobile operator's firewall and gateway devices, skip the overhead charges like per-message SMS fees normally associated with "mobile push", and make it easy to create mobile applications as web pages, therefore usually without programming. Thus, push application command services according to the invention can be modified from existing or newly designed mobile applications, modules and application features simply by creating an application command, module or feature entry point compatible with the embodiment of the push client. This embodiment can be, but is not limited to, a separate application, operating system background process or application module bundled in a suite with the mobile application and kept running such that it may receive IP push events and command applications, modules or significant application features.

These messages may be routed automatically to a nearby computing device with faster or less expensive network connectivity such as a broadband access point, server or workstation in the proximity of the mobile terminal. Both mobile terminal and access point must be equipped with a wireless communication network such as RFID, Bluetooth, ultrawideband, Zigbee, Wi-Fi, Wi-Max or other personal area, local area, metropolitan area or large area network. This ability embodied in the invention alleviates speed and economic problems that affects the present-day mobile messaging solutions which are either limited in bandwidth or cost based on volume of use. When in proximity to a more

capable network, if a connection can be established to this alternative broadband network access point, it is used preferentially and intelligence normally located in this broadband access point would route messages received on behalf of the wireless terminal to and from the server. Further, the availability of a broadband access point would be notified to mobile terminal application software and server alike allowing either to initiate bandwidth intensive operations such as application update and application data update.

IP push command messages may be triggered either by events which the server is aware of, or events which a similar mobile application on another mobile terminal is aware of and which that push client makes the server aware of, the server in turn initiating a push command event. Such client-side events of which the server is informed for the possibility of delivering new web services include the making, receiving, and termination of telephone calls, the transition from presence-to-absence or absence-to-presence of other devices in a geographical area, metropolitan area, local area or personal area wireless communications network, mobile terminal creation of photographs video audio or other media, mobile terminal presence of files matching a given filtering criteria, changes in internal state of electronic equipment, taking of measurements, processing of such measurements to see if they match certain criteria, video game actions or events, application calculation events and messaging actions. In some cases a separate server such as a telecommunication service server, gaming environment server, and original equipment manufacturer device or business information system may notify the server directly of events or messages to be sent using an external application programming interface (API). Such a device connected by external API may also receive information including the results and responses of messages which the user has replied to by terminal software or web page interaction.

The invention provides content and application producers with a cost effective, network-independent and device-independent service for the rapid creation and distribution of interactive information alerts and Internet launch-at-will and present-when-needed features for their users.

The invention provides techniques for launching and changing the operating state such as information currently presented to the user of applications on a mobile terminal at will by a server computer and without the need for any additional action by the recipient, and more specifically to a system, method and program for sending, receiving and acting upon messages sent to a mo-

bile client device over a packet switched data network for the purpose of initiating application launch, application command, module or feature display and user direct or indirect user interface state change based on server-side events and mobile terminal events including user interface events, phone calls and network proximity events, and local resource changes.

Brief description of the drawings

In the following description, reference is made to the accompanying drawings which form a part hereof, and which illustrate several embodiments of the present invention. In the drawings:

Figure 1 shows an implementation known in the art for pushing information including commands to an application through existing telecommunications SMS infrastructure to a mobile terminal and if necessary launching the application;

Figure 2 shows an implementation known in the art for pushing information including commands from inside the mobile network firewall to an application to a mobile terminal and if necessary launching the application;

Figure 3 shows an implementation known in the art for pushing information including commands from outside the mobile network firewall through a firewall hole opened by the mobile network operator to an application to a mobile terminal and if necessary launching the application;

Figure 4 is a block diagram illustrating an exemplary implementation of the system for pushing mobile application launch commands to a mobile terminal based on server side events in accordance with one embodiment of the present invention;

Figure 5 is a block diagram illustrating an exemplary implementation of the system for a wire terminal with an alternative broadband connection to use the alternative broadband access point for receiving push messages including launch events, application updates and data updates;

Figure 6 is a flow chart illustrating an exemplary process by which the mobile terminal maintains an open path for network communication with the server (6A), keeps the connection to the server open and tests the integrity of the connection while automatically reestablishing the connection as necessary (6B), responds to messages from the server (6C), and notifies the server of locally monitored events on the mobile terminal (6D) in accordance with one embodiment of the present invention;

Figure 7 illustrates an example of a "COMMAND PUSH" procedure;

Figure 8 illustrates an example of a "COMMAND PUSH WITH BIDIRECTIONAL DATA" procedure;

Figure 9 illustrates an example of a "BROADBAND COMMAND PUSH WITH BIDIRECTIONAL DATA" procedure;

Figures 10A, 10B and 10C are flow diagrams showing examples of processes by which the server maintains a connection with the client and communicates commands and data to it.

Detailed Description of the Invention

In the above description, the invention is described using various embodiments of the present invention as examples but the invention is not intended to be restricted to these embodiments. It is understood that other embodiments may be utilized and structural and operational changes may be made without departing from the scope of the present invention.

Figure 4 illustrates an exemplary implementation of the system for pushing application commands to a mobile terminal based on server side and client side events in accordance with one embodiment of the present invention. More specifically, Figure 4 illustrates a Push Client basic functionality 405 implemented in a mobile terminal 404 as part of a message passing system in cooperation with a Push Server 401. The push client 405 is preferably implemented in the form of a software run in a processor of the mobile terminal 404. The mobile terminal 404 is preferably a commercially available mobile terminal, such as a portable phone or appliance capable of running software or personal digital assistant, which provides a platform for downloading application software over air or from a personal computer, installing the software in the terminal, and running the software in the terminal. For example, the Symbian operating system is an open operating system, application framework, and application suite optimised for the needs of wireless information devices such as intelligent phones, smartphones and communicators, and open operating system and drives standards for the interoperation of data-enabled mobile phones with mobile networks, content applications, and services. The Symbian operating system also includes connectivity software for synchronisation with data on personal computers and servers. The application may be in an appropriate programming language, such as C++ or Java, which is an object-oriented pro-

gramming language and virtual machine platform developed by Sun Microsystems.

The push client software 405 preferably launches automatically when the mobile terminal 404 is turned on. The push client software 405 runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 401 as in FIG 3. If successful, operations can continue as described in FIG 3 or as below. Arrow B illustrates a login attempt.

The wireless network 403 may be protected by means of a firewall or other appropriate security entity 402, which connects the wireless network to the public networks and filters and selectively discards the data packets entering and exiting the wireless network according to predefined rules. Thus, a firewall is a gateway device which operates at the same time as a connector and a separator between networks in a sense that it keeps track of the traffic that passes through it from one network to another and restricts connections and packets that are defined as unwanted by the administrator of the system. Additionally, IP connection techniques, such as Internet Protocol 4 (IP4) network address translation and Internet Protocol 6 (IP6) security, encryption and/or authentication, may hide the actual address of the mobile terminal or make it otherwise difficult for a server to connect to directly to the mobile terminal. The public telecommunications operator, company, organization, or private individual setting up this mobile Internet space with one or more users having greater ability to contact one another and included servers does this to protect the mobile terminals from hostile or unwanted contact common on the public Internet. They may also do so to reduce background traffic, protect their control over this space and limit outside or undesired service providers through technical connection policies set in the firewall, gateway or other security device. Physically a firewall may be a machine with appropriate software to do the task assigned to it. 402 can be a firewall, router, a personal computer, a gateway server or whatever similar computing device, infrastructure or secure protocol such as IP6 can be used for such purposes.

Consequently to the existence and configuration of 402, if a fixed or mobile computing device from outside the controlled mobile network tries to contact a mobile terminal 404 inside the controlled mobile network, the incoming connection is not generally permitted unless the organization or individual controlling the firewall, gateway server or security device 24 makes an adjustment to their security policy and explicitly allows the connection.

From point of view of the wireless packet data network 23, the login attempt B and associated connection establishment C is one or more normal IP connections established at the request of the mobile terminal. If optional attempt A to open a path for the push server 401 to initiate the IP connection now or in the future, use from that point forward may proceed as in FIG 3. However this firewall and/or security device or protocol opening A is unsuccessful, the login attempt B is in either case normally allowed to pass the firewall or security infrastructure 402 since the connection is from the inside the firewall to outside.

After attempting login, if the login is not successful the client 405 may automatically or under supervision of the user of the mobile terminal 404 try several different techniques for contacting the server 401. Since there are numerous variations between mobile terminals and in network connectivity, more than one connection attempt may be needed to contact the push server 401. Differences in wireless network terminal characteristics, individual push server availability and accessibility, firewall and gateway permissions, connection timeouts, connection level stateful (such as TCP) and stateless (such as UDP) transport protocol availability (such as IP4, IP6, SSL, HTTP, HTTPS) and alternate mobile terminal network settings mean that a sequence of connection establishment attempts and connection maintaining steps may required to be made using variations in relevant network parameters.

A second reason for automatically adjusting the network parameters is not simply establishing the connection but establishing the connection in the most cost effective or highest performance configuration available for the individual mobile terminal user account service level. Connection types differ in speed, customer expensive, server resource usage and hence scalability, security and ability to penetrate gateway security devices. Some customers may be offered more secure or higher performance service while others receive the least cost connectivity from either the mobile terminal's or the push server's standpoint.

