DESCRIPTION

SYSTEMS AND METHODS FOR STREAMING DATA THROUGH A WIRELESS MOBILE DEVICE

FIELD OF THE INVENTION

The present invention relates generally to data transfer over a network, and more particularly to systems and methods for streaming data through a wireless mobile device.

BACKGROUND

Wireless Application Protocol (“WAP”) is a standard for providing content on the Internet to wireless mobile devices, e.g. WAP mobile phones. Typically, a network for providing Internet access to a WAP mobile phone includes a mobile phone network for transmitting data to and receiving data from the mobile phone and a WAP gateway for routing data between the mobile phone network and the Internet.

The Internet includes content written especially for wireless transmission to WAP enabled mobile phones. Such content may be written in a Wireless Markup Language (“WML”), which takes into account the generally small screen size of mobile phones.

In order to access content on the Internet, a WAP mobile phone transmits a request for the content to the mobile phone network. The mobile phone network sends the received request to the WAP gateway, which then routes the request to the Internet. A server on the Internet storing the requested content retrieves the content and sends the content to the WAP gateway. Preferably, the content is already written in WML. If the content is written in a language other than WML, such as HyperText Markup Language (“HTML”), the WAP gateway may reformat the content into WML. The WAP gateway then routes the content to the mobile phone network, which transmits the content to the mobile phone. The mobile phone displays the received content to a user using an embedded microbrowser.

The WAP standard may work with many different cellular phone standards, including but not limited to, GMS and CDMA. In addition, the WAP standard may work with various bearer service protocols provided by a cellular phone standard. For example, the WAP standard may work with various bearer service protocols provided by GMS including General Pack Radio Services (“GPRS”), Unstructured Supplementary Services Data (“USSD”) and Short Message Services (“SMS”).

Another example of a standard, i.e., protocol, for providing content on the Internet to wireless mobile devices is i-Mode, which is currently deployed primarily in Japan. The Internet includes content written especially for i-Mode enabled mobile phones. Such content is typically written using a subset of HTML called compact HTML.

In a data network, data is typically transferred between two entities over the network using an established protocol supported by both entities. The entities may include, but are not limited to, mobile phone networks, mobile phones, gateways, computers, and the like. For example, WAP is a protocol used for transmitting data between a mobile phone and a WAP gateway. WAP further includes a stack of several protocol layers, with each protocol layer handling a different aspect of data transmission between the mobile phone and the WAP gateway. In addition, WAP may run on top of any one of a number of different bearer service protocols provided by a mobile phone network, including, but not limited to, GPRS, USSD, SMS, and the like. For example, in a GMS mobile phone network providing GPRS, the GMS mobile phone network may transmit data to the mobile phone by running WAP on top of GPRS.

Another example of a protocol is Open Systems Interconnect (OSI), which includes a stack of seven protocol layers for transmitting data over the Internet. The topmost protocol layer is an application layer that uses HyperText Transfer Protocol (“HTTP”). Other OSI protocol layers operating below the HTTP layer include a presentation layer, a session layer, a transport layer, a network layer, a data link layer, and a physical layer.

The transport layer and the network layer use Transaction Control Protocol (“TCP”) and Internet Protocol (“IP”), respectively.

HTTP specifies request messages and response messages for facilitating the transfer of data between computers on the Internet. The request message according to HTTP enables a computer to pull, i.e., retrieve, data from another computer. A request message for pulling data from another computer typically includes a “GET” command followed by a Uniform Resource Identifier (“URI”) identifying the location of the data to be retrieved. The request message according to HTTP also enables a computer to push, i.e., store, data onto another computer. A request message for storing data onto another computer typically includes a “PUT” command followed by a URI identifying the location into which the data is to be stored. The data to be stored onto the other computer is typically included in a body of the request message.

The request message may also include authorization information for a requester. The authorization information may include a user ID and a password. The computer receiving the request message uses the received authorization information to authenticate the requester and determine whether or not the requester is authorized to access the URI in the request message.

The response message according to HTTP enables a computer to respond to a request message sent from another computer. The response message typically includes a status code indicating the computer's ability to understand and satisfy a received request message. The response message may further include any requested data in a body of the response message.

