METHOD AND SYSTEM FOR NETWORK ACCESS OVER A LOW BANDWIDTH LINK

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to data communication networks. More particularly, the invention relates to transmitting information from a faster network to a data terminal via a slow network connection.

DESCRIPTION OF THE PRIOR ART

Electronic commercial and academic computer networks have been in existence for many years. An example of a commercial network is America On-Line. The largest and best-known electronic network is commonly known as the Internet. The Internet is a world-wide "network of networks" that is composed of thousands of interconnected computers and computer networks.

The preferred protocol of communications on the Internet is a set of standards and protocols, commonly referred to as TCP/IP. These TCP/IP protocols provide the means to establish a connection between two nodes on the network, and to subsequently transfer messages and data between these nodes. TCP (Transmission Control Protocol) provides the logical connection to ensure delivery of an entire message or file, while IP (Internet Protocol) provides the routing mechanism.

The majority of residential and commercial users of such computer networks (the clients) do not have a direct connection to the network. They subscribe to the services of an access provider, commonly called "Internet Service Provider" (ISP) in the case of the Internet. Clients use a personal computer or other terminal that is equipped with a data modem, to dial into the ISP connection service.

Fig. 1 is a diagram of the topology of an Internet connection via an ISP, according to the prior art. The ISP maintains a network 10 that connects its clients to the Internet 26. This network includes the ISP backbone 12, which is an internal set of connected nodes. ISP backbones are typically connected to points of presence, known as POPs 14, and management (control) centers 16. The POP is typically the local

exchange that users dial into via modem. There are usually few management centers and many POPs in an ISP backbone.

The ISP typically connects to the Internet via a Local Area Network (LAN) 28 at the management center. However, the ISP host may also be connected to the Internet via a Wide Area Network (WAN) such as the X.25 style Public Data Network (PDN). The LAN communicates with the Internet 26 through a physical, wide-bandwidth connection 30. However, the client does not connect to the LAN through such wide-bandwidth connection. Rather, the client accesses the Internet by using a computer 18 with a modem 20 to dial up, through the public telephone system 22, another modem in a local POP 24. This modem-to-modem connection is a relatively slow, low bandwidth two-directional link.

The ISP usually provides a relatively slow point-to-point (serial) two-dimensional link, through which the client communicates directly to the Internet, using the Internet standard TCP/IP protocols. Such serial link transfers digital data one bit after the other. The Recommended Standard-232 (RS-232) is the standard commonly followed for serial data transmission. (See, for example, *Electronic Industries Association*, EIA Standard RS-232-C, "Interface Between Data Terminal Equipment and Data Communications Equipment Employing Serial Binary Data Interchange", August 1969.)

Point-to-point links are among the oldest methods of data communications and almost every host supports point-to-point connections. A long-distance point-to-point link is achieved by using a modem to establish a dial-up link between a display terminal, such as a personal computer, and the other host, such as a host located on the ISP site.

The communication over the serial point-to-point line is in the form of encapsulated (framed) TCP/IP datagrams (data packets) using either Serial Line Internet Protocol (SLIP) or Point-to-Point protocol (PPP) packet framing.

The SLIP protocol is described in *IETF Network Working Group (J. Romkey)*, "A NONSTANDARD FOR TRANSMISSION OF IP DATAGRAMS OVER SERIAL LINES: SLIP ," Request for Comments: 1055 - June 1988. The PPP protocol is described in *IETF Network Working Group (D. Perkins)*, "The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams Over Point-to-Point Links," Request for Comments: 1171. SLIP transmits IP data packets over any serial link, such as a telephone line. It is generally used to provide Internet access to networks that support TCP/IP. PPP is a more common data link protocol that provides dial-up access over serial lines.

Fig. 2 is a diagram of a traditional Internet connection according to the prior art. To communicate on the Internet 26, an appliance or terminal (the client) 18 establishes a logical connection with a content provider. This connection is typically made using dial-up equipment, such as a modem 46 to connect to the public telephone system 22. The information transmitted over the telephone line is composed of PPP-wrapped TCP/IP data packets. Essentially, the ISP 10 is a transparent channel through which the client directly communicates with content providers, such as Web sites 51.

There are several disadvantages to this prior art approach. One major disadvantage is that of cost and complexity. Each terminal must be equipped with the ability to handle TCP/IP and PPP or SLIP protocols. This can require as much as 90 Kbytes of code and 70 Kbytes of data. Another major disadvantage is that of bandwidth. The Internet protocol overhead must be transmitted through the slow serial link, thereby reducing the actual useful bandwidth.

Yet another disadvantage is the need to duplicate, in each terminal, the code to deal with every possible type of data that may be transferred. Additionally, the established connection between the client and the Web server is recreated for every required file. This prevents a more general solution where commonly fetched files or messages are cached.

The World-Wide Web (Web) is an Internet client-server distributed information retrieval system. On the Web documents, menus, and indices are represented to the user as hypertext objects. Hypertext is a collection of documents containing cross-references or "links". These links enable the user, with the aid of an interactive browser program, to move from one document to another.

The Web may be accessed through other types of devices than a computer, including personal data assistants, fax machines, and Internet-capable telephones. One device that can provide Internet access is the terminal described in M. Valentaten, B. Moeschen, Y. Friedman, Y.-T. Sidi, Z. Bikowsky, Z. Peleg , *Multi-Mode Home Terminal System that Utilizes a Single Embedded General Purpose/DSP Processor and a Single Random Access Memory*, U.S. Patent No. 5,259,940 (October 5, 1993).