As an example, such network parameters may include but are not limited to one or more of the following:

- Use of a connection-oriented protocol suitable to the wireless network such as TCP/IP protocol, which is generally preferred in the common situation where an intermediate firewall, Network Address Translation (NAT) or gateway device 402.

- Use of a connectionless protocol suitable to the wireless network 403, such as UDP/IP, which is generally preferred when no network device restricting terminal contact 402 is present as this reduces resource usage on the server when large number of mobile terminals is served.

- Use of an alternate connection-oriented protocol such as HTTP suitable for mobile push operation through more strict firewalls and proxy Internet connections.

- Use of a secure protocol such as HTTPS, IP-Sec, secure IP6 or a proprietary security protocol to identify the communicating parties, prevent message interception by 3rd parties, and prevent message modification by 3rd parties.

- Use of a periodic message from the client to the server or server to the client to the effect that client socket, server socket, intermediate Network Address Translation (NAT) devices, intermediate firewalls, and intermediate gateway devices do not time expire the connection.

- Use of configuration messages to the firewall (FIG 4, A) using a firewall or other security configuration protocol, such as Universal Plug and Play, to open a path for server-initiated connection-oriented or connectionless messaging to the client.

- Use of periodic messages initiated by the client 405 or by the server 401 to the effect that the receiving application is informed that their messages are continuing to be received, as indicated by response data from the receiving party to the sending party.

- Use of longer or shorter timing between such periodic messages to match the combined needs of all such network devices and protocols involved in transmitting communication, as adjusted based on the success or failure of earlier messages to successfully traverse the open network connection.

- Use of an alternate push server network address such as an address accessible with fewer security and performance restrictions which may be more available, more accessible, lower cost, lower latency, less busy, higher

bandwidth or otherwise preferred at the present time and from the present network location of mobile terminal 404.

- Use of one or more alternative mobile terminal network connection settings, or variation of the parameters within the settings.

- Use of an alternate connection route through a second mobile or fixed terminal accessible to the mobile terminal over an alternate wireless network or network connection and capable of communication with a push server.

In an embodiment of the invention, the mobile terminal 404 keeps a prioritized data structure listing the preferred communication protocols, server addresses, periodic message timings, network access settings, and alternate communication routes which it should use. Automated and manually assisted use of this structure should then result in a connection to one or more push servers, however if one of these connections which the mobile terminal is programmed to expect is not available, the mobile terminal can choose the sleep that function of the client software and attempt re-connection at a later time, repeating in like fashion with the same or different network connection parameters as described above until a connection is established or other user intervention.

At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client. This is illustrated by means of the arrow C in Figure 4. These messages through a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS arrive quite rapidly, more rapidly than if the connection where established only when needed. The associated protocol overhead such as 3-way handshake during connection establishment thus occurs before the actual need for message transmission rather than needing to occur at time or transmission and thus delaying the transmission.

A connectionless protocol, such as UDP/IP, may also used in cases where the connection-oriented protocol is not possible or not desirable because of the push server resources used by every static instance of a connection-oriented protocol. Connectionless communication is also used when a message needs to get through to the server 401 quickly, such as when occurs when the client 405 notifies the server 401 of an terminal event, user interface event or application event but the connection-oriented protocol connection is not connected or is already busy transferring information. Such a selective use

of this high priority, low latency communication channel from terminal 404 to server 401 gives the server the opportunity to in return push to the client 405 a service triggered by the event begin communicated by the client 405 as will be described in more detail below. The sending of such a return message is frequently time critical and the fastest available combination of techniques will be used.

In the preferred embodiment of the invention, the most basic message pushed from the server 401 to the client 405 is a data packet containing no information. Additional binary or text information specifying which mobile application 406 from a plurality of such applications may optionally be included. Additional binary or text information to be passed to the default or specified mobile application may optionally be included. Additional binary or text information specifying how upon receiving such a message the client 405 will notify and optionally ask the user if the action should be performed may optionally be included. Upon receiving such a message, appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401 are executed by the push client 405. The most common case is playing a sound, asking the user if the mobile application should be launched, and if affirmative passing the command details (arrow D) to the specified mobile application, module or additional feature within the same application suite, launching the application, module or feature if needed. Such action may include optionally proceed without audio, with network downloaded audio, or without first requesting confirmation from the user as specified by user preferences and command details.

The push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system. With a push message, the server 401 may direct the mobile application 406 to make use of additional network resources either on the push server or other server. The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly for redirection through push server API to other servers, mobile terminals, push clients and computing devices.

Figure 7 illustrates an example of an “COMMAND PUSH” procedure wherein the push server 701 is triggered by a trigger event as for example from

an application server 702 to push an application command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705. The trigger event from push client 704 to mobile application 705 may optionally include commands or data affecting the aspect of visual presentation of mobile application 705. The even trigger mechanism may be any electronic process mechanism as for example computer software calls to load or configure a class, library, dynamic link library, MIDlet, Symbian application, Linux daemon or other executable object. The trigger mechanism may be between separate parts of a larger application or application suite, or may be an IP-triggered application launch and data transfer as in the case of mobile application launched by contact to a specified IP port. The potential applications of the COMMAND PUSH include but are not limited to mobile applications for enterprise connectivity, back office integration, database access, scientific and industrial process and equipment monitoring and control, pictures transfer, mobile gaming, application data transfer, mobile electronic commerce, application alarms, reminders, multimedia interactivity, corporate calendaring, personal messaging and group messaging, etc.

The push server 701 may in combination with user preferences stored in either the push server 701 or push client 704, optionally request the push client 704 to display or otherwise as appropriate for the mobile terminal or appliance notify the user asking permission for launching mobile application 705 prior to application launch. The preferred embodiment is for the behavior, language and content of this information to be configured on the server 701 based on the service and preferences of the individual receiving the message. In such cases an optional informational message will be presented to the user and such information messages are associated for instance with the menu option transmitted to the client 704 by the server 701.

In an embodiment of the invention, the client 704 is also capable of receiving and storing variables which configure behavior of the client 704. The server 701 may thus specify the network connection preferences, notification sounds, and other behavior of the client 704. These changes facilitate the efficient and manageable operation of the entire system and can be used for automated remote maintenance of push client 704, mobile applications as 705, and configurable aspects of underlying mobile terminal 703, for example

changing the username, password, and security certificates stored in the mobile terminal.

Figure 8 illustrates an extension of the procedure in Figure 7, one known as "COMMAND PUSH WITH BIDIRECTIONAL DATA". This embodiment of the invention includes all aspects of "COMMAND PUSH" with the additional step that a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804. An IP or other API (application programming interface) connection between application server 802 and push server 801 simplifies the work and cost of creating application server 802 by taking care of network details already established for the COMMAND PUSH procedure. Similarly, an IP or API connection between push client 804 and mobile application 805 completes the connection and reduces the number, cost and resource usage of a possible plurality of such connections for a corresponding plurality of mobile applications. Further, each mobile application 805 is now free to sleep, exit or otherwise release the limited resources of the mobile terminal 803 with the assurance it may be remotely commanded back into action provided at least one push client 804 from a possible plurality of such push clients remains active, connected to the push server 801 and configured to launch the mobile applications 805.

Referring now to Figure 5, users of mobile terminals 507 are frequently but not continuously in range of alternative and more desirable wireless network access (Figure 5, arrow A). Such access is shown in the exemplary embodiment as broadband access point 509 within alternate and more desirable for reasons such as speed, cost and security broadband wireless network 507. Such an access point 509, which may also be mobile, is referred to herein as a broadband access terminal. The push client network connection functionality according to the invention may also be installed on such a broadband access point to facilitate routing and connection over alternative wireless network 507 such as a bluetooth, Wi-Fi, Wi-Max, Zigbee, radio frequency, infrared, ultrawideband, grid network or similar opportunistic or ad-hoc wireless network. Thus, both devices 508 and 509 have the minimal capabilities of client software 405 necessary for establishing and maintaining the already established contact with push server 501. Further, the two devices detect one another and so noting inform the server 501 through connections B or C and are thereafter

temporarily associated with one another in the server for routing and mobile application launch and data transfer purposes. Further, the push server 501 on becoming aware of such a situation may inform (arrow D) the application server 502 of the availability of a more desirable network connection (C plus A) to the client and on behalf of the application server 502 or independently decide to push additional commands and data to the wireless terminal using this connection. Further, the push client in the broadband access point has the additional capability to route such information to the wireless terminal. The result benefits customers using the wireless terminal in that data is managed and updated automatically for their mobile applications in the most cost efficient manner for non-time-critical capabilities such as lazy application code update and lazy application data update. While such alternative routing may also be achieved through normal IP routing and path cost without the presence in the broadband access point 509 of push-aware client software, the push server 501 and application server 502 would in this known-in-the-art implementation not be aware of when the low cost alternative route was available and thus able to at such time trigger the push of lazy application code and data installation and updates.

Figure 9 illustrates an example of a "BROADBAND COMMAND PUSH WITH BIDIRECTIONAL DATA" extending the earlier discussed "COMMAND PUSH WITH BIDIRECTION DATA" disclosed in Figure 8. This procedure wherein the push server 901 has been earlier informed by either the mobile terminal push client 906 or broadband access point 904 of the available alternative broadband network 507 and associated route (Figure 5 D, C and A, corresponding to Figure 9 A, B and C). Based on this notification and user preferences the push server 901 may choose to route future messages to either device, however the preferred remote application command push and data route is indicated by arrows B and C. Then at some later time the push server 901 is triggered by a trigger event of which the application server 902 is aware to send push message to the mobile terminal 905, and procedure steps A, B, C are traversed thereby pushing application launch commands, lazy application updates, and lazy application data updates through push clients 904 and 905 to mobile application 907 through local command and data path D (also Figure 5, arrow A).