HTTP is not limited to running on top of the protocol layers specified by OSI, and may, in principle, run on top of other protocol layers. For example, HTTP may run on bearer service protocols used by mobile phone networks, including GPRS, USSD, and SMS. HTTP may also run on top of mobile phone protocols, including i-Mode, which already supports an HTTP layer. In addition, although HTTP is typically used to transfer web pages written in HTML on the Internet, HTTP may be used to transfer many different file formats, including text files, video files, audio files, Word documents, and the like.

TCP/IP, which operate below HTTP, handle the transport of data between computers in the form of data packets. Specifically, IP groups data into data packets. Each data packet includes a source address and a destination address identifying the source and the destination of the data packet, respectively. The destination address ensures that the data packet is routed to the appropriate computer. TCP ensures that the data packet is correctly received at its destination. If the data packet is incorrectly received at its destination, TCP enables the computer at the destination to signal the computer at the source to retransmit the data packet.

SUMMARY OF THE INVENTION

The present invention is directed to systems and methods for transferring data between a networked computer and a local device by streaming the data through a wireless mobile device.

A data network, in accordance with one aspect of the present invention, includes a remote server connected to the Internet, a mobile phone network, and a gateway connected between the mobile phone network and the Internet for routing data between the mobile phone network and the Internet. The data network further includes a wireless mobile device and a local device.

The mobile device includes a Radio Frequency (RF) transceiver for receiving data from and transmitting data to the mobile phone network using a bearer service protocol, e.g., GPRS, provided by the mobile phone network. The mobile device also includes an Infrared Data Application (“IrDA”) transceiver for transmitting data to and receiving data from the local device, which is also equipped with an IrDA transceiver. The mobile device further includes a memory buffer and a proxy for streaming data through the mobile device.

To stream a file from the server to the local device through the mobile device, the mobile device initiates a “GET” request message according to HTTP to pull the file from the server. The “GET” request message includes a “GET” command, the URI of the file on the server, and authorization information for the mobile device, e.g., a user ID and password. The mobile device transmits the “GET” request message to the mobile phone network by running an HTTP layer on top of a bearer service protocol, e.g., GPRS.

The mobile phone network passes the received “GET” request message to the gateway, which routes the “GET” request message to the Internet. The server receives the “GET” request message via the Internet. Upon receiving the “GET” request message, the server may check the authorization information included in the “GET” request message to determine whether or not the mobile device is authorized to access the file identified by URI in the “GET” request message. If the mobile device is authorized to access the file, the server retrieves the file identified by the URI. The server then generates a response message according to HTTP. The response message includes the requested file in the body of the response message. The server transmits the response message to the Internet. The gateway receives the response message via the Internet and routes the response message to the mobile phone network.

The mobile phone network transmits the response message to the mobile device by running an HTTP layer on top of the bearer service protocol. The mobile phone receives the response message from its RF transceiver and temporarily stores the file in the received response message in its memory buffer. The proxy on the mobile device then generates a “PUT” request message according to HTTP to push the received file onto the local device. The “PUT” request message includes a “PUT” command, a URI on the local device, and the file stored in the memory buffer. The mobile device transmits the “PUT” request message to the local device using its IrDA transceiver by running an HTTP layer on top of an IrDA protocol.

The local device receives the “PUT” request from its IrDA transceiver. The local device stores the file in the received “PUT” request message into a memory location identified by the URI in the “PUT” request message.

Other objects and features of the present invention will become apparent from consideration of the following description taken in conjunction with the accompanying drawings.

BREIF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a data network, according to an embodiment of the present invention.

FIGS. 2A and 2B are flowcharts showing steps for streaming a file from a remote server to a local device through a mobile device, according to one embodiment of the present invention.

FIGS. 3A and 3B are flowcharts showing steps for streaming a file from a local device to a remote server through a mobile device, according to one embodiment of the present invention.

FIG. 4 is a block diagram of a network for streaming data from one server to another server, according to an embodiment of the present invention.

FIG. 5 is a block diagram of the network in FIG. 4 further including a caching device, according to an embodiment of the present invention.

FIG. 6 is a block diagram of a network for streaming data from a remote server to a local device through another local device, according to an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 1 shows an exemplary data network 5 according to an embodiment of the present invention. The network 5 includes a remote server 10, e.g., a PC, connected to the Internet 15, via a communications link 12 including, but not limited to, a dialup link, a DSL link, an Ethernet link, and the like. The network 5 also includes a mobile phone network 20. The mobile phone network 20 may be, but is not limited to, a GMS or a CDMA mobile phone network. The network 5 also includes a gateway 25, e.g., a WAP gateway, connected between the mobile phone network 20 and the Internet 15 for routing data between the mobile phone network 20 and the Internet 15. The network 5 further includes a wireless mobile device 30, including, but not limited to, a WAP mobile phone, a Personal Digital Assistant (“PDA”), and the like.