An internet access device, such as a modem-connected personal computer, generally uses a software application known as a Web browser to access the Web information available on the Internet. Such Web browsers, including Navigator, manufactured by Netscape Communications Corporation of Mountain View, California, and Mosaic, owned by the National Center for Supercomputing Applications (NCSA) at the University of Illinois, Urbana-Champaign, use a direct connection to the Web over SLIP/PPP. Thus, all of the overhead, in particular the parallel protocol overhead, is over the slow link.

To mitigate the above-mentioned disadvantages, prior art solutions that do not require the display terminal to handle the TCP/IP protocol set have been used to access the Internet. One such solution is for the user to acquire a shell account on the ISP host computer. With a shell account, a simple text transfer protocol and a terminal-emulator program are used to permit the user to communicate with the Internet through the display terminal, using a text-only Web browser program. One such text-only Web browser is Lynx, developed by the University of Kansas and currently maintained by Foteos Macrides at the Worcester Foundation for Biological Research. However, the shell account approach is subject to the major disadvantage that only plain-text information, and not images, colors, and sounds can be viewed.

Another prior art approach is that of the commercial Internet Web browser SlipKnot, developed by MicroMind, Inc. SlipKnot permits Web browsing through a serial link to a server, typically a Unix server, using a simple dial-in program. A TCP/IP stack is not required. When a user selects a hyperlink using SlipKnot, a textual description of the hyperlink is transferred to the ISP. SlipKnot uses the capabilities of the shell account to execute an "agent" program (such as Lynx) on the ISP machine. This agent fetches the requested multimedia file and uses a data-transfer program, such as xmodem, to transfer the file to the display terminal for display.

However, SlipKnot can be inconvenient to use and install. Furthermore, because the agent program is invoked separately for each separate request, information transfer is inefficient and slow. The SlipKnot method can handle only one request at a time and allows only local caching on the client.

It would therefore be an advantage to provide a method and system that improves the transfer of information from a faster network to a data terminal via a low-bandwidth link. It would be a further advantage if such method and system filters irrelevant Internet information prior to its transfer over the low-bandwidth link. It would be yet another advantage if such method and system reduces, if not entirely eliminates, protocol negotiation over the low-bandwidth link.

SUMMARY OF THE INVENTION

The invention provides a method and system for transmitting information from a faster network to a data terminal via a slower network connection. The preferred embodiment of the invention is adapted for use with any Internet terminal or access device, such as a telephone. A client connects to the Internet via an intermediary software program, known as the Gateway (GW). In the preferred embodiment of the

invention, the GW executes on a host computer of an ISP's Local Area Network (LAN.

The client dials up a Remote Access Server located at the ISP's local point-of-presence. The Remote Access Server communicates, via the ISP's backbone and using the TCP/IP network, with an authentication server. Initial authentication on the incoming calls is performed and the logical connection to the GW is established. The GW, in turn, uses the ISP's communication network and the TCP/IP protocol to connect to the Internet.

The GW thus mediates the data transfer between the Internet, such as the Web, and the client Internet terminal. The GW employs a point-to-point Internet protocol, the Gateway Interface Protocol (GWIP) to communicate with the client over the low-bandwidth link. The invention shifts the entire overhead of the Internet protocol stack to the GW, and does not involve the Internet terminal or the slow link between Internet terminal and GW. The Internet terminal needs no IP address. A single IP address is used by the GW to represent all Internet terminals.

The GW makes and negotiates multiple Internet requests, in parallel, for information to be fetched and loaded from the Internet using the GWIP protocol. The GW parses all requests and forwards them to the appropriate Internet server for execution. The GW filters each received file according to file type, and multiplexes the resulting data streams efficiently over the single link to the Internet terminal, based on the current priority of each stream. This allows documents to be loaded in parallel with their associated images, resulting in a much improved perceived speed. These streams can be paused and resumed as desired, according to a predetermined or dynamic priority. Such parallel retrieval of multiple objects is performed over the slow link without the use and overhead of Internet protocols.

The GW may also be used to conveniently customize the Internet terminal. In one embodiment of the invention, a profile of the Internet terminal user is stored in the Internet terminal. The GW uses this profile to provide customized services, such as sending only thumbnail views of images, or not sending certain material. In another embodiment of the invention, the GW serves as a software upgrade server. Using the GWIP protocol, a new firmware version can easily be uploaded, or a specialized device driver such as a printer driver uploaded as necessary. The invention may be used to upgrade or modify the graphical user interface of the Internet terminal.

In an alternative embodiment of the invention, the GW performs off-line services, such as collecting e-mail, or conducting intelligent off-line searches. The GW can be used to cache commonly used information fetched from the Internet. In

one embodiment of the invention, the GW is used as a mechanism for ISPs to track and bill customers for the use of the internet connection.

The invention is also readily adapted for use with Internet access devices that require different document formats and have different display capabilities, such as an Internet-compatible telephone, computer, a cellular phone, or a personal digital assistant having a wireless phone. In the preferred embodiment of the invention, the HTML layout is converted to a compressed equivalent of the HTML layout. Alternative embodiments of the invention do not convert the HTML layout, or convert the HTML to different screen formats.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a diagram of the topology of an Internet connection via an ISP according to the prior art;

Fig. 2 is a diagram of a traditional Internet connection according to the prior art;

Fig. 3 is a diagram of the GW architecture according to the invention;

Fig. 4 is a diagram of the topology of an Internet connection via an ISP according to the invention;

Fig. 5 is a diagram showing the topology of an Internet terminal connection to the Internet according to the invention; and

Fig. 6 is a flow chart of the filtering process according to the invention.

DETAILED DESCRIPTION OF THE INVENTION

The invention provides a method and system for transmitting information from a faster network to a data terminal via a slower network connection.