Figures 6A, 6B, 6C and 6D are flow diagrams showing examples of processes of the preferred embodiment by which the client 405 maintains a connection with the server 401 and communicates messages to it.

Firstly, in step 602 the push client 405 initiates an attempts to log into the push server 401 over data network 403, as indicated by the arrow B in Figure 4. Success of the connection attempt is monitored in step 603, and if the network connection shows an error or does not get a response from the server in a reasonably short time the network settings are adjusted in 604 (for example in the manner describe above if the network connection fails). If the server does connect but login fails at 605, the push client may change the login settings in step 606 (for example in the manner described above if the login is rejected) and retry the login. If the login to the push server 401 is successful, a connection is already opened to the client 405 and the application server 405 changes state informing applications server 502 from a plurality of such server of the availability of the client 401 and relevant details. Thereafter the mobile applications 406 of a plurality of such applications can be contacted until the server 401 is otherwise notified or decides based on exceptions in network communications that the terminal 404 has disconnected.

In association with the login procedure the client 405 may attempt to communicate with the mobile network firewall, gateway or security device 402 it is behind using a firewall configuration interface, such as the Universal Plug and Play (UPnP) Internet gateway configuration protocol (the arrow A in Figure 4). In an alternative embodiment t may do so either before step 601. In either case the client 405 requests the firewall 402 to open an IP hole allowing the server 401 to initiate contact with the client 405 whenever there is need a need thereafter allowing classic prior art mobile application launch as illustrated and described in Figure 3. In the preferred embodiment, the configuration of the firewall 401 to allow incoming messages in this manner is attempted in step 607 immediately after login, and if the security opening operation appears successful (step 608) the server 401 is notified by the client 405. The server 401 then attempts this type of direct IP connection to the client 405 or mobile application 406 (step 609), and if that too is successful (step 610), then one of these test connections is closed and both server 401 and the client 405 mark the second connection as closable (step 611) since they now know that they may save bandwidth charges and resources by closing the connection knowing that server 401 has now gained the ability reopen the connection when

needed for a classic command push procedure (Figure 3, arrow A) or other data communication use. At any time after this point the server 401 or the client 405 may choose to close the remaining network-level connection (usually a TCP or SSL connection) while keeping the application-level "connection" in the form of knowledge of how the server 401 can contact the client 210. The advantage of such an application-level connection without the network-level connection is lower use of network resources particularly on the server side thus significantly increasing server scalability for infrequent command and data pus applications by decreasing overhead cost per active user.

Referring to steps 620 and 621 in Figure 6B, after the login and connection setup in Figure 6A, the push client 405 may send a periodic message (the arrow C in Figure 4) from the client 405 to the server 401 to the effect that client socket, server socket, intermediate Network Address Translation (NAT) devices, intermediate firewalls, and intermediate gateway devices do not time expire the connection. The period of time (measured in seconds, minutes or hours and varying from mobile network to mobile network) between such periodic messages is set to match the combined needs of all such network devices and network protocols involved in transmitting communication (step 604). The client 405 may then wait for a reply or acknowledgement from the server 401, and if the acknowledgement is received (step 622), the client 405 returns to step 620 to await the next combined connection refresh and test. If the no acknowledgement is received, the client 405 may return to step 604 in Figure 6A to possibly automatically adjust network settings or, if it deems the settings correct but the network not available, simply re-attempt login to the server 401.

Referring to Figure 6C, after the login and connection setup, the push client 405 is ready to receive the push messages over the connection from the server outside the operator's firewall (step 631). If a push message is received, the client 405 executes the corresponding function, such as launch or pass commands and data a mobile application (step 632). If required, the client 405 may also send a message to the server 401 (step 633), such as a command execution acknowledgement or result.

Referring to Figure 6D, after the login and connection setup, the push client 405 may monitor the local resources for a state change (a trigger event) in step 641. If a state change of other event is detected in step 642, the push client 405 may send a message to the server 401 (step 643).

The push server 401 can be server software run on any appropriate computing device. Figures 10A, 10B and 10C are flow diagrams showing examples of processes by which the server 401 maintains a connection with the client 405 and communicates push commands and other data to it.

In step 1001 of Figure 10A, the push server 401 receives a login from the client 405 initiates an attempt to log into the push server 401 over a packet switched network 403, as indicated by the arrow B in Figure 4. In step 1002, a connection is opened to the server 401 and the server 401 is informed of the client's its availability and the details by which the client 405 can be contacted until the server 401 is otherwise notified or detects a network fault or shutdown. In step 1003, the server 401 attempts this type of classic IP push connection to the client 405, and if that is successful (step 1004), then one of these connections is closed and both the server 401 and the client 405 mark the second connection as closable (step 1005).

Referring to steps 1010 and 1011 in Figure 10B, after the login and connection setup, the push server 401 may receive a periodic message (the arrow C in Figure 4) from the client 405 to the effect that the server updates the respective server object associated with a given client so that the connection, server socket or client socket is not time expired as being stale. The server 401 may send a reply or acknowledgement in step 1012 to notify client 405 of successful network, firewall and security traversal.

Referring to step 1021 Figure 10C, after the login and connection setup, the push server 401 monitors for trigger events and event messages. If the server 401 is triggered by a trigger event (the result "yes" in step 1023), it decides based on application-specific logic and/or user preferences whether the resulting push procedure is a BROADBAND COMMAND PUSH WITH BIDIRECTIONAL DATA procedure in step 1024. In case of BROADBAND COMMAND PUSH WITH BIDIRECTIONAL DATA, the server 401 sends the command or data push message to the push client 508 through the push client 509 (step 1025). If not using this alternative network connection selected in step 1024, the push server 401 proceeds to step 1026 to perform a more classic COMMAND PUSH or COMMAND PUSH WITH BIDIRECTIONAL DATA. In case of COMMMAND PUSH, the server 401 sends an command push message to the push client 405 and thereby forces a mobile application 406 from a plurality of such push-aware mobile applications in the mobile terminal as specified in the command (step 1026). In the case of COMMAND PUSH WITH

BIDIRECTIONAL DATA, the push server 401 sends the client 405 push commands or data and may return response through the same IP connection (step 1026).

If the server 401 is not triggered by a trigger event (the result "no" in step 1022), it checks whether an event message has been received from the client 405 (step 1023). If not, the process returns to step 1021. If an event message has been received from the client 405, the server checks whether a proximity detection is involved indicating that the mobile terminal 508 can connect to a broadband access point 509 (step 1024). If the broadband access point 509 is connectable, the server 501 performs accordingly, typically by sending an command or data push directly through the broadband access point in a manner described with reference to Figures 5 and 10C (step 1025). If the alternate broadband connection is not available or not preferred in preferences, the server delivers the message directly as above (step 1026).

The above specification, examples and data provide a complete description of the manufacture and use of the system, method, and article of manufacture, i.e., computer program product, of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.

Claims

1. A method of pushing commands and data to a wireless terminal from a push server located outside a wireless system serving the wireless terminal, comprising

the push client initiates and maintains over an extended period of time a packet-based data connection through at least one communication security or inter-network connection device of the serving wireless system to the push server using Internet technologies;

the push server sends, in response to predetermined trigger events, at any time push application installation, updates, commands or data messages to be directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection, whereby avoiding blocking of said server-originated push messages by the security device, and launching said application, application module or dynamically loaded module or user interface feature as needed,

the push client, in response to receiving a push message containing application installation, updates, application commands or data from the push server, forwards such application commands or data or stores such application installation or updates within the mobile terminal to a mobile application from among a plurality of such applications so as to initiate launch optionally to a specified state or if already running initiate or change presentation to the user according to the push message.

2. A method as claimed in claim 1, comprising

the push client sends configuration messages of a security device configuration protocol to the security device to open a path for later server-initiated connection-oriented or connectionless messaging to the push client.

3. A method as clamed in claim 1 or 2, comprising

the security device is a security protocol such as IP6 whereby the client may open the possibility for the server to contact it by exchanging security keys or otherwise enabling subsequent server-initiated contact.

4. A method as claimed in claim 1, 2 or 3, comprising

the push server attempts server-initiated messaging through the security device to the push client after having being informed of the client-initiated connection.

5. A method as claimed in claim 1, 2, 3 or 4, comprising

the push server and the push client after testing the ability of the server to open a connection to the client mark the maintained network-level connection closeable,

the push server and the push client can close said network-level connection while keeping an application level connection information indicating future possibility for the push sever to reopen said network-level connection when needed.

6. A method as claimed in any one of claims 1-5, wherein the step of maintaining further comprises that the push client maintains the permanent connection through the security device by sending periodic keep-alive messages.

7. A method as claimed in any one of claims 1-6, comprising

the push client or the push server sends a periodic reply to messages initiated by the push server or by the push client, respectively, to the effect that the receiving end is informed that the messages are continuing to be received.

the push client or the push server upon failure to receive acknowledgement of said periodic reply may choose to open a new connection.

8. A method as claimed in claim 6 or 7, comprising

use of varying timing between the periodic keep-alive or reply messages to match the combined needs of various network devices involved in transmitting and maintaining communication.

