

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
19 September 2002 (19.09.2002)

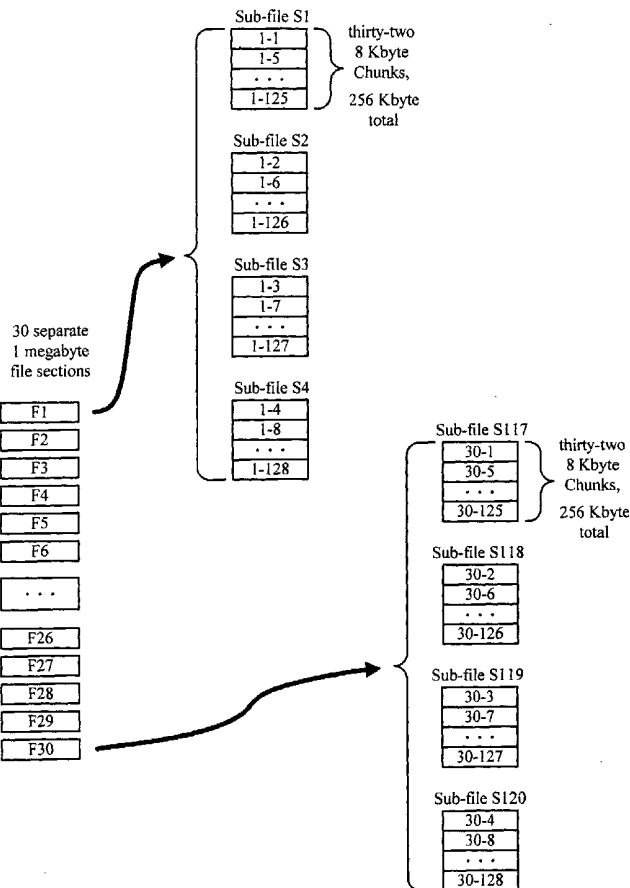
PCT

(10) International Publication Number
WO 02/073391 A1

- (51) International Patent Classification⁷: G06F 3/14, 13/00, 17/30
- (72) Inventor; and (75) Inventor/Applicant (for US only): CHUNG, Randall, M. [US/US]; 30542 Via Estoril, Laguna Niguel, CA 92677 (US).
- (21) International Application Number: PCT/US02/07658
- (74) Agent: REDJAIAN, Babak; Irell & Manella LLP, Suite 400, 840 Newport Center Drive, Newport Beach, CA 92660 (US).
- (22) International Filing Date: 12 March 2002 (12.03.2002)
- (25) Filing Language: English
- (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, 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.
- (26) Publication Language: English
- (30) Priority Data: 60/275,409 12 March 2001 (12.03.2001) US
- (84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW),
- (71) Applicant (for all designated States except US): EDGESTREAM, INC. [US/US]; Suite 231, 22931 Triton Way, Laguna Hills, CA 92653 (US).

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR STREAM CACHING



(57) Abstract: A system and method is described for stream caching content that is being provided to one or more users over a network. The streaming content may be split into a plurality of file sections (F1, F30), which may then be further split into a number of subfiles (S1, S4). These subfiles may then be streamed to a plurality of users over a network. In one embodiment, file sections may be cached on a common caching server and accessed by subsequent users without having to wait until the entire requested file has been streamed to a preceding user.

WO 02/073391 A1



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).

Published:

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

Declarations under Rule 4.17:

- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii)) for all designations*
- *of inventorship (Rule 4.17(iv)) for US only*

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.

SYSTEM AND METHOD FOR STREAM CACHING

CROSS-REFERENCES TO RELATED APPLICATIONS

The present invention claims priority to U.S. provisional patent application serial number 60/275,409, filed on March 12, 2001 and assigned to the assignee of the present application.

The present invention is related to U.S. Patent Application Serial number 09/475,807, entitled "Method and Apparatus for Data Delivery Using Distributed Storage System," filed December 30, 1999, based on Provisional Patent Application Serial number 60/146,893 filed August 2, 1999, also assigned to the assignee of the present application, the subject matter of which is incorporated herein by reference. The present invention is also related to U.S. provisional patent application serial number 60/275,408, entitled "Splitting and Redundant Storage on Multiple Servers", and U.S. Provisional Patent Application serial number 60/275,407, entitled "Re-assembly of Streaming Files from Separate Connections," both of which were filed on March 12, 2001, assigned to the assignee of the present application, and incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to data communication networks and more particularly to a method and apparatus for providing stream caching.

2. Background Information

Currently, the first user to access a streaming video for viewing will cause the video content to be cached, if the user's Internet Service Provider uses a caching server that supports the particular streaming video format. The caching server typically will store a file and its file name, as the data comes from the originating server. If the same file is requested again, and the file is still in the cache memory, the caching server will provide the data to the requestor in place of the originating server.

Present caching server are not be able to provide a copy of the file to another requestor until the entire file has been received and stored, particularly if the file is sent using the standard Hyper Text Transfer Protocol (HTTP). This can be a problem with a long video, such as an hour-long movie, or for a live broadcast. In the case of a one hour-long movie, the caching server can provide its cached copy only to viewers who start watching one hour after the first viewer has started watching. In the case of a live broadcast, it isn't possible to cache the broadcast until the broadcast is complete, which may be many hours or even days.

Accordingly, there is a need in the technology for a method and apparatus for overcoming the aforementioned problems.

SUMMARY OF THE INVENTION

The invention is a technique and apparatus for providing a cached copy to many viewers of pre-stored information such as a video or a live broadcast without having to wait until the video is complete. In one embodiment, the original file may

be divided into X file sections or pages, where each file section may be, Y bytes length, where X and Y are integers. Then each file section or page may be divided into sub-files. All of the sub-files may then be stored and possibly replicated onto one or more servers. When an end user makes a request for the playback of the file, a first group of sub-files may be simultaneously transmitted to the end user. The subsequent sub-files will then be transferred, and so forth until all sub-files have been transferred. It should be noted that the sub-files do not necessarily have to be transferred four at a time, nor is there a need to wait until all of the sub-files of a group have been transferred before transferring other sub-files. That is, as soon as one sub-file is transferred, the next one in line starts, without waiting for the other sub-files in the same "rank" to finish.

In the case where a caching server is used by the end user's Internet Service Provider, the first group sub-files may be cached and available from the caching server N seconds after the start of playback. The next group of sub-files may be cached and available from the caching server another N seconds later. Thus, subsequent end users, viewing the same file as the first end user, will receive data from the caching server as long as the viewers started viewing at least N seconds after the first viewer. Various embodiments are described.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1A is a system block diagram of one embodiment of a network system in which the system and method of the invention may be implemented.

Figure 1B is a system block diagram of another embodiment of a network system in which the system and method of the invention may be implemented

Figure 2 is a system block diagram of one embodiment of a computer system which implements the embodiments of the invention.

Figure 3A illustrates the splitting of a file according to one embodiment of the invention.

Figure 3B is an illustration of file splitting according to another embodiment.

Figure 4 is a flow diagram of one embodiment for performing the file splitting operations of Figures 3A-3B.

Figure 5 is a flow diagram of one embodiment for stream caching content consistent with the principles of the invention.

Figure 6 is a flow diagram of another embodiment for stream caching content consistent with the principles of the invention.

Figure 7 is a flow diagram of one embodiment for determining stream caching user groups.

DETAILED DESCRIPTION

One aspect of the invention relates to a system and method for caching streaming content that is being provided to one or more users over a network. In one embodiment, the content is split into a plurality of file sections, which may then be further split into a number of sub-files. The sub-files may then be streamed to a plurality of users over a network. In one embodiment, each sub-file has a file name and other file attributes, and is treated by the operating system's file system as just another file.

Another aspect of the invention relates to caching the plurality of file sections on a caching server. In one embodiment, a first file section, which may have been previously split into a plurality of sub-files, is streamed to a caching server in response to a user request. This first file section is then streamed to a first user from the caching server, according to another embodiment. Once the first user has received the completed first file section, a second file section may then be streamed to the first user over the network. If there is then a second user request, the first file section, which had previously been streamed to the caching server, is then streamed to the second user, while the second file section is being streamed to the first user.

Yet another aspect of the invention relates to providing stream caching for real-time presentations. In one embodiment, a real-time presentation is made over a network connection to an early view group and one or more common cache groups. Users may be assigned to either the early view group or one of the common cache groups depending on their shared characteristics. In another embodiment, real-time content is streamed to a caching server as a series of file section and/or sub-files. Thereafter, the content may be streamed to the early view group. After a given file section or sub-file has been streamed to the early view group, it may then be streamed to one of the common cache groups.

Definitions

As discussed herein, a "computer system" is a product including circuitry capable of processing data. The computer system may include, but is not limited to, general purpose computer systems (e.g., server, laptop, desktop, palmtop, personal electronic devices, etc.), personal computers (PCs), hard copy equipment (e.g.,

printer, plotter, fax machine, etc.), banking equipment (e.g., an automated teller machine), and the like. Content and/or media elements refers to application programs, driver programs, utility programs, file, payload, and combinations thereof, as well as graphics, informational material (articles, stock quotes, etc.) and the like, either singly or in any combination. A "communication link" refers to the medium or channel of communication. The communication link may include, but is not limited to, a telephone line, a modem connection, an Internet connection, an Integrated Services Digital Network ("ISDN") connection, an Asynchronous Transfer Mode (ATM) connection, a frame relay connection, an Ethernet connection, a coaxial connection, a fiber optic connection, satellite connections (e.g. Digital Satellite Services, etc.), wireless connections, radio frequency (RF) links, electromagnetic links, two way paging connections, etc., and combinations thereof.