The preferred embodiment of the invention is adapted for use with the Web technique described above. However, alternative embodiments of the invention are adapted for use with any Internet access device. Thus, the following discussion is provided for purposes of example and not as a limitation on the scope of the invention.

The Web is usually accessed via telephone lines by a modem-connected computer, or Internet access terminal. The client dials up an ISP, for example using the PPP protocol. In turn, the ISP host establishes the connection to the Internet.

In the invention, however, the client connects to the Internet via an intermediary software program, known as the Gateway (GW). The GW mediates the

data transfer between the Internet, such as the Web, and the client computer/Internet access terminal. The GW employs a point-to-point Internet protocol, the Gateway Interface Protocol (GWIP) to communicate with the client over the low-bandwidth link. The invention thereby shifts the entire overhead of the Internet protocol stack to the GW, and does not involve the Internet terminal or the slow link between Internet terminal and GW.

Fig. 3 is a diagram of the GW architecture according to the invention. The invention framework includes a GW 42 that functions as an intermediary between client devices 30, such as an Internet-compatible telephone, and content providers, such as Web sites 51. Therefore, the GW, integrated into the ISP network, mediates access to the Internet on behalf of its clients.

Fig. 4 is a diagram of the topology of an Internet connection via an ISP according to the preferred embodiment of the invention. In the invention, the Internet terminal 30 uses the public telephone system 22 to dial up a local POP 24 within the ISP backbone 12. The GW, which is integrated into the ISPs network 10, mediates access to the Internet on behalf of its clients. The GW is a seamless software addition to standard ISP network configurations.

The ISP's E-mail servers 32, authentication servers 34, network management control 36, and proxy servers 38 are typically located at a management center 16. In the preferred embodiment of the invention, the GW 42 is also located at the management center. The GW preferably executes on a host computer of the ISP's Local Area Network (LAN) 40 with other computers and servers, such as the E-mail servers, network management servers, and authentication servers. However, the GW may also be executed by a dedicated server mediating between the LAN and the client.

The client uses the Internet terminal 30 to dial and connect to a dial-up Remote Access Server 46 which is located at the ISP's local POP 24. This Remote Access Terminal Server communicates, via the ISP's backbone 12 and using the TCP/IP network 44, with an authentication server 34. Initial authentication on the incoming calls is performed and the logical connection to the GW is established. The GW, in turn, uses a router 48 to connect to the Internet 26.

Fig. 5 is a diagram showing the topology of an Internet terminal connection to the Internet according to the invention. The Internet terminal 30 uses a modem 50 to dial-up a modem 52 at the ISP's local POP. This modem can be a hardware device that is located internal or external to the Internet terminal, or can be an integrated software modem. This modem 52 transmits information from the client to the GW 42 on the LAN. The GW uses Hypertext Transfer Protocol (HTTP) and TCP/IP protocol 60 to communicate, via a datalink 54 to the Internet 26. The Internet terminal

displays the information retrieved from the Internet on its display screen 56 using a browser application 58.

The GW executes on a host computer with a physical, wide-bandwidth connection to the Internet. However, the client computer does not have such a wide-bandwidth connection to the Internet. Rather, the client computer must connect to the Internet through a low-bandwidth slow link.

The GW employs an extremely simple point-to-point Internet protocol, referred to hereafter as the Gateway Interface Protocol (GWIP) to communicate with the Internet terminal over this relatively slow, low-bandwidth link. The GW then negotiates Internet requests on behalf of the Internet terminal over the fast, wide-bandwidth link. The function of the GW is, therefore, to serve as an agent or representative of the Internet terminal, and to negotiate Internet access on its behalf. As a result, the entire overhead of the Internet protocol stack is handled by the GW, and does not involve the Internet terminal or the slow link between Internet terminal and GW.

A request is a signal sent from a transmitting station to a receiving station requesting permission to transmit information. This permission is given with a response. The GW examines, pre-processes and modifies (filters) the Internet terminal's Internet requests and responses to improve performance and utility. The invention is particularly useful in the context of Internet display terminals, such as the Internet-compatible telephone, having a low bandwidth connection and limited storage capabilities.

A Web page is encoded in Hypertext Markup Language (HTML). An HTML document is a plain-text (ASCII) file that uses tags to denote the various elements in the document. An element may include an attribute, which is additional information that is included between tags.

HTML can be used to link text and/or images, such as icons, to another document or section of a document. The user activates a link by clicking on it, and the linked database is directly accessed. Links are used to access related information, or to contact a person or entity. However, information on a Web page must have the requisite HTML tags to be an active link.

In the invention, the client selects a hyperlink request for a text or image file on the Web. The Internet terminal forwards the request to the GW using the GWIP protocol. If the page in question contains embedded images, the Internet terminal forwards additional, parallel requests for the images as well.

The GW simplifies the transmission of requested Internet data, as compared to the prior art. Because the GW handles the protocol negotiation with the Internet data

server, it does not need to be negotiated over the slow link. The design of the Internet terminal may therefore be simplified and the amount of traffic over the slow link reduced. Experimental data has shown that the GW is approximately 20% more efficient than a prior art TCP/IP stack connection over PPP.

In the preferred embodiment of the invention, the GW also pre-processes, filters and simplifies the actual data by reducing the amount information that is irrelevant to the display environment that has been identified by the GW. This information can be reduced by such methods as stripping out information, optimizing, compressing, and by generally reducing size. Fig. 6 is a flowchart of the filtering process according to the invention. The client initiates a request to the GW for information from the Internet (100). The GW either identifies (105) or is pre-programmed with a client's attributes.