9. A method as claimed in any one of claims 1-8, comprising

if an attempt to contact the push server during initiation of the permanent connection and login is not successful, the push client automatically or under supervision of the user of the wireless terminal tries one or more other techniques, protocols and/or network parameters for contacting the push server.

10. A method as claimed in claim 9, wherein said trying of one or more other techniques and/network parameters include one or more of

using an alternate push server network address which may be more available or more accessible at the present time and from the present location,

using one or more alternative mobile terminal network connection settings, or variation of the parameters within the settings,

using an alternate connection route through a second wireless or fixed terminal accessible to the wireless terminal over a wireless network and capable of communication with the push server,

using a TCP connection,
using a UDP connection,
using an SSL connection,
using an HTTP or tunneling HTTP connection,
using an SHTTP or tunneling HTTPS connection,
using an XML-based connection over other transport protocol.

11. A method as claimed in any one of claims 1-10, comprising

the push client stores in the wireless terminal a prioritized data structure listing preferred communication protocols, server addresses, periodic message timings, network access settings, and/or alternate servers and communication routes and uses this data structure to attempt reconnection to the push server if the connection to the server fails.

12. A method as claimed in any one of claims 1-11, wherein the connected networks on which the push server and mobile terminal are located are private networks which allow greater interconnectivity than they both offer to the public Internet.

13. A method as claimed in any one of claims 1-12, wherein the message the push server sends to the push client includes an application launch or reconfiguration push message directing the immediate or subsequent actions and data of the application.

14. A method as claimed in any one of claims 1-13, wherein the push server sends to the push client include an application command or data or reconfiguration message redelivered by the push client as an API call or IP connection between processes running on the same mobile terminal.

15. A method as claimed in any one of claims 1-13, wherein the push server sends to the push client is new application code, application code updates or application data updates.

16. A method as claimed in claim 13-15, wherein

the push client is a library or process installed independently or together with but containing or enabling the functionality to supply push messages to one or more of a plurality of applications, modules or dynamically loaded module or user interface features on the mobile terminal.

17. A method as claimed in claim 16 wherein only one of the possible plurality of push clients need operate at a given time, providing push services for a plurality of push-aware applications on the mobile terminal.

18. A method as claimed in any one of claims 1-17, wherein the message the push server sends to the push client includes an optional application launch or event push message triggering the wireless terminal to play a sound notifying the recipient of the message and/or to display notification information such as text, rich text, graphics and rich media from which the user may select or decline to launch the application, dynamically loaded program module or user interface feature based on push message or messages recently received.

19. A method as claimed in 18, wherein the notification sound, text, rich text, graphics and rich media presented by the terminal is selected by and deliverable from the server and may be retrieved from the server by the terminal, thereby allowing the push server to indicate though audio and media signals including note sequences, audio samples sound modification applied to note sequences and audio samples, spoken speech, color, motion, shape and esign the message sender, nature, priority, content and other attributes of the received message, message sender and/or message sender current state.

20. A method as claimed in 19, wherein the specified notification information and media is played from a cache in the terminal and the cache is automatically updated over the air as needed with new notification information and media.

21. A method as claimed in any one of claims 1-20, wherein said trigger events include events which the push server is aware of and/or events which the push client is aware of and which a push client from among a plurality of such push clients makes the push server aware of.

22. A method as claimed in any one of claims 1-21, wherein said trigger events include one or more of: the making, receiving, and termination of telephone calls, the transition from presence-to-absence or absence-to-presence of other devices in a physical contact, RF, near field, infrared, personal area, local area, metropolitan area or geographical area wireless communications network, an alarm, notification, videogame action or event, remote application user or application server event, remote user action or inaction, electronic or computing equipment change of state or measurement result.

23. A method as claimed in any one of claims 1-22, comprising

the push client in the wireless terminal detects availability of an alternate wireless network connection to a broadband access terminal, or the broadband access terminal detects ability to initiate contact to the wireless terminal,

the push client is responsive to receipt of command and data, application installation and update push messages through said broadband access terminal.

24. A method as claimed in any one of claims 1-23, comprising

the push client in the wireless terminal detects network connection availability of a terminal device with a higher bandwidth capacity or otherwise specified as preferred, or the device with a higher bandwidth capacity detects such ability to connect to the wireless terminal

the push client transmits connection availability information of the higher bandwidth device to the push server, or the large bandwidth terminal transmits connection availability information of the mobile terminal to the push server,

the push server uses the connection availability information as a trigger event for sending lazy or low priority messages including application updates and application data updates routed through the higher bandwidth capacity device to the wireless terminal.

25. A method as claimed in any one of claims 1-24, comprising

the push client in the wireless terminal detects connection availability of a higher bandwidth or more desirable alternate network connection device,

the push client transmits connection availability information of the higher bandwidth device to the push server,

the push server uses the connection availability information as a trigger event for routing messages directed to the user of the wireless terminal through the higher capacity device.

26. A method as claimed in claim 25, comprising

the push server stops directing messages to the higher capacity device when the wireless terminal is no longer connectable to the higher capacity or more desirable alternate wireless network connection device.

27. A method as claimed in any one of claims 24-26, comprising

providing the higher network connectivity device with a push client with server connectivity similar to that in the wireless terminal.

28. A method as claimed in any one of claims 1-27, comprising

the push client in the wireless terminal detects connection availability of a device with a better alternative network connection, or a push client in the device with a better alternative network connection detects connection availability of the wireless terminal,

establishing a data-transfer capability between the push clients,

routing communication to and from the push client in the wireless terminal through the push client and the network connection of the better alternative network connection device.

29. method as claimed in any one of claims 1-28, comprising

customizing the appearance, notification sound or look and feel of the messages received to based on message source, message source state, message parameters or message recipient preferences.

30. A method of pushing messages to a wireless terminal from a push server, comprising

the push server sends, in response to predetermined trigger events, at any time application push messages to a push client in the mobile terminal,

the push client in the wireless terminal detects connection availability of a device with a preferable alternative network connection, or the device with a preferable connection detects connection availability of the wireless terminal,

the push client is responsive to receipt of the application push message to push commands to an application in said device with a larger screen so as to bring forward view an application view identified by, modified by, contained in or otherwise affected in presentation by the received application push message.

31. A method of routing messages to a wireless terminal from a push server, comprising

the push server sends, in response to predetermined trigger events, at any time application push messages to a push client in the mobile terminal,

a push client in the wireless terminal detects a device with a preferred alternative network connection, or a push client in the device with a network connection detects physical proximity of the wireless terminal,

establishing a data-transfer capability between the push clients,

routing communication to and from the push client in the wireless terminal through the push client and the network connection of the device.

32. A method as claimed in any one of claims 1-31 wherein the push client accepts compressed information and optionally decompresses such information for the mobile application.

33. A method as claimed in any one of claims 1-32, wherein the push client and push server establish a secure connection and unwrap such security on behalf of the mobile application.

34. A server comprising means for performing the server steps as claimed in any one of method claims 1-33.

35. A wireless terminal comprising means for performing the push client steps as claimed in any one of method claims 1-34.

36. A program product comprising program code means which, when run in a computing unit of a wireless terminal, performs the push client steps as claimed in any one of method claims 1-35.

37. A program product comprising program code means which, when run in a computing device, performs the push server steps as claimed in any one of method claims 1-37.

Figure 1

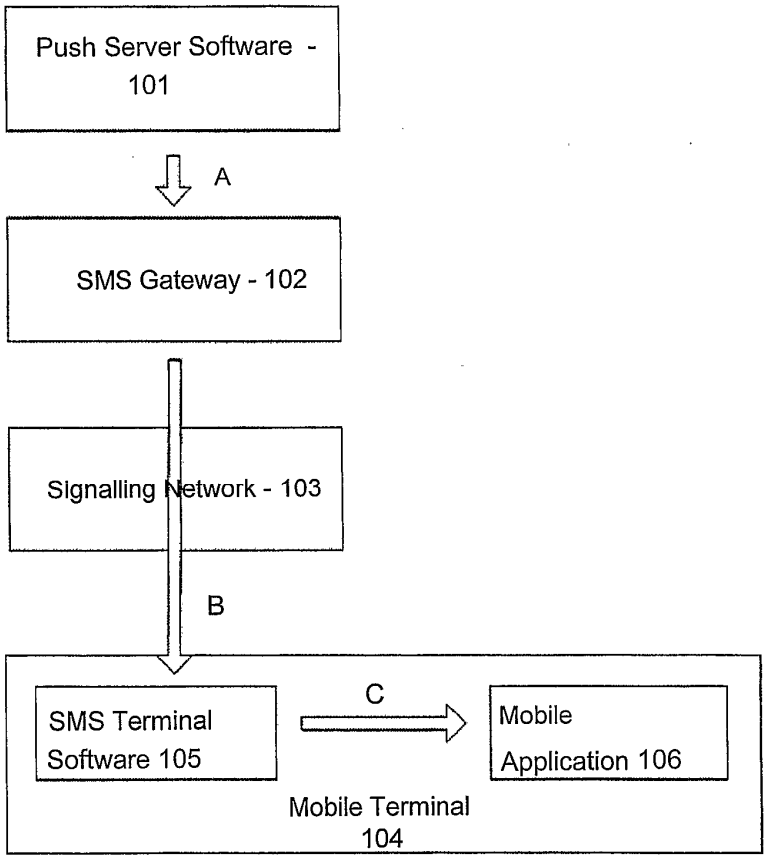

Push Server Software - 101
A
SMS Gateway - 102
Signalling Network - 103
B
SMS Terminal Software 105
C
Mobile Application 106
Mobile Terminal 104

