

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
7 February 2002 (07.02.2002)

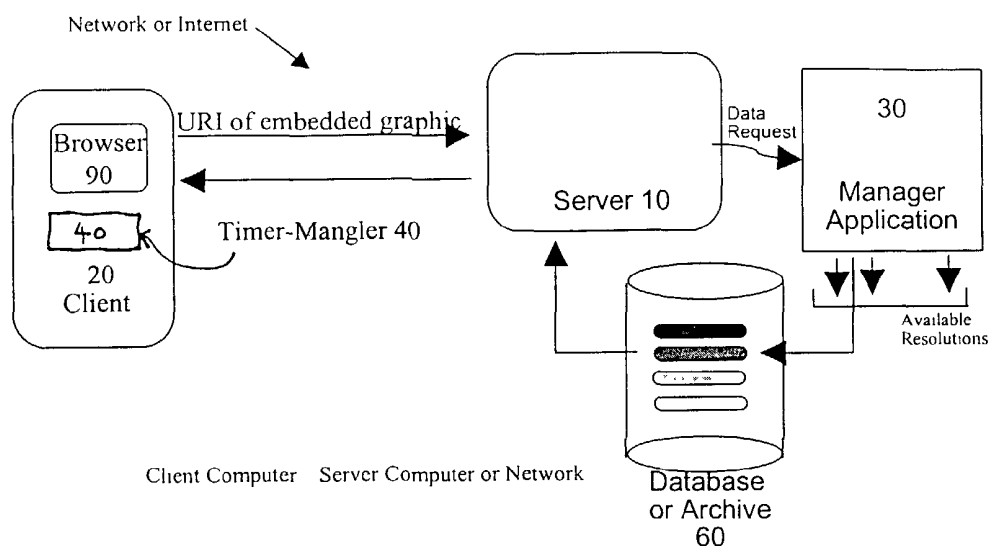
PCT

(10) International Publication Number
WO 02/10943 A1

- (51) International Patent Classification⁷: G06F 15/16
- (74) Agents: CHANG, Ya-Chiao et al.; Darby & Darby P.C., 805 Third Avenue, New York, NY 10022-7513 (US).
- (21) International Application Number: PCT/US01/23902
- (81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.
- (22) International Filing Date: 30 July 2001 (30.07.2001)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data: 09/628,593 28 July 2000 (28.07.2000) US
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- (71) Applicant (for all designated States except US): DAVIS ENGINEERING [US/US]; 225 Kensington Way, San Francisco, CA 94127 (US).
- (72) Inventor; and
- (75) Inventor/Applicant (for US only): DAVIS, Emery [US/US]; 225 Kensington Way, San Francisco, CA 94127 (US).
- Published: — with international search report

[Continued on next page]

(54) Title: ADAPTIVE DOWNLOADING TECHNOLOGY



(57) Abstract: A downloading timer-mangler application (40) at the client (20) and a manager application (30) at the server (10) measuring the bandwidth between the server and client to select an appropriate image resolution for the requested information to ensure a generally constant page impression rate. The system allows for the adaptive downloading of a web page displaying information that includes an embedded resource of a third party server such as banner ad. If the third party server is a cooperating server, then the timer-mangler application appropriately selects the image resolution in conjunction with that manager application. If the third party server is a non-cooperating server, the timer-mangler application replaces the third party IP address with that of the alias server. The virtual queue stores the requested information of the third party server in a plurality of image resolutions. The alias server then accesses the virtual queue for the appropriate image resolution.

WO 02/10943 A1



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

5

10

ADAPTIVE DOWNLOADING TECHNOLOGY

FIELD OF THE INVENTION

The invention generally relates to downloading information from the Internet, and more particularly, to an adaptive downloading technology for downloading information from a server at a website to a client over the Internet.

15

DESCRIPTION OF THE RELATED ART

The Internet has become a popular venue for gathering and exchanging information for research, entertainment or business. An increasing number of websites are coming online over the Internet for communicating information to end users or customers. HTML, or hypertext markup language, is the basic mode for storing information or documents on the server(s) for the websites accessible by online end users. In particular, hypertext is textual material with pointers to other text. By activating the pointers (e.g., by pointing a cursor and clicking) the supporting text is presented. An HTML document can include graphics, text, sound, or video links to other HTML documents on third party servers. The websites in a client-server architecture transfer the HTML document or information stored in the server, e.g., mainframe or personal computer (PC) servers, to the client at the end user location, e.g., personal computers, web television or appliances. The client-server architecture, a basic concept in computer networking, allows servers to retrieve information requested by clients who in turn

20
25
30

display that information to the end user. An end user at the client runs a client program called a browser, such as Netscape Navigator[] or Internet Explorer[], that communicates with a server program running on the server of the website. Through the client program or browser, the end user sends a user request to the server of the website. The server then
5 handles the request and sends a response to the end user. The information at the website is generally displayed in web page form in HTML, for viewing or browsing by the end user. Using a computer keyboard, mouse, voice activated input device, or other input devices, the end user(s) can choose to go to a certain page for viewing as desired. A typical web page contains zero to one megabyte of downloadable graphical information,
10 depending on the density of the graphics content in the web page. As the end user selects a page for viewing, the information of that page is impressed and downloaded to the client at the end user location. The browser at the client then displays the requested information for viewing by the end user.

Ideally, the downloading of information from the server to the client at the
15 end user location should be as quick and efficient as possible. However, many factors in existing Internet application environments hinder expedient downloading. One such factor is the slow speed of the Internet link at the client, e.g, a slow 56 kilo-byte modem link. Because of the relatively large sizes of graphics files, downloading of web pages is even more time consuming when the pages have heavy graphic content.

20 Therefore, there is a need in the art for a speedy and efficient system and method for downloading information in a computer network or over the Internet.

SUMMARY OF THE INVENTION

The adaptive downloading technology according to the present invention
25 provides a downloading timer-mangler application at the client and a manager application at the server for downloading information over the Internet or another network. The timer-mangler application may be resident at the client, or be downloaded to the client as the end user requests information from the website at the server. The timer-mangler application monitors the timing information, e.g., the bandwidth between the server and
30 client for a download of information. Based on the timing information, the timer-

mangler application, in conjunction with the manager application, adaptively selects an appropriate image resolution for the requested information to ensure a generally constant page impression rate, i.e., a generally constant timing rate for a download of a web page. Using the selected image resolution, the information requested by the client is
5 downloaded to the client.

The timer-mangler application also supports network caching to reduce network traffic between the server and the client over the Internet or another network. The timer-mangler application takes the URL (uniform resource locator) or URI (uniform resource identifier) addressing the requested information and undertakes object renaming,
10 i.e., renames the URI to represent different resolutions of the requested image. Because each resolution or image object is unique to the server, the adaptive downloading technology according to the invention operates in conjunction with existing distributed caching resources, which avoids unnecessary additional transmissions between the server and client.

15 In addition, the adaptive downloading technology according to the invention provides adaptive downloading for a web page displaying information that includes an embedded resource of a third party server, such as a third party banner ad. If the third party server is a cooperating third party server, i.e., a server that includes a manager application functionally equivalent to the manager application at the server, then
20 the timer-mangler application appropriately selects the image resolution in conjunction with that manager application to ensure a generally constant page impression rate. That is, the manager application ensures a generally constant page impression rate, and the timer-mangler application undertakes object renaming but addresses the renamed object within the IP domain of the third party server. The cooperating third party server then
25 adaptively provides the appropriate image resolution for the requested information to ensure the generally constant page impression rate.

If the third party server is a non-cooperating third party server, i.e., a server that does not have a manager application, then an alias server and a virtual queue of images at various resolutions are provided at the server. In the server, an alias server
30 with an Internet protocol (IP) address known to the timer-mangler and the manager

applications is provided for the third party server. The alias server serves as an addressing substitute as the timer-mangler addresses the third party server. The timer-mangler application replaces the third party IP address with that of the alias server while it undertakes object renaming. The virtual queue is provided to store the requested information of the third party server in a plurality of image resolutions. The first-in-first-out (FIFO) nature of the virtual queue ensures that the order of images accessed from the alias server remains identical to that which would have been accessed from the third party server. The calculation of the various image resolutions is not done during periods of peak server activity, which significantly reduces the server load during those periods. Because the alias server is located in the same logical domain as the server, additional control is provided over the network routing and overall server performance. The alias server then accesses the virtual queue for the appropriate image resolution, as determined by the timer-mangler and manager applications, corresponding to the requested information of the third party server, which ensures a generally constant page impression rate.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other features, aspects, and advantages of the invention will become more readily apparent with reference to the following detailed description of a presently preferred, but nonetheless illustrative, embodiment when read in conjunction with the accompanying drawings. The drawings referred to herein will be understood as not being drawn to scale, except if specifically noted, the emphasis instead being placed upon illustrating the principles of the invention. In the accompanying drawings:

Figures 1 and 1A are diagrams that respectively illustrate the method and system according to the invention in general;

Figure 2 illustrates exemplary functional relationships between graphics resolution and bandwidth as configurable to ensure a constant page impression rate according to the invention;

Figure 3 is a block diagram illustrating the timer-mangler application according to the invention in further detail;

Figures 3a and 3b illustrate exemplary methods whereby the timer-mangler application intercepts and processes incoming data from, and outgoing data to, the network according to the invention;

Figure 3c illustrates an exemplary TCP/IP stack for Windows™ applications;

Figure 4 is a block diagram that illustrates the operation of the manager application according to the invention in further detail;

Figure 5 is a block diagram that illustrates the operation of the manager application according to the invention with a JPEG 2000 database or archive;

Figure 6 is a block diagram that illustrates the operation of the adaptive downloading technology according to the invention in conjunction with a non-cooperating third party server;

Figure 7 is a flow diagram that illustrates the adaptive downloading technology according to the invention in further detail;

Figures 7a and 7b respectively illustrate exemplary methods of operation for the virtual queue and alias server according to the invention; and

Figure 8 is a block diagram that illustrates another embodiment of the invention with multiple third party servers.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The general method and system according to the invention are respectively illustrated in Figures 1 and 1A. A server 10 storing information for a website is accessible to a plurality of clients (not entirely shown in Figure 1A), and particularly client 20, connected thereto over the Internet or a network such as LAN (local area network), WAN (wide area network), MAN (metropolitan area network). The adaptive downloading technology according to the invention provides a downloading timer-mangler application 40 at the client 20 and a manager application 30 at the server 10 for downloading information over the Internet or another network. Server 10 and client 20 are general purpose computers. For example, server 10 can be a mainframe, workstation or a personal computer (PC). Client 10 can be a PC, web television or

appliance, personal data assistant or pager or telephone with Internet access. In step 101, an end user at the client 20 requests information from the website, e.g., by activating a voice input device, clicking on an icon in the web page being displayed in browser 90 or pressing certain keys on a keyboard (not shown). For commercially available websites, the information requested by the client 20 generally include images or graphic objects. Graphic objects include pictures or data files that are referenced by the source (HTML, or SMIL (Synchronized Multimedia Integration Language) or XML (Extensible Markup Language), or other source) of a web page via a reference locator or reference indicator, such as URL (uniform resource locator) or URI (uniform resource indicator) which is a universal naming scheme of the World Wide Web for identifying and addressing documents and other data and image sources on the Internet. When the end user at the client 20 requests display of a web page on the browser 90, the page source references the embedded graphic object by name, which are in turn requested from the server 10 (or its network proxies, if applicable). Note that "page source" is often referred to herein as the "HTML source", but the term encompasses any source such as HTML, XML, SMIL, or any other type of page source that is interchangeable in this context. The adaptive downloading technology according to the invention described herein is not limited to HTML page implementations, even though for brevity in the figures and the descriptions herein HTML alone is used to represent a page source.

The timer-mangler application may be resident at the client 20, or be downloaded to the client 20 as the end user requests information from the website at the server 10. In step 103, it is determined whether the timer-mangler application is resident at the client 20. This is accomplished by examining the HTTP, WAP (wide area protocol) or other protocol headers appropriately marked to indicate the presence of the timer-mangler application, or by downloading an applet (that is a browser applet) to determine the presence of the timer-mangler application as an applet, plugin or in some other form at the client 20, or by other means known to the art but not materially effecting the present invention per se. If it is determined that the timer mangler application is not resident at the client 20, the timer-mangler application 40 is downloaded to the client 20 (step 105) and the control flow goes to step 107. If the

timer-mangler application is resident at the client 20, the control flow goes directly to step 107.

In step 107, the timer-mangler application 40 monitors the timing information, e.g., the bandwidth from the server 10 to the client 20, i.e. a bandwidth measurement is a point-to-point measurement between the client 20 and the server 10. The timing information is typically not directly communicated to the server or manager, only indirectly as discussed below with respect to object name mangling. The server 10 and client 20 are connected to the network using links such as a wireless Internet link, T1 line, digital subscriber line (DSL), telephone or cable modem. The timer-mangler application 40 is typically implemented in an applet language such as JAVA or Javascript, if the timer-mangler application is downloaded, but may also be resident on the client 20 in some other form. An applet is a software program that can be included in a web or HTML (hypertext markup language) page, similar to the way a graphic image is included therein. When an end user at a client (such as client 20) uses browser 90 (such as a Netscape Navigator[®] or Internet Explorer[®] browser) to view a web page that contains an applet, the applet's program code is transferred to the client and executed by browser 90. The timer-mangler application applet 40 is thus downloaded to client 20 where it is executed by a browser 90 program resident in client 20. The timer-mangler application 40 can also be embedded as a component of individual web pages by utilizing features of the website protocol, such as HTTP (hypertext transfer protocol) of the World Wide Web (WWW).

Based on the timing information known to and measured by the timer-mangler application 40, in conjunction with the manager application 30, and based on the information provided by the manager application 30 for determining the desired constant page impression rate, an appropriate image resolution is adaptively selected for the requested information to ensure a generally constant page impression rate, i.e., a constant timing rate for downloads seen on a per page basis by the client 20. Typically, the bandwidth measure consists of a value in terms of bytes divided by download transfer time. If a new web page represents a stand-alone graphic (e.g., a click-through thumbnail graphic which can be enlarged by clicking thereon), the timer-mangler application selects

the image resolution such that the graphic is downloaded at the generally constant page impression rate. Based on the timing information, the timer-mangler application 40, in conjunction with the manager application 30, adaptively selects an appropriate image resolution for the requested information (step 109) to ensure a generally constant page impression rate, i.e., a generally constant timing rate for a download. The generally constant page impression rate can be a predetermined value set at server 10.

To ensure a generally constant timing rate for a download, the adaptive downloading technology according to the invention performs a transformation of eligible graphics objects addressing the requested information or web page:

$$G \rightarrow G^r \quad (\text{Eq. 1})$$

where G represents the source referenced graphics object (URI or URL) and the superscript r indicates the modified resolution. For every source referenced graphics object, the server 10 has a set of graphics objects $\{G^r\}_{r=0,n}$ available at the database or archive 60. This set $\{G^r\}_{r=0,n}$ may involve storage at a plurality of image resolutions in the archive 60 or may be calculated on an as-needed basis by the manager application 30. Typically, G^n is the graphics object with maximum resolution, and G^0 the graphics object with minimum resolution, with the intervening G^r 's chosen according to a predetermined function defined at the server 10. In an embodiment according to the invention, the predetermined function is $G^r = 2 * G^{r-1}$, among other possible functions of resolution versus bandwidth that may be configured on the server 10. The total number in the set $\{G^r\}_{r=0,n}$ is also server-side definable. Figure 2 illustrates three possible functional definitions for the set $\{G^r\}_{r=0,n}$ that can be used in implementing the adaptive downloading technology according to the invention. The spacing between each version of the graphics object in terms of image resolution (i.e., the increase in abscissa for each G) is constant. "Size" in Figure 2 refers to the file size or number of bytes for the requested information to be transmitted over the Internet or the network, which is a measure of image resolution. "Version" in Figure 2 denotes the number of available versions for the graphics object, and can be interchanged with "bandwidth" according to a piecewise functional definition.