The GW then negotiates the request with the Internet (110) and retrieves the requested information (115). The GW filters the Internet information by reducing the amount of information that is not relevant to the client's attributes (120). The GW then sends to the client only that information which is needed (120). Therefore, bandwidth is not wasted on irrelevant or useless information and the client's processor does not have to process the unnecessary information. The size or power of the Internet terminal processor can therefore be reduced as compared to the prior art, resulting in cost and maintenance savings to the client.

For example, extra (redundant) color information is removed from images, or is reduced according to the physical characteristics of the display. Thus, if the receiving internet access terminal has a gray scale display, there is no need to transmit the color-related information over the slow link.

The GW reduces the amount of this irrelevant information in the HTML coding of a Web page before the page is transmitted over the low bandwidth link to the client. Data pages can also be stripped of redundant blank characters, or converted to a more efficiently compressed equivalent data format. This results in improved usage of the bandwidth of the slow link. Experimental data has shown that by pre-processing, filtering, and simplifying the actual data, the GW is approximately 10% more efficient than the prior art.

The invention uses a low-power compression/ decompression scheme, such as run-length encoding (RLE). This data compression scheme converts a "run" of identical characters into a code. Using this two-dimensional run-length compression, the GW compresses data on the fly without a lot of complex looking-back or looking-ahead. Such two-dimensional run-length compression looks in both vertical and horizontal directions at any given coordinate.

The GW makes and negotiates multiple Internet requests, in parallel, for information to be fetched and loaded from the Internet using the GWIP protocol. The GW parses all requests and forwards them to the appropriate Internet server for execution. The GW filters each received file according to file type, and multiplexes the resulting data streams efficiently over the single link to the Internet terminal, based on the current priority of each stream. This allows documents to be loaded in parallel with their associated images, resulting in a much improved perceived speed. These streams can be paused and resumed as desired. Such parallel retrieval of multiple objects is performed over the slow link without the use and overhead of Internet protocols.

The GW may also be used to conveniently customize the Internet terminal. In one embodiment of the invention, a profile of the Internet terminal user is stored in the Internet terminal. The GW uses this profile to provide customized services, such as sending only thumbnail views of images, or not sending certain material. In another embodiment of the invention, the GW serves as a software upgrade server. Using the GWIP protocol, a new firmware version can easily be uploaded , or a specialized device driver such as a printer driver uploaded as necessary.

Furthermore, the GW may be used with the integrated graphical user interface and slide-up window described in *Method And Apparatus For Organizing And Displaying Internet And Telephone Information* , also assigned to InfoGear Technology Corp. of Redwood City, CA, filed in the U.S. Patent and Trademark Office herewith and incorporated as a part hereof. The invention may be used to upgrade or modify the integrated GUI as described in the abovementioned application.

In an alternative embodiment of the invention, the GW is used as an off-line agent. Thus, in situations where the Internet terminal is connected via a link which is not always available, as with dial-up phone lines, the GW performs off-line services, such as collecting e-mail, or conducting intelligent off-line searches.

The GW can be used to cache commonly used information. A GW that serves multiple Internet terminals can cache the information fetched from the Internet. This reduces overall network access and improves responsiveness. For example, the GW can maintain a local name-to-IP address table.

In one embodiment of the invention, the GW is used as a mechanism for ISPs to track and bill customers for the use of the internet connection. For example, the GW can keep track of charges that the user accepts for the privilege of accessing certain copyrighted material.

The GWIP protocol is a high-level (session layer), full duplex, point-to-point data transfer protocol. It is used between a single client, such as the Internet-

compatible telephone, and a single server running the GW. The entire GWIP protocol may be encapsulated by lower-level network transport protocols, such as V42/V42.bis, X.25, or even PPP/TCP/IP. These encapsulating protocols may add compression, encryption and reliability, as required.

GWIP assumes that the underlying transport is error free. It is optimized to allow efficient asynchronous data transfer over a serial sequential medium, such as a relatively slow serial modem line.

The GWIP protocol consists of small data packets, typically about 250 bytes, which provide the capability for multiple interleaved data streams in both directions. There is no concept of "embedded," "escaped," or "out-of-band" control characters; all control information is in the form of complete GWIP packets. The GWIP described below is optimized for use with an Internet-compatible telephone. However, one skilled in the art will readily appreciate that the GWIP may be adapted to other Internet terminals, including different types of Internet-compatible telephones.

Table 1 describes the GWIP protocol structure. All GWIP packets start with a DLE character, followed by a byte, designated "len," with the length of the rest of the packet. The DLE character is a non-printable character that is used to indicate a beginning of a message. This convention provides a small measure of error-detecting redundancy over a normally error-free transport layer.

DLE	*len*	*t*	*data*

TABLE 1

The third byte of a GWIP packet ("*t* " in Table 1, above) always indicates the packet type. Additional data fields are optional. The packet type determines the packet destination and structure. Multiple-byte integer values are transmitted in network byte order (MOST significant byte first).

In most cases, the Internet terminal issues a request (a GWIP command), to which the GW responds by issuing one or more GWIP response packets. A request is always associated with a new stream. A stream comprises multiple packets that are logically associated via a common stream handle. A handle is an unsigned integer between 0 and 254 (255 is reserved). Responses use the same handle as the original request. Streams are sometimes also known as virtual channels. The GW may also issue spontaneous status and service messages that are related to one of the open streams.

A connection between the Internet terminal and the GW host (server) is always initiated by the client. As soon as a data connection is established, a configuration dependent exchange takes place between the Internet terminal and the GW host. This results in the establishment of a logical link between the Internet terminal and the GW software.