Figure 2

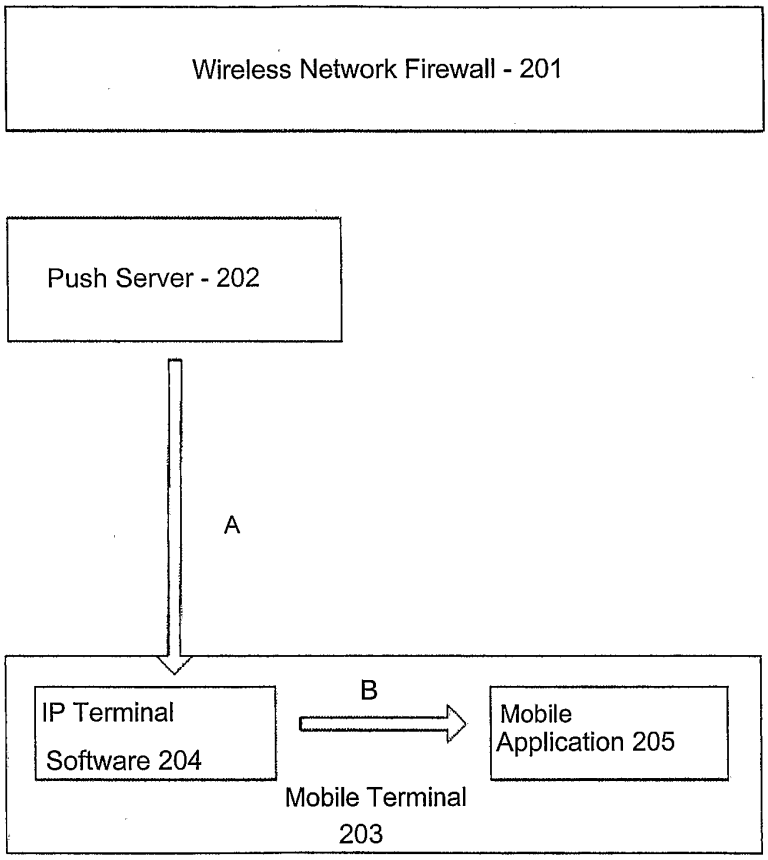

Wireless Network Firewall - 201
Push Server - 202
A
IP Terminal
Software 204
B
Mobile
Application 205
Mobile Terminal
203

Figure 3

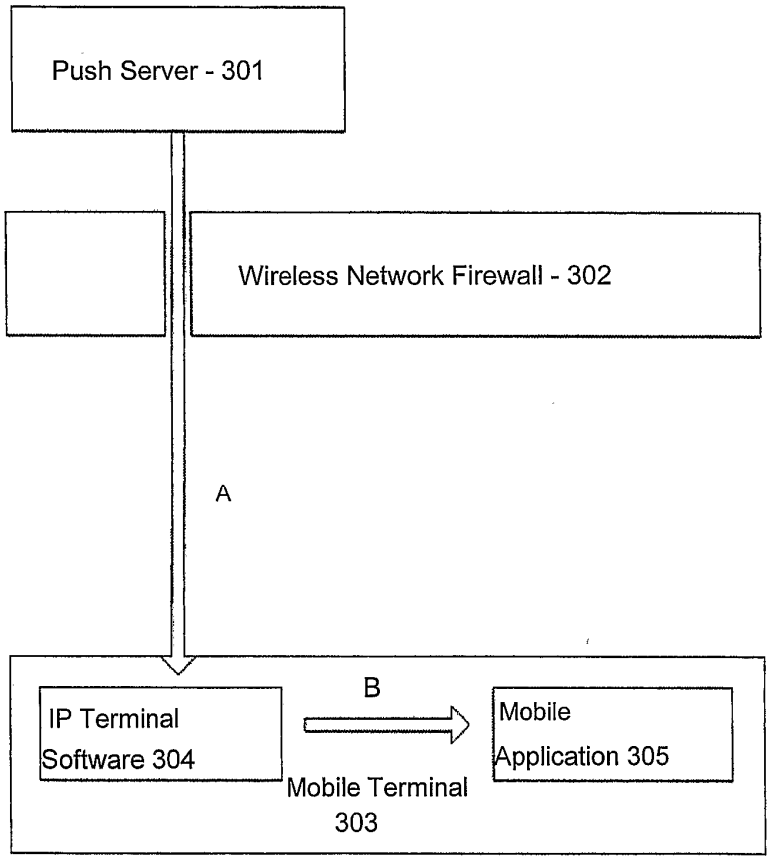

Push Server - 301
Wireless Network Firewall - 302
A
B
IP Terminal Software 304
Mobile Application 305
Mobile Terminal 303

Figure 4

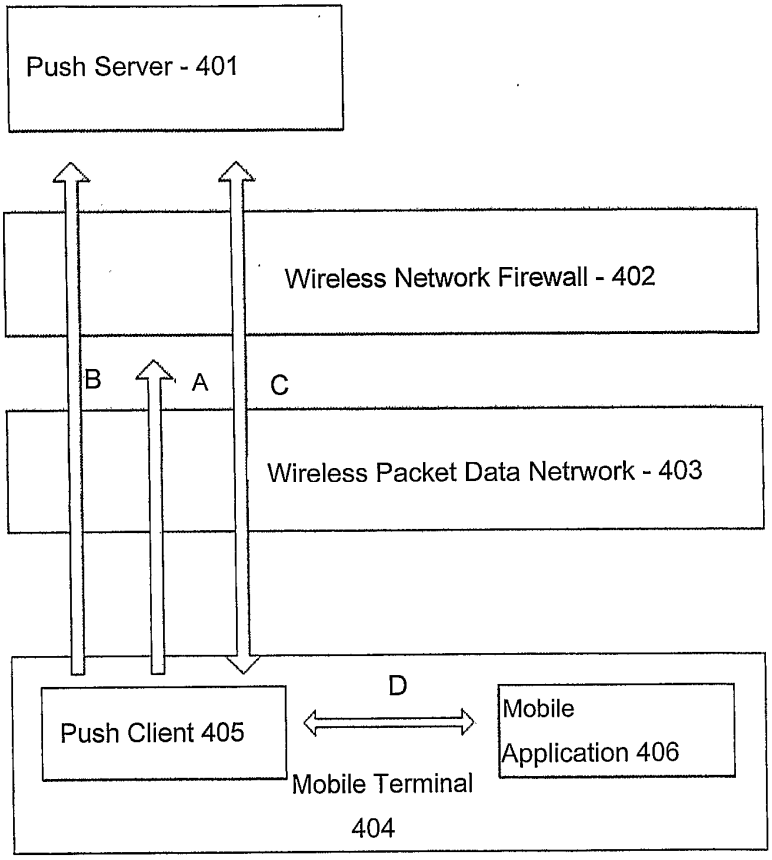

Push Server - 401
Wireless Network Firewall - 402
B
A
C
Wireless Packet Data Netrwork - 403
D
Push Client 405
Mobile
Application 406
Mobile Terminal
404

Figure 5

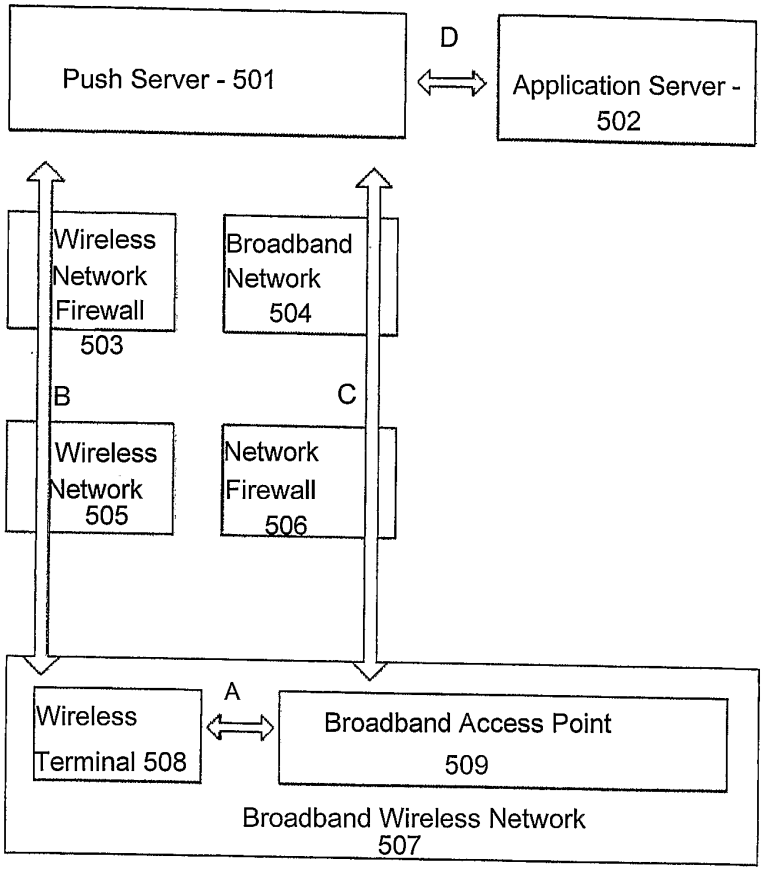

Push Server - 501
D
Application Server - 502
Wireless Network Firewall 503
Broadband Network 504
B
C
Wireless Network 505
Network Firewall 506
A
Wireless Terminal 508
Broadband Access Point 509
Broadband Wireless Network 507