System Overview

A description of an exemplary system, which incorporates embodiments of the invention, is herein described. Figure 1A shows a system block diagram of one embodiment of a network system 10 in which the system and method of the invention is used. Referring to Figure 1A, the network system 10 comprises a service center 12 that is connected over one or more communication links 20 to a remote network 30 (e.g., a wide area network or the Internet) or a remote site (e.g., a satellite, which is not shown in Figure 1A) to one or more user computer systems 40₁-40_N ("40"). In one embodiment the service center 12 is a website. The service center 12 includes one or more servers 22 and one or more databases 24. In one embodiment, the server 22 includes software modules that may be downloaded for performing the processes

of the invention, as described in detail in the following sections. In addition, more than one service center 12 may be used, each having one or more servers.

The service center 12 may also include one or more computers 26₁-26_M. If a plurality of computers are used, then the computers 26₁-26_M may be connected by a local area network (LAN) or any other similar connection technology. However, it is also possible for the service center 12 to have other configurations. For example, a smaller number of larger computers (i.e. a few mainframe, mini, etc. computers) with a number of internal programs or processes running on the larger computers capable of establishing communication links to the user computers.

The remote network 30 or remote site allows the service center 12 to provide peer-to-peer information and services to the user computers 40₁-40_N, using software that is stored at the service center 12. In one embodiment, the user computers 40 may be any computer system or any other device that processes and/or presents audio and/or video information. The one or more databases 24 connected to the service center computer(s), e.g., computer 26₁, are used to store data. Each user computer 40 is connected via network connection 32₁-32_N over a corresponding communication link 42₁-42_N such as a local carrier exchange to a respective ISP 44₁-44_N, through which access to the remote network 30 is made.

By inputting the URL address of the target website with which the user desires to interact, the user may be connected to various target websites, such as websites 50₁-50_{NN}. In an alternate embodiment, each user computer 40 may be connected via network connection 32₁-32_N over a corresponding communication link 48₁-48_N to the service center 12, which provides internet access and service to the user computer(s)

40. In a further embodiment, the display screen for viewing the content or media elements may be located on a television coupled to the network 30. In this case, navigation through the content or media elements may be provided through the use of control buttons on a remote control unit for controlling viewing of the television, or by other means known in the art.

One aspect of the present invention relates to streaming of content and/or media elements over a network connection. The software for providing such processes may be developed and/or stored on a computer 40, or may be developed using one of the computers 26₁-26_M. Upon completion of the development process, the software may be stored in the computer 40 or in the database 24. Alternatively, the software may be stored on a machine-readable medium, in any of the computers 40 or computers 26₁-26_M.

Figure 1B depicts another embodiment of the network system 10 of Figure 1A. Rather than the single server structure of Figure 1A, the network system 10 of Figure 1B has user computers 40 connected to and in communication with a plurality of service centers 12a-12d. In this embodiment, each of service centers 12a-12d includes servers 22a-22d and databases 24a-24d. While Figure 1B is shown with four service centers 12a-12d, it should be appreciated that more or fewer service centers may also be used in network system 10.

As will be discussed in more detail below, content may be streamed from one or more of service centers 12a-12d to user computers 40. In one embodiment, one or more of service centers 12a-12d stream individual segments of media content for

caching at ISPs 44₁-44_N ("44"). ISPs 44 may then provide the cached media content to user computers 40 as requested.

Referring now to Figure 2, the computer system 100 (representing either of computer 26 or 40) comprises a processor or a central processing unit (CPU) 110. The illustrated CPU 110 includes an Arithmetic Logic Unit (ALU) for performing computations, a collection of registers for temporary storage of data and instructions, and a control unit for controlling operation for the system 100. In one embodiment, the CPU 110 includes any one of the x86, Pentium™, Pentium II™, and Pentium Pro™ microprocessors as marketed by Intel™ Corporation, the K-6 microprocessor as marketed by AMD™, or the 6x86MX microprocessor as marketed by Cyrix™ Corp. Further examples include the Alpha™ processor as marketed by Digital Equipment Corporation™, the 680X0 processor as marketed by Motorola™; or the Power PC™ processor as marketed by IBM™. In addition, any of a variety of other processors, including those from Sun Microsystems, MIPS, IBM, Motorola, NEC, Cyrix, AMD, Nexgen and others may be used for implementing CPU 110. The CPU 110 is not limited to microprocessor but may take on other forms such as microcontrollers, digital signal processors, reduced instruction set computers (RISC), application specific integrated circuits, and the like. Although shown with one CPU 110, computer system 100 may alternatively include multiple processing units.

The CPU 110 is coupled to a bus controller 112. The bus controller 112 may include a memory controller (not shown) integrated therein, though the memory controller may be external to the bus controller 112. The memory controller provides an interface for access by the CPU 110 or other devices to memory 116 via memory bus 114. In one embodiment, the system memory 116 includes synchronous dynamic

random access memory (SDRAM). System memory 116 may optionally include any additional or alternative high speed memory device or memory circuitry. The bus controller 112 is coupled to a system bus 120 that may be a peripheral component interconnect (PCI) bus, Industry Standard Architecture (ISA) bus, etc. Coupled to the system bus 120 are a graphics controller, a graphics engine or a video controller 132, a mass storage device 150, a communication interface device 152, and one or more input/output (I/O) devices 168₁-168_N. The video controller 132 is coupled to a video memory 136 (e.g., 8 Megabytes) and video BIOS 140, all of which may be integrated onto a single card or device, as designated by numeral 144. The video memory 136 is used to contain display data for displaying information on the display screen 148, and the video BIOS 140 includes code and video services for controlling the video controller 132. In another embodiment, the video controller 132 is coupled to the CPU 110 through an Advanced Graphics Port (AGP) bus.

The mass storage device 150 includes (but is not limited to) a hard disc, floppy disc, CDRW-ROM, DVDRW-ROM, tape, high density floppy, high capacity removable media, low capacity removable media, solid state memory device, etc., and combinations thereof. The mass storage device 150 may include any other mass storage medium. The communication interface device 156 includes a network card, a modem interface, etc. for accessing network 164 via communications link 160. The I/O devices 168₁-168_N include a keyboard, mouse, audio/sound card, printer, and the like. The I/O devices 168₁-168_N may be disk drive, such as a compact disc (CD) drive, a hard disc drive, a tape drive, a zip drive, a jazz drive, a digital versatile disc (DVD) drive, a magneto-optical disk drive, a high density floppy drive, a high

capacity removable media drive, a low capacity media device, and/or any combination thereof.

The information, content and/or application software may be stored in the database 24, on the computers 40, or on a machine-readable medium. The information, content and/or application software may also be made available to users such as those located at user computer 1 – N, i.e., computers 40₁–40_N, through service center 12 or by means of the machine-readable medium.

In accordance with the practices of persons skilled in the art of computer programming, the invention is described below with reference to symbolic representations of operations that are performed by a computer system or a like electronic system. Such operations are sometimes referred to as being computer-executed. It will be appreciated that operations that are symbolically represented include the manipulation by a processor, such as a central processing unit, of electrical signals representing data bits and the maintenance of data bits at memory locations such as in system memory, as well as other processing of signals. The memory locations where data bits are maintained are physical locations that have particular electrical, magnetic, optical, or organic properties corresponding to the data bits. Thus, the term “server” is understood to include any electronic device that contains a processor, such as a central processing unit.

Referring now to Figure 3A, in which an original file 300 is shown being split into file sections F1-F30. In this embodiment, original file 300 is 30 Mbytes large and is split into 30 individual file sections each being 1 MByte long. However, it

should be appreciated that original file 300 may be any number of bytes in size and may be split into more or fewer file segments.

In one embodiment, software may take continuous data from a camera and form the short file segments (e.g., file sections F1-F30). The software could run on a computer, such as computer 26 or user computer 40, in the vicinity of the camera, and may be connected via a digital or analog video capture card. In another embodiment, this computer performs the segmenting, as well as the video capture and compression functions. In yet another embodiment, the segmenting software is executed on one of servers 22a-22d. It should be appreciated that other configurations may also be obvious to those skilled in the art.

Once original file 300 is split into file sections F1-F30, the file sections may be further split into a plurality of sub-files, as shown in Figure 3B. One example of a method for splitting content into sub-files is provided in related PCT Application No. PCT/US___/_____, entitled "Splitting and Redundant Storage on Multiple Servers," which describes the division of a single file into multiple sub-files which are then stored on one or more servers. Such a file may include text, data, graphics, video clips, JPEG elements or images, static photographs, web pages, audio clips, animation, any type of informational material or any combination thereof. The servers may be distributed over different geographical locations. In one embodiment, the sub-files can be transmitted in parallel from one or more servers in response to a user request, thereby increasing the rate at which data can be delivered.