The archetypal graphic G , pre-existing at the website, and forming the basis of the set $\{G^r\}_{r=0,n}$ need not correspond to the highest available resolution G^n . By allowing higher resolutions as bandwidth conditions permit, the adaptive downloading technology according to the invention allows existing websites to scale for future network infrastructure improvements as and when they arrive, without penalizing existing lower bandwidth clients.

Although the adaptive downloading technology according to the invention typically varies the resolution of images without changing their on-screen dimensions, the manager application 30 can also be configured to scale down (or up) the on-screen dimensions of graphics while varying the resolution, with respect to the device type of the client as communicated in the communication protocols. This allows a pre-existing site to support a plurality of client devices without the creation of special graphics to support each client type.

To reduce server load or data overhead at the server 10, copies of multiple resolutions of the requested information can be, if necessary, generated in advance and stored in the database or archive 60 at the server 10, i.e., embedded in the server 10. In an SQL (structured query language) environment, manager application 30 intercepts user data requests for information from server 10, and translates them into requests for currently appropriate data with a corresponding image resolution. The manager application 30, as dictated by the timer-mangler application 40 modifies the SQL requesting the information to an SQL request for the information in a corresponding resolution stored in a database or archive 60 at server 10. SQL, known in the art, allows end users to access information in a relational database management system (RDMS), such as Oracle[], Sybase[], Informix[], Microsoft SQL Server[], Access[], and others, by allowing end users to describe the requested information, as well as define and manipulate the requested information in the database. In an environment where the server 10 reads directly from an archive, the manager application 30 intercepts the read command, for example a read command is a file from a disk, and modifies the read command to a command to read data at the corresponding resolution as dictated by the timer-mangler application 40.

Referring again to Figures 1 and 1A, the end user at client 20 can optionally select a specific image resolution as desired from a plurality of options of image resolutions displayed at browser 90 at client 20 (step 111). The options of image resolutions can be displayed at client 20 in a selection menu. For example, the selection menu displayed in browser 90 can be a text- or icon-based slide bar showing poor image resolution to the left and gradually increasing to full image resolution to the right. An exemplary selection menu shows four or more different levels of JPEG compression, which is a graphical file format and compression scheme developed by the Joint Photographic Experts Group, with varying image resolutions from 5 to 250 kilobytes. Other file formats available for implementing the adaptive downloading technology according to the invention include TIFF (tagged image file format), PNG (portable network graphics), BMP (Windows™ bitmap), GIF (graphical interface format), Macintosh™ PICT format, and many others known in the art. Another file format compatible with the adaptive downloading technology according to the invention is JPEG 2000 International Standard (or IS15444-1), as well as other incremental or data stream-based image compression formats, such as the FlashPix format. Additionally, each embedded graphic may be shown as an HTML link to the next lower or higher image resolution. In step 113, if the end user overrides manager application 30, the requested information is downloaded to client 20 with the specific image resolution selected by the end user. If the end user chooses not to override the manager application 30, then the requested information is downloaded to client 20 from server 10 with the image resolution selected by timer-mangler application 40 and forwarded to the manager application 30 (step 115).

Figure 3 is a block diagram that illustrates the timer-mangler application according to the invention in further detail. The timer-mangler application 40, which is the client-side component of the adaptive downloading technology according to the invention, includes two primary functions. The first is to measure the current network bandwidth conditions as experienced by the end user at the client 20. This is a point-to-point bandwidth measurement between the server 10 and the client 20, and as such reflects effects on bandwidth conditions, such as server load. The second is to transform

the names of graphics objects (i.e., name mangling) in selecting appropriate image resolutions to ensure a generally constant page impression rate for a download. Since most of the timer-mangler application 40 resides (which can be downloaded or embedded) at the client 20, only client-side details are shown in Figure 3.

5 Bandwidth monitoring by the timer-mangler application provides timing information as the basis for selecting an image resolution for a download to ensure a generally constant page impression rate. In the present embodiment, the timer-mangler application 40 is implemented in JAVA[], Javascript, ActiveX[] or some other applet language. The timer-mangler application can also be embedded at the client 20 as a
10 component of individual web pages, provided as a plug-in (i.e., externally supplied user agent) for the browser 90, or as a separately downloadable applet to avoid multiple downloads of the same applet. When the timer-mangler application 40 is an applet, it is downloaded to the client 20 from the server 10 at the beginning of a session, unless it already resides at the client 20. The term "session" herein refers to the first of a series of
15 page downloads by the client 20 from the server 10. The size of the timer-mangler application is relatively small which does not add burdensome overhead when it is a downloadable applet.

 Name mangling by the timer-mangler application 40 generally eliminates cache aliasing, as well as providing a method for the timer-mangler application 40 to
20 communicate the desired resolution to the manager application 30. Cache aliasing is when network caches become confused when different graphics objects, for example the same object represented by multiple different resolutions, have the same names. Network caches are useful in their ability to cache the files as they are retrieved. Subsequent requests for the same files are met by retrieving from the network caches rather than from
25 the remote server, thereby avoiding unnecessary traffic over the Internet or the network, as well as additional server load. The timer-mangler application 40 generally avoids cache aliasing by giving each of the graphics objects in the set {G} a unique name. Because each eligible graphics object is provided a uniquely mangled name, the adaptive downloading technology according to the invention optimally utilizes and operates in
30 conjunction with the network caches, such as Server RAM[], Akamai[]-type server

5 caches, proxy server caches or distributed caches such as those provided by the vendors Inktomi™, InfoLibria™ and others. A proxy is a program that acts as an agent for its clients, making requests on their behalf. In the case of an Internet proxy server, the clients are the browsers, other Internet servers and client programs. Proxies are useful where the browser programs cannot access remote servers, typically when direct connections between internal and outside computers are prohibited.

The timer-mangler application 40 also de-mangles names of graphics objects as they arrive at the client 20, so that HTML integrity is preserved within the browser 90. Accordingly, the timer-mangler application 40 supports network caching to 10 reduce network traffic between the server 10 and the client 20 over the Internet or another network. The timer-mangler application 40 takes the URI (uniform resource identifier) addressing the requested information and undertakes object renaming, i.e., renames the URI to represent different resolutions of the requested image, in this fashion communicating the desired resolution to the manager application 30 on the server 10. 15 Because each resolution or image object is uniquely named, the adaptive downloading technology according to the invention operates in conjunction with existing distributed caching resources, which avoids unnecessary additional transmissions between the server 10 and client 20. In this manner, the invention advantageously overcomes the problem where several resolutions of an embedded object "G" are all represented by the same name (i.e., "G") in which a distributed cache that is asked to serve 'G' may respond with 20 a version that has a resolution (i.e., size) which does not ensure the generally constant page impression rate, and may be a greater or lesser resolution than that selected by the timer-mangler application based on the information provided by the manager application.

As shown in Figure 3, the URI or URL of an embedded graphics object in 25 the HTML source 310 for a web page being displayed at the browser 90 is called G. The timer-mangler application transforms the name of the graphics object according to current bandwidth conditions to obtain a URI or URL designated as G^{n-1} chosen from the set of resolutions $\{G\}_{0,n}$ which is in turn sent to the server 10. When G^{n-1} arrives in response, the timer-mangler application 40 reverse transforms it back into G (if required

by the browser 90) and passes the response along to the browser 90 for display or additional processing.

The control block 320 receives information on a per page basis from the manager application 30 at the server 10. The control block 320 handles the decisions regarding the timing information and name mangling, e.g., which graphics objects (URI or URL) are eligible for name mangling, and hence the adaptive downloading according to the invention, and the options in terms of image resolution that are available from the server 10.

In some cases, the HTML source 310 may have multiple embedded images that comprise a single larger image, where the graphic representation seen by the client 20 is made up of multiple image tiles, each tile being a separate graphics object. The graphic representation is adversely affected if the multiple image tiles are downloaded at varying image resolutions. The control block 320 receives control information 322 from the manager application 30 as to which embedded graphics are image tiles, as well as an identifier (e.g., URI) for each group of image tiles. The control information 322 is relatively small and generally compressed, which does not add significant burden to the download of the requested information. If the bandwidth changes sufficiently, the timer-mangler application 40 requests image tiles with different resolutions, and then reloads the previous image tiles already loaded at another resolution.

The timer block 330 measures current bandwidth conditions, with a granularity determined by the control block 320 as dictated by the server 10 via control information 322. Granularity is the measuring rate per page, or per embedded graphic, or amount of time which is server-side defined. In general, the bandwidth measure consists of a single number in a form such as bytes received divided by time:

$$B_i = (\delta b_i) / (\delta t_i) \quad (\text{Eq. 2})$$

with

$$\delta b_i = b_i - b_{i-1} \text{ and } \delta t_i = t_i - t_{i-1}$$

where B represents bandwidth, δb_i is the number of bytes (or packets of data) received since the last bandwidth measurement and δt_i is the granularity. When the time t between

measurement samples is non-constant, the subscript i represents the sample number. When the granularity δt_i is constant, the subscript i represents both the sample number and the specific time in which an integral increase in i is equal to a temporal increase in granularity.

5 Aside from monitoring the bandwidth conditions between the server 10 and the client 20, bandwidth prediction is also used. Depending on the granularity, different methods of bandwidth prediction can be utilized. The appropriate bandwidth prediction method, as configured by the server 10, is downloaded to the client 20 with the timer-mangler application 40.

10 When the granularity is large and δt is non-constant, on the same order as the time required to download an embedded graphic, or even an entire page, it is sufficient to predict that:

$$B_{i+1} \approx B_i \quad (\text{Eq. 3})$$

thereby replacing the current bandwidth B_i with the new measurement.

15 When the bandwidth measurement is more frequent, e.g., on an order of less than several seconds, a more calculated approach in bandwidth prediction is taken, whether the granularity is constant or not. A weighted average which considers several recent B_i values is used. For example, previous three bandwidth measurements are considered:

20
$$B_{i+1} \approx [3B_i + 2B_{i-1} + B_{i-2}] / 6 \quad (\text{Eq. 4})$$

which can be generalized for bandwidth measurement intervals (α) as follows:

$$B_{i+1} \approx \frac{\sum_{k=0}^{a-1} (a-k) \cdot B_{i-k}}{\sum_{k=1}^a k} \quad (\text{Eq. 5})$$

where B_{i+1} is a local extrapolation using a finite number of nearest neighboring points, and the predicted value is replaced with the measurement for the next iteration of Eq. 5.

25 When the granularity is constant and sufficiently small, the bandwidth function approximates a continuous function having derivatives, i.e.:

$$B \approx B(t) \quad (\text{Eq. 6})$$

A Taylor's series can be constructed using the left-handed derivatives of B:

$$B_{i+1} \approx B_i + B'_i \cdot \delta t + \frac{B''_i}{2} \cdot \delta t^2 + \frac{B'''_i}{6} \cdot (\delta t^3) + \dots \quad (\text{Eq. 7})$$

where the left-handed derivatives are defined as:

$$B'_i \equiv \frac{B_i - B_{i-1}}{\delta t} \quad (\text{Eq. 8})$$

5 This particular bandwidth prediction method is applicable when B(t) is relatively continuous across the sample space, i.e., the bandwidth B varies smoothly with time t.

Another bandwidth prediction method involves the historical averaging based on time of day, day of week and month of year. According to empirical observation, for example, bandwidth tends to drop off for a client of the east coast of the
 10 United States, when accessing a server located on the west coast of the U.S. at noon EST (eastern standard time). This drop-off is due to the fact that network traffic in greatly increased with the advent of the west coast business day. By logging such bandwidth fluctuations on the client side, the timer block 330 can build up a history for use with bandwidth extrapolation and calculation.

15 The bandwidth prediction methods discussed herein are not exhaustive. Other bandwidth prediction methods that can be implemented include Neville's algorithm for polynomial extrapolation, and Bulisch and Stoer's for rational function extrapolation. Many other algorithms known in the art that are relevant to bandwidth measurement and prediction can also be used within the scope of the invention. The
 20 adaptive downloading technology according to the invention can be used with various bandwidth prediction methods which are not critical to the implementation of the invention per se.

The routing of the requested information from the server 10 to the client
 20 is typically immaterial to bandwidth monitoring and prediction, since the bandwidth is determined by the narrowest part of the routing path in terms of bandwidth. The routing
 25 involves the selection of points in the network or the Internet, represented by IP (Internet protocol) addresses, as the routing or transmission path for transferring the requested information in packets from the server 10 to the client 20.

At times, changing network architectures and conditions require modification in bandwidth prediction. In that case, the timer block 330 can employ a heuristic method, e.g., by recording a recent history of data packet routing information to predict the most likely bandwidth conditions based on the next likely route. Such a heuristic method associates the routing information, represented by a set of IP addresses, with bandwidth measurements. After having built up a sample or history, simple pattern matching algorithms pick out which parts of a route are likely to be fast or slow. The heuristic method can be represented as follows:

$$\{R\}_i = P, Q, A, F, L \quad (\text{Eq. 9})$$

where $\{R\}_i$ represents the route (i.e., a set of IP addresses), and each element of $\{R\}_i$ (i.e., P, Q, A, F or L) represents a route fragment. These route fragments that have pattern commonality with the historical sample can be further decomposed as:

$$P = p, n, z \quad (\text{Eq. 10})$$

$$A = a, b, d \quad (\text{Eq. 11})$$

$$L = l, m, q \quad (\text{Eq. 12})$$

where each of the lower case letters p, n, z, a, b, d, l, m and q represents an IP address. Each $\{R\}_i$ containing the route fragment P is a relatively high bandwidth sample, each $\{R\}_i$ containing A is a relatively low bandwidth sample, and each $\{R\}_i$ containing L is a sample which is not reliably high or low bandwidth. Occurrences of P, A and L serve as yardsticks in the bandwidth prediction in tempering the granularity of the bandwidth measurement, and in weighting subsequent predictions.