Figure 6A

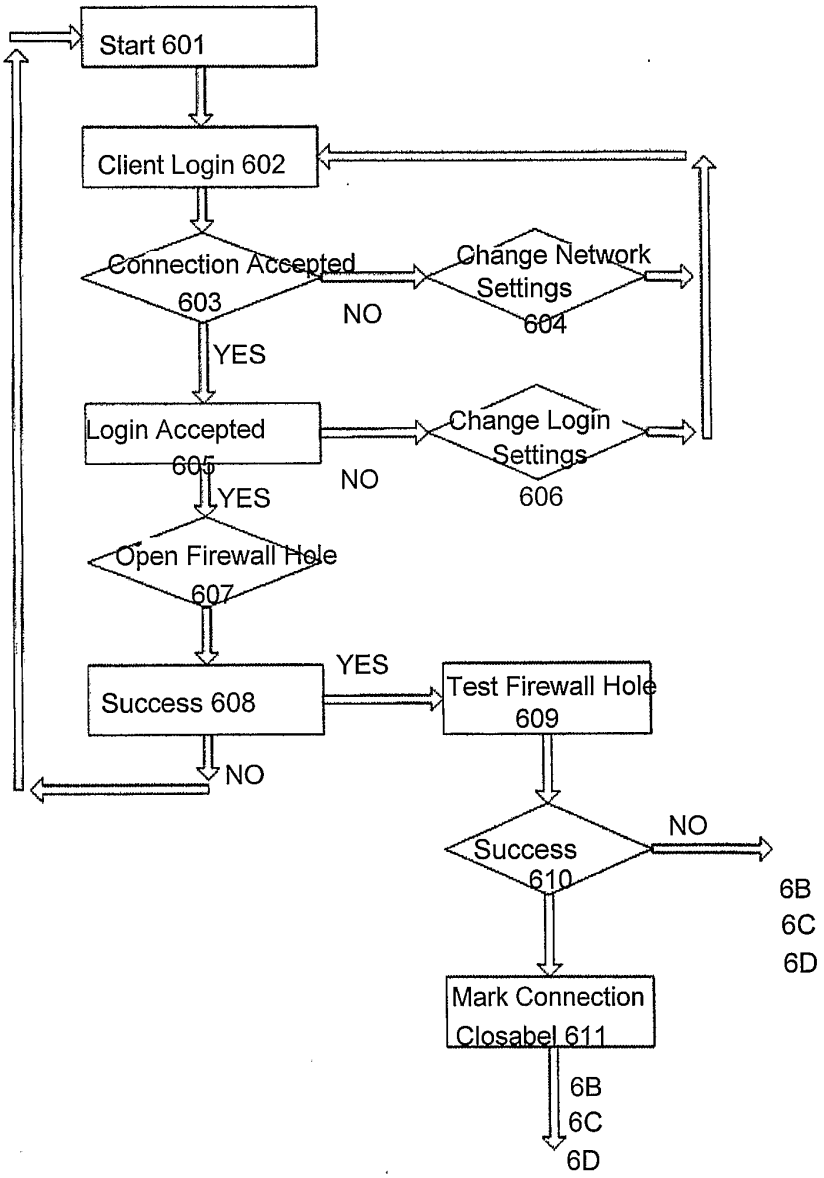

Start 601
Client Login 602
Connection Accepted 603
NO
Change Network Settings 604
YES
Login Accepted 605
NO
Change Login Settings 606
YES
Open Firewall Hole 607
Success 608
YES
Test Firewall Hole 609
NO
Success 610
NO
6B
6C
6D
Mark Connection Closabel 611
6B
6C
6D

Figure 6B

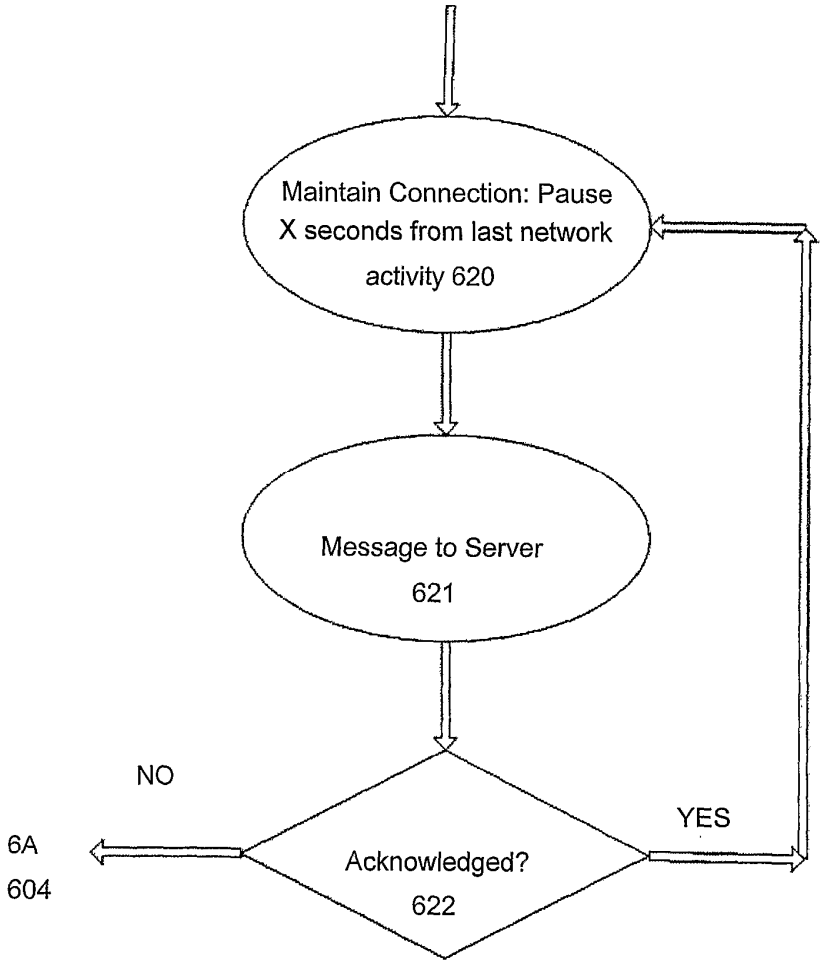

Maintain Connection: Pause X seconds from last network activity 620
Message to Server 621
Acknowledged? 622
NO
YES
6A
604

Figure 6C

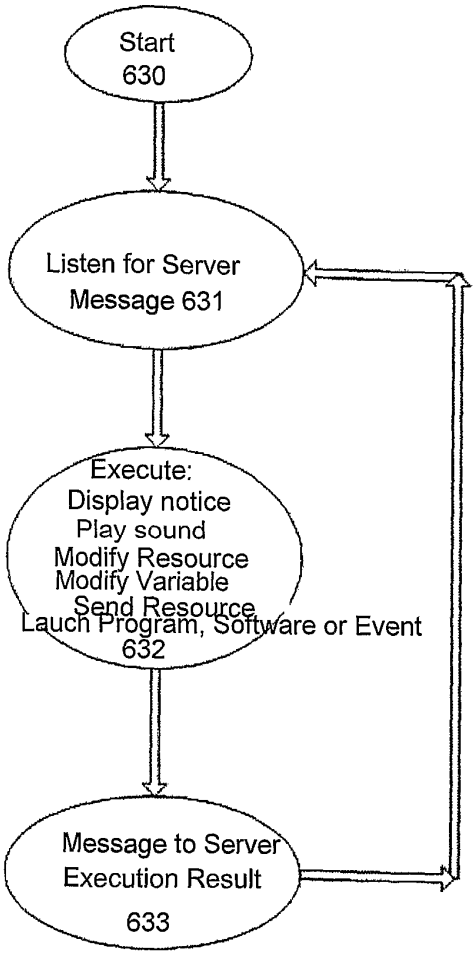

Start
630
Listen for Server
Message 631
Execute:
Display notice
Play sound
Modify Resource
Modify Variable
Send Resource
Lauch Program, Software or Event
632
Message to Server
Execution Result
633

Figure 6D

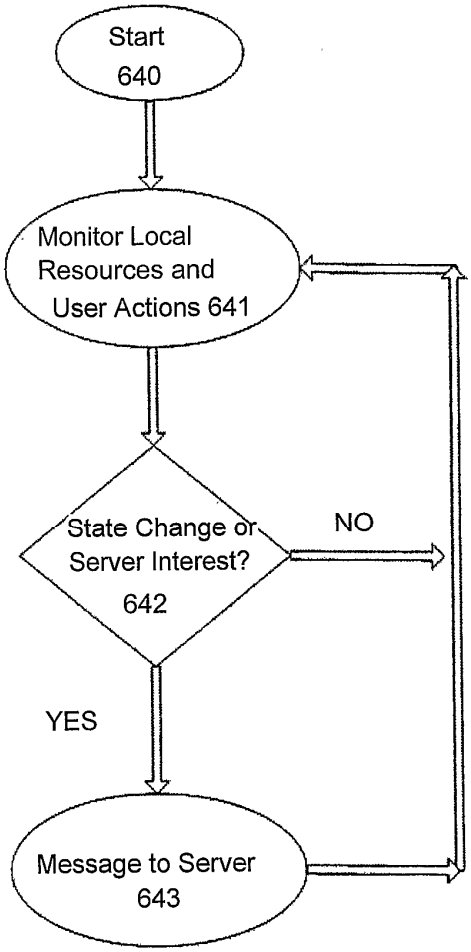

Start
640
Monitor Local
Resources and
User Actions 641
State Change or
Server Interest?
642
NO
YES
Message to Server
643