In the embodiment of Figure 3B, each file section is further subdivided into four sub-files, where the first sub-file S1 holds the first 8 Kbyte block ("1-1"), the

second sub-file S2 holds the second 8 KByte block (“1-2”), and so on. Since there are only four sub-files, the fifth 8 KByte block would be directed back to the first sub-file S1, and the assignment process would repeat in this fashion until all file section chunks have been assigned to one of sub-files S1-S4. Although Figure 3B is depicted with each of the file sections F1-F30 being split into four sub-files, it should be appreciated that the file sections may similarly be split into more or fewer sub-files.

It should further be noted that the above-described file splitting procedure may be used even if the last file section is smaller than the other sections. By way of a non-limiting example, if the data for sub-file S120 was only 1 KByte long, seven Kbytes of a null value (e.g., zero) could be used to “pad” the data out to a length of 8 Kbytes. Similarly, the remaining three chunks of 8 Kbytes could be formed using null-valued bytes.

One process for splitting original file 300 into a plurality of sub-files is provided in Figure 4. In particular, process 400 starts with a determination at block 405 of how long to make each file section. For example, in the embodiment of Figure 3A-3B, each file section F1-F30 is 1 MByte long. In addition, the length (Y) of the file sections may be a function of the size of the original file 300 and/or a function of how many sub-files, if any, the file sections will be split into. Once the length of the file sections has been determined, the number of file sections (X) can then be determined by simply dividing the length of the original file 300 by the file section length determined at block 405. The original file 300 is then split into X file sections at block 410. While in the embodiment of Figures 3A-3B “Y” was 1 MByte and “X” was 30, X and Y may be greater or less than these values.

The number of sub-files (M) per file section is then determined at block 415. While in the embodiment of Figures 3A-3B each file section was divided into four sub-files, it should be appreciated that there may be more or less sub-files per file section. At block 420, each file section is divided into Z segments, where Z is any integer and may be a function of the length of each segment (e.g., 8 Kbytes) and the number of sub-files (M). By way of example, Z was equal to 128 in the embodiment of Figures 3A-3B. This is due to the fact that each file section F1-F30 was divided into four sub-files S1-S4, where each sub-file was made up of 8 KByte segments. Thus, thirty-two 8 KByte segments (256 Kbytes total) would make up each of the four sub-files, for a total of 1 MBytes for each file section.

Once the file sections have been divided up into Z segments each, a counter variable is set to 1 at block 425. In another embodiment, counter variable n is set at any point along process 400 prior to block 430. At block 430, the first Z segments are distributed to the first sub-file group (e.g., S1-S4), which corresponds to the first file section. In one embodiment, the manner of distribution corresponds to the manner of distribution depicted and described above with reference to Figure 3B. In any event, the term "sub-file group" will be used to refer to that set of sub-files that comprise a particular file section. Thus, in the embodiment of Figure 3B, the first sub-file group would be comprised of sub-files S1-S4, while the final sub-file group is comprised of sub-files S117-S120.

At decision block 435, variable n is checked to see if it is equal to X , where X is the previously determined number of file sections (see block 410). In one embodiment, variable n equals X when all of the file section segments have been assigned to their corresponding sub-file groups. However, until then, decision block

435 directs process 400 to block 440, where variable n is incremented. The next Z segments may then be distributed to the n th sub-file group at block 445.

Once $n=X$ and all of the file section segments have been assigned to corresponding sub-file groups, process 400 continues to block 450 where the various sub-files may be loaded onto more than one server. For example, to provide storage redundancy it may be desirable to store one or more of sub-files S1-S120 on more than one server, such as servers 22a-22d. Providing storage redundancy may enable a sub-file to be streamed to a given user in parallel and simultaneously from one or more servers, thereby increasing the data delivery rate. In another embodiment, sub-files are replicated so that the most frequently accessed files and sub-files are replicated onto more servers, while the less frequently accessed files and sub-files are replicated onto fewer servers. The sub-files may be replicated onto multiple servers based on anticipation or actual demand.

Referring now to Figure 5, in which a process 500 for providing stream caching to a plurality of users accessing the same caching server is depicted. Where a caching server is used by the end user's Internet Service Provider ("ISP"), a portion of the original file 300 may be cached and available from the caching server some period of time after the start of playback. For illustrative purposes only, this period of time may be 8 seconds. Moreover, while this period of time will be referred to below as being 8 seconds long, it should be appreciated that it may obviously be longer or short.

This caching function may occur without explicit instruction to the ISP 44, where ISP 44 caches standard files transmitted using the HTTP protocol. In one

embodiment, the first four sub-files S1-S4, corresponding to the first file section F1, may be streamed to a caching server of user ISP 44. Thereafter, the next four sub-files S5 through S8 may be cached and available from the caching server another 8 seconds later. It should be appreciated, however, that this period of time may be more or less depending on the quality of the network connection and/or the size of the file sections being provided to the ISP 44. Subsequent users, desiring to view the same file as the first user, may be streamed the content without delay from the caching server as long as the subsequent viewer started viewing at least 8 seconds after the first viewer. This is due to the fact that, after 8 seconds, at least the first file section will have been cached by the ISP, according to one embodiment.

Where the original file 300 is not segmented or split as described herein, subsequent users may not be able to take advantage of the caching server until the entire file has been viewed by the first viewer. In the example of the original file 300, which contains 30 file sections each taking approximately 8 seconds to stream, it would take approximately 240 seconds to stream the unsplit original file 300 to a user. Thus, a subsequent user would have to wait approximately 240 second, as opposed to 8 seconds, before they would begin to receive streaming content from their ISP. In this manner, the invention may be used to reduce the delay in receiving streaming content from a caching ISP server.

Continuing to refer to Figure 5, process 500 begins with a first user request for content presentation at block 505. Variables n and m are set to a starting value, which in the embodiment of Figure 5 is a value of 1. At block 510, the n th file section of original file 300 is streamed to user ISP 44 from one of servers 22a-22d. Where ISP 44 uses caching servers, the n th file section is cached by the ISP 44 at block 515.

Thereafter, the cached *n*th file section may be streamed to the user from the ISP at block 520. Since the *n*th file section must be first streamed from to ISP 44, and then from ISP 44 to the user, some delay will exist between when the first user makes the request at block 505 and when the user actually begins receiving the *n*th file section at block 420. It should be appreciated that this delay may be a function of the quality of the network connection, the size of the *n*th file section, and the type of caching server. In one embodiment, a caching server will simultaneously cache the data and transmit the data. As a result, there will be a very slight delay going through a caching server. It is only after the entire file has been received correctly that the file is committed to cache and will then be available for subsequent use. It should also be appreciated that content may be streamed to a caching ISP 44 and/or a user one file section at a time, more than one file section at a time, or a portion of a file section at a time. For sake of brevity, the following discussion will assume that content is streamed one file section at a time, although this need not be the case.

At decision block 530, a determination is made as to whether the counter variable *n* is equal to *X*, where *X* is the number of file sections determined at previous block 410. When $n=X$, all file sections will have been streamed to the first user and process 500 will proceed to decision block 555. However, until $n=X$, process 500 will move to block 535 where the counter variable *n* is incremented.

At block 540, the next file section is provided to the ISP 44. Again, where the ISP 44 uses a caching server, the *n*th file section streamed at block 540 is cached at block 545. As with the first file section, the cached *n*th file section may then be streamed to the user at block 550.

At this point, decision block 555 makes a determination as to whether a second user has requested the content. If no additional user has made such a request, the process 500 returns back to decision block 530 to determine if the first user has received all of the file sections for the requested content. Where a second user is determined to have made a content request, process 500 moves to block 560 where the m th file section is streamed from the ISP to the second user. Since m has not yet been incremented (as n was at block 535), the second user will be receiving a file section which has already been streamed to the first user. In other words, when the second user makes its request immediately following the first user, $m = n - 1$ and the second user will be streamed the file section immediately preceding the file section currently being streamed to the first user. It should further be appreciated that, when the second user makes a content request after the first user has been streamed a plurality of file sections, the second user will still be streamed the content beginning at the first file section. In this case, the content streaming to the first user will be completed in advance of the streaming to the second user. When this occurs, decision block 530 will relay process 500 down to block 560 until m also equals X and the second user has also received all of the file sections.

While not shown in Figure 5, the same process 500 may be applied when more than two users request the same content, such as original file 300, from the same caching server. For example, decision block 555 may be modified to determine when any additional user makes a request for the same content. Cached content may then be streamed to any additional users one file section at a time. Where the additional users are connected to network 30 through the same caching ISP 44, previously cached file sections may be streamed to the additional users in the order

such requests are made, or according to some other criteria. In this manner, subsequent users requesting the same content from the same server need not wait until all of the content has been streamed to a previous user. Rather, subsequent users requesting content need only wait until a given file section be streamed to a previous user, at which point that cached file section will be available for streaming to another user.

Process 500 terminates when decision block 570 indicates that $m=X$, meaning that the second user has received all file sections of original file 300. Where more than two users have made requests as is likely, process 500 would terminate when the latest requester has been streamed all of the requested content.