The mangler block 340 in Fig. 3 determines which graphics object is requested. The mangler block 340 name-mangles the URI (or URL) of the requested eligible graphics object (i.e., G) into another unique URI (or URL) which is known to the manager application 30 at the server 10 (e.g., G^{n-1}). Name mangling is a way of changing URI (or URL) names. Name mangling changes the URI names according to a common protocol understood by all parts of the adaptive downloading application, analogous to C++ name mangling. Depending on the current bandwidth conditions, the sizes of the graphics objects and corresponding image resolutions available at database 60, the mangler block 340 renames the graphics object to select an appropriate resolution to

ensure a generally constant page impression rate for a download between the server 10 and the client 20. The correspondence between a generally constant page impression rate (server-side defined) and the name-mangling in selecting an appropriate resolution can be represented as follows:

$$5 \quad T_p = T_f + T_a + T_r \quad (\text{Eq. 13})$$

where T_p denotes the time to completely download a web page, i.e. the time for a page impression, T_f is the fixed (and measured) time to download the page source (e.g., HTML source 310), T_a is the time to be used for adaptive downloading, and T_r is the remainder which is estimated from T_f and the ratio of the sizes (S_f / S_r) by the manager application and communicated to the timer-mangler application. In general, when adaptive downloading is called for,

$$T_f + T_r \ll T_a \quad (\text{Eq. 14})$$

and it follows that

$$T_p \cong T_m + T_1 \quad (\text{Eq. 15})$$

15 Where T_m is the time measured prior to the bandwidth prediction in question and T_1 is the time remaining such that the generally constant page impression rate is ensured. Accordingly, when T_m is zero, $T_1 = T_a$. Provided that S_a denotes the total number of bytes for a download during the time period T_a , then the fraction of the total associated with each graphics object for adaptive downloading is defined as:

$$20 \quad c_i = S_i / S_a \quad (\text{Eq. 16})$$

Each bandwidth prediction is then scaled at the remaining time T_1 by the set of dimensionless normalized constants $\{c_i\}$, given that the set is renormalized after the download of each graphic object eligible for adaptive downloading, effectively dividing the remaining T_1 into a series of fractions proportional to the sizes remaining to be downloaded. Since by definition,

$$\Sigma c_i = 1 \quad (\text{Eq. 17})$$

It follows that,

$$T_1 = \Sigma c_i \cdot T_1 \quad (\text{Eq. 18})$$

and as

$$30 \quad T_1 = \Sigma T_i \quad (\text{Eq. 19})$$

Thus,

$$T_i = c_i \cdot T_1 \quad (\text{Eq. 20})$$

Recalling the definition of bandwidth $B = \delta b / \delta t$, where δb is a metric of size S , and δt is a metric of time, then

$$B_i = S_i / T_i \quad (\text{Eq. 21})$$

and by substitution,

$$S_i = c_i \cdot T_1 \cdot B_i \quad (\text{Eq. 22})$$

Since there is a known functional relationship, as configured by the server 10, between the size of each graphics object and the number of available versions,

$$S_i^k = f(k, S_i^n) \quad (\text{Eq. 23})$$

and the inverse solution is also known:

$$k = g(S_i^n, S_i) \quad (\text{Eq. 24})$$

The name mangling that ensures a generally constant page impression rate is the nearest (allowed discrete) value of k such that

$$k \leq g(S_i^n, S_i) \quad (\text{Eq. 25})$$

The LCI (local cache intervention) 350 shown in Fig. 3 resides further up the browser 90 from the other blocks, and intervenes in local disk cache management. Depending on the browser 90 and the network design, the LCI 350 may not always be required. In a typical object-oriented implementation according to the invention, the LCI 350 uses the mangler block 340 to select the appropriate image resolution. The LCI 350 is not required to return the current image resolution dictated by the mangler block 340. Rather, any graphics object of greater or equal resolution is returned to the browser 90.

The de-mangler block 360 reverse transforms the mangled name of the graphics object (i.e., G^{n-1}) back into the URI (or URL) originally requested in the HTML source 310 (i.e., G). The de-mangler block 360 then passes the graphics object, with the original name, to the browser 90. When the manager application 30 includes compression of page source, the de-mangler block is also tasked with decompression.

Figure 3a is an exemplary description of the operation of the timer-mangler application according to the invention in further detail, which processes outgoing

requests from the client browser or other user agent and modifying (name mangling) the request in order to ensure a generally constant page impression rate. This particular embodiment considers only a session between a client and a server, i.e., it is assumed that the user agent does not access another server (not configured with adaptive downloading technology according to the invention) but stays on the single server. In this exemplary implementation, the timer-mangler application interposes itself between the browser or user agent and the port at the level of the TCP/IP stack (as illustrated, e.g., in Figure 3c which shows an exemplary TCP/IP stack), where TCP stands for “transmission control protocol” and IP for “Internet protocol.” The TCP supplies a connection-based protocol over IP which ensures delivery of packets and proper sequencing of the data, and validates the accuracy in both the packet headers and their data. To travel to and from the physical network, packets travel through each layer of the TCP/IP stack. For Windows™ applications, the Windows Sockets API (application programming interface) provides a common interface that communicates with the application layer, e.g., FTP (file transfer protocol), SMTP (simple mail transfer protocol), or WWW (world wide web), and passes packets to and from the next layer, the transport layer with TCP or UDP (user datagram protocol). The lowest level of the network shown in Figure 3c, the link layer, is used in determining how to get packets out onto the physical network, which is associated with the network interface card, e.g., a serial port connected to a modem. The network layer deals with moving packets around the network, while IP is responsible for routing the packets at this layer which wholly or partly requires unique IP addresses. The TCP/IP stack is known in many forms in the art and is not limited to the one illustrated in Figure 3c.

Referring to Figure 3a, in step 3001 a request for a resource (e.g., URI) arrives at the TCP/IP stack and is intercepted by the timer-mangler application. A decision is made as to the nature of the resource in step 3003. If the resource is not embedded, then a new page will be started, and the measured time T_m is reset in step 3005. The request is marked as coming from a user agent configured with the timer-mangler application in step 3007, typically in the protocol header. In step 3009 a

request is made for the control information pertaining to the new page. The flow then proceeds from A to step 3035.

When the resource is embedded in the web page, a determination must be made as to its eligibility for adaptive downloading according to the invention (step 3011).

5 If the resource is not eligible according to the control information received previously in step 3009 for the page currently being parsed, the flow proceeds from A to step 3035.

If the resource is eligible, a determination is made in step 3015 as to whether a user override has been asserted by one of the methods described herein or by some other method known in the art. If there is a user override, then in step 3016 it is
10 determined whether the resource belongs to a non-cooperating 3rd party server. If this is not the case, the user-selected resolution is substituted for the graphics object referenced in the source, by way of mangling the object name appropriately (step 3017). The flow then proceeds from A to step 3035. If the user override is for a 3rd party server resource, the domain of the 3rd party server is replaced with that of the alias server provided for the
15 web page in block 3009 (step 3018). Then in step 3019 the nearest allowed user-selected resolution is appended to this new URI, and the flow proceeds from A to step 3035.

When there is no user override, in step 3020 the time remaining before the generally constant page impression rate time is exceeded is calculated according to Eq. 13. In step 3021, the next predicted bandwidth B_i is calculated according to methods
20 described herein in equations 3 through 12, or by some other method known in the art.

Next, the decision is made in step 3023 as to whether the URI in question belongs to a non-cooperating 3rd party server. If the requested URI is a 3rd party server, the domain is replaced in step 3025 with that of the alias server provided for the web page in block 3009. The appropriate size S_i is picked in step 3027 based on the expected
25 3rd party average size either provided in 3009 or provided globally with the download of the timer-mangler application. Then in step 3029 the resource name is mangled to pick the nearest S according to Equation 25. This branch of the flow then proceeds to step 3035.

If the URI for the graphics object belongs to the adaptive server according
30 to the invention, step 3031 picks the desired size S_i according to the methodology in Eq.

22 using the control fraction values c_i provided in 3009 for this web page. Next, the object name is mangled with respect to Equations 23-25 (step 3033).

Step 3035 downloads the resource, which may or may not be name mangled according to its nature and eligibility, and measures the time required for the download of the resource. In 3037, the amount of time measured is added to the total measured download time for the page, T_m . The timer-mangler application is then ready to intercept the next resource request from the user agent.

Figure 3b demonstrates an exemplary process by which the timer-mangler applet processes incoming data from the network port. The data is intercepted by the timer-mangler application, which is interposed between the port and the user agent in step 3101. In step 3103, the determination is made as to whether the incoming resource (e.g., URI) has been name mangled. If this is not the case, step 3105 determines whether the incoming resource represents control data pertaining to the current or subsequent page. If the incoming resource represents control data sent by the manager application for the control block of the timer-mangler application, such data may be compressed, as determined in step 3107, and is forwarded to the control block in step 3109 or first decompressed in step 3108 prior to being sent to step 3109. If the resource does not represent control data, program flow continues from step 3105 to step 3117.

If it is determined in step 3103 that the resource is name mangled, it is subsequently de-mangled in step 3111. The next step 3113 determines if the resource was compressed by the manager. If compressed by the manager the resource is decompressed in step 3115, then sent on to the browser or user agent for display or parsing in step 3117. If the resource is not compressed it is passed directly for display or parsing in step 3117. The timer-mangler application is then ready to intercept the next incoming resource.

Although the descriptions herein are laid out in a temporally cohesive order, it is important to note that because of the multi-threaded nature of user agents and indeed computers and operating systems in general, a process flow chart may not always accurately reflect the actual temporal nature of the execution. Thus, for example, step 3035 in Fig. 3a may be a separate, continually executing process that accepts resource

names to download and times such downloads with programmable granularities as described herein and above. Similarly, while step 3035 is shown in Figure 3a in order to present a conceptually cohesive process flow, this step may also be represented in Figure 3b as a sub-step to step 3117 or step 3101.

5 Returning to Fig. 1A, the manager application 30 provides the appropriate image resolutions, in concert with the timer-mangler application 40 on the client side, to ensure a generally constant page impression rate for a download. In terms of server development, the manager application 30 is entirely non-invasive on the server 10, i.e. it does not effect maintenance and content development on the server 10. The manager
10 application 30 acts as a broker between the server 10 and the user-requested information. That is, the manager application 30 intercepts requests for information from the server 10, and translates them into requests for currently appropriate data with a corresponding image resolution. The data lookup is performed by whatever method is indigenous to the server 10, e.g., a lookup from the database 60 or a read from a hard disk or RAM cache.
15 In an SQL (structured query language) environment, the manager application 30 modifies the SQL requesting information to an SQL request for the information in a corresponding resolution stored in the database 60.

In other embodiments according to the invention, the role of manager application 30 is not restricted to brokering user requests for information. On a server
20 with sufficient computing and storage resources, manager application 30 can be configured to handle impression of the requested information on the fly at the instant of the user request. Further, the invention is applicable in many operating system (OS) environments, including various UNIX and Microsoft Windows™ operating systems.

The manager application 30 interposes itself typically at the level of the
25 operating system between the server 10 and the data, or between the data and the network port when the data is stored at database 60 using an incremental compression scheme such as JPEG 2000. For example, in a UNIX operating system the manager application 30 can replace the open system call so that when the server 10 requests the file handle for a multiple resolution graphics object, the handle of the appropriate current resolution is
30 automatically returned. There are many possible designs of the interposition of the

manager application 30 between the server 10 and the data, which all fall within the scope and spirit of the invention and may be used to implement the adaptive downloading technology according to the invention.

Depending on the configuration of the server 10, the function of the manager application 30 may not be restricted to passing on data requests. On a server equipped with JPEG 2000 graphics, the manager application 30 controls the amount of data sent out over the network. On a server with sufficient computing power and more limited storage resources, the manager application 30 can be configured to handle data compression on the fly at the instant of the data request.

Figure 4 is a block diagram that illustrates the operation of the manager application according to the invention in further detail. The manager application 30 includes two parts: the front manager 31 and the back manager 32. In the present embodiment, the graphics objects are stored in a name-mangled form, i.e., standard non-incremental compression such as JPEG or GIF (Graphics Interchange Format) compression. The front manager 31 is interposed between the network port to the Internet and the server 10. The front manager 31 de-mangles a name request and passes that request to the server 10. Under certain circumstances, the front manager 31 can simply intercept the mangled name and pass it directly to the back manager 32.

The back manager 32 is interposed between the server and the data. The back manager 32 intercepts the data lookup, re-mangles the URI and passes the request to the database or archive 60.

The end user at the client 20 requests the embedded graphics object G. The timer-mangler application 40 of the client 20 determines that the appropriate image resolution for this graphics object is G^{n-1} . The mangler block 340 renames the graphics object appropriately (as described herein and above). The request for the graphics object is then passed out the TCP/IP stack to the Internet. A distributed Internet cache 41 (such as those available from vendors like Inktomi™ and InfoLibria™) is encountered along the route between the client 20 and the server 10. The requested graphics object is not present in this Internet cache 41 (although there are other versions of G), which results in a cache miss. If no other Internet caches contain G^{n-1} , the request arrives at the server

port and is subsequently intercepted by the front manager 31. The front manager 31 demangles the named request G^{n-1} into G and passes the demangled object G to the server 10, where it is recognized as one of the original objects originating on that server. The front manager 31 also passes the name of the mangled object G^{n-1} to the back manager 32. The server 10 in turn requests the object G from the archive 60, and this request is intercepted by the back manager 32 and transformed (re-mangled) into an SQL request or other read of G^{n-1} from the archive 60. The object G^{n-1} is subsequently returned by the archive 60 and served to the port, the distributed Internet Cache 41 and the client side demangler block 360. Since the object G^{n-1} is now present in the distributed Internet cache 41, subsequent requests from the client 20 or other clients is appropriately served according to the adaptive downloading technology according to the invention without further server intervention. Because of the name-mangling by the timer-mangler application 40, the names of the graphics objects are unique at different image resolutions. Internet caches can return requested objects at the appropriate image resolution without intervention from the server 10.

The back manager 32 is also responsible for communication of all of the control information 322 to the timer-mangler application 40. When a web page is downloaded to the client 20, the back manager 32 sends control information 322 to the timer-mangler application 40 regarding which graphics objects are eligible for name mangling, i.e., which graphics objects are available in multiple resolutions. The control information 322 also includes information on how many image resolutions are available. The number of resolutions can be provided globally (per server), regionally (per web page), or locally (per graphics object), depending on the server-side defined configuration.

Figure 5 is a block diagram that illustrates the operation of the manager application according to the invention with a JPEG 2000 database. When graphics objects (e.g., G, H, I, J and K) are stored in an incremental compression scheme such as JPEG 2000 in database or archive 560, the image resolution of a graphics object is a function of its size. With a JPEG 2000 compressed image, a relatively small amount of data gives a fuzzy (relatively low resolution) image and a relatively large amount of data

gives a sharp (relatively high resolution) image. The image resolution visibly increases as more data is received. Each successive higher resolution includes the data from the lower resolution, with more data that increases the sharpness of the resolution. As shown in Figure 5, the back manager 32 further includes a counter 532 for counting the number of bytes sent out from the server 10. Once the threshold (in terms of number of bytes) dictated by the timer-mangler application 40, the back manager 32 discards the remaining data. Because the URI (or URL) is mangled by the timer-mangler application 40, cache aliasing is generally avoided as described herein.

In addition, the adaptive downloading technology according to the invention provides adaptive downloading for a web page displaying information that includes an embedded resource of a third party server, such as a third party banner ad. If the third party server is a cooperating third party server, e.g., a server that includes a manager application functionally equivalent to the manager application at the server, then the timer-mangler application appropriately selects the image resolution in conjunction with that manager application to ensure a generally constant page impression rate. That is, the manager application ensures a generally constant page impression rate, and the timer-mangler application undertakes object renaming but addresses the renamed object within the IP domain of the third party server. The timer-mangler application performs name mangling on each eligible URI (or URL) into an external lookup at the appropriate resolution. In this case, the third party server is either a server that has incorporated the adaptive downloading technology according to the invention, or a server that has been configured with multiple resolution versions of the graphics objects of the requested information. The cooperating third party server then adaptively provides the appropriate image resolution for the requested information to ensure the generally constant page impression rate.