Figure 7

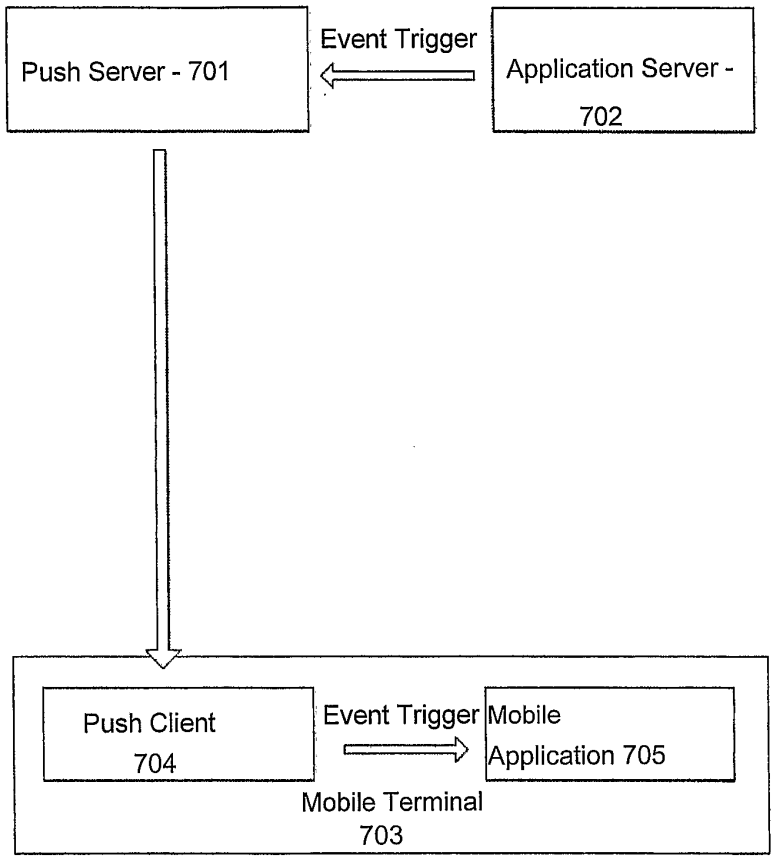

Push Server - 701
Event Trigger
Application Server - 702
Push Client 704
Event Trigger
Mobile Application 705
Mobile Terminal 703

Figure 8

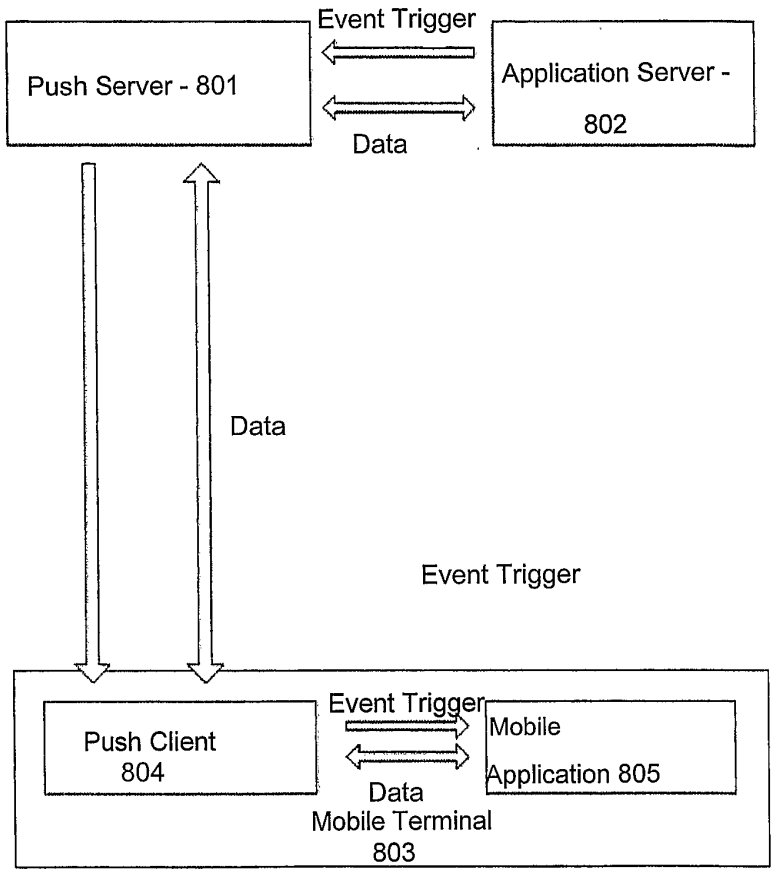

Event Trigger
Push Server - 801
Application Server - 802
Data
Data
Event Trigger
Event Trigger
Push Client 804
Mobile Application 805
Data
Mobile Terminal 803

Figure 9

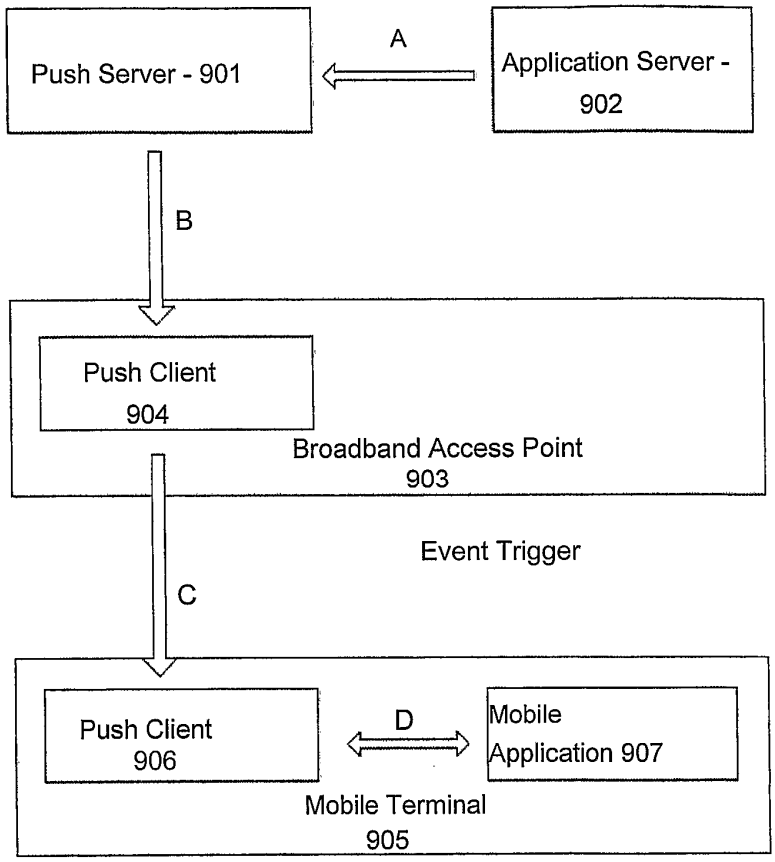

Push Server - 901
A
Application Server - 902
B
Push Client 904
Broadband Access Point 903
Event Trigger
C
Push Client 906
D
Mobile Application 907
Mobile Terminal 905

Figure 10A

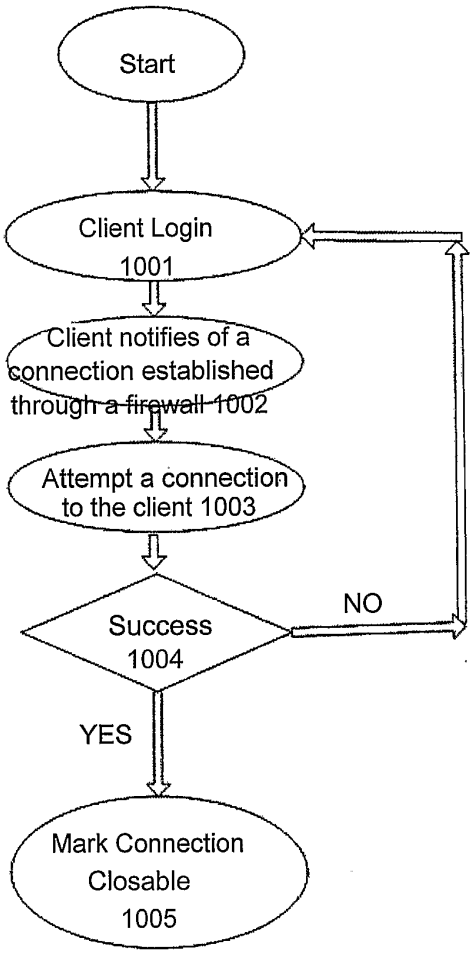

Start
Client Login
1001
Client notifies of a connection established through a firewall 1002
Attempt a connection to the client 1003
Success
1004
NO
YES
Mark Connection Closable
1005