As described above, the first user to request content will cause the content to be cached by a caching server. In this manner, the first user causes a given file to be cached for use by subsequent users. However, for "live" content, all users typically will be watching the same content at the same point in time. Thus, with live content there is no "first user" to cause the content to be cached for subsequent users since all users are receiving the live content at roughly the same time. While one alternative solution to caching live content for many simultaneous users is to use the multicast Internet protocol, not all Internet routing equipment supports multicast.

As mentioned above, one aspect of the invention is to provide a system and method for stream caching real-time presentations. In one embodiment, a real-time presentation is made over a network connection to an early view group and one or more common cache groups. Users may be assigned to either the early view group or one of the common cache groups depending on their shared characteristics. In

another embodiment, real-time content is streamed to a caching server as a series of file section and/or sub-files. Thereafter, the content may be streamed to the early view group. As will be described in more detail below with reference to Figure 6, after a given file section or sub-file has been streamed to the early view group, it may then be streamed to one of the common cache groups.

Referring now to Figure 6, in which a process 600 for stream caching live content is described. The live content "broadcast" begins at block 605. It is at this point that one or more of service centers 12 stream content over network 30. In addition, variables n and t are each set to 1, where n is used to indicate the current file section and t is used to indicate a particular point in time. Thereafter, users are assigned to either an early view group or a common cache group, as determined at block 610. One process for determining which users are placed into an early view group or a common cache group is described below with reference to Figure 7.

Once all users have been properly assigned, process 600 continues to block 615 where the n th file section (the 1st file section) is streamed to the common caching server at $t=1$. At decision block 620, a determination is made as to whether the live content/broadcast has ended. If so, the currently cached file section, which at this point is the 1st file section, is streamed to the early view group(s) at block 625 and $t=2$. The currently cached file section may thereafter be streamed to the common cache group(s) at block 630 and $t=3$, after which the process 600 ends since all the users will have received all of the broadcast content.

If, on the other hand, it is determined at decision block 620 that the live content has not ended, process 600 proceeds to block 635 where the current file

section is streamed to the early view group at $t=2$. In addition, the variable n is incremented. Still at $t=2$, the new current file section (which at this point is the 2nd file section) is streamed to the common caching server at block 640. Moving to $t=3$ and to block 645, a determination is made as to whether the live broadcast has now ended. If so, process 600 will proceed to block 625 where the n th file section (which at this point would be the 2nd file section) is streamed to the early view group(s) at $t=3$. The process 600 would then proceed to $t=4$ and to block 630 where the current file section would be streamed to the common cache group. As mentioned above, the process 600 would end at this point since all the users would have received all of the broadcast content.

If it is determined at block 645 that the broadcast has not ended (which at this point would mean that the broadcast is more than two file sections long), process 600 continues to block 650 at $t=3$. At block 650, the $n-1$ file section (i.e., the 1st file section) would be streamed to the common cache group(s), the n th file section (i.e., the 2nd file section) would be streamed to the early view group(s), and the $n+1$ file section (i.e., the 3rd file section) would be streamed to the common caching server, all at roughly the same time ($t=3$).

It may be desirable to keep all users, whether in an early viewer group or a common cache group, in sync such that they all receive the same content at the same time. Even though an early view group would normally receive a given file section before those in a common caching group, it is possible to delay the streaming display of the content such that both groups would receive content with the same total delay, and thus both groups would watch or listen to the same portion of the content at the same time.

Continuing to refer to Figure 6, process 600 would then continue to block 655 where users may be reassigned to a different group than what they were initially assigned to. It may be desirable to dynamically reassign users from one group to another when, for example, the number of users in an early view group dwindles down to a low level. In such a case, one or more users from a common caching group may be assigned to the early view group to ensure that the flow of content continues uninterrupted. The functionality of reassigning users will be discussed in greater detail below with reference to Figure 7.

At this point, process 600 returns to decision block 645 to determine if the end of the broadcast has been reached. If so, process 600 proceeds to blocks 625 and 630 where, as mentioned before, the final cached file section is streamed to both user groups. If the broadcast has not ended, the events of block 650 are repeated for the next file section. In this manner, a sustained level of stream caching is achieved in which various sequential portions of content are being streamed to a common caching server, to an early view group and to a common caching group.

Figure 7 is a flow diagram of one embodiment for a process of assigning users to either early view group(s) or common caching group(s). Process 700 begins with the issuing of a content request for the *i*th user for particular live content. This request may be made by having a user on user computer 40 make a selection from one of target websites 50. Alternatively, the user may issue the request to one of servers 22a-22d. It should be appreciated that other configurations may also be used. In one embodiment, the content request at block 405 includes the user's Internet Protocol ("IP") address and port number.

Once there has been a user content request, process 700 moves to decision block 710 where a determination is made as to whether there are any current users with the same IP address as the requesting user. Users with the same IP addresses but different port numbers are typically sharing a common caching server. In one embodiment, one of servers 22a-22d maintains a log of all users and their IP addresses. As new users make content requests, the server may reference the user log to determine if there are any matches to the IP address of the requesting user. If there is no match, process 700 moves to block 715 where the *i*th user is assigned to an early view group. As discussed above with reference to Figure 6, users in an early view group receive content prior to users in a common cache group.

After the assigning the *i*th viewer has been assigned per block 715, process 700 moves to decision block 735 where a determination is made as to whether another user has issued a content request for the particular live content. If not, process 700 loops until there is an additional request. Once the subsequent user makes the content request, process 700 moves to block 740 where the user counter variable is incremented. Process 700 then moves back to block 705, where the content request is made by the new user.

Referring now back to block 710, if there was a match to the requesting user's IP address, process 700 would move to block 725, where the *i*th user would instead be assigned to the corresponding common cache group. Since there would be a match to the *i*th user's IP address, there would be at least two users accessing network 30 using the same caching server. Moreover, since the first user with a particular IP address is assigned to the early view group at block 715, a subsequent user with the same IP address may be automatically assigned to the common cache group. The

content request by the early view group user(s) will cause the live content to be cached by the common caching server, and subsequently available for streaming to the common cache group.

It may be undesirable to have only one user assigned to the early view group since content caching may be interrupted if that early user stops streaming the content. To this end, process 700 makes a determination at decision block 730 as to whether the i th user, who has been already assigned to the common cache group, should be re-assigned to the early view group. In one embodiment, every k th user in the common cache group is assigned to the corresponding early group, where k is any integer. In one embodiment, k equals 10 such that every 10th user in the common cache group is re-assigned to the early view group. However, k may be any other number of users. In another embodiment, the number of users to be re-assigned to the early view group may be adjusted dynamically as a function of the total number of users requesting the live content.

Where it is determined at block 730 that the i th user should be reassigned, process 700 moves to block 715 where the re-assignment takes place. Alternatively, where no re-assignment is to take place, process 700 moves to block 735 where, as mentioned above, process 700 loops until a subsequent user makes a request.

In yet another embodiment, it may be necessary to re-assign users where no new user has made a content request. As mentioned above, users may be dynamically re-assigned where, for example, the number of users in an early view group dwindles down to a low level and there is a danger that streaming continuity may be interrupted. In such a case, a second determination may be made at block 735

relating to the number of users in a given early view group. For example, in addition to checking for new user requests at block 735, process 700 may also check for any user drops at block 735. Where users drops have caused the number of users in the early view group to drop below a predetermined level, process 700 may be redirected to block 715 where another user (e.g., i th-1 user) is re-assigned to the early view group. In this manner, this process 700 may be used in conjunction with process 600 of Figure 6 to provide stream caching for live content.

As the requested content is received by the user, stitching software may be used to re-join the file sections before providing the re-joined file to content presentation software, such as a media player. In one embodiment, the re-joining process is as described in U.S. Provisional Patent Application serial number 60/275,407, entitled "Re-assembly of Streaming Files from Separate Connections."

Regardless of the re-joining process used, the re-joined file may be provided to media presentation software on user computer 40. In addition, the media presentation software may include any of Windows Media Player, Real Player, QuickTime Player, or any other media players capable of receiving streaming content. In one embodiment, the client-side software of client system 610 includes stitching software 620, which may be used to create reassembled file 630.

While the preceding description has been directed to particular embodiments, it is understood that those skilled in the art may conceive modifications and/or variations to the specific embodiments described herein. Any such modifications or variations which fall within the purview of this description are intended to be

included therein as well. It is understood that the description herein is intended to be illustrative only and is not intended to limit the scope of the invention.

CLAIMS

What is claimed is:

1. A method for streaming content over a network comprising:
splitting said content into a plurality of file sections;
caching a first file section of said plurality of file sections on a caching server in response to a first user request;
streaming said first file section from said caching server to a first user, where said first user accesses the network using said caching server; and,
streaming said first file section from said caching server to a second user in response to a second user request, said second user to access the network using said caching server.
2. The method of claim 1, further comprising streaming a second file section from the caching server to the first user at the same time as the first file section is being streamed to the second user.
3. The method of claim 1, further comprising splitting each of said plurality of file sections into a plurality of sub-files.
4. The method of claim 1, further comprising storing at least one of said plurality of file sections on a plurality of content servers connected to said network, said plurality of content servers to provide said plurality of file sections to the caching server over the network.