At this point, the GW waits for a special sequence of four bytes, that indicates the version of the GWIP protocol supported by the Internet terminal. The structure of this sequence is shown in Table 2, as follows:

DLE	*major*	minor	ETX

TABLE 2

The first byte of the sequence is the DLE character and the second byte is a (positive) major version number. The third byte is the minor version number, and the fourth byte the ETX character. ETX is a non-printable character that is frequently used to indicate the end of a message.

The discussion in this application describes an exemplary version of the GWIP protocol specification. One skilled in the art will readily recognize that other versions of the GWIP may be implemented in the invention. Thus, the following discussion is provided for purposes of example and not as a limitation on the scope of the invention.

The GW ignores all bytes that precede this version sequence. When it receives a valid version sequence, it returns a STAT_OK GW_STATUS packet (see below) to the Internet terminal and prepares to handle a regular GWIP session. The Internet terminal repeats the version sequence at, for example, 5-second intervals, until it receives the GW_STATUS packet. The Internet terminal disconnects if no valid response is received after three attempts. This disconnection resets the GW.

Unless specified differently, the word "command" indicates a packet that originates from the Internet terminal terminal; the word "response" refers to a GW initiated packet. types. Most GWIP packet types are used both as commands and response, but may have different formats.

Some commands, such as GW_LOGIN, GW_CONFIG and GW_ANCHOR, have additional command fields. Table 3 shows the GW_LOGIN command. Command fields are pairs of the field "name" (a one byte enumerator), followed by the field "value." Each command field is preceded by a one-byte length field, indicating the total length of the command field.

DLE	*len*	type	authentication fields

TABLE 3

Every GWIP session starts with a GW_LOGIN Internet terminal command. This authenticates the Internet terminal. The authentication includes the fields listed in Table 4:

field	na`me	value length (bytes)
Internet terminal serial ID (unique)	LOG_SERID	4
boot-section/Internet terminal version id	LOG_BOOTVER	4
firmware version id	LOG_FIRMVER	4
user name (id)	LOG_USER	string
password	LOG_PASSWD	string
Ö. .		

TABLE 4

The GW replies with a GW_STATUS response message, with a status STAT_OK, followed by an optional GW_SERVICE message. Any other response, or no response at all, causes the Internet terminal to disconnect.

The only valid commands after GW_LOGIN are GW_CONFIG, GW_LOGOUT, and GW_UPGRADE. The GW_LOGOUT command packet is shown in Table 5.

DLE	*len*	type

TABLE 5

The proper way for an Internet terminal to terminate a session is to send a GW_LOGOUT command, and then to wait for the GW_LOGOUT response. A session may also end as a

result of the Internet terminal disconnecting deliberately, or as a result of a communication fault or other error.

The GW_LOGOUT response to the Internet terminal GW_LOGOUT command is shown in Table 6. The log out message is optional. After receiving the GW_LOGOUT response, the Internet terminal disconnects from the line.

DLE	*len*	type	logout msg

TABLE 6

A GW_CONFIG command is sent by the Internet terminal to the GW immediately after the GW_LOGIN command. The GW_CONFIG command may be repeated at any time after the GW_LOGIN command. GW_CONFIG informs the GW about the user's preference/configuration table, including the language in which service messages must be sent, image preference options, and time out values. All values must be valid. The GW_CONFIG command is shown in Table 7.

DLE	*len*	type	PI_type	configuration fields

TABLE 7

The one-byte PI_type field determines the plug-in or GW sub-system that is concerned with these configuration values. Typically, the Internet terminal will send a number of GW_CONFIG commands, one for each type of sub-system or plug-in, whenever the configuration on the Internet terminal is changed by the user. Accepted values for this field include PI_EMAIL, PI_WEB and PI_GEN. The configuration fields listed in Table 8 are defined:

field	name	value length(bytes)	used by
font size	CONFIG_FONT	1	PI_GEN
image display method	CONFIG_IMG	1	PI_WEB
max image size to receive	CONFIG_MAX_IMG_SIZE	2	PI_WEB
cache refresh options	CONFIG_CACHE	4	PI_GEN
time out preference	CONFIG_TIMEOUT	4	PI_GEN
language	CONFIG_LANG	1	PI_GEN
date& time style	CONFIG_DATE	1	PI_GEN
email user id	CONFIG_EUSER	string	PI_EMAIL
email password	CONFIG_EPASSWD	string	PI_EMAIL
Ö.			

TABLE 8

The GW responds with a GW_CONFIG response to confirm receipt of the GW_CONFIG command. The Internet terminal does not issue any other type of commands before it receives the GW_CONFIG response. The GW_CONFIG command is thus a synchronization point between the Internet terminal and the GW.

A GW_ANCHOR command is an Internet terminal request for a new hypertext/image file. In an HTML document, the anchor is the format codes that are used to define a link to another page. The handle (*h*) field identifies the data stream and relates the request with subsequent GW response packets. The 2-byte "width" field indicates the width of the pane, in pixels, in which the information will be displayed. This command packet is followed by one or more GW_DATA packets that include the anchor Uniform Resource Locator (URL), and is terminated by a GW_EOF packet. The GW responds with a GW_ANCHOR response, followed by GW_DATA packets, terminated by GW_EOF. The GW_ANCHOR command is given in Table 9.

DLE	*len*	type	h	width

TABLE 9

A GW_REFRESH command is an Internet terminal request for the re-transmission of a hypertext/image document that is partially or fully in the Internet terminal cache. The handle (*h*) field identifies the data stream and relates the request with subsequent GW response packets. The 2-byte "width" field indicates the width of the pane, in pixels, in which the information will be displayed. THe GW_REFRESH command is shown in Table 10.