Figure 10B

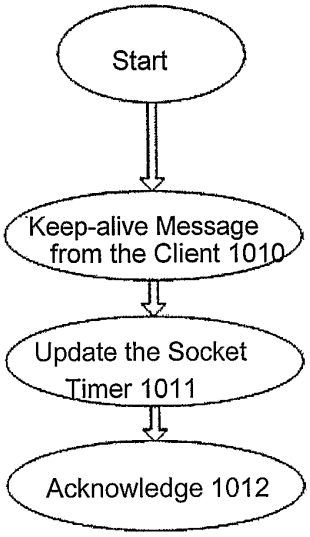
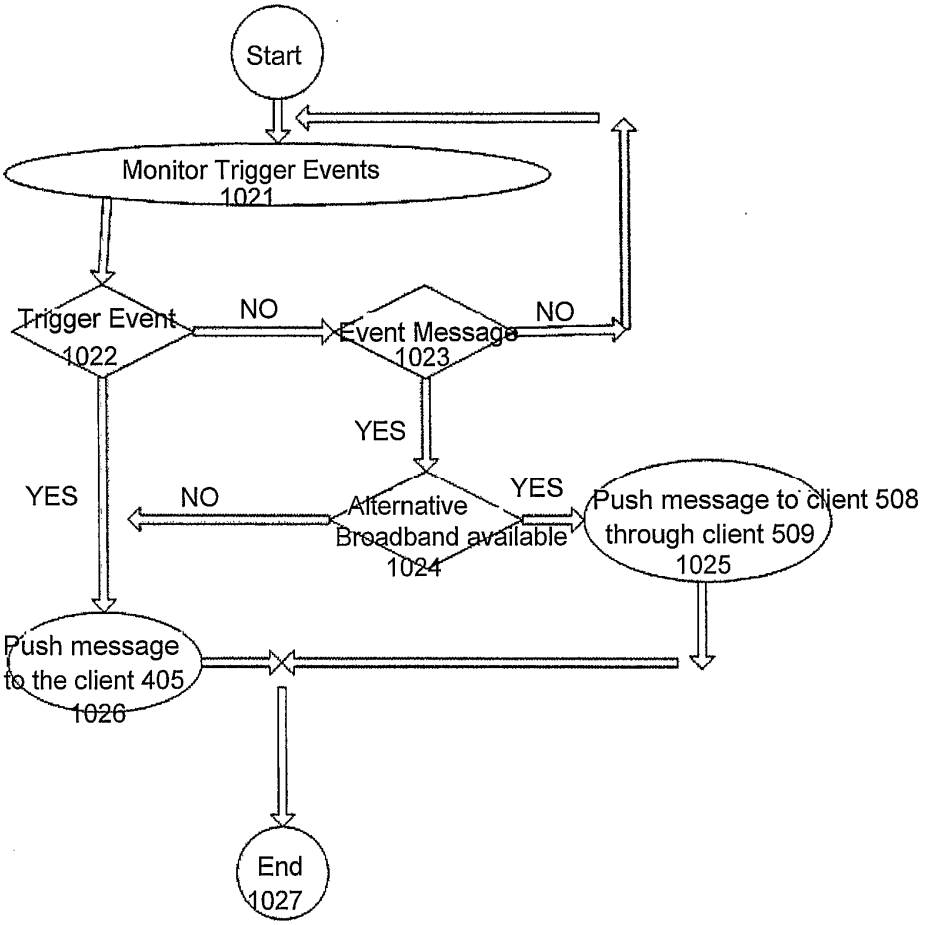

Start
Keep-alive Message from the Client 1010
Update the Socket Timer 1011
Acknowledge 1012

Figure 10C


Start
Monitor Trigger Events
1021
Trigger Event
1022
NO
Event Message
1023
NO
YES
YES
NO
Alternative
Broadband available
1024
YES
Push message to client 508
through client 509
1025
Push message
to the client 405
1026
End
1027

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2006/050032

A. CLASSIFICATION OF SUBJECT MATTER

See extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC8: H04L, H04Q

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

FI, SE, NO, DK

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-INTERNAL, WPI, PAJ, INSPEC, IEEE Xplore

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2001/0005864 A1 (MOUSSEAU et al.) 28 June 2001 (28.06.2001), Abstract, paragraphs [0003], [0013], [0018], [0071], [0073], [0075]-[0082], fig. 6-9	1-4, 6-7, 13, 21-22, 28, 30-31, 34-37
Y	US 2003/0009561 A1 (SOLLEE) 09 January 2003 (09.01.2003), paragraphs [0007], [0027], [0076]	1-4, 6-7, 13, 21-22, 28, 30-31, 34-37
Y	WO 02067538 A2 (CELLTICK TECHNOLOGIES LTD et al.) 29 August 2002 (29.08.2002), Astract, claim 6	1-4, 6-7, 13, 21-22, 28, 30-31, 34-37
A	JP 2003134566 A (AICON KK et al.) 09 May 2003 (09.05.2003), Abstract	1-37
A, P	US 2006/0009198 A1 (KASAI et al.) 12 January 2006 (12.01.2006), whole document	1-37

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
28 March 2006 (28.03.2006)	10 April 2006 (10.04.2006)

Name and mailing address of the ISA/FI	Authorized officer
National Board of Patents and Registration of Finland P.O. Box 1160, FI-00101 HELSINKI, Finland	Seppo Ojala
Facsimile No. +358 9 6939 5328	Telephone No. +358 9 6939 500

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2006/050032

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2001/0005864 A1	28/06/2001	US 2005148356 A1	07/07/2005
		JP 2005100427 A	14/04/2005
		JP 2005100426 A	14/04/2005
		JP 2005056437 A	03/03/2005
		US 2004252727 A1	16/12/2004
		JP 2004220622 A	05/08/2004
		US 2004073619 A1	15/04/2004
		US 2004024824 A1	05/02/2004
		US 2003187938 A1	02/10/2003
		US 2003050987 A1	13/03/2003
		US 2003005066 A1	02/01/2003
		US 2003018816 A1	23/01/2003
		US 2002194285 A1	19/12/2002
		US 2002049818 A1	25/04/2002
		US 2002029258 A1	07/03/2002
		EP 1282957 A2	12/02/2003
		EP 1273136 A2	08/01/2003
		EP 1273135 A2	08/01/2003
		CN 1436413 A	13/08/2003
		WO 0178342 A2	18/10/2001
		WO 0178341 A2	18/10/2001
		WO 0178319 A2	18/10/2001
		CA 2343932 A1	10/10/2001
		CA 2343905 A1	10/10/2001
		CA 2343648 A1	10/10/2001
		AU 5206001 A	23/10/2001
		AU 5020401 A	23/10/2001
		AU 5020101 A	23/10/2001
		US 2001054115 A1	20/12/2001
		US 2002120696 A1	29/08/2002
		US 2001005857 A1	28/06/2001
		US 2001005861 A1	28/06/2001
		US 2001004744 A1	21/06/2001
		US 2001005860 A1	28/06/2001
		US 2001013071 A1	09/08/2001
		US 2001009015 A1	19/07/2001
		MX PA00011592 A	17/10/2002
		NO 20005917 A	26/01/2001
		US 6463463 B1	08/10/2002
		US 6779019 B1	17/08/2004
		US 6463464 B1	08/10/2002
		US 6701378 B1	02/03/2004
		NZ 528050 A	28/01/2005
		NZ 508305 A	30/04/2004
		JP 2002517947T T	18/06/2002
		EP 1082839 A2	14/03/2001
		EP 1320221 A2	18/06/2003
		EP 1315336 A1	28/05/2003

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/FI2006/050032

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
		EP 1206073 A2	15/05/2002
		EP 1126662 A2	22/08/2001
		EP 1124352 A2	16/08/2001
		EP 1098481 A2	09/05/2001
		EP 1096727 A2	02/05/2001
		EP 1096726 A2	02/05/2001
		EP 1096725 A2	02/05/2001
		CN 1304608 A	18/07/2001
		WO 9963709 A2	09/12/1999
		CA 2333881 A1	09/12/1999
		AU 3924499 A	20/12/1999
		AU 758302B B2	20/03/2003
		CA 2511594 A1	29/11/1999
		CA 2367135 A1	29/11/1999
		CA 2356324 A1	29/11/1999
		CA 2356073 A1	29/11/1999
		CA 2356046 A1	29/11/1999
		CA 2356038 A1	29/11/1999
		CA 2356004 A1	29/11/1999
		CA 2245157 A1	29/11/1999
		US 6219694 B1	17/04/2001
US 2003/0009561 A1	09/01/2003	WO 02103981 A2	27/12/2002
		EP 1446929 A2	18/08/2004
WO 02067538 A2	29/08/2002	RU 2004116910 A	10/10/2005
		US 2004078427 A1	22/04/2004
		EP 1362491 A2	19/11/2003
		CN 1493165 A	28/04/2004
JP 2003134566 A	09/05/2003	None	
US 2006/0009198 A1	12/01/2006	JP 2005108012 A	21/04/2005

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2006/050032

CLASSIFICATION OF SUBJECT MATTER

Int.Cl.
H04L 12/56 (2006.01)
H04L 29/08 (2006.01)
H04Q 7/22 (2006.01)