If the third party server is a non-cooperating third party server, i.e., a server that does not have a manager application, then an alias server and a virtual queue of images at various resolutions are provided at the server. In the server, an alias server with an Internet protocol (IP) address known to the timer-mangler and the manager applications is provided for the third party server. The alias server serves as an

addressing substitute as the timer-mangler addresses the third party server. The timer-mangler application replaces the third party IP address with that of the alias server while it undertakes object renaming. The virtual queue is provided to store the requested information of the third party server in a plurality of image resolutions. The first-in-first-
5 out (FIFO) nature of the virtual queue ensures that the order of images accessed from the alias server remains identical to that which would have been accessed from the third party server, while the calculation of the various image resolutions is decoupled from periods of peak server activity, thereby avoiding added encumbrance to the server during such periods. The alias server then accesses the virtual queue for the appropriate image
10 resolution, as determined by the timer-mangler and manager applications, corresponding to the requested information of the third party server, which ensures a generally constant page impression rate.

Figure 6 is a block diagram that illustrates the operation of the adaptive downloading technology according to the invention in conjunction with a non-
15 cooperating third party server. When the third party server 600 is a non-cooperating server an alias server 610 is provided at the server side. The alias server 610 is a logical server block located within the logical domain of the server 10 that serves as a surrogate for the third party server 600. When the control information 322 is sent to the timer-mangler application 40, a list 622 of eligible third party servers and their aliases is
20 included. When the URI (or URL) within the domain of an eligible server is traversed, the timer-mangler application 40 transforms it into an URI (or URL) located on the appropriate alias server 610.

A compression engine 620 is dedicated to concurrently building an image cache for the appropriate alias server 610, i.e., virtual queue 630. The compression
25 engine 620 may be a dedicated device or a software program running within the logical domain of the server 10 either on common or dedicated hardware resources. The virtual queue 630 includes images at various resolutions at the server side. No special knowledge of server 600's content is needed. Management of the virtual queue 630 takes place within the logical domain of the server 10 equipped with the adaptive downloading

technology according to the invention. An alias server is provided for each third party server. The alias server also resides within the logical domain of the server 10.

Non-cooperating advertising servers generally include two model which respectively provide advertising targeting the site (i.e., demographics based), and
5 advertising targeting the individual (i.e., content selected based on input from the server 10). When the advertising is demographics based, a single queue is sufficient for each
• third party server. Such a virtual queue can be built or restocked during periods of relative server inactivity. When the compression engine 620 shares hardware resources with the server 10 and the alias server 610, it allows the virtual queue (or queues) 630 to
10 be built and maintained without encumbering the server's computing resources during periods of high activity. In this case, the virtual queue is configured with d slices and n choices to accommodate peak activity times. In an alternative, the virtual queue 630 can be maintained with dedicated compression hardware. However the virtual queue 630 is built and maintained, a queue slice (out of the d slices) contains n choices of available
15 resolutions for the requested image 640 for each accessed image on the third party server 600. When the image 640 is already incrementally compressed, then there is no need for the compression engine 620. Likewise, when the number or available resolutions n is one, there is no need for the compression engine 620. In this case, the virtual queue 630 decouples the website served by the server 10 from the potentially slower 3rd party server,
20 as well as insulating the server 10 from the vagaries of changing network routing, adding control to ensure the generally constant page impression rate. The alias server 610 controls the resolution of the graphics object A which is sent to the client over the Internet or the network, where the resolutions are selected by the timer-mangler application 40. No particular knowledge of the content at the third party server 600 is
25 required, since the building mechanism of the virtual queue 630 accesses the third party server 600 in the same order as if it were actual end user choices. Busy servers retain reasonable temporal locality with respect to traditional third party banner-type graphics. To support advertising vendors that offer strict temporal control, e.g., so that an advertisement is served between certain hours, the depth of the virtual queue 630 can be
30 configured variably as a function of time.

The slices (d slices) of the virtual queue 630 retain generally most of the aspects of the original content of the third party server 600. Only the graphic resolutions are potentially modified in accordance with the invention. To further reduce download time, the source text associated with each queue slice of the virtual queue 630 can also be
5 compressed. In this case, the de-mangler block 360 of the timer-mangler application 40 is tasked with client side decompression.

To access the virtual queue 630, a list of aliases for third party servers is sent with the control information 322 to the control block 320 of the timer-mangler application 40 at the client 20. When an URI (or URL) located on the third party server
10 600 is detected, the timer-mangler transforms the lookup to one located at the address of the third party alias server 610 which is coupled with the virtual queue 630. The alias server 610 reads the appropriate resolution from the top slice of the d slices in the virtual queue 630. The requested information in the appropriate image resolution is routed through the alias server 610 to the client 20. The top slice of the d slices is discarded
15 after having been read by the alias server 610. The alias server 610 can also perform data collection 650 for the third party server 600 by gathering server log information, such as streaming data per click, referring web page, log history, and other information which is useful for user data collection and analysis. This hidden information regarding the client consisting of frequency and number of website visits, data mining and marketing
20 information are forwarded to the third party server 600 in data collection 650 or maintained in logs for later retrieval by the third party server 600.

Since the manager application acts as a broker between the server and data, conversion of existing websites for use in conjunction with the adaptive downloading technology according to the invention is simple to implement. In general, a
25 group of scripts (such as PERL scripts) of a type specific to the website implementation are run to create the appropriate data constructs for adaptive downloading. When the server uses non-incremental data compression schemes, the conversion scripts modify the hard disks or databases to create different resolutions which are server-side defined. When the server uses incremental graphics compression (such as JPEG 2000), there is no
30 need to modify the image data stored in the database. In addition, other scripts build the

control tables that the manager application requires for transmission to the timer-mangler application at the client, set the number of available resolutions, the granularity of the timer-mangler bandwidth measurement, the bandwidth prediction methods, the functional relationship of $\{G\}_t$, the end user override, local cache intervention, and the desired page
5 impression rate. Furthermore, websites are often in a state of flux, which may require constant updating of content. When eligible graphics objects are stored in a database or a recognizable tree hierarchy, a script running constantly in the background monitors the data for changes and automatically updates the archive and control structures accordingly.

10 An end user at the client 20 is given the option of overriding the manager application's selection of the appropriate image resolution, when, e.g., the end user desires to see a particular graphic in better detail. An implementation of the use override is to affix a relatively small and easily downloadable graphic to the bottom or periphery of, or overlaid across the bottom of each eligible graphics object. The downloadable
15 graphic can be text or an icon, showing poor resolution in one direction but gradually increasing to full resolution in the other direction. The downloadable graphic, or logical slide bar, provides a resolution value and load request to the server when the end user at client 20 clicks on or about the logical slide bar. The timer-mangler application 40 then overrides bandwidth considerations and requests the nearest available resolution of the
20 desired graphics object per the end user's selection. In an alternative embodiment, each graphics object downloaded to the client 20 can include a link to the next higher resolution. The end user can also be given the option of overriding the desired data impression rate implemented on the server side. Such an override, which represents a relatively direct path to the control block 320 of the timer-mangler application 40, is
25 implemented as a separate applet.

Figure 7 is a flow diagram that illustrates the adaptive downloading technology according to the invention in further detail. In step O701 an end user at the client 20 requests certain information from the server 10. In step O703, it is determined whether the timer-mangler application according to the invention is resident at the client
30 20. If not, the timer-mangler application 40 is downloaded to the client 20 (step O705)

and the control flow goes to step O707. If the timer-mangler application is resident at the client 20, the control flow goes directly to step O707.

In step O707, the timer-mangler application 40 monitors the timing information, e.g., the bandwidth between the beginning and end of a download of information from the server 10 to the client 20, (a point-to-point measurement), but also the bandwidth between the beginning and end of a download of information from a distributed cache 41 or other proxy for the server 10. Additionally the size of the download of information that forms the baseline of a measurement, that is the granularity of the bandwidth measurement, may be varied so that bandwidth is measured with a per time fixed granularity or a per size fixed granularity, as described herein and above. Furthermore, the timing is a generally constant process that may or may not be strictly represented in temporal fashion in the context of Figure 7.

In step O709 a determination is made of whether the information is embedded in the source. If not, then the requested information is a new page, and the timer for the generally constant page impression rate is reset in step O711. The new page request is passed out the client 20 network port to the network in step O713. In step O715 the control information required to ensure the generally constant page impression rate is requested from the server by the timer-mangler application 40.

If the information is embedded in the resource, the flow continues to step O721, where a determination is made by the timer mangler 40 of whether the embedded resource is eligible for adaptive downloading according to the invention. If the timer-mangler 40 determines that the object is not eligible, the URI is passed unmodified to the network port in step O723, and the information is subsequently downloaded from the network. If the resource is eligible for adaptive downloading according to the invention (step O721), the timer-mangler 40 predicts the bandwidth conditions (step O725) expected during the upcoming period as described herein and above. Using this prediction the timer-mangler 40 selects a resolution, and hence name-mangles, thereby ensuring a constant page impression rate (step O727) as described herein and above.

In step O729 a determination is made of whether the information is located at a cooperating server with a manager application 30, e.g. server 10, or at some

other server that is eligible for adaptive downloading according to the invention, but not a cooperating third party server as defined and described herein.

If the server is one that cooperates directly with the timer-mangler application, the control flow proceeds to step O731 where it is determined by the timer mangler 40 whether the user is overriding the resolution selection of the timer mangler application. If there is a user override, the timer mangler application mangles the name of the graphics object to use the user-selected resolution in step O733. The control flow then proceeds to step O737. If there is no user override, then the resolution selected by the timer-mangler application, which ensures the generally constant page impression rate, is used to mangle the name of the graphics object in step O735. The mangled name is passed onto the network port and the network in step O737. The server 10, in conjunction with the manager application 30, or the appropriate network resource, e.g. a distributed cache 41, serves the information to the network in step O739 for downloading by the client, which information is subsequently downloaded to the client.

If the server is one that does not cooperate directly with the timer mangler application 40, e.g. a third party server 600, the flow proceeds from step O729 to step O741, where the IP address of the third party server 600 is replaced with that of the alias server 610. In step O743 a determination is made as to whether there is a user override of the resolution selected by the timer-mangler application. If this is the case the user-selected resolution is used for the name mangling in step O745. If there is no user override, the resolution selected by the timer mangler 40, which ensures the generally constant page impression rate, is used in step O749. The flow along both branches then proceeds to step O749, where the request is passed via the network port to the network. In step O751, the request is received by the alias server 610, which serves the requested information from the top of the virtual queue 630 as described herein and above, and the information is subsequently downloaded to the client.

Figure 7a and 7b, in conjunction with Figure 3a, are flow diagrams that illustrate the method according to further embodiment of the invention using virtual queue and an alias server. Figure 7a shows an exemplary embodiment of the process whereby the virtual queue 630 is maintained on a server architecture where the server and

compression engine 620 share hardware resources. In such situations it is desirable that the compression engine 620 not tax the common resources during periods of peak server activity. Initially, the current virtual queue depth is examined in step 701. In step 705 a determination is made as to whether the current depth is less than a configurable threshold F. If not, the flow proceeds to step 703, where a configurable period of time of 5 J seconds is allowed to pass before repeating the previous steps. This represents a period when the virtual queue has a depth greater than its minimal allowable size, so that no action need be taken to maintain it. If the current depth is less than the threshold F in step 705, the server load is checked in step 707. If in step 711 it is determined that the load is 10 greater than a configurable threshold, flow proceeds to step 709 for a timeout of configurable T seconds, before the load is rechecked in step 707. This represents a period where common hardware resources are not available without degrading server performance, so that the potentially computationally expensive step 717 is not performed. The values of F, J, T and the threshold in step 711 are chosen with care to avoid emptying 15 the virtual queue. If computational resources are limited, a large value \underline{d} is selected for the depth of the virtual queue 630. Server 10 load permitting, the control flow proceeds to step 713 where the embedded graphic is requested from the network and received from the third party server. Subsequently, a new bottom queue slice is created in step 715. In practice a virtual queue 630 may be constructed in many ways known in the art, such as a 20 database application or simply a linked list of memory blocks. In this latter implementation, creating a new bottom slice as in step 715 involves allocating memory to hold the slice structure and the various copies of the resource at various resolutions, as well as updating the pointers of the new slice and those of the previous bottom slice. In step 717, the various resolutions are created. The compression engine 620 is invoked to 25 create a plurality of \underline{n} resolutions of the 3rd party graphic, according to a definable functional relationship analogous to those demonstrated in Figure 2. These resolutions are then stored in or linked to the new queue slice in step 719. The control flow then proceeds to step 701, resources permitting, to build the next slice.

Figure 7b demonstrates an exemplary process for the alias server 610 30 according to the invention. The alias server is idle until it receives a request for a

resource in step 751. This request comes from the timer-mangler application 40 as described herein and above. The alias server reads the object at the requested resolution from the top slice of the virtual queue 630 in step 755. It subsequently collects data 650 regarding the client 20, which is either logged for later retrieval or served directly to the
5 third party server 600 (Fig. 6), in step 757. The object with the appropriate resolution is served to the network port in step 759. Next, the top queue slice, which now contains stale data, is discarded in step 763. This may require interfacing to a separate database application or a simple pointer modification and memory de-allocation as described herein relating to step 715 of Figure 7a, to which this step is linked. The data being
10 served by the alias server 610 proceeds in step 767 to the next request, reverting to the waiting state in step 751.

The timer-mangler application determines, in step 3023 of Figure 3a, that the user agent has parsed the URI for an eligible third party server. The timer-mangler application received, with the control information for the page in step 3009, the URI of
15 the alias server 610, as well as the expected average of sizes of embedded graphics from this third party server 600. The timer-mangler application can therefore, in step 3025, replace the domain of the third party server 600 with that of the alias server 610 and retain that part of the entire URI as is pertinent to the alias server, where the relative path on the third party server 600 is then discarded. Using the control information obtained in
20 step 3009, the timer-mangler estimates (step 3027) the desired size S_i that enables the third party graphic to be downloaded while ensuring a generally constant page impression rate according to the invention described herein. In step 3029, the timer-mangler application selects the nearest size S that corresponds to the existing copies and resolutions of the desired object according to the configured functional relationship for
25 this third party alias server 610, as described herein and above, to ensure a generally constant page impression rate. This mangled name is passed to the network port in step 3035, which is, upon transmission across the network, serviced by the alias server 610 in step 751 of Figure 7b.

Figure 8 is a block diagram that illustrates the operation of another
30 embodiment of the invention with multiple third party servers. In the present

embodiment, the website being displayed includes banner advertising that is controlled by three third party servers 600A, 600B and 600C. There are n different levels of JPEG compression available.