The mobile device 30 includes an RF transceiver 35 for transmitting data to and/or receiving data from the mobile phone network 20 using a bearer service protocol provided by the mobile phone network 20. For example, in the case of a GMS mobile phone network, the mobile device 30 may use GPRS to communicate data with the mobile phone network 20.

The mobile device 30 also includes an Infrared Data Application (“IrDA”) transceiver 40 for transmitting data to and/or receiving data from a local device 45, which is also equipped with an IrDA transceiver 50. Many commercially available mobile phones already come with a built-in IrDA transceiver for providing a point-to-point communications link between the mobile phone and a nearby local device.

Alternatively, the transceiver 40 may use other wireless communications methods, such as short-range Radio Frequency (RF) signals, e.g., Bluetooth signals, to communicate with the local device 45. The local device 45 may include an electronics device within close proximity to the mobile device 30 but not directly coupled to mobile device 30, e.g., configured to communicate with the mobile device using wireless communications methods. Exemplary local devices include, but are not limited to, printers, FAX machines, video projectors, WAP mobile phones, and the like.

Preferably, the local device 45 includes one or more software-based modules and/or hardware-based components (not shown) that enable the local device 45 to transfer data over an IrDA communications link using HTTP. This may be done, for example, by having the software-based modules and/or hardware-based components generate and interpret messages, e.g., request and response messages, according to HTTP. The local device 45 also includes memory (not shown) for storing data. Preferably, addresses in the memory of the local device 45 may be identified with URIs.

In order to discover the local device 45, the mobile device 30 may broadcast a discovery request using its IrDA transceiver 40. The local device 45 responds to the discovery request by transmitting its Media Access Control (“MAC”) address to the mobile device 30. This enables the mobile device 30 to establish an IrDA communications link with the local device 45. In addition to transmitting its MAC address, the remote 45 device may also transmit a URI identifying a memory location into which files may be stored. If more than one local device responds to the discovery request, then the mobile device 30 may display a list of the local devices to the user, and allow the user to select a local device.

The mobile device 30 further includes a proxy (not shown) for streaming data through the mobile device 30. The term “proxy” is used generally herein to refer to any combination of software-based modules and/or hardware-based components that may perform the features described herein. The mobile device 30 also includes a memory buffer (not shown) for temporarily storing data.

Steps for streaming a file from the server 10 to the local device 45 through the mobile device 30 will now be described with reference to FIGS. 2A and 2B.

In step 210, a user of the mobile device 30 selects one or more file(s) on the server 10 that he or she wishes to transfer to the local device 45. This may be done, for example, by having the mobile device 30 display a table listing files on the server 10 and allowing the user to select the desired file(s) from the table. The table may be stored locally on the mobile device 30 or downloaded onto the mobile device 30 from the server 10.

In step 220, the mobile device 30 determines the URI of the selected file. This may be done, for example, by including the

URI of the selected file in the table or, if the user already knows the URI of the selected file, having the user enter the URI directly into the mobile device 30.

In step 225, the mobile device 30 initiates a “GET” request message according to HTTP to pull the selected file from the server 10. The “GET” request message includes a “GET” command and the URI of the selected file. The “GET” request message may also include authorization information, e.g., a user ID and password, for the mobile device 30. In step 227, the mobile device 30 transmits the “GET” request message to the mobile phone network 20 by running an HTTP layer on top of a bearer service protocol, e.g., GPRS, provided by the mobile phone network 20. Intermediate protocol layers may be inserted between the HTTP layer and the bearer service protocol, including, but not limited to, TCP/IP and Wireless Transaction Protocol (“WTP”).

In step 230, the mobile phone network 20 receives the “GET” request message and passes the “GET” request message to the gateway 25. In step 235, the gateway 25 routes the “GET” request message to the Internet 15. The gateway 25 may also translate some of the protocol layers underlying the “GET” request message from protocol layers used by the mobile phone network 20 to protocol layers used by the Internet 15.