5. The method of claim 1, wherein caching a first file section of said plurality of file sections comprises streaming said first file section over the network from a content server to an ISP caching server in response to a first user request.

6. The method of claim 1, further comprising:
streaming, to said first user from said caching server, a series of individual file sections of said plurality of file sections; and,
streaming each of said series of individual file sections to said second user from said caching server after being streamed to said first user.

7. The method of claim 1, further comprising when said content is live content:
comparing an address of a new user to a log of current users;
assigning said new user to an early group where the address of the new user does not match a current user address; and,
assigning said new user to a common cache group where the address of the new user does match the current user address.

8. The method of claim 7, further comprising:
streaming said plurality of file sections to said early group from said caching server; and,
streaming said plurality of file sections to said common cache group from said caching server, each of said plurality of file sections to be streamed to the common cache group after being streamed to the early group.

9. A system for streaming content over a network, where said content has been split into a plurality of file sections, comprising:

a first user computer coupled to the network;

a second user computer coupled to the network;

a content server in communication with said first user computer and second user computer; and,

a caching server coupled to the first user computer and the second user computer, said first and second user computers to communicate with the content server using said caching server, said caching server to,

be provided with a first file section of said plurality of file sections by the content server in response to a first user request,

cache said first file section,

stream said first file section to a first user, and

stream said first file section to a second user in response to a second user request.

10. The system of claim 9, wherein said caching server is further to stream a second file section to the first user computer at the same time as the first file section is being streamed to the second user computer.

11. The system of claim 9, where each of said plurality of file sections is further split into a plurality of sub-files.

12. The system of claim 9, further comprising a plurality of content servers where each of said plurality of content servers is connected to the network, and at least one of said plurality of file sections is stored on said plurality of content servers.

13. The system of claim 9, wherein said caching server is an ISP caching server through which said first user computer and second user computer access the network.

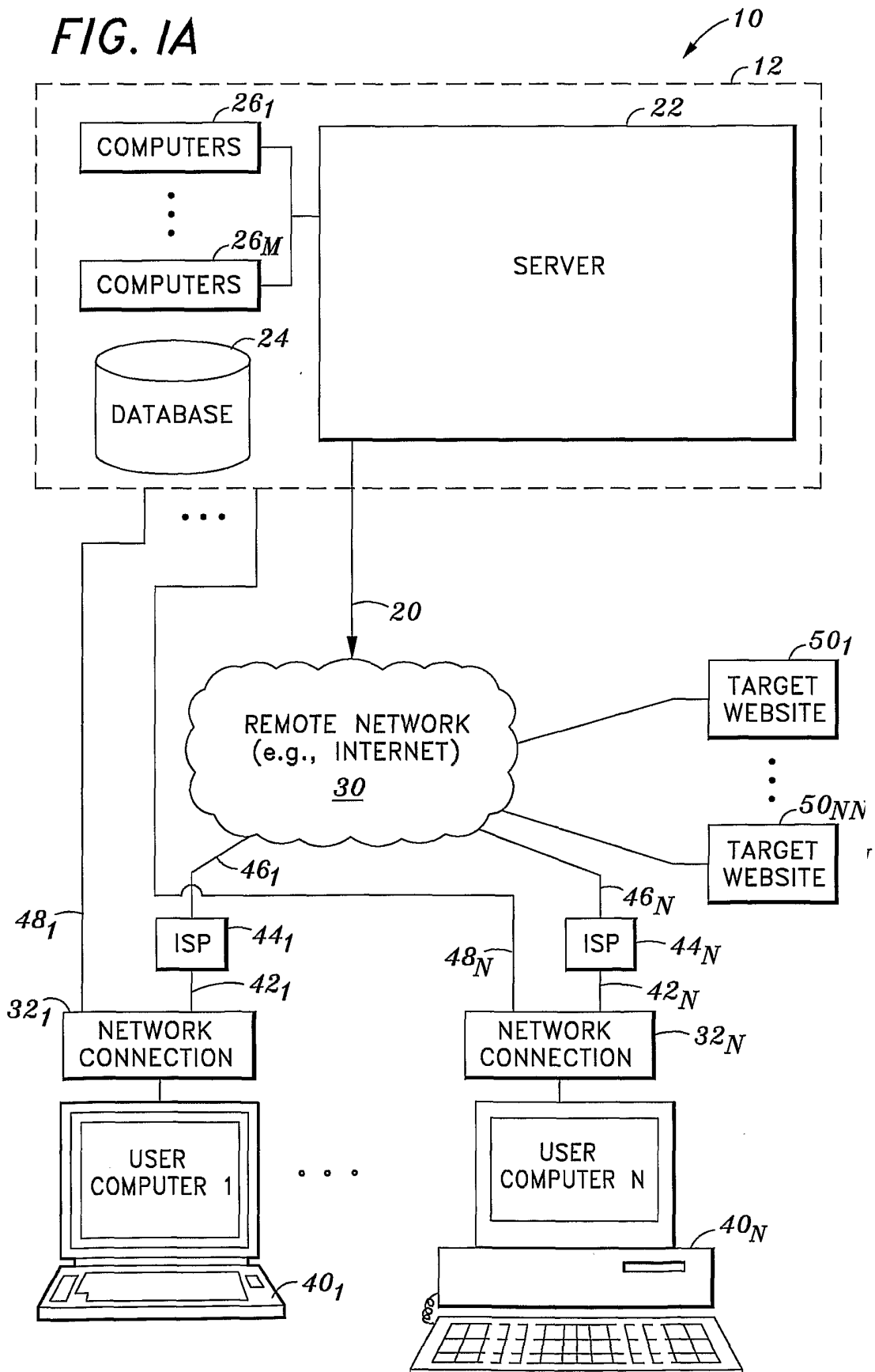
14. The system of claim 9, wherein said caching server is further to, stream, to said first user computer, a series of individual file sections of said plurality of file sections, and, stream each of said series of individual file sections to said second user computer after being streamed to said first user computer.

15. The system of claim 9, wherein said content server is to, compare an address of a new user computer to a log of current user computers, assign said new user computer to an early group where the address of the new user computer does not match a current user computer address, and assign said new user computer to a common cache group where the address of the new user computer does match the current user computer address.

16. The system of claim 15 wherein said caching server is further to, stream said plurality of file sections to said early group, and

stream said plurality of file sections to said common cache group, each of said plurality of file sections to be streamed to the common cache group after being streamed to the early group.