DLE	*len*	type	h	width	skip-len	creation date

TABLE 10

The 2-byte "skip-len" field indicates how many data bytes must be skipped by the GW before sending the actual data. This is useful if a file was partially received and cached by the Internet terminal in a previous transmission, and does not need to be fully re-transmitted.

The 6-byte "creation date" field indicates the creation date of the currently cached document. This command packet is followed by one or more GW_DATA packets that include the anchor URL, and is terminated by a GW_EOF packet.

The GW responds with a GW_ANCHOR response. If the file in the cache is up-to-date, the AN_CACHE_VALIDITY field of the GW_ANCHOR header will indicate that the cache is up-to-date (CACHE_VALID). Otherwise, the AN_CONTENT field will be CACHE_REPLACE or CACHE_DONT, and the value of skip-len will be ignored and the entire file will be re-transmitted. The GW_ANCHOR response will be followed by zero or more appropriate GW_DATA packets, and terminated by a GW_EOF.

A GW_POST command is a form-request, transmitted via the HTTP POST method. The handle (*h*) field identifies the data stream and optionally relates the request with GW response packets. The GW responds with a GW_ANCHOR response, with a AN_CACHE_VALIDITY field of value CACHE_DONT, followed by GW_DATA packets. Responses to GW_POST are never cached. This command packet is followed by one or more GW_DATA packets that include the anchor URL and the posted text, and is terminated by a GW_EOF packet. The GW_POST command is given in Table 11.

DLE	*len*	type	h	width

TABLE 11

The GW indicates its intention to send the requested GW_ANCHOR or GW-POST information by responding with a GW_ANCHOR response. A relatively long time may pass between the GW_ANCHOR or GW_POST command and the GW_ANCHOR response, due to network delays. The header information may include the creation data/time, content type, compression and display method. The GW_ANCHOR response is given in Table 12.

DLE	*len*	type	h	hdr info fields

TABLE 12

The following anchor fields are defined:

field	name	value length (bytes)
creation time-stamp	AN_CREATION	4 (time_t)
cache validity	AN_CACHE VALIDITY	1
content type	AN_CONTENT	1
Ö.		

TABLE 13

The GW then follows this with a sequence of GW_DATA response packets, terminated by a GW_EOF response. (Also see GW_ACK, described below). if a file cannot be retrieved, or if there is an unreasonably long time-out, the Internet terminal will usually cancel the GW_ANCHOR request via a GW_ENDSTREAM command.

GW_DATA packets are used to send data streams, such as bitmap data, from the Internet terminal to the GW, or vice-versa. Each GW_DATA packet contains a handle that relates this packet to the specific stream. Data streams are terminated by a GW_EOF packet.

An informational percent byte, with a value between 1 and 100, indicates an estimated percentage of how much has been transmitted, including the package. A percent of 0% indicates that the sender does not know, or does not care to compute, the estimated percentage. The GW_DATA packet is given in Table 14.

DLE	*len*	type	h	%	data...

TABLE 14

The Internet terminal acknowledges each received data-packet of type GW_DATA by sending a GW_ACK command. This allows the GW to send another packet, without overflowing the Internet terminal input buffers, and prevents unwanted data from piling up in internal UNIX (host) serial driver and modem buffers. The Internet terminal may also, at regular time intervals, send GW_ACK commands to the GW to indicate that it is alive. GW_DATA packets sent by the Internet terminal to the GW are not acknowledged by the GW. The GW_ACK command is shown in Table 15.

DLE	*len*	type	h

TABLE 15

The GW_PRI command is provided to the GW to recommend the priority order for the transmission of packets of concurrent existing streams. The priority is provided in "pri," a one-byte non-negative value. The GW does not echo these commands. Initially, a stream is of priority ten. The GW_PRI command is given in Table 16.

DLE	*len*	type	h	*pri*

TABLE 16

A priority of zero indicates that the stream is out of sight, This is usually true for images, when the user scrolls the screen. The GW sends data of all higher priority data streams first, in round-robin fashion, and only then sends data packets for lower priority streams. In the preferred embodiment of the invention, the values zero and ten only are used. However, alternative embodiments of the invention may use any appropriate priority values.

While the Web uses HTML coding for documents, the invention is also readily adapted for use with Internet access devices that require different document formats. In the preferred embodiment of the invention, the HTML layout is converted to an equivalent of the HTML layout. Alternative embodiments of the invention do not convert the HTML layout, or convert the HTML to different browser formats. For example, an HTML page can be converted into formats accessible by a Web telephone, a cellular phone, or a personal digital assistant having a wireless phone.

The layout conversion is performed, for example, by the server, client, Internet-capable telephone, Internet access device, or by a computer networked to the Internet-capable telephone or Internet access device. Further, such conversion may be an HTML-to-HTML conversion, for example as provided by a plug-in that operates in a connection with a conventional Web browser, such as Navigator, manufactured by Netscape Communications Corporation of Mountain View, California.

One embodiment of the invention is upgradeable via downloading from the Web. The HTML coding may thereby be readily modified to provide new features, update existing features, or to comply with different protocols. This HTML is then translated, as necessary.

The telephone numbers on a displayed Web page may be iconified and dialed, for example, as described in *Method And Apparatus For Iconifying And Automatically Dialing Telephone Numbers Which Appear On A Web Page*, also assigned to InfoGear Technology Corp. of Redwood City, CA, filed in the U.S. Patent and Trademark Office herewith and incorporated as a part hereof. These iconified telephone numbers may be added to an address book maintained as part of the invention.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention.

For example, while the Preferred embodiment of the invention is adapted for use with a serial link, the teachings of the invention are readily applicable to any type of slow link.