In step 240, the server 10 receives the “GET” request message via the Internet 15. In step 243, the server 10 checks the authorization information in the “GET” request message to determine whether or not the mobile device 30 is authorized to access the file identified by the URI in the “GET” request message. If the mobile device 30 is not authorized to access the file, the server 10 may transmit a response message to the mobile device 30 informing the user of the mobile device 30 that he or she is “forbidden” from accessing the request file. Otherwise, in step 245, the server 10 retrieves the requested file identified by the URI in the “GET” request message. In step 250, the server 10 generates a response message according to HTTP, which includes the retrieved file in the body of the response message. In step 255, the server 10 sends the response message via the Internet 15. In step 260, the gateway 25 receives the response message via the Internet 15 and routes the response message to the mobile phone network 20. In step 265, the mobile phone network 20 transmits the response message to the mobile device 30 by running an HTTP layer on top of the bearer service protocol provided by the mobile phone network 20.

In step 270, the mobile device 30 receives the response message including the requested file and temporarily stores the file in its memory buffer. In step 275, the proxy on the mobile device 30 generates a “PUT” request message according to HTTP to push the received file onto the local device 45. The “PUT” request message includes a “PUT” command, a URI on the local device 45, and the file temporarily stored in the memory buffer. In step 280, the mobile device 30 establishes an IrDA communications link with the local device 45 and transmits the “PUT” request message to the local device 45 by running an HTTP layer on top of the IrDA protocol. In step 285, the local device 45 receives the “PUT” request message from the mobile device 30 and stores the file included in the received “PUT” request message at a memory location identified by the URI in the “PUT” request message. In step 290, the local device 45 transmits a response message to the mobile device 30 including a status code indicating whether or not the file was successfully stored in the local device 45.

In another embodiment, the mobile device 30 begins streaming file data from a response message to the local device 45 before receiving all of the file data in the response message. In this embodiment, the mobile device 30 stores incoming file data from the response message into its memory buffer. As the mobile device 30 receives the file data from the response message, the proxy begins transmitting the file data already stored in the memory buffer to the local device 45 using a “PUT” request message. As file data is transmitted to the local device 45, the proxy clears file data that has already been transmitted to the local device 45 from the memory buffer. This frees up space in the memory buffer, enabling the mobile device 30 to receive more file data from the response message. An advantage of this embodiment is that the memory buffer does not have to be large enough to store an entire file at one time, allowing the mobile device 30 memory to be more compact.

In yet another embodiment, the proxy on the mobile device 30 controls the data flow between the server 10 and the mobile device 30. In one example, the mobile device 30 receives data from the mobile phone network 20 at a faster data rate than it can transmit the data to the local device 45. As a result, the memory buffer in the mobile device 30 may overflow. In this case, the proxy may transmit TCP/IP layer signals to the server 10 in order to control the data flow between the mobile device 30 and the server 10. For example, when the proxy on the mobile device 30 detects that the memory buffer is near capacity, the proxy may transmit a TCP/IP layer signal to the server 10 to temporarily suspend data transmission to the mobile device 30. When sufficient space in the memory buffer has been cleared, the proxy may then transmit a TCP/IP layer signal to the server 10 to resume data transmission to the mobile device 30.

An advantage of the mobile device 30 according to the present invention is that it may be used to stream data to the local device 45 in cases where the local device 45 itself is not authorized to access a file on the server 10. For example, the server 10 including the file may be part of a local network that is protected by a firewall. The firewall may restrict access to the file to authorized users that do not include the local device 45. In this case, the mobile device 30, which is an authorized user, may initiate a “GET” request message for the file and stream the file from the server 10 to the local device 45 through the mobile device 30. The mobile device 30 in effect becomes part of the data path between the server 10 and the local device 45.

This advantage of the present invention may be illustrated by way of an example. Suppose a salesperson visiting a customer wishes to print a file on the customer’s printer 45. The file is stored on a company server 10 protected by a firewall that the customer’s printer 45 is not authorized to access. In this case, the salesperson may use the mobile device 30 to request the file from the company server 10 and stream the file from the company server 10 to the customer’s printer 45 via the mobile device 30.

The invention may be used for streaming data from a remote server, e.g., a PC, to a variety of local devices 45. For example, the mobile device 30 may be used to stream a document from a remote PC connected to the Internet 15 to a local computer or FAX machine. The mobile device 30 may also be used to stream a media file, such as a video file, e.g., a Quicktime file, from a remote PC connected to the Internet 15 to a display device, such as a video projector. The examples given above are for illustrative purposes and are not intended to be exhaustive.