The end user at the first client 20G requests the embedded graphics object
5 G. The LCI (not shown in Figure 8 but described herein and above) determines that the graphics object is not in the local cache of the client 20G and passes along the request. The timer-mangler application 40G (described herein and above) of the client 20G determines that the appropriate image resolution for this graphics object is G^{n-1} . The mangler block 340G renames the graphics object appropriately, as described herein and
10 above. The request is passed along as a query (such as an HTTP query) to the first distributed Internet cache 81 along the route between the client 20G and the server 10. The requested graphics object is not present in this Internet cache 81 (although there are other versions of G), which results in a cache miss. Further along the route, or in one of the other Internet caches queried by the first Internet cache, an Internet cache 82 is found
15 to contain the object G^{n-1} . This Internet cache 82 returns G^{n-1} , which is subsequently loaded and returned to the client 20G. Because of name-mangling by the timer-mangler application 40G, the names of the graphics objects are unique at different image resolutions. Internet caches can return requested objects at the appropriate image resolution without intervention from the server 10.

20 The end user at the second client 20H requests the embedded graphics object H. The timer-mangler application 40H (described herein and above) determines that the appropriate resolution given the bandwidth conditions at the client 20H is H^{n-2} . This particular data request results in cache misses in all of the Internet and local caches, so that the highest level Internet cache 89 passes the request to the server 10. The front
25 manager 31 de-mangles the name of the graphics object into a URI (or URL) known to the server 10. Since the server 10 never sees the mangled names, it is not necessary to modify server function or HTML source code (or SMIL source code, etc.) to implement the adaptive downloading technology according to the invention.

The server 10 then requests the data by generating an SQL lookup to a
30 database 60, e.g., an ORACLE or SYBASE database application. The database 60 can

also be a directly accessible, local hard disk resident in the server 10, or a RAM cache or other archival method associated with the server 10, since the back manager 32 (described herein and above) receives data requests from the server 10 by intercepting requests for the archive sources (e.g., database 60). The back manager 32 re-mangles the name of the graphics object before passing the request onto the appropriate archive. That is, the back manager 32 modifies the SQL according to the control information 323 communicated by the front manager 31, to request the appropriate resolution from the database 60. The data, in this case H^{n-2} , is served to the requesting cache, then passed back to the requesting client 20H where it is de-mangled to H and displayed for viewing by the end user at the client 20H.

In the third client 20A, the browser has parsed and subsequently requested display of a third party advertisement, in this case an advertising banner identified as A. The control block 320A of the timer-mangler application 40A received a list of aliasable third party servers along with the download of the page source that contains the embedded banner. Accordingly, the name of the requested URI is mangled to reference the alias server 610A by the virtual queue 630A associated with the appropriate third party server 600A. The appropriate image resolution A^s for the banner A is also selected in conjunction with the name mangling.

The alias server 610A for the third party server 600A delivers the appropriate A^s and that slice of the virtual queue 630A is discarded (as described herein and above). Because the alias server 610A interacts closely with the virtual queue 630A, it is not necessary to de-mangle names between the alias server 610A and the virtual queue 630A. The alias server 610A also delivers the appropriate data to the third party server 600A (data collection 650A, described herein and above), or logs it for later collection by the third party server 600A.

The virtual queue (630A, 630B or 630C) processes the data request from the alias server (610A, 610B or 610C) and return one of the graphics objects from the leading slice in the virtual queue (described herein and above). The virtual queue then discards the leading slice and bumps the succeeding slices up one slot. As described herein and above, the virtual queue is FIFO (first in first out) so that third party banners

(A, B or C) are accessed from the third party server (600A, 600B or 600C) in the same order as if there were no adaptive downloading technology according to the invention present. The compression engine 620 supplies the virtual queue (630A, 630B or 630C) with new slices of data based on images from the third party server (600A, 600B or
5 600C), also described herein and above.

Although the invention has been particularly shown and described in detail with reference to the preferred embodiments thereof, the embodiments are not intended to be exhaustive or to limit the invention to the precise forms disclosed herein. It will be understood by those skilled in the art that many modifications in form and detail may be
10 made without departing from the spirit and scope of the invention. Similarly, any process steps described herein may be interchangeable with other steps to achieve substantially the same result. All such modifications are intended to be encompassed within the scope of the invention, which is defined by the following claims and their equivalents.

I CLAIM:

- 1 1. A method for downloading information from a server to a client,
2 comprising the steps of:
3 causing the client to request information from the server;
4 monitoring download timing information between the server and the client
5 using a time-mangler application located at the client;
6 selecting an image resolution corresponding to the requested information
7 based on the download timing information; and
8 downloading the requested information with the selected image resolution.

- 1 2. The method of claim 1 further comprising the step of
2 allowing a user at the client to override the selection based on the
3 download timing information and to choose a user selected resolution; and
4 wherein the step of downloading causes downloading of the requested
5 information with the user selected resolution.

- 1 3. The method of claim 1 wherein the client requests the information by
2 name, and further comprising the steps of:
3 mangling the name of the requested information based on the selected
4 resolution with the timer-mangler application;

5 transmitting the mangled name from the timer-mangler application to the
6 server;

7 storing the information at the server at different resolutions and under
8 unique respective mangled names based on the resolution; and

9 downloading the information corresponding to the mangled name in the
10 request received by server from the timer-mangler application at the client.

1 4. The method of claim 1 wherein the client and server are connected over
2 the Internet.

3

1 5. The method of claim 1 wherein the server includes a manager application
2 and the monitoring of the download timing information between the server and the client
3 is based on the time-mangler application at the client and the manager application at the
4 server.

1

1 6. The method of claim 5 wherein the step of selecting an image resolution
2 corresponding to the requested information based on the download timing information is
3 set to give a certain page impression rate.

4

1 7. The method of claim 3 further including the steps of:
2 using a front portion of the manager application to de-mangle the mangled
3 name received from the client at the server;
4 presenting the de-mangled name to the server;

5 causing the server to generate a database information request based on and
6 including an indication of the name of the requested information;

7 using the a back portion of the manager application to mangle the name of
8 the requested information in the database information request;

9 accessing a particular resolution image to download based on the mangled
10 name of the information in the database information request.

1

1 8. The method of claim 1 wherein the timer-mangler application is one of an
2 applet, a plug-in and an embedded component of a web page downloaded from the server
3 to the client.

1 9. The method of claim 4 wherein the step of causing the client to request
2 information from the server involves placing the request in an Internet browser.

1 10. The method of claim 1 wherein the on screen dimensions of the
2 downloaded information do not change based on the resolution selected.

1 11. The method of claim 1 wherein the on screen dimensions of the
2 downloaded information change based on the resolution selected.

1

1 12. The method of claim 3 wherein the client and server are connected over
2 the Internet and wherein the selected information is stored in an Internet cache under the
3 mangled name, comprising the steps of:
4 searching the Internet cache for the selected information listed under the
5 mangled name prior to sending the request to the server;
6 if the selected information is found in the Internet cache, downloading it to
7 the client and stopping the process; and
8 if the selected information is not found in the Internet cache, sending the
9 request to the server.

1 13. The method of claim 2 wherein the step of allowing the user to override
2 the selection involves allowing the user to move a marker on an icon-based slide bar on a
3 display screen at the client to indicate the desired resolution.

1

1 14. The method of claim 3 wherein the downloaded information is related to a
2 mangled name and the timer-mangler application de-mangles the name and applies the
3 information and the name to the client.

1

1 15. The method of claim 3 wherein the client and server are connected over
2 the Internet, the timer-mangler takes the uniform resource identifier or locator of the
3 address of the requested information and renames it to the address of the information at a
4 selected resolution which is stored separately from the information at other resolutions.

1

1 16. The method of claim 1 wherein the requested information is made up of
2 multiple image tiles, and the resolution is selected so that each of the tiles is downloaded
3 with the same resolution.

4

1 17. The method of claim 6 wherein in selecting an image resolution based
2 based on downloading timing information, the bandwidth between the server and the
3 client is predicted.

1

1 18. The method of claim 17 wherein the prediction is based on measurements
2 a particular times.

1

1 19. The method of claim 18 wherein the prediction is based one of a weighted
2 average of past measurements, historical averaging, Neville's algorithm for polynomial
3 extrapolation, Bulisch and Stoer's rational function extrapolation, and a heuristic method.

1

1 20. The method of claim 1 further comprising the steps of:
2 accessing a third party server if the requested information is not located in
3 the database of the server;
4 building a virtual queue with images from the third party server which are
5 compressed into different image resolutions if the third party server does not include the
6 manager application; and
7 accessing the virtual queue to retrieve the requested information.

1 21. The method of claim 20 further comprising the steps of:

2 downloading the requested information from the virtual queue with the
3 selected image resolution if there is no user override; and
4 downloading the requested information from the virtual queue with user-
5 selected image resolution if there is a user override.

1 22. The method of claim 20 further comprising the step of creating an alias
2 server for the third party server.

1 23. The method of claim 20 further comprising the steps of:
2 accumulating hidden information regarding the client including frequency
3 and number of website visits, data mining and marketing information; and
4 forwarding the hidden information to the third party server or maintaining
5 local logs thereof for future retrieval by the third party server.

1 24. A system for downloading information from a server to a client
2 comprising:
3 a database of information stored at the server at different resolutions and
4 having a particular version of a name associated with each resolution;
5 a timer-mangler application at the client, said timer-mangler application
6 detecting the bandwidth between the server and the client and automatically mangling the
7 name of information requested by the client from the server to a particular version of the
8 name based on a resolution selected to give a particular page impression rate from the
9 server to the client based on the bandwidth; and
10 a manager application at the server for receiving the mangled name and
11 selecting the information at the resolution associated with the mangled name and
12 arranging it for download to the server.

1 25. The system of claim 24 in which the information is downloaded over the
2 Internet.

1

1 26. The system of claim 24 further including an override which allows a user
2 at the client to select a resolution other than that set by automatically mangling the name
3 of the information requested, said manage application selects the override resolution
4 instead in the automatic resolution.

1

1 27. The system of claim 24 in which the downloaded information and
2 mangled name are returned to the client and the timer mangler de-mangles the name
3 before presenting it and the information to the client.

1

1 28. The system of claim 24 wherein the manager application receives the
2 mangled name and has a front manager portion that de-mangles it and presents it to the
3 server;

4 the server generates a database query based on the de-mangled name; and

5 the manager application has a back manager portion that mangles the
6 name associated with the database query based on the mangled name received from the
7 client.

1

1 29. The system of claim 24 in which the timer-mangler and the manager
2 application jointly determine the bandwidth.

1

1 30. The system of claim 24 in which the timer-mangler application is one of
2 an applet, a plug-in and an embedded component of a web page downloaded from the
3 server to the client.

1

1 31. The system of claim 25 in which the client further includes a browser
2 application in which requests for information are made by inserting the uniform resource
3 locator or identifier address in the browser, and the timer-mangler application modifies
4 the address to a location having a version of the information at the selected resolution.

1

1 32. The system of claim 24 in which the selected resolution is based on a
2 predicted bandwidth.

1

1 33. The system of claim 31 wherein a third party server has information to be
2 downloaded to the client when the server is accessed and does not have a manager
3 application, further including:

4 an alias server; and

5 a virtual queue associated with the alias server for storing images of the
6 contents of the third party server at different resolutions; wherein the manager application
7 accesses the alias server instead of the third party server and selects information from the
8 virtual queue at a resolution based on the mangled name.

1

1 34. A system for downloading information over the Internet, the system
2 comprising:

3 a server;
4 a client requesting information from the server wherein the server
5 downloads a timer-mangler application to the client, the timer-mangler application
6 monitoring download timing information between the server and the client;
7 a manager application resident in the server that selects an image
8 resolution in conjunction with and as instructed by the timer-mangler application and
9 corresponding to the requested information based on the download timing information to
10 ensure a constant page impression rate, and downloads the requested information with the
11 selected image resolution to the client.

1 35. The system of claim 34 further including a user override, and wherein the
2 requested information is downloaded to the client with the selected image resolution if
3 there is no user override, and with user-selected image resolution if there is a user
4 override.

1 36. The system of claim 34 further comprising:
2 a third party server wherein the client accesses the third party server by
3 accessing a third party banner presented by the server in a browser of the client; and
4 a virtual queue resident in the server storing images from the third party
5 server which are compressed in different image resolutions by a compression engine if
6 the third party server does not include the manager application;
7 wherein the virtual queue is accessed to retrieve the requested information.

1 37. The system of claim 36 wherein the requested information is downloaded
2 from the server to the client if the requested information is embedded in the server.

1 38. The system of claim 36 wherein the requested information is downloaded
2 from the virtual queue with the selected image resolution if there is no user override, and
3 with user-selected image resolution if there is a user override.

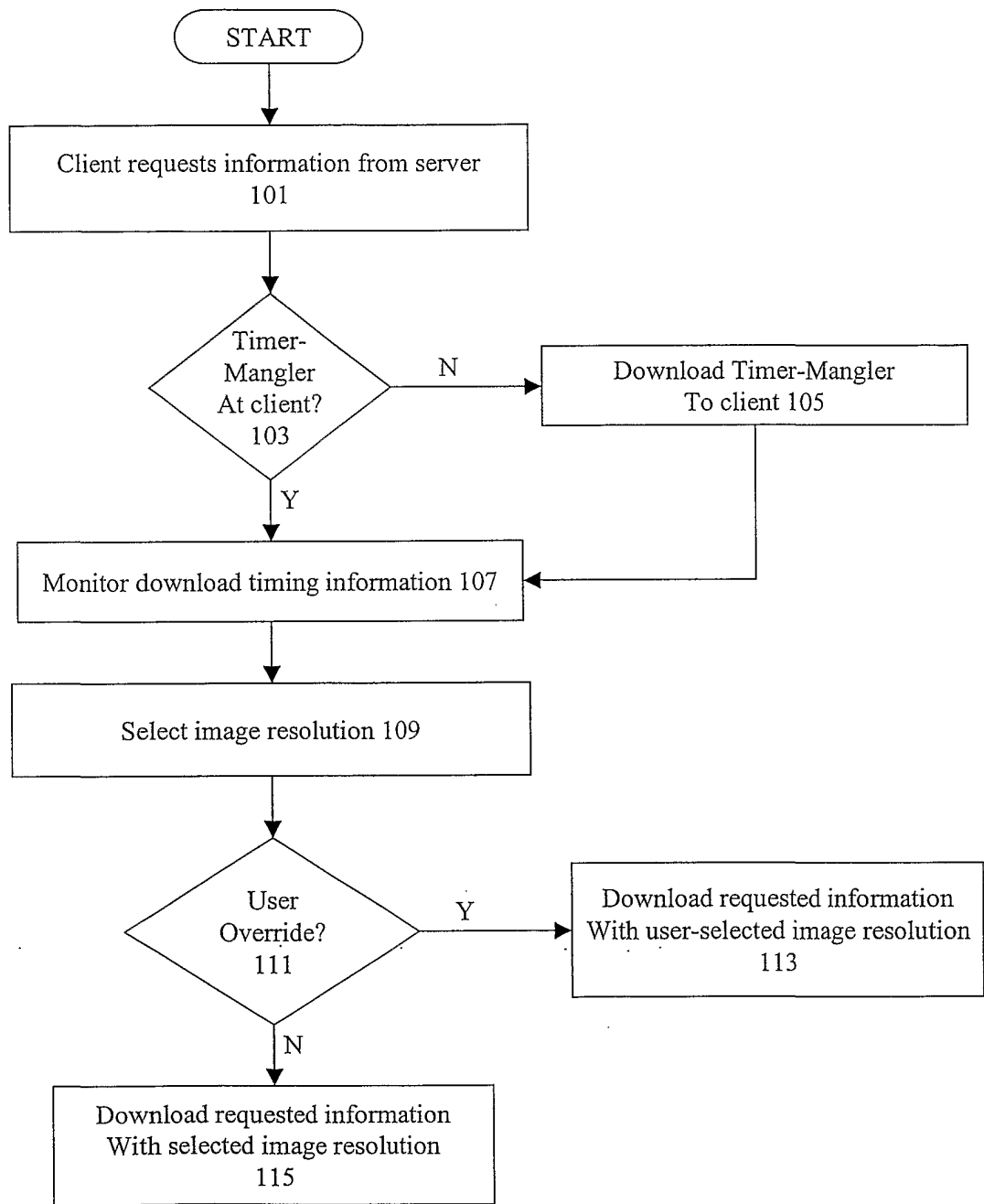
1 39. The system of claim 36 wherein the manager application comprises:
2 a front manager de-mangling a name of the requested information and
3 passing the request to the server; and
4 a back manager mangling the name of the requested information and
5 passing the request to the database.