Accordingly, the invention should only be limited by the Claims included below.

CLAIMS

1. A method for accessing network information during a communications session, comprising the steps of:

accessing a host via a slow link using an Internet terminal;

the host negotiating an Internet connection on behalf of said Internet terminal with a software application integrated into said host; and

said software application using an interface protocol to transfer data from said Internet connection to and from said internet terminal, thereby mediating data transfer between the Internet and said Internet terminal.

2. The method of Claim 1, further comprising the step of said software application filtering said Internet data from said Internet connection to reduce the amount of data that is irrelevant to said Internet terminal's attributes.

3. The method of Claim 1, further comprising the step of said software application stripping redundant characters from said Internet data.

4. The method of Claim 1, further comprising the step of said software application converting said Internet data to a compressed equivalent data format.

5. The method of Claim 1, further comprising the steps of:

said software application using said interface protocol to make and negotiate multiple Internet requests, in parallel;

said software application parsing and forwarding said requests to an Internet server for execution;

said software application filtering data received from the Internet in response to said request and according to file type; and

said software application multiplexing said filtered data over said slow link to said Internet terminal.

6. The method of Claim 1, further comprising the steps of:

said Internet terminal storing a profile of a user of said Internet terminal; and

said software application using said profile to customize said Internet data sent to said internet terminal.

* C L M S *

7. The method of Claim 1, further comprising the step of said software application using said interface protocol to upload data to upgrade said Internet terminal.

8. The method of Claim 1, wherein said software application is an off-line agent for said Internet terminal.

9. The method of Claim 1, further comprising the step of said software application caching information commonly used by said Internet terminal.

10. The method of Claim 1, further comprising the step of said software application tracking charges accruing to said Internet terminal as a result of said Internet connection.

11. The method of Claim 1, wherein said Internet terminal is an Internet-compatible telephone.

12. A method for accessing network information during a communications session, comprising the steps of:

accessing a host via a slow link using an Internet terminal;

a software application integrated into said host negotiating an Internet connection on behalf of said internet terminal;

said software application using an interface protocol to transfer data from said Internet connection to and from said Internet terminal;

said software application using said interface protocol to make and negotiate multiple Internet requests, in parallel;

said software application parsing and forwarding said requests to an Internet server for execution;

said software application filtering data received from the Internet in response to said request according to file type;

said software application reducing the amount of said Internet data that is irrelevant to said Internet terminal's attributes; and

said software application multiplexing said filtered data over said slow link to said Internet terminal.

13. The method of Claim 12, further comprising the steps of:

said Internet terminal storing a profile of a user of said Internet terminal; and

said software application using said profile to customize said Internet data sent to said Internet terminal.

This Sheet Has Been Replaced

14. The method of Claim 12, further comprising the step of said software application using said interface protocol to upload data to upgrade said Internet terminal.

15. The method of Claim 12, wherein said software application is an off-line agent for said Internet terminal.

16. The method of Claim 12, further comprising the step of said software application caching information commonly used by said Internet terminal.

17. The method of Claim 12, further comprising the step of said software application tracking charges accruing to said user of said Internet terminal as a result of said Internet connection.

18. The method of Claim 12, further comprising the step of said software application stripping redundant characters from said Internet data.

19. The method of Claim 12, further comprising the step of said software application converting said Internet data to a compressed equivalent data format.

20. The method of Claim 12, wherein said Internet terminal is an Internet-compatible telephone.

21. A system for accessing network information during a communications session, comprising:

an Internet terminal for accessing a host via a slow link;and

a software application integrated into said host for negotiating an Internet connection on behalf of said Internet terminal;

wherein said software application uses an interface protocol to transfer data from said Internet connection to and from said Internet terminal.

22. The system of Claim 21, further comprising a filtering module in said software application for filtering said Internet data from said Internet connection to strip out data that is irrelevant to said Internet appliance's attributes.

23. The system of Claim 21, further comprising a stripping module in said software application for stripping redundant characters from said Internet data.

This Sheet Has Been Replaced

24. The system of Claim 21, further comprising a compression module in said software application for converting said Internet data to a compressed equivalent data format.

25. The system of Claim 21, further comprising:

a parallel request module in said software application for using said interface protocol to make and negotiate multiple Internet requests, in parallel;

a parsing module in said software application for parsing and forwarding said requests to an Internet server for execution;

a file type module in said software application for filtering data received from the Internet in response to said request according to file type; and

a multiplexer module in said software application for multiplexing said filtered data over said slow link to said Internet terminal.

26. The system of Claim 21, further comprising:

a storage module in said Internet terminal for storing a profile of a user of said Internet terminal; and

a customizing module in said software application for using said profile to customize said Internet data sent to said Internet terminal.

27. The system of Claim 21, further comprising an upgrade module in said software application for using said interface protocol to upload data to upgrade said Internet terminal.

28. The system of Claim 21, wherein said software application is an off-line agent for said Internet terminal.

29. The system of Claim 21, further comprising a cache in said software application for caching information commonly used by said Internet terminal.

30. The system of Claim 21, further comprising a tracking module in said software application for tracking charges accruing to said client as a result of said Internet connection.

31. The system of Claim 21, further comprising a stripping module in said software application for stripping redundant characters from said Internet data.

32. The system of Claim 21, further comprising a compression module in said software application for converting said Internet data to a compressed equivalent data format.

This Sheet Has Been Replaced

33. The system of Claim 21, wherein said Internet terminal is an Internet-compatible telephone.

34. An access terminal, comprising:

an Internet-compatible telephone for receiving and displaying a Web page; and

a software module integrated into an Internet Service Provider for connecting said Internet-compatible telephone to the Internet;

wherein said software module mediates data transfer between the Internet and said Internet-compatible telephone.