In addition to streaming files to the local device 45, the mobile device 30 may also transmit commands to the local device 45 for instructing the local device 45 on how to process a received file. For example, in the case of a printer, the mobile device 30 may command the printer via an IrDA communications link to print a received file. The mobile device 30 may transmit the command to the printer using an HTTP layer or a protocol established by the printer for receiving remote commands.

Although an IrDA communications link was used in the above examples between the mobile device 30 and the local device 45, other wireless communications links may be used, such as a short-range Radio Frequency (RF) link, e.g., a Bluetooth communications link. In this case, the mobile device 30 and the local device 45 each may be equipped with a short range RF transceiver, e.g., configured to use a Bluetooth protocol. The mobile device 30 and local device 45 may transfer data between each other by running an HTTP layer on top of the Bluetooth protocol. An advantage of using Bluetooth is that it may provide a secure low-power data transport mechanism between the mobile device 30 and the local device 45.

The mobile device 30 according to the present invention may also be used to stream data from the local device 45 to the remote server 10 through the mobile device 30.

Steps for streaming data from the local device 45 to the server 10 will now be described with reference to FIGS. 3A and 3B.

In step 310, a user on the mobile device 30 selects a file on the local device 45 that he or she wishes to transfer to the server 10. This may be done, for example, by having the mobile device 30 display a table listing files on the local device 45 and allowing the user to select a desired file from the table. The table may be stored locally on the mobile device 30 or downloaded onto the mobile device 30 from the local device 45.

In step 320, the mobile device 30 determines the URI of the selected file. This may be done, for example, by including the URI of the selected file in the table. In step 325, the user selects the location on the server 10 in which he or she wishes to store the file. The corresponding URI may be stored locally on the mobile device 30 or entered directly into the mobile device 30 by the user. Alternatively, the mobile device 30 may contact the server using a URI and allow the server 10 to determine in which location to store the selected file.

In step 330, the mobile device 30 initiates a “GET” request message for the selected file according to HTTP to pull the selected file from the local device 45. The “GET” request message includes a “GET” command and the URI of the selected file on the local device 45. In step 332, the mobile device transmits the “GET” request message to the local device 45 by running an HTTP layer on top of the IrDA protocol, or if a Bluetooth communications link is being used, on top of the Bluetooth protocol.

In step 335, the local device 45 receives the “GET” request message, and retrieves the requested file identified by the URI in the “GET” request message. In step 340, the local device 45 generates a response message according to HTTP including the requested file. In step 345, the local device transmits the response message to the mobile device 30 by running an HTTP layer on top of the IrDA protocol.

In step 350, the mobile device 30 receives the response message and temporarily stores the file in the received response message into its memory buffer. In step 353, the proxy on the mobile device 30 generates a “PUT” request message according to HTTP to push the received file onto the server 10. The “PUT” request message includes a “PUT” command, the URI of the selected location on the server 10 and the file temporarily stored in the memory buffer. In step 355, the mobile device 30 transmits the “PUT” request message to the mobile phone network 20 by running an HTTP layer on top of a bearer service protocol, such as GPRS. Intermediate protocol layers may be inserted between the HTTP layer and the bearer service protocol, including, but not limited to, TCP/IP and Wireless Transaction Protocol (“WTP”).

In step 360, the mobile phone network 20 receives the “PUT” request message and passes the “PUT” request message to the gateway 25. In step 365, the gateway 25 routes the “PUT” request message to the Internet 15. The gateway 25 may also translate some of the protocol layers underlying the “PUT” request message from protocol layers used by the mobile phone network 20 to protocol layers used by the Internet 15.

In step 370, the server 10 receives the “PUT” request message from the Internet 15. In step 375, the server 10 stores the file included in the “PUT” request message at the location identified by the URI in the “PUT” request message. The server 10 may then send a response message to the mobile device 30 including a status code indicating whether or not the file was successfully stored in the server 10.

FIG. 4 shows a data network 405 according to another embodiment of the invention. The network 405 includes a second mobile device 410 and a second server 450 connected to the Internet 15. The second mobile device 410 includes an IrDA transceiver 420 for communicating data with the first mobile device 30 and an RF transceiver 430 for communicating data with the mobile phone network 20. The second mobile device further includes a memory buffer (not shown), and a proxy (not shown) according to the invention.