1 40. The system of claim 36 wherein the manager application accumulates
2 hidden information regarding the client consisting of frequency and number of website
3 visits, data mining and marketing information, and forwards the hidden information to the
4 third party server or maintains logs of said information for eventual retrieval by the third
5 party server.

1 41. The system of claim 36 further comprising a local Internet protocol (IP)
2 alias and server serving as a substitute for the third party server when accessing the third
3 party server if the third party server does not include the manager application.

1 42. The system of claim 36 wherein the timer-mangler application comprises:
2 a control block receiving control information from the manager
3 application resident in the server;
4 a timer block measuring current bandwidth conditions;
5 a mangler block mangling a name of the requested information;
6 a de-mangler block de-mangling the mangled name of the requested
7 information; and
8 a local cache intervention (LCI) selecting the image resolution in
9 conjunction with the mangler block.

Figure 1



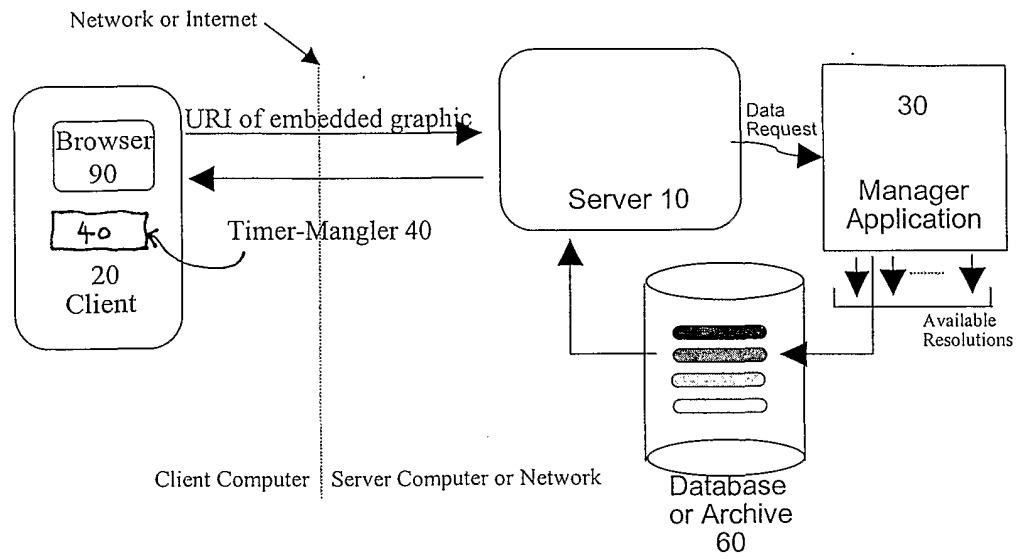


Fig 1A.

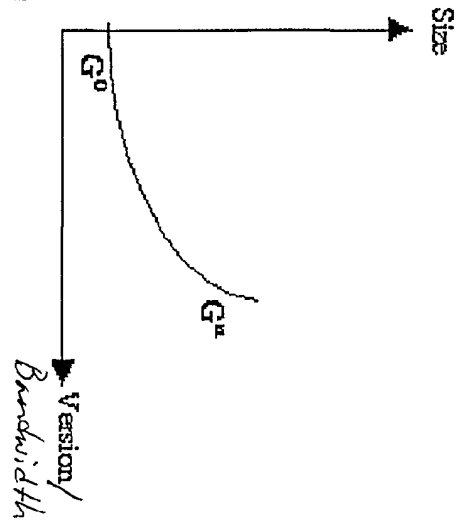
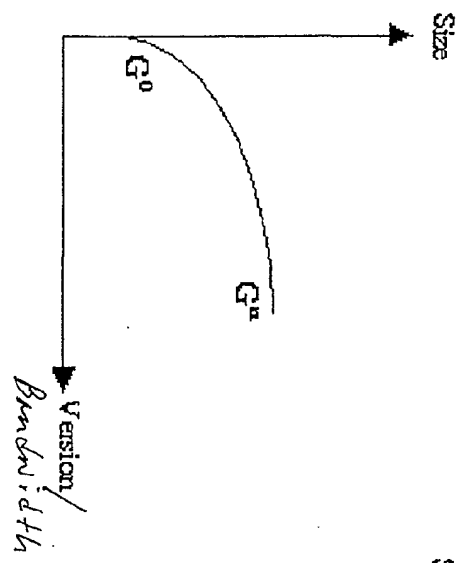
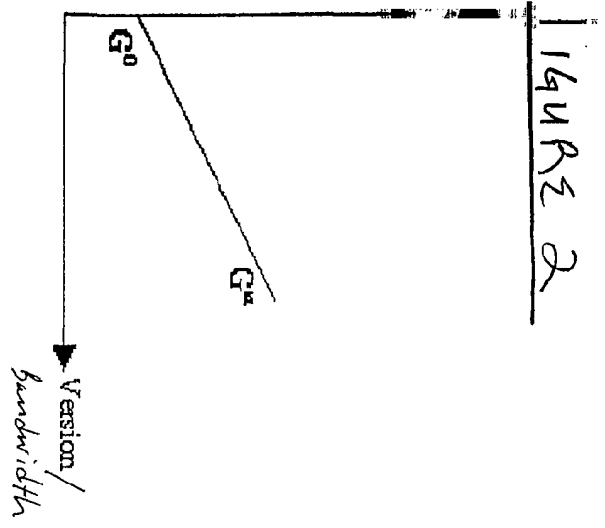


FIGURE 3

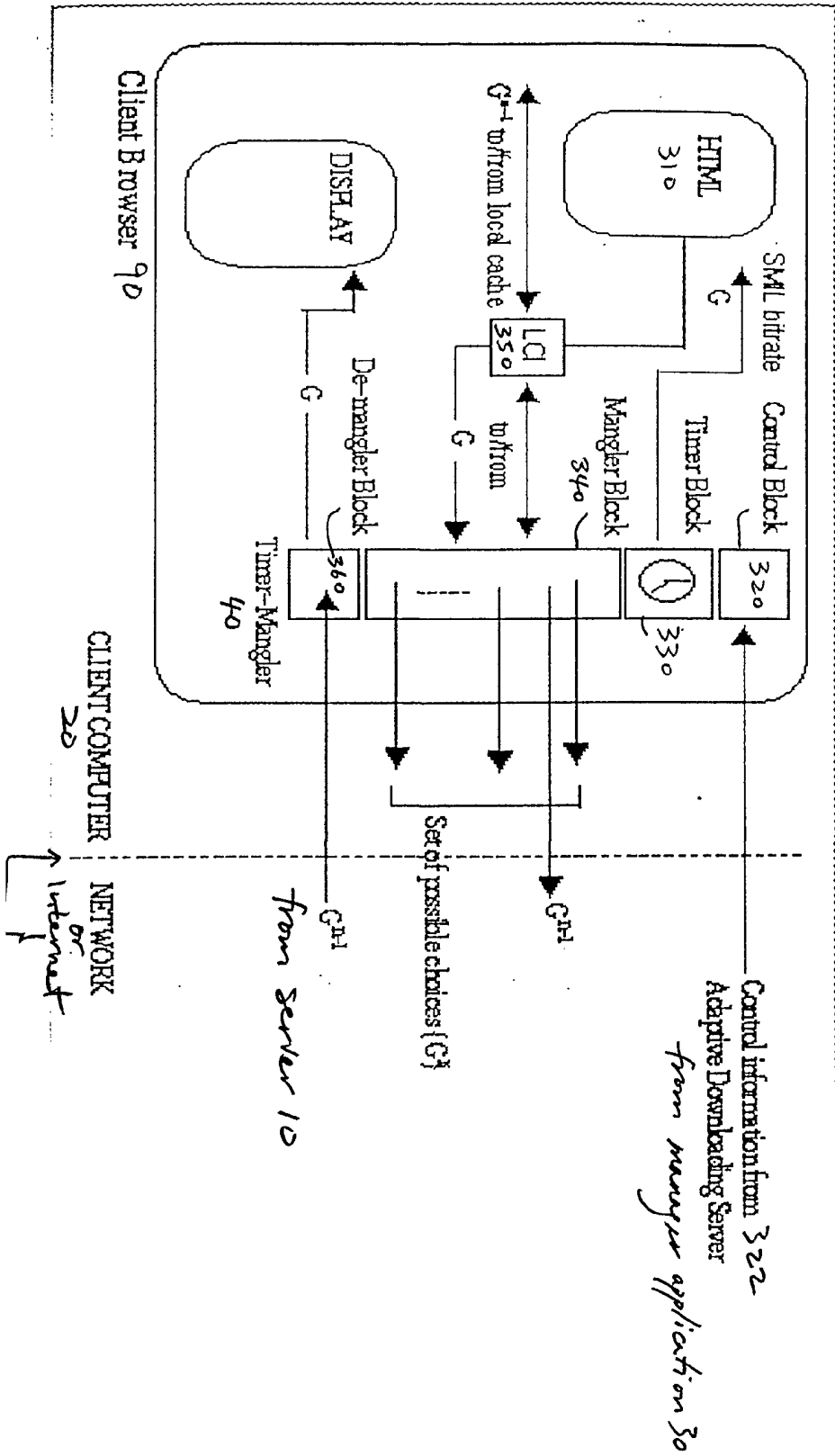


Fig 3a: Timer-Mangler Operational Flow, Out going

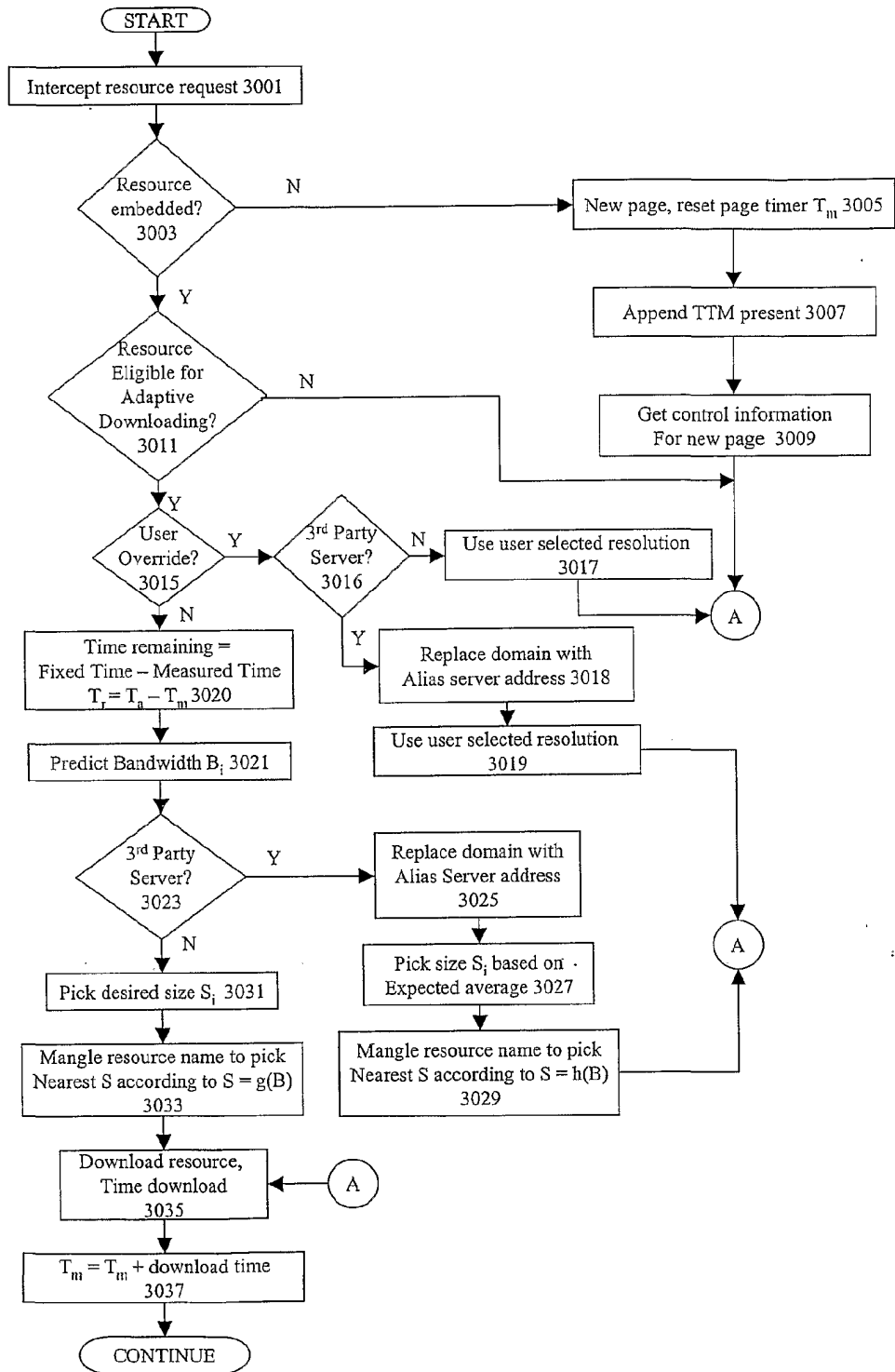
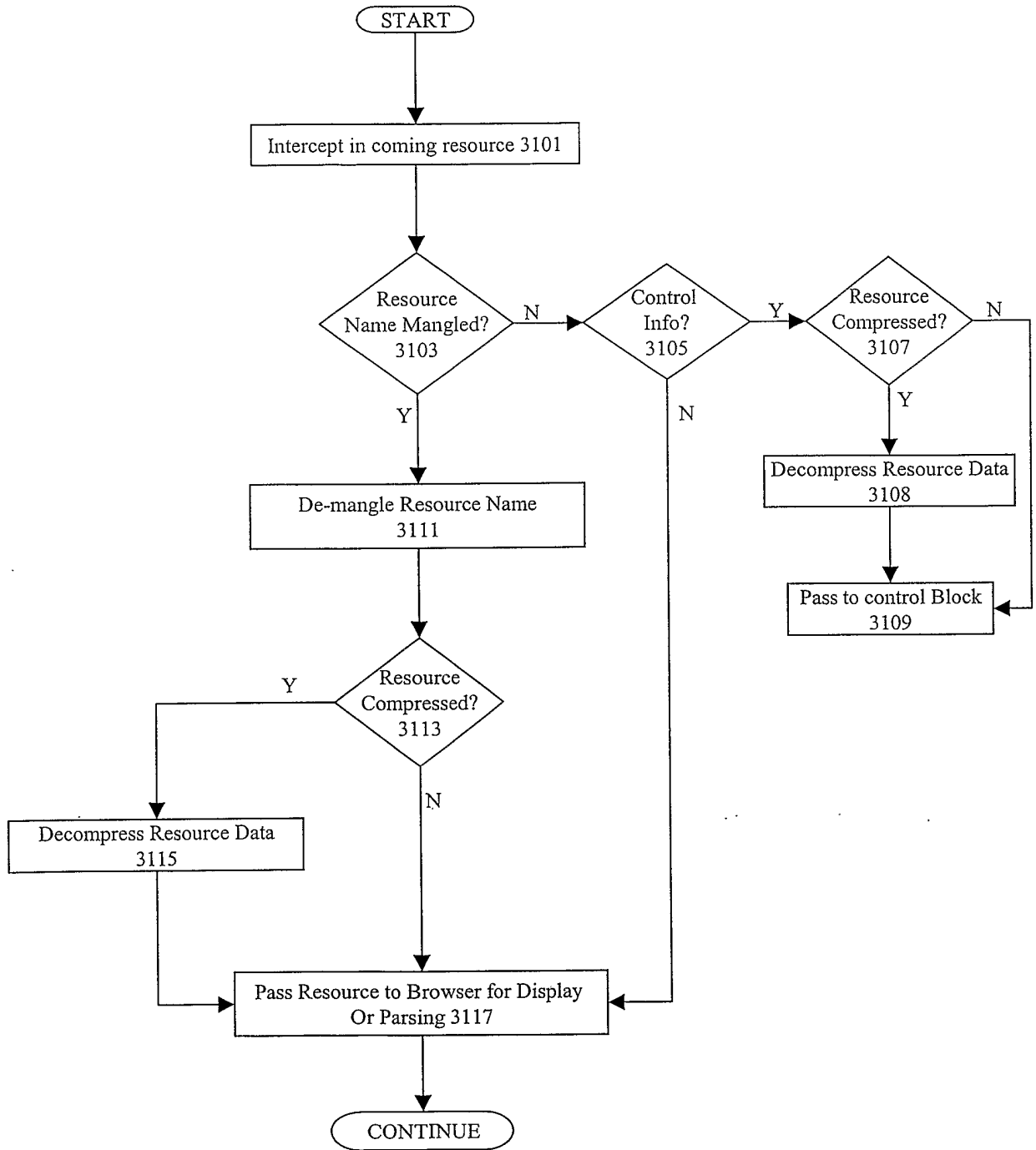