FIG. 1A



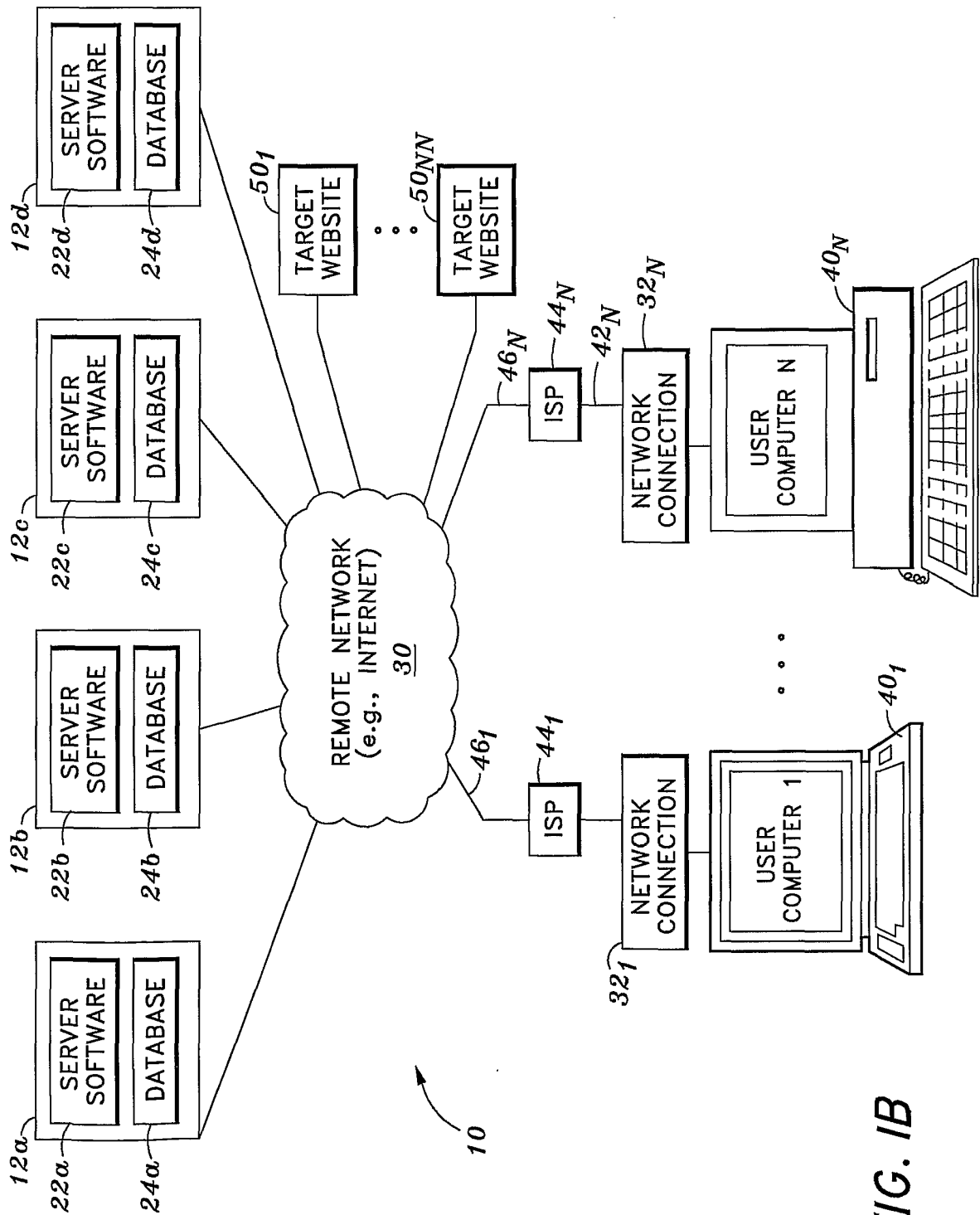


FIG. 1B

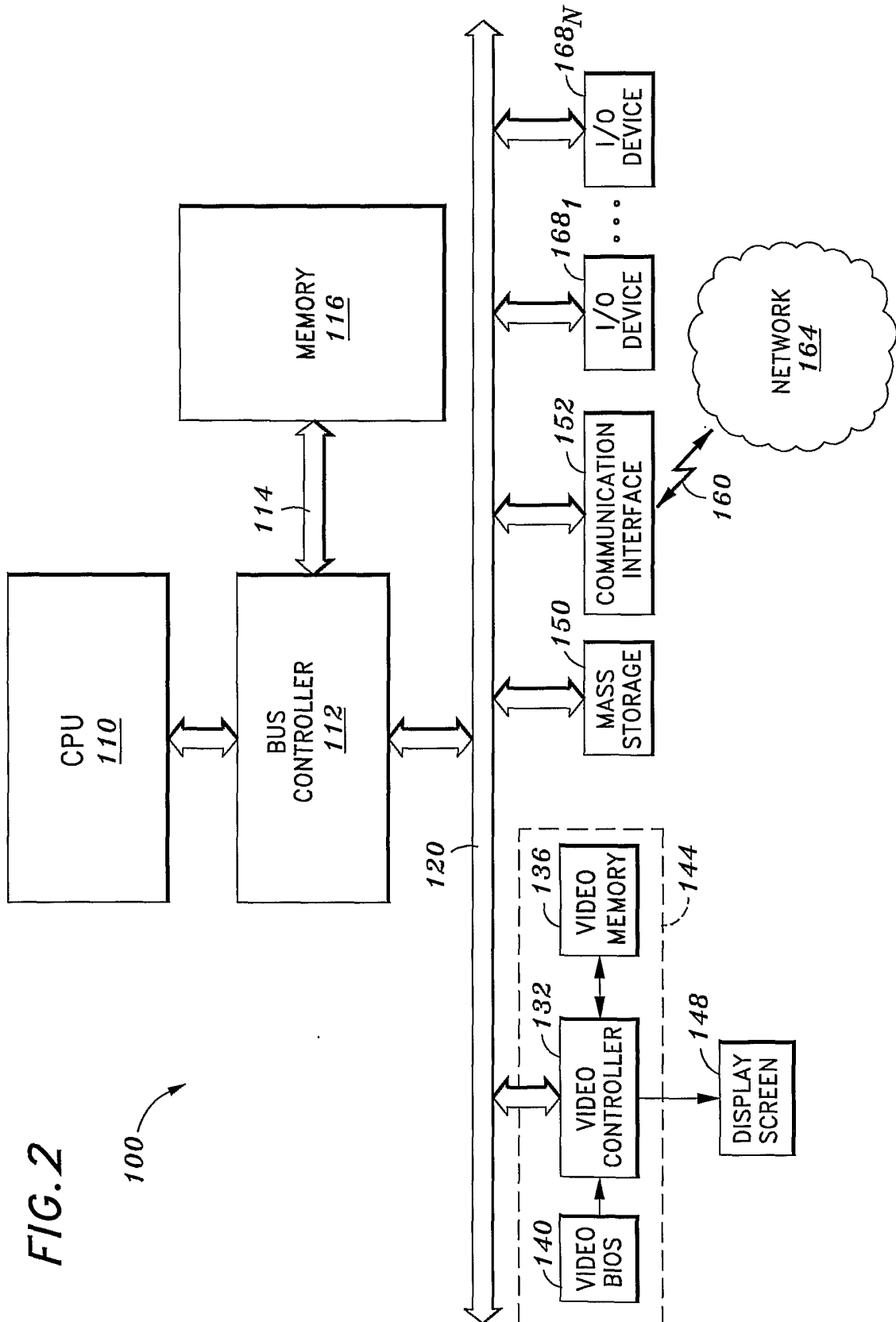


FIG. 2

100

FIG 3A

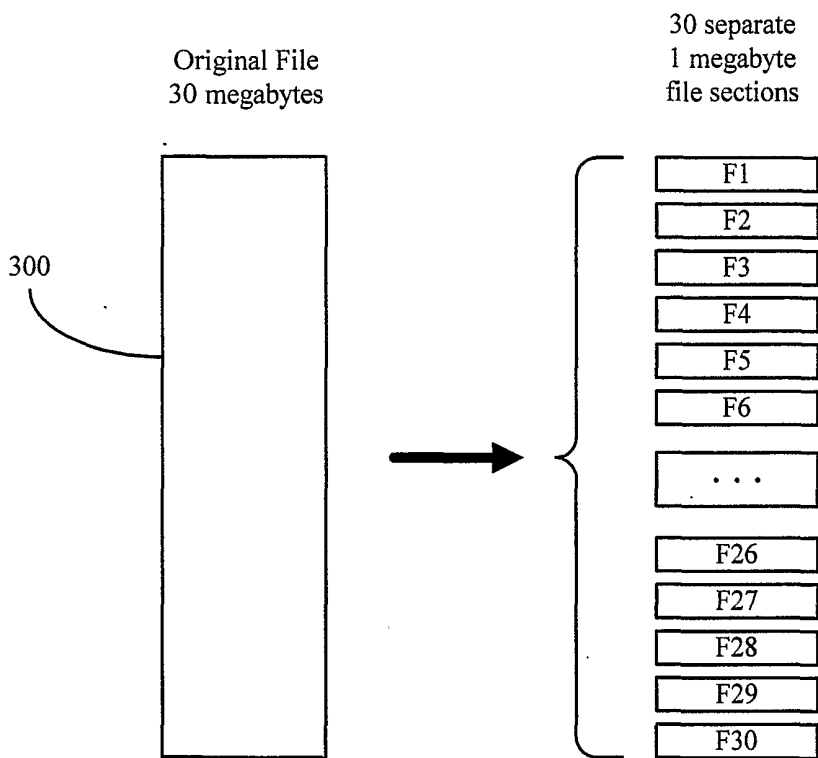


FIG 3B

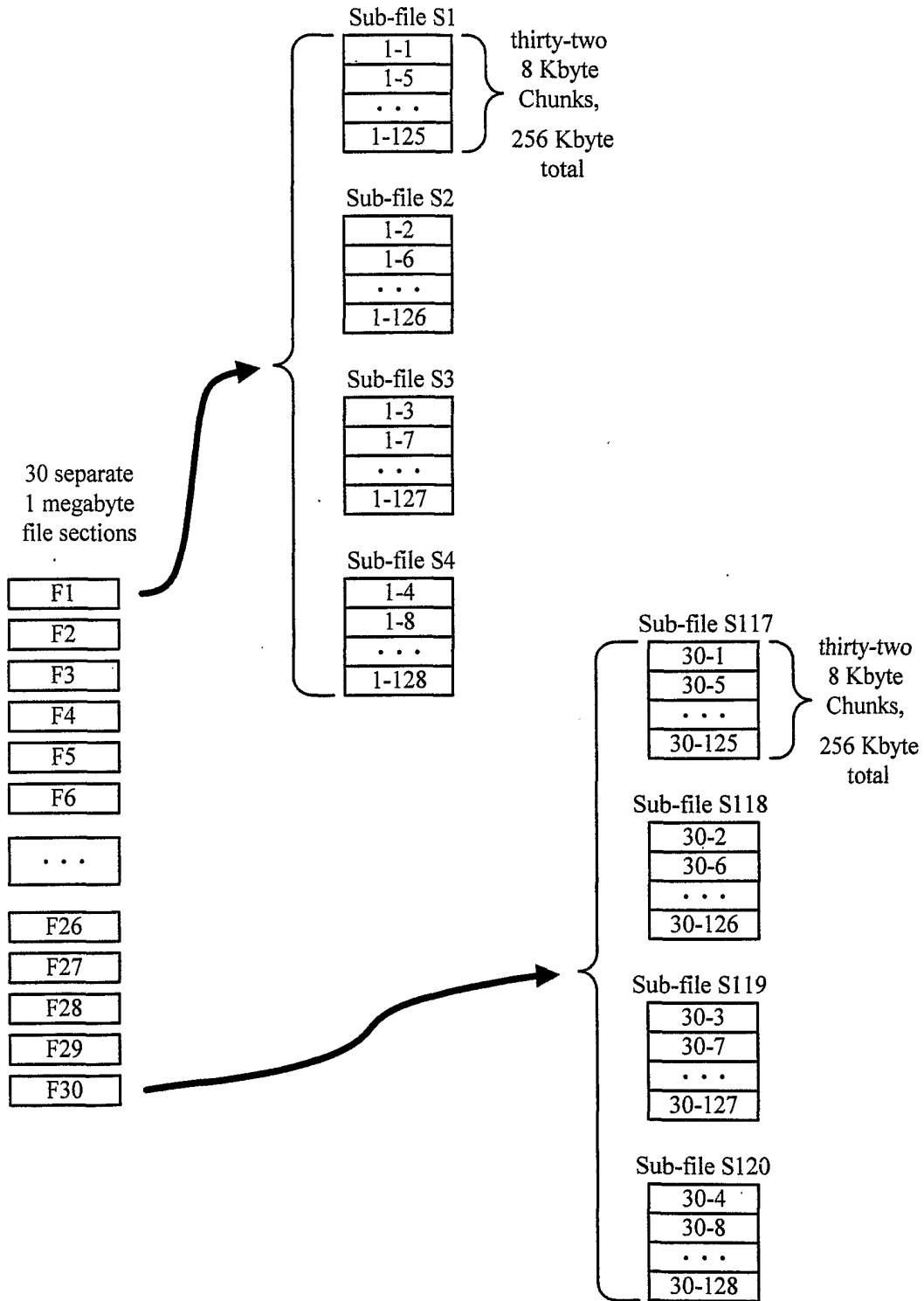


FIG 4

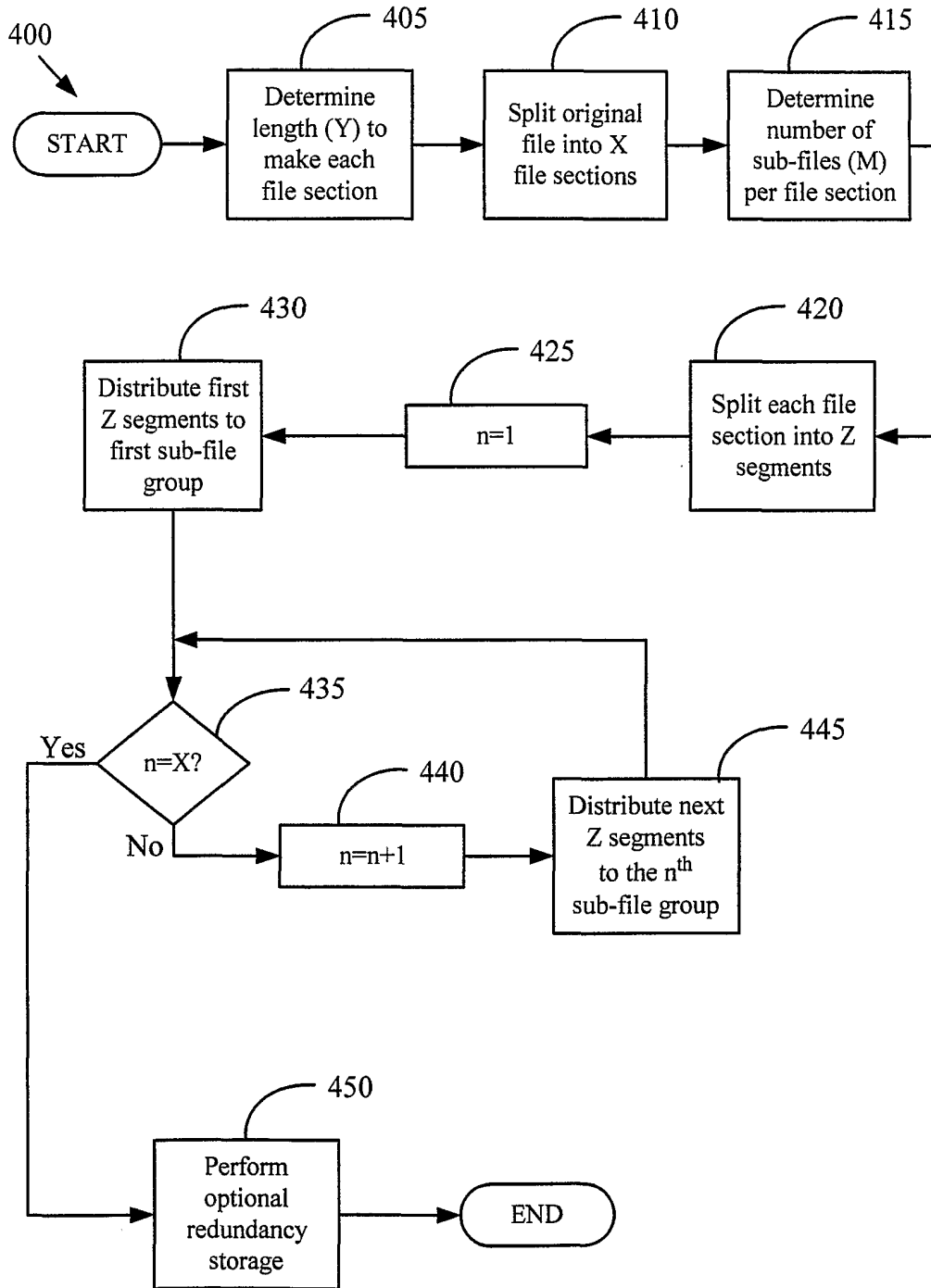


FIG 5

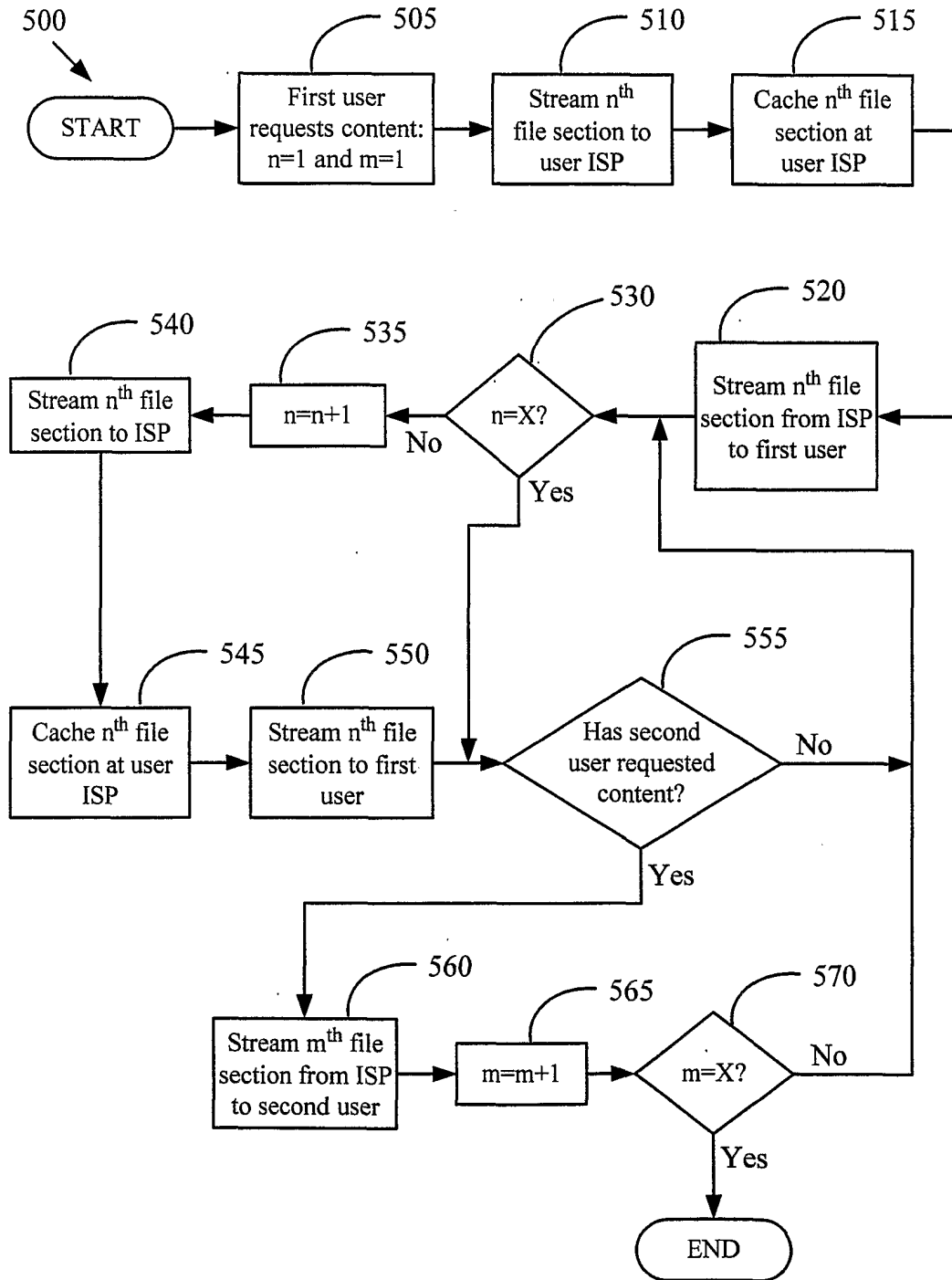


FIG 6

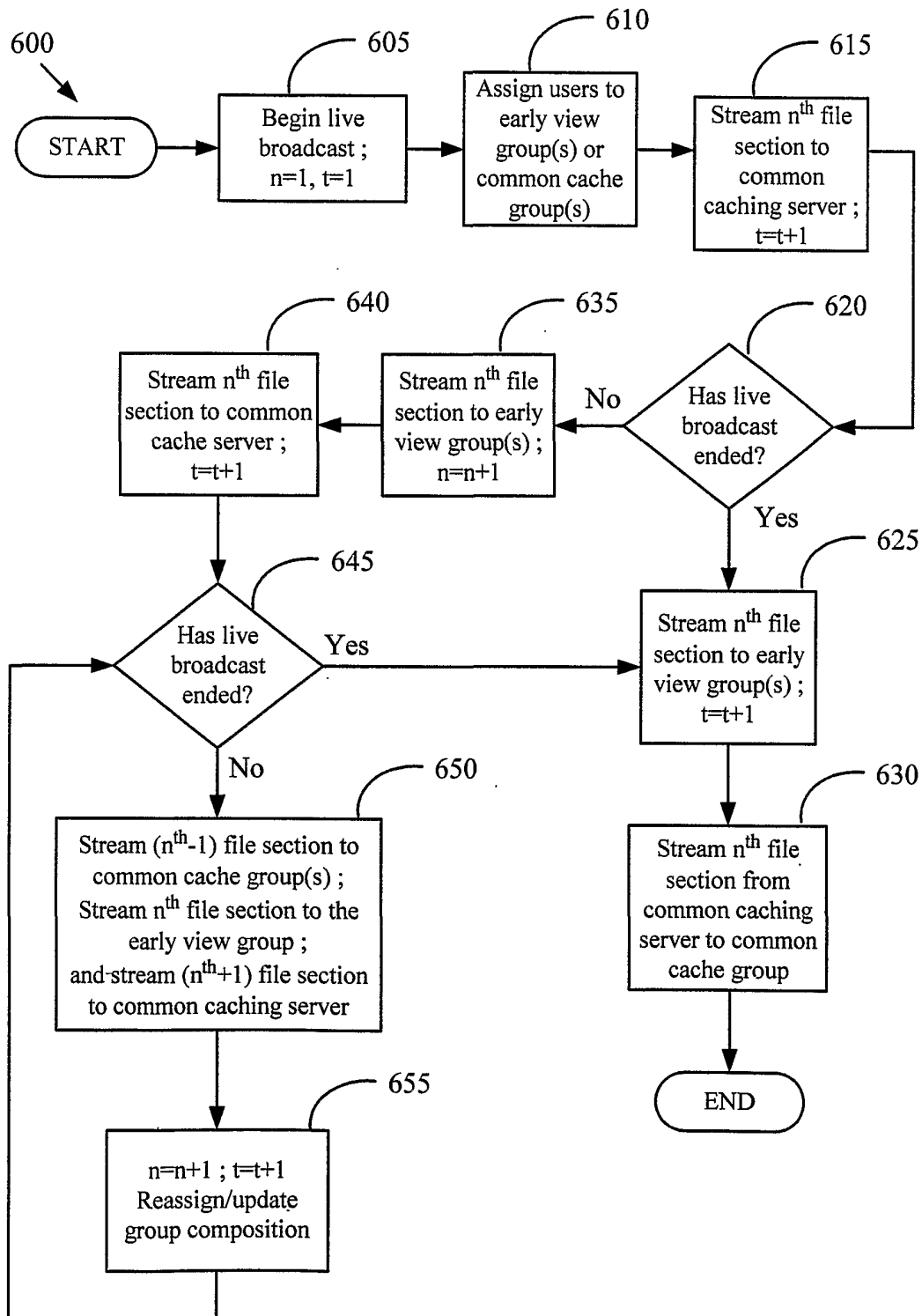
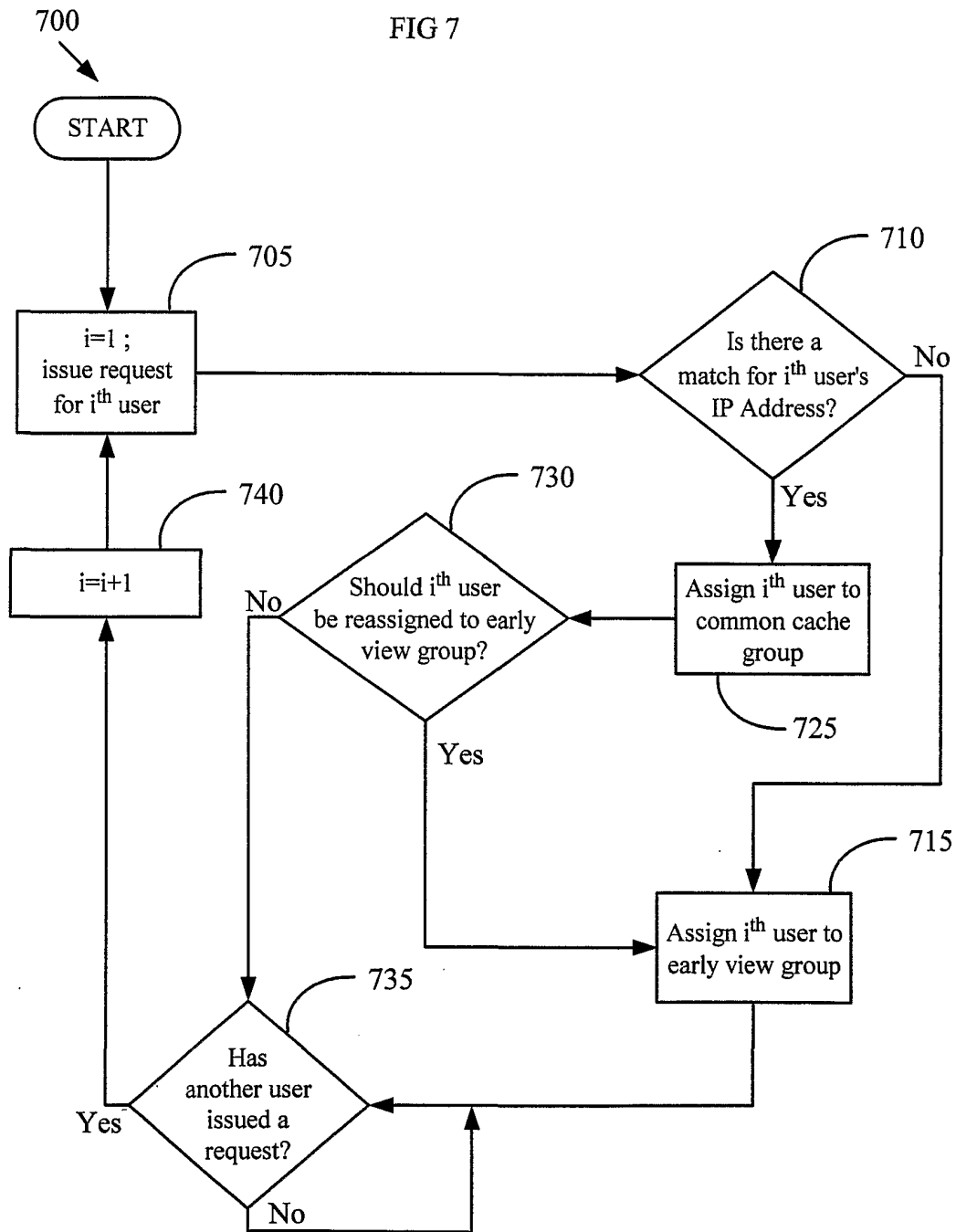
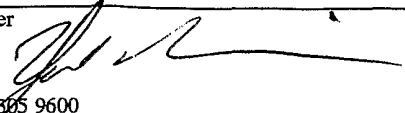


FIG 7



INTERNATIONAL SEARCH REPORT

International application No.
PCT/US02/07658

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(7) : G06F 3/14, 13/00, 17/30 US CL : 709/231; 707/10		
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. : 709/231; 707/10		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) uspat		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,751,968 A (COHEN) 12 May 1998 (12.05.1998), col. 2 lines 20-45.	1-16
Y	US 5,583,994 A (RANGAN) 10 Dec. 1996 (10.12.1996), col.2 lines 17-39).	1-16
Y	US 6,105,029 A (MADDALOZZO, JR. et al.) 15 Aug. 2000 (15.08.2000), abstract	1-16
<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	"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
"E"	earlier application or patent published on or after the international filing date	"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
"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)	"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
"O"	document referring to an oral disclosure, use, exhibition or other means	"&" document member of the same patent family
"P"	document published prior to the international filing date but later than the priority date claimed	
Date of the actual completion of the international search 12 June 2002 (12.06.2002)		Date of mailing of the international search report 11 JUL 2002
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 Dung Dinh  Telephone No. 305 9600