The data network 405 according to this embodiment enables a file to be streamed from the first server 10 to the second server 450 through the first mobile device 30 and the second mobile device 410. In this embodiment, the user of the first mobile device 30 selects a file on the first server 10. The first mobile device 30 then streams the selected file from the first server 10 to the second mobile device 410 by following steps 210-290 described above, in which the second mobile device 410 acts as the local device 45 and the received file is temporarily stored in the memory buffer of the second mobile device 410. A user of the second mobile device 410 selects a location on the second server 450 in which to store the file. Alternatively, the second server 450 may select where to store the file in its memory. The second mobile device 410 then streams the file received from the first mobile device 30 to the second server 450 by following steps 353 to 375 described above.

Although, the second mobile device 410 is shown communicating with the same mobile phone network 20 as the first mobile device 30, those skilled in the art will appreciate that the second mobile device 410 may communicate with a different mobile phone network. For example, the user of the second mobile device 410 may subscribe to a different phone service than the user of the first mobile device 30.

FIG. 5 shows a data network 505 according to yet another embodiment. The network 505 includes a caching device 510, wherein the caching device 510 includes one or more IrDA transceivers 520 for communicating data with the first and second mobile devices 30, 410. The caching device 510 also includes a memory buffer (not shown), and a proxy (not shown) according to the invention. The caching device 510 may be a PDA, a laptop, or the like. Preferably, the caching device 510 has more memory space than either mobile device 30, 410. The caching device 505 facilitates the temporary caching, i.e., storing, of files that are streamed through the first and second mobile devices 30, 410.

In this embodiment, the user of the first mobile device 30 selects a file on the server 10 and streams the selected file to the caching device 505 by following steps 210-290 described above, in which the caching device 510 acts as the local device 45 and the received file is temporarily stored in the memory buffer of the caching device 510. Once the file is stored on the caching device 510, the caching device 510 initiates a “PUT” request message to push the file onto the second mobile device

410. The caching device 510 transmits the “PUT” request message including the file to the second mobile device 410. A user of the second mobile device 410 selects a location on the second server 450 at which to store the file. The second mobile device 410 streams the file received from the caching device 510 to the second server 450 by following steps 353 to 375, described above.

In this embodiment, the caching device 510 may use various methods for communicating data with the first and second mobile devices 30, 410. For example, the caching device 510 may include a short-range RF transceiver for communicating data with the first and/or second mobile devices 30, 410, e.g., using a Bluetooth communications link.

An advantage of this embodiment is that the caching device 510 may be used to simplify data flow between the first and second mobile devices 30, 410, especially for the case in which one or both of the mobile devices 30, 410 does not have enough memory buffer space to store an entire file. Without the caching device 510, the data flow between the first and second mobile devices 30, 410 may have to be tightly controlled to make sure that the memory buffer of either of the mobile devices 30, 410 does not overflow. The caching device 510 simplifies data flow between the first and second mobiles 30, 410 device by temporarily caching, i.e. storing, a complete file received from the first mobile device 30. Once the complete file is stored in its memory buffer, the caching device 510 may then transmit the file to the second mobile device 410. Preferably, the caching device 510 transmits the file to the second mobile device 410 at a data rate that enables the second device to stream the file to the second server 450 without overflowing its memory buffer.

FIG. 6 shows a data network 605 according to still another embodiment. The network 605 includes a first local device 610 and a second local device 640, wherein the second local device 630 is similar to local device 45. The first local device 610 includes an IrDA transceiver 620 for communicating data with the mobile device 30, a memory buffer (not shown), and a proxy (not shown) according to the invention. The first local device 610 includes a second transceiver 630 for communicating data with the second local device 640, which is equipped with a similar transceiver 650. The second transceiver 630 may be an IrDA transceiver or a short-range RF transceiver.

The data network 605 according to this embodiment enables data to be streamed from the server 10 to the second local device 640 through the mobile device 30 and the first local device 610. To do this, the mobile device 30 streams a desired file from the server 10 to the first local device 610 by following steps 210-290 described above. The first local device 610 then streams the file received from the mobile device 30 to the second local device 640 by following steps 275-290 described above.