Fig. 3b: Timer-Mangler Operational Flow, In coming



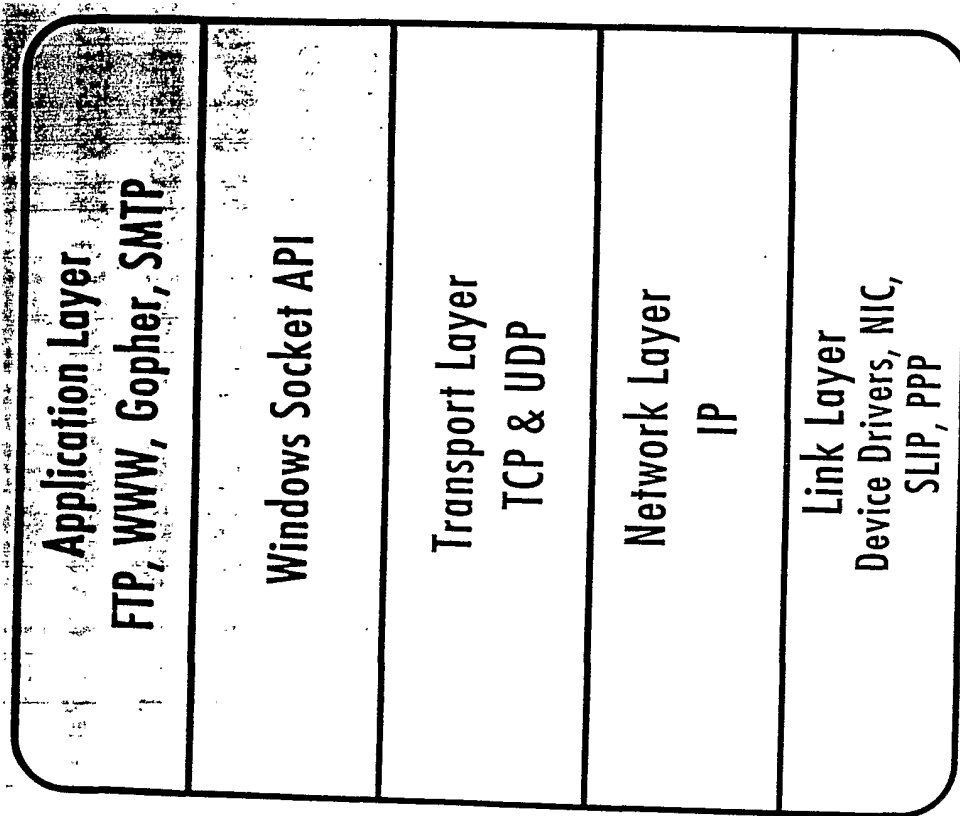


FIGURE 3C

FIGURE 4

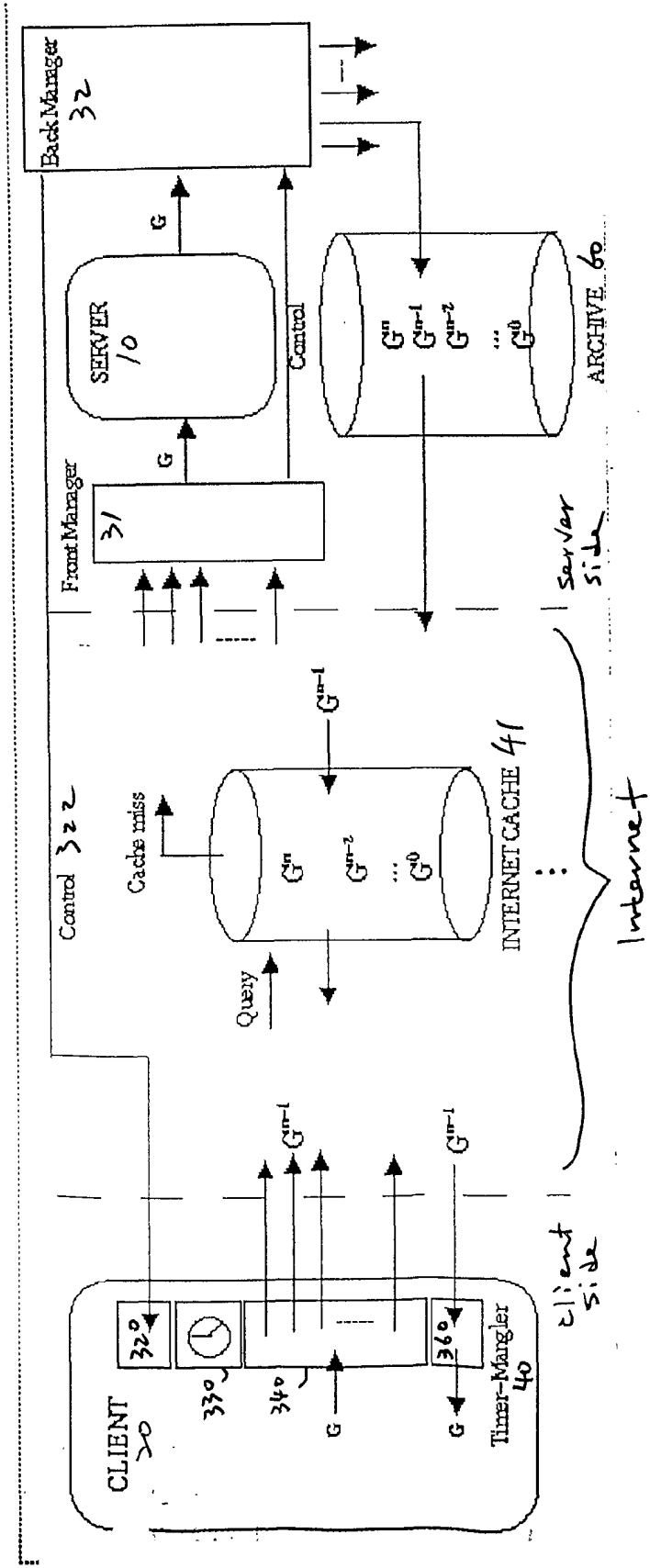


FIGURE 5

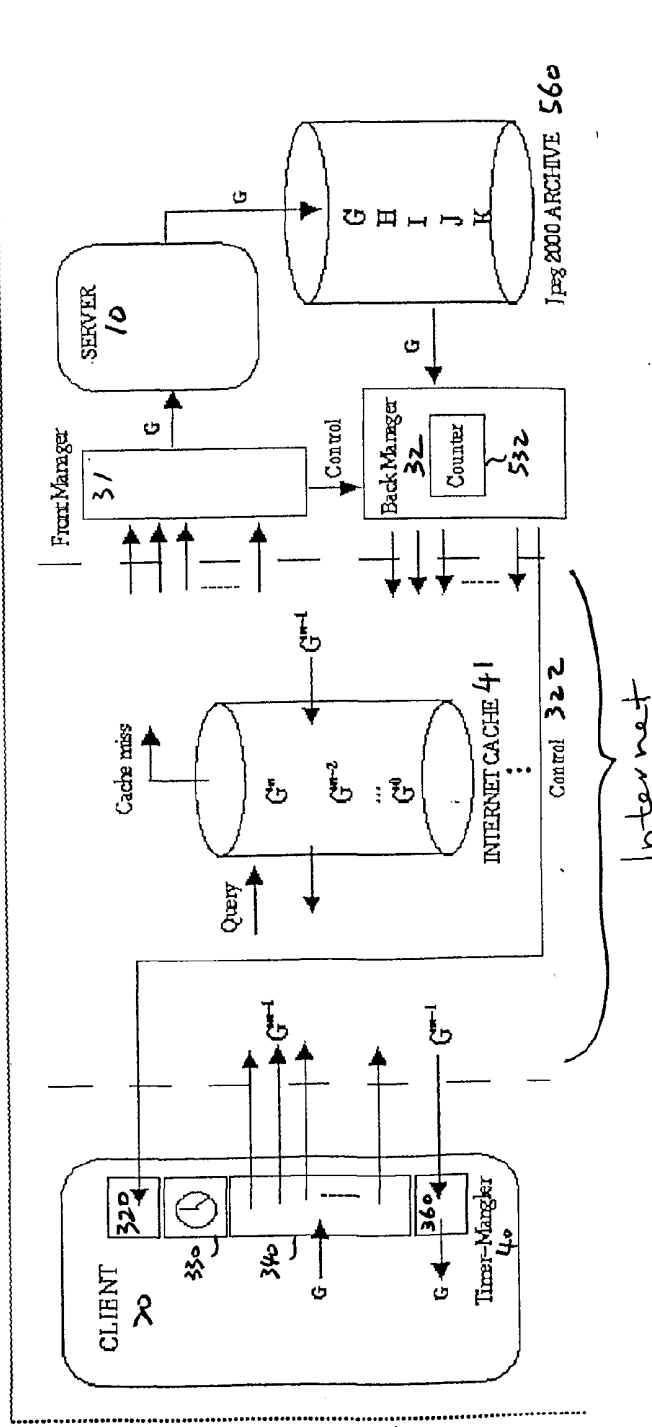


FIGURE 6

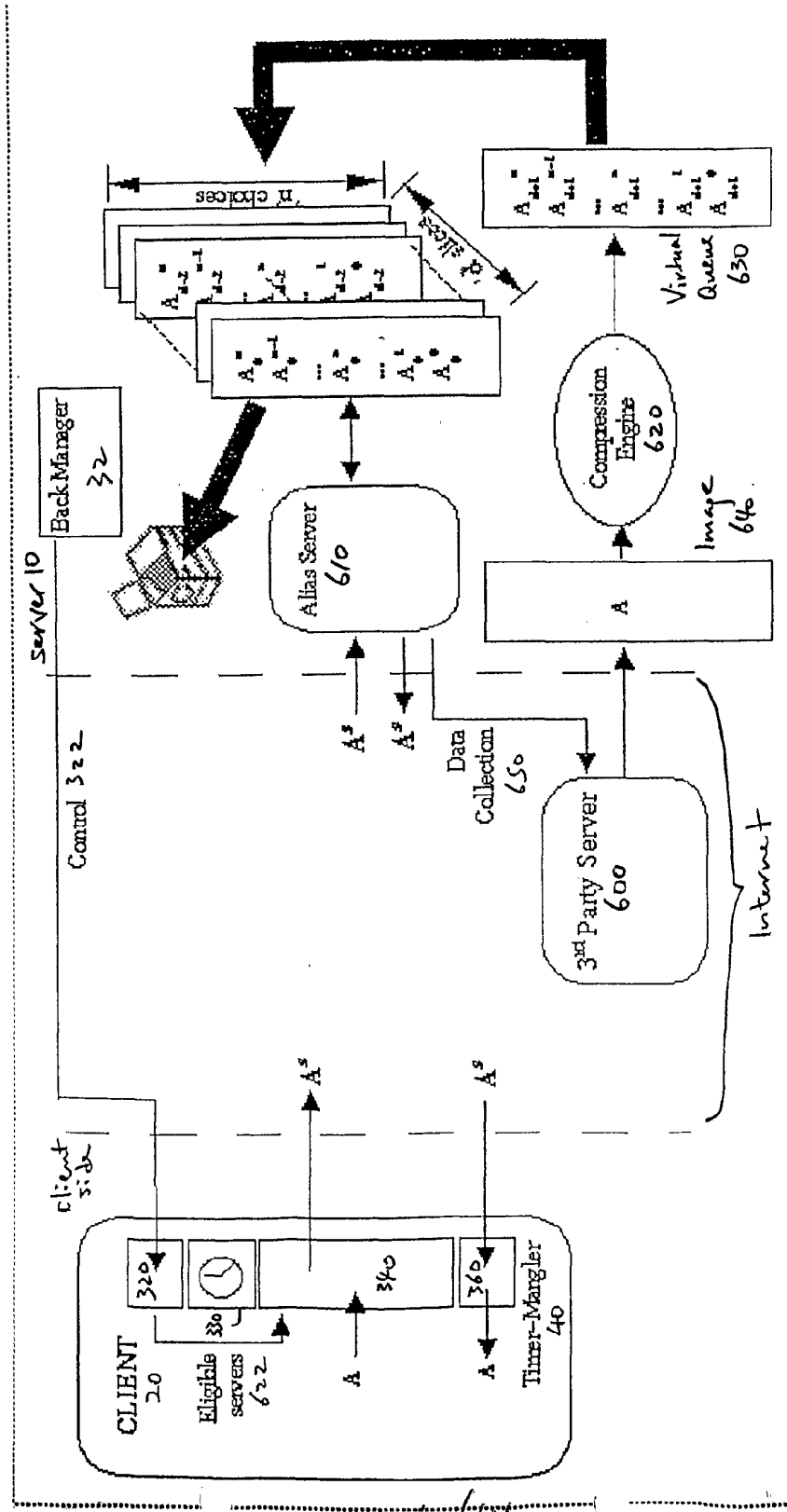


Figure 7

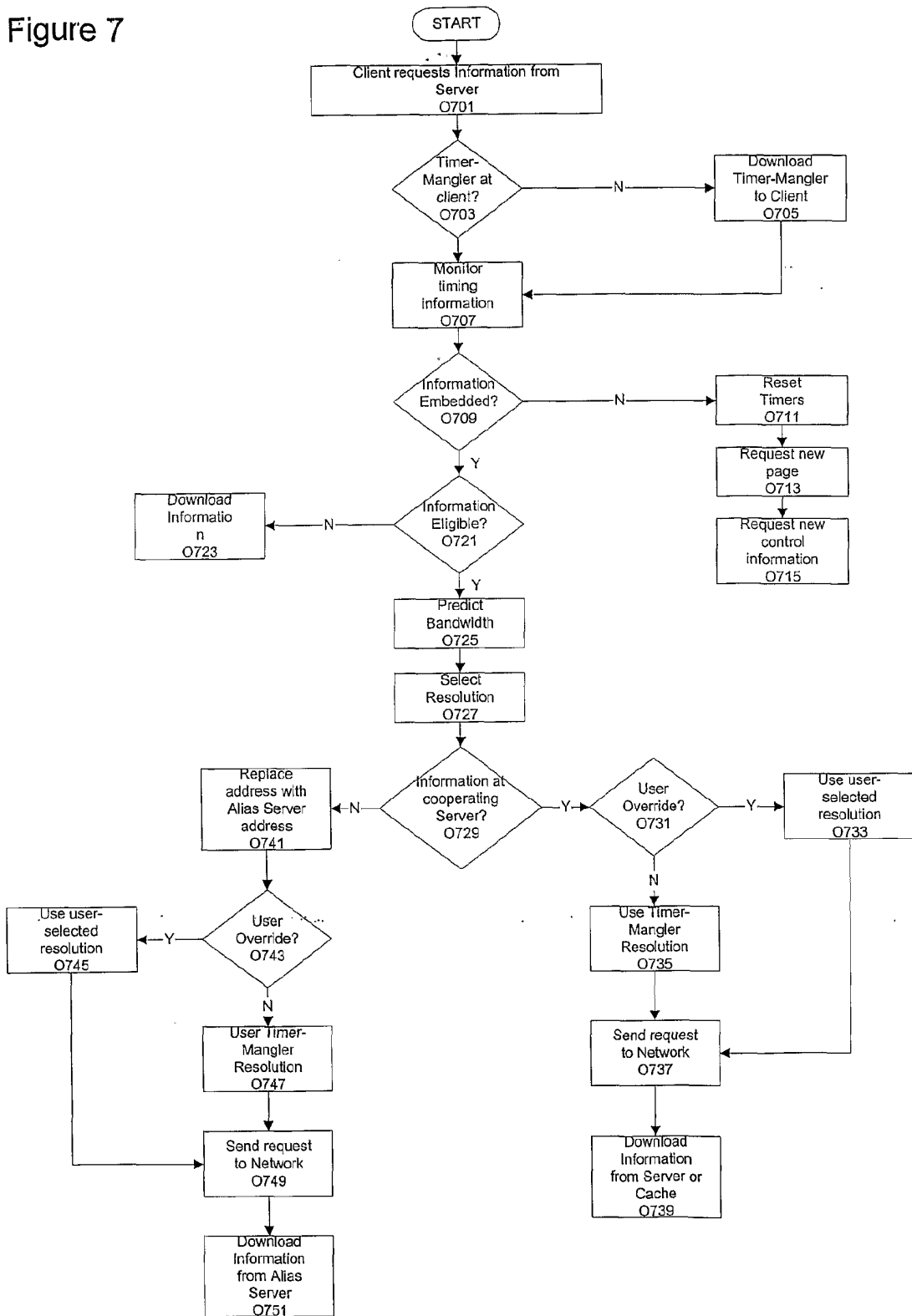


Figure 7a: Maintaining a Virtual Queue, Shared Hardware

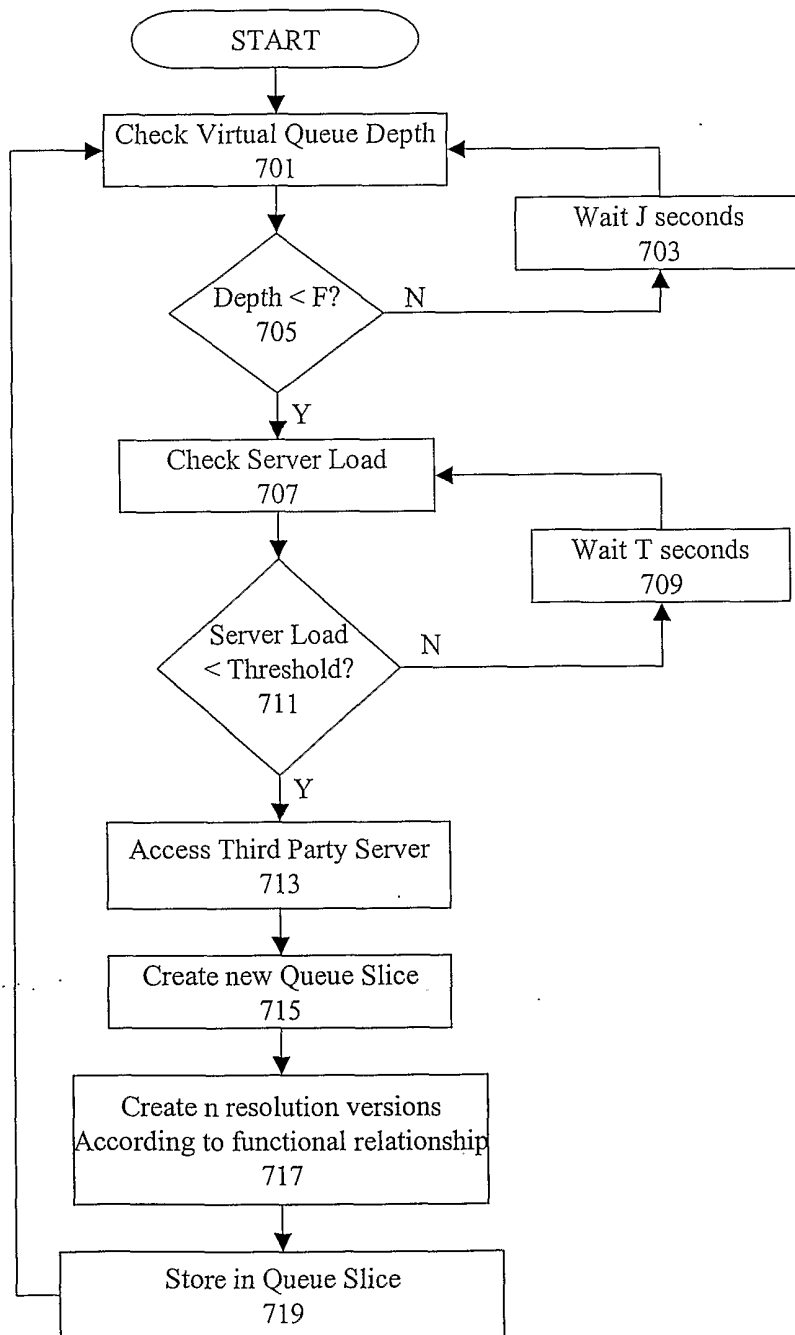
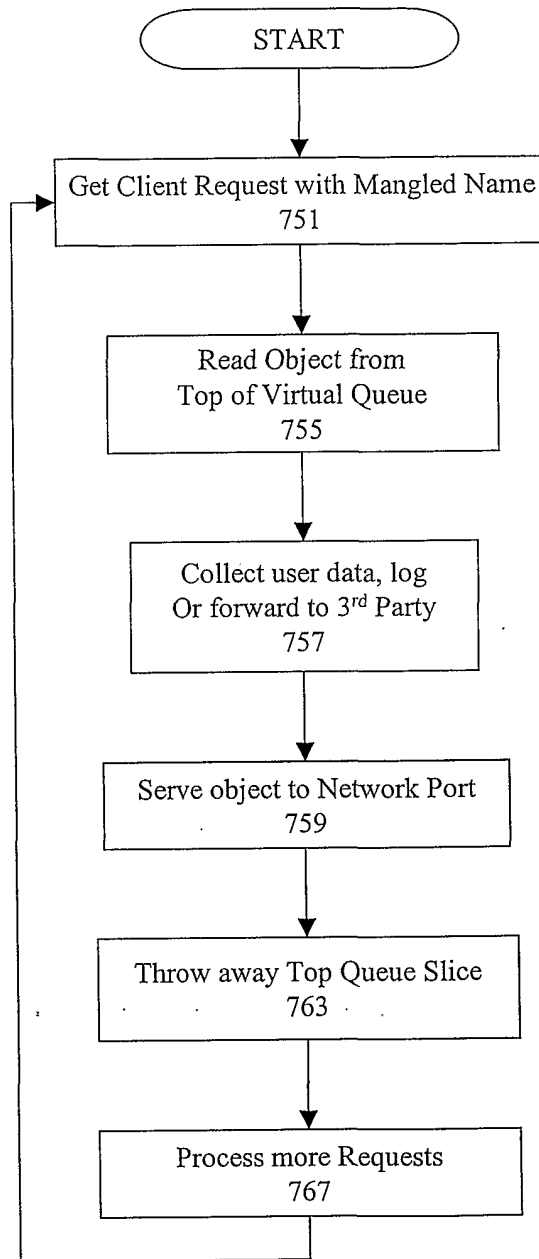


Figure 7b: The Third Party Alias Server



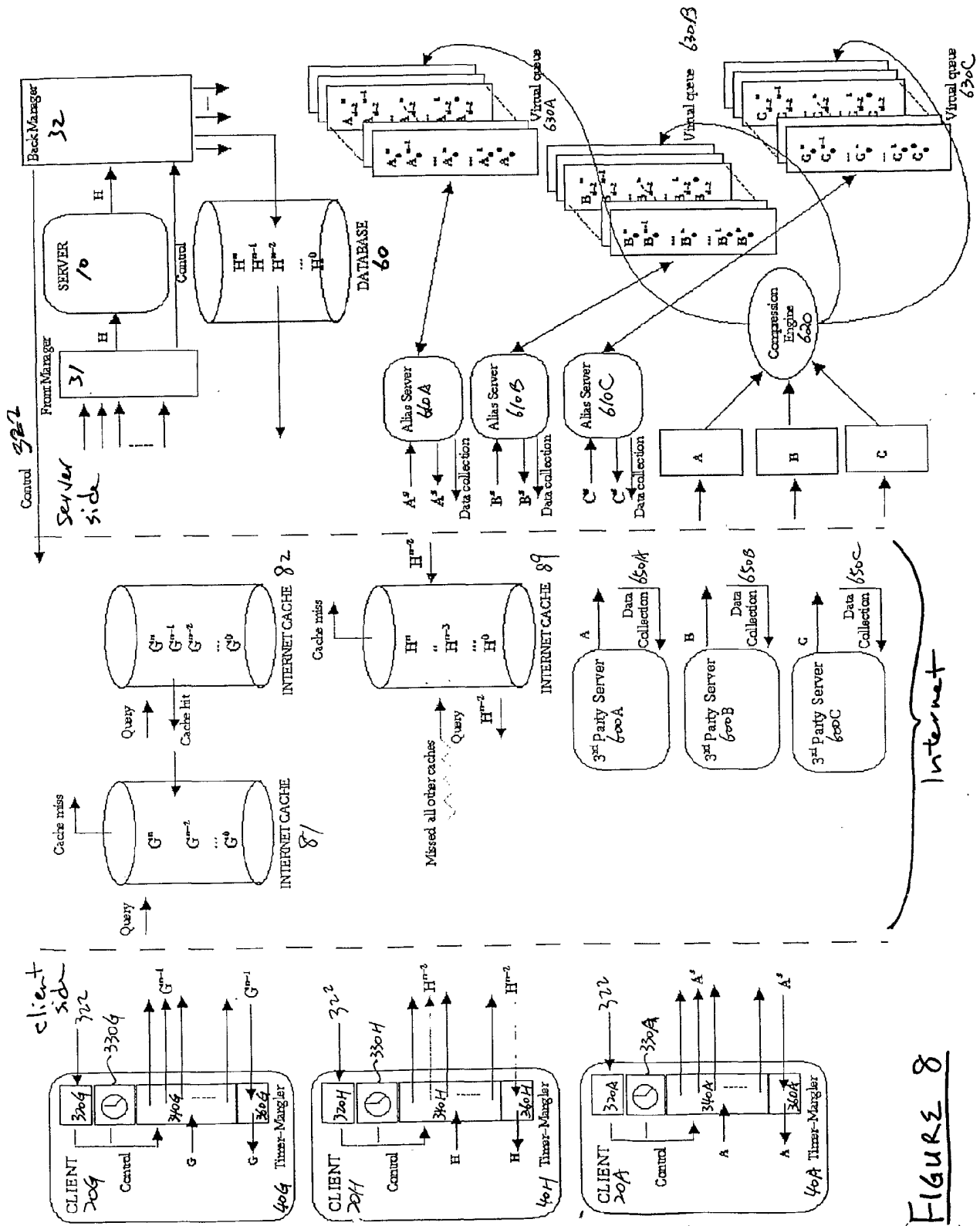


FIGURE 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/23902

| A. CLASSIFICATION OF SUBJECT MATTER | | |
|---|--|---|
| IPC(7) :G06F 15/16 US CL :709/233 According to International Patent Classification (IPC) or to both national classification and IPC | | |
| B. FIELDS SEARCHED | | |
| Minimum documentation searched (classification system followed by classification symbols) U.S. : 709/233, 232, 228, 229. | | |
| Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched NONE | | |
| Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) STN and WEST Search terms : download rate, resolutions, selection, bit rate, download time. | | |
| C. DOCUMENTS CONSIDERED TO BE RELEVANT | | |
| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
| Y | US 5,764,235 A (HUNT et al) 09 June 1998, Abstract, Col.2, Line 15 - Col. 4, Line 6. | 1-42 |
| Y,P | US 6,175,740 B1 (SOUISSI et al) 16 January 2001, Abstract, Figs. 2-4, Col. 1, Line 53 - Col. 2, Line 47. | 1-42 |
| Y,P | US 6,281,874 B1 (SIVAN et al) 28 August 2001, Abstract, Fig.1, and Col. 2, line 40 - Col. 3, Line 64. | 1-42 |
| A | US 5,983,263 A (ROYHROCK et al) 09 November 1999, See the whole reference. | 1-42 |
| A,P | US 6,243,761 B1 (MOGUL et al) 05 June 2001, see the whole reference. | 1-42. |
| <input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex. | | |
| * Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family | | |
| Date of the actual completion of the international search 18 SEPTEMBER 2001 | | Date of mailing of the international search report 01 NOV 2001 |
| Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230 | | Authorized officer AYAZ R. SHEIKH <i>James R. Matthews</i> Telephone No. (703) 305-9648 |