This Sheet Has Been Replaced



AMENDED CLAIMS

[received by the International Bureau on 24 August 1999 (24.08.99);
original claims 1-34 replaced by amended claims 1-25 (4 pages)]

1. A method for accessing network information during a communications session, comprising the steps of:

accessing a host via a slow link using an Internet terminal;

the host negotiating an Internet connection on behalf of said Internet terminal with a software application integrated into said host;

said software application using an interface protocol to transfer data from said Internet connection to and from said internet terminal, thereby mediating data transfer between the Internet and said Internet terminal;

said software application filtering said Internet data from said Internet connection to reduce the amount of data that is irrelevant to said Internet terminal's attributes;

said software application stripping redundant characters from said Internet data; and

said software application converting said Internet data to a compressed equivalent data format.

2. The method of Claim 1, further comprising the steps of:

said software application using said interface protocol to make and negotiate multiple Internet requests, in parallel;

said software application parsing and forwarding said requests to an Internet server for execution;

said software application filtering data received from the Internet in response to said request and according to file type; and

said software application multiplexing said filtered data over said slow link to said Internet terminal.

3. The method of Claim 1, further comprising the steps of:

said Internet terminal storing a profile of a user of said Internet terminal; and

said software application using said profile to customize said Internet data sent to said internet terminal.

4. The method of Claim 1, further comprising the step of said software application using said interface protocol to upload data to upgrade said Internet terminal.

5. The method of Claim 1, wherein said software application is an off-line agent for said Internet terminal.

This Sheet Has Been Replaced



6. The method of Claim 1, further comprising the step of said software application caching information commonly used by said Internet terminal.

7. The method of Claim 1, further comprising the step of said software application tracking charges accruing to said Internet terminal as a result of said Internet connection.

8. The method of Claim 1, wherein said Internet terminal is an Internet-compatible telephone.

9. A method for accessing network information during a communications session, comprising the steps of:

accessing a host via a slow link using an Internet terminal;

a software application integrated into said host negotiating an Internet connection on behalf of said internet terminal;

said software application using an interface protocol to transfer data from said Internet connection to and from said Internet terminal;

said software application using said interface protocol to make and negotiate multiple Internet requests, in parallel;

said software application parsing and forwarding said requests to an Internet server for execution;

said software application filtering data received from the Internet in response to said request according to file type;

said software application reducing the amount of said Internet data that is irrelevant to said Internet terminal's attributes;

said software application multiplexing said filtered data over said slow link to said Internet terminal;

said software application stripping redundant characters from said Internet data; and

said software application converting said Internet data to a compressed equivalent data format.

10. The method of Claim 9, further comprising the steps of:

said Internet terminal storing a profile of a user of said Internet terminal; and

said software application using said profile to customize said Internet data sent to said Internet terminal.

This Sheet Has Been Replaced



11. The meth of Claim 9, further comprising the step of s. software application using said interface protocol to upload data to upgrade said Internet terminal.

12. The method of Claim 9, wherein said software application is an off-line agent for said Internet terminal.

13. The method of Claim 9, further comprising the step of said software application caching information commonly used by said Internet terminal.

14. The method of Claim 9, further comprising the step of said software application tracking charges accruing to said user of said Internet terminal as a result of said Internet connection.

15. The method of Claim 9, wherein said Internet terminal is an Internet-compatible telephone.

16. A system for accessing network information during a communications session, comprising:

an Internet terminal for accessing a host via a slow link;and

a software application integrated into said host for negotiating an Internet connection on behalf of said Internet terminal;

wherein said software application uses an interface protocol to transfer data from said Internet connection to and from said Internet terminal;

a stripping module in said software application for stripping redundant characters from said Internet data; and

a compression module in said software application for converting said Internet data to a compressed equivalent data format.

17. The system of Claim 16, further comprising a filtering module in said software application for filtering said Internet data from said Internet connection to strip out data that is irrelevant to said Internet appliance's attributes.

18. The system of Claim 16, further comprising:

a parallel request module in said software application for using said interface protocol to make and negotiate multiple Internet requests, in parallel;

a parsing module in said software application for parsing and forwarding said requests to an Internet server for execution;

This Sheet Has Been Replaced

a file type module in said software application for filtering data received from the Internet in response to said request according to file type; and

a multiplexer module in said software application for multiplexing said filtered data over said slow link to said Internet terminal.

19. The system of Claim 16, further comprising:

a storage module in said Internet terminal for storing a profile of a user of said Internet terminal; and

a customizing module in said software application for using said profile to customize said Internet data sent to said Internet terminal.

20. The system of Claim 16, further comprising an upgrade module in said software application for using said interface protocol to upload data to upgrade said Internet terminal.

21. The system of Claim 16, wherein said software application is an off-line agent for said Internet terminal.

22. The system of Claim 16, further comprising a cache in said software application for caching information commonly used by said Internet terminal.

23. The system of Claim 16, further comprising a tracking module in said software application for tracking charges accruing to said client as a result of said Internet connection.

24. The system of Claim 16, wherein said Internet terminal is an Internet-compatible telephone.

25. An access terminal, comprising:

an Internet-compatible telephone for receiving and displaying a Web page;

a software application integrated into an Internet Service Provider for connecting said Internet-compatible telephone to the Internet;

wherein said software application mediates data transfer between the Internet and said Internet-compatible telephone;

a stripping module in said software application for stripping redundant characters from said Internet data; and

a compression module in said software application for converting said Internet data to a compressed equivalent data format.

This Sheet Has Been Replaced