The first local device 610 according to this embodiment does not need to include a second transceiver 630. In this case, the first local device 610 may communicate data with the mobile device 30 and the second local device 640 using the same transceiver 620. This may be done by having the first local device 610 completely receive and temporarily store an entire file before streaming the file to the second local device 640. Alternatively, this may be done by having the first local device 610 toggle, i.e., switch, between receiving file data from the mobile device 30 and transmitting the file data to the second local device 640. In this case, the mobile device 30 may divide a file into portions and transmit the file to the first local device 610 a portion at a time. Each time the first local device 610 receives a portion of the file from the mobile device 30, the first local device 610 may transmit the received portion of the file to the second local device 640. Once the received portion of the file has been transmitted to the second local device 640, the first local device 610 may receive a next portion of the file from the mobile device 30.

While the invention is susceptible to various modifications, and alternative forms, specific examples thereof have been shown in the drawings and are herein described in detail. It should be understood, however, that the invention is not to be limited to the particular forms or methods disclosed, but to the contrary, the invention is to cover all modifications, equivalents and alternatives falling within the spirit and scope of the appended claims.

CLAIMS

What is claimed is:

1. A method for streaming data through a wireless mobile device comprising a first transceiver and a second transceiver, the method comprising:

receiving data from the first transceiver using an application protocol; and

transmitting the received data from the second transceiver using the application protocol used by the first transceiver for receiving the data.

2. The method of claim 1, wherein the application protocol is a HyperText Transfer Protocol (HTTP).

3. The method of claim 1, wherein the first transceiver is a Radio Frequency (RF) transceiver.

4. The method of claim 3, wherein the second transceiver is an Infrared Data Application (IrDA) transceiver.

5. The method of claim 3, wherein the second transceiver is a short-range Radio Frequency (RF) transceiver.

6. The method of claim 3, wherein the first transceiver receives the data by running the application protocol on top of a mobile phone network protocol.

7. The method of claim 6, wherein the application protocol is a HyperText Transfer Protocol (HTTP).

8. The method of claim 6, wherein the mobile phone network protocol is General Pack Radio Services (GPRS).

9. The method of claim 6, wherein the mobile phone network protocol is i-Mode.

10. The method of claim 4, wherein the second transceiver transmits the received data by running the application protocol on top of an IrDA protocol.

11. The method of claim 10, wherein the application protocol is a HyperText Transfer Protocol (HTTP).

12. The method of claim 5, wherein the second transceiver transmits the received data by running the application protocol on top of a Bluetooth protocol.

13. The method of claim 12, wherein the application protocol is a HyperText Transfer Protocol (HTTP).

14. A method for streaming data through a wireless mobile device, comprising:

generating a “GET” request message for pulling data from a server connected to the Internet, the “GET” request message comprising a Uniform Resource Identifier (URI) identifying the location of the data on the server;

transmitting the “GET” request message from the mobile device;

receiving a response message including the data identified by the URI in the “GET” request message;

generating a “PUT” request message for pushing the received data onto a local device, the “PUT” request message comprising a URI identifying a memory location on the local device; and

transmitting the “PUT” request message from the mobile device.

15. The method of claim 14, wherein the “GET” request message is transmitted from a Radio Frequency (RF) transceiver on the mobile device using a mobile phone network protocol.

16. The method of claim 15, wherein the mobile phone network protocol is General Pack Radio Services (GPRS).

17. The method of claim 15, wherein the mobile phone network protocol is i-Mode.

18. The method of claim 14, wherein the “PUT” request message is transmitted from an Infrared Data Application (IrDA) transceiver on the mobile device using an IrDA protocol.

19. The method of claim 14, wherein the “PUT” request message is transmitted from a short-range Radio Frequency (RF) transceiver on the mobile device using a Bluetooth protocol.

20. The method of claim 14, wherein the “GET” request message comprises authorization information for the mobile device.

21. The method of claim 20, wherein the authorization information comprises a password.

22. The method of claim 14, wherein the data is a media file, and wherein the method further comprises:

receiving the “PUT” request message including the media file at the local device; and

playing the received media file on the local device.

23. The method of claim 22, wherein the media file is a video file.

24. The method of claim 14, wherein the data is a document, and the method further comprises:

receiving the “PUT” request message including the document at the local device; and

printing the received document on the local device.

25. A method for streaming data through a wireless mobile device, comprising:

generating a “GET” request message for pulling data from a local device, the “GET” request message comprising a Uniform Resource Identifier (URI) identifying the location of the data on the local device;

transmitting the “GET” request message from the mobile device;

receiving a response message including the data identified by the URI in the “GET” request message;

generating a “PUT” request message for pushing the received data onto a server connected to the Internet, the “PUT” request message comprising a URI identifying a memory location on the server; and

transmitting the “PUT” request message from the mobile device.

26. The method of claim 25, wherein the “GET” request message is transmitted from an Infrared Data Application (IrDA) transceiver on the mobile device using an IrDA protocol.

27. The method of claim 25, wherein the “GET” request message is transmitted from a short-range Radio Frequency (RF) transceiver on the mobile device using a Bluetooth protocol.

28. The method of claim 25, wherein the “PUT” request message is transmitted from a Radio Frequency (RF) transceiver on the mobile device using a mobile phone network protocol.

29. The method of claim 25, wherein the mobile phone network protocol is General Pack Radio Services (GPRS).

30. The method of claim 25, wherein the mobile phone network protocol is i-Mode.

31. A method for streaming a data file through a mobile device, comprising:

sending a request for the data file from the mobile device to a remote server;

receiving the requested data file on the mobile device; and

transmitting the data file from the mobile device to a local device.

32. The method of claim 31, wherein the step of sending the request to the remote server further comprises:

transmitting the request from the mobile device to a mobile phone network; and

routing the request from the mobile phone network to the remote server.

33. The method of claim 32, wherein the request includes a Uniform Resource Identifier (URI) identifying the location of the data file on the remote server, and wherein the method further comprises retrieving the data file from the location identified by the URI in the request.

34. The method of claim 32, wherein the data file is transmitted to the local device from an Infrared Data Application (IrDA) transceiver on the mobile device.

35. The method of claim 32, wherein the data file is transmitted to the local device from a short-range Radio Frequency (RF) transceiver on the mobile device.

36. The method of claim 31, wherein the data file is a media file, and wherein the method further comprises playing the media file on the local device.

37. The method of claim 31, wherein the data file is a document, and wherein the method further comprises printing the document on the local device.

38. A method for streaming a data file through a mobile device, comprising:

transmitting a request for the data file from the mobile device to a local device;

receiving the requested data file from the local device on the mobile device; and

sending the data file from the mobile device to a remote server.

39. The method of claim 38, wherein the step of sending the data file to the remote server further comprises:

transmitting the data file to a mobile phone network; and

routing the data file from the mobile phone network to the remote server.

40. The method of claim 38, wherein the request includes a Uniform Resource Identifier (URI) identifying the location of the data file on the local device, and wherein the method further comprises retrieving the data file from the location identified by the URI in the request.

41. The method of claim 38, wherein the request for the data file is transmitted to the local device from an Infrared Data Application (IrDA) transceiver on the mobile device.

42. The method of claim 38, wherein the request for the data file is transmitted to the local device from a short-range Radio Frequency (RF) transceiver on the mobile device.

43. A communications system, comprising:

a wireless mobile device including a first transceiver and a second transceiver, wherein the first transceiver is configured to transmit requests for data files to a wireless communications network;

a local device including a transceiver configured to receive data files from the second transceiver of the mobile device; and

a remote server including an interface and a database comprising a plurality of data files, wherein the interface is configured to receive requests from a computer network that are routed from the wireless communications network.

44. The communications system of claim 43, wherein the mobile device comprises a memory buffer.

45. The communications system of claim 43, wherein the local device is a printer.

46. The communications system of claim 43, wherein the local device is a display.

47. The communications system of claim 43, wherein the local device is a media player.

48. The communications system of claim 43, wherein the local device includes memory for storing data files.

49. The communications system of claim 43, wherein the mobile device includes memory storing Uniform Resource Identifiers (URIs) associated with data files stored in the database of the remote server.

50. The communications system of claim 49, wherein each request for data files comprises at least one of the URIs.

51. The communications system of claim 45, wherein the remote server includes a processor configured to retrieve a data file from the database based on a URI in a received request.

ABSTRACT

Systems and methods for streaming data through a wireless mobile device are provided. The mobile device generates a “GET” request to pull data from a server connected to the Internet. The “GET” request includes a Uniform Resource Identifier (“URI”) identifying the location of the data on the server. The mobile device transmits the “GET” request to a mobile phone network. The “GET” request is routed from the mobile phone network to the server, which retrieves the data identified by the URI in the “GET” request. The server generates a response including the requested data. The response is routed to the mobile phone network, which transmits the response to the mobile device. Upon receiving the response, the mobile device generates a “PUT” request for pushing the data received in the response to a local device. The mobile device then transmits the “PUT” request including the data to the local device.