METHOD AND SYSTEM FOR PROVIDING STREAMING MEDIA SERVICES

Abstract: The invention provides a system for the delivery of continuous media based on a hierarchical architecture of multiple servers to support a large number of streaming media sessions at any given time over data networks. The data networks may include the Internet, the Intranet or a network of other private networks. In an exemplary embodiment, each server is connected to a local media depository, with some of the servers additionally connected to a shared media depository. A server can supply continuous media to the network from any of the media depositories to which it is connected. Service providers are assigned to servers within the system based upon their requirements as well as the relative abilities of the servers. Once a service provider has been allocated to the servers, requests from a user are then decided based upon this allocation.
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.
CROSS REFERENCE TO RELATED APPLICATIONS

This application claims priority from provisional patent application 60/155,000, entitled METHOD AND SYSTEM FOR PROVIDING WORLD WIDE STREAMING MEDIA SERVICES, filed September 21, 1999.

The above referenced application is incorporated herein by reference for all purposes. The prior application, in some parts, may indicate earlier efforts at describing the invention or describing specific embodiments and examples. The present invention is, therefore, best understood as described herein.

FIELD OF THE INVENTION

The present invention relates generally to networked multimedia systems. More particularly, the invention relates to a method and system architecture for providing global streaming media service over a data network using multiple media servers to ensure that streaming signals are delivered with a high quality of service (QoS).

Copyright Notice

A portion of the disclosure of this patent document may contain materials that are subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyright rights whatsoever.
Prior Publications

The following publications may be related to the invention or provide background information. Listing of these references here should not be taken to indicate that any formal search has been completed or that any of these references constitute prior art.

BACKGROUND OF THE INVENTION

The Internet is a rapidly growing communication network of interconnected computers around the world. Together, these millions of connected computers form a vast repository of hypermedia information that is readily accessible by users through any of the connected computers from anywhere and anytime. As there is an increasing number of users who are connected to the Internet and surf for various information, they meanwhile create tremendous demands for more content to be available and methods to deliver them on the Internet. Currently, the most commonly available information that is deliverable on the Internet may include text information, images and graphics, videos and audio clips.

Continuous information such as continuous videos, audio clips, audiovisual or other multimedia works (referred to below collectively as “media”) may be one of the frequently requested network resources. It is therefore not uncommon to experience thousands of access requests simultaneously to a piece of popular video, audio or audiovisual program. Most of the existing streaming technologies are based on a single media server architecture that ultimately suffers from resource/performance limitation. The single media server architecture may support a few hundreds of streaming sessions at one time, but can hardly in the scale of thousands of streaming sessions simultaneously. Furthermore, the single server architecture suffers a server pitfall on the reliability issue. It represents a single point of failure when a computer system crashes or unexpected power disruption happens. Once the single server is down, the entire media streaming service would be discontinued so do the revenue and profits for the service providers. A reliable, fault-tolerance, and 24-hour streaming media service is of most importance to a
streaming media services provider such as Internet Service Provider (ISP) or Internet Content Provider (ICP). There is therefore a great need for solutions to media delivery systems that can be configured to support hundreds thousands of streaming sessions at the same time meanwhile keeping minimum impact on the quality of service.

Two possible paradigms have been suggested to scale-up the media streaming server: use a single powerful server machine or have multiple servers working together to provide the aggregated performance. The first approach may leads to a special-build proprietary-oriented machine platform, which translates into high cost and maintenance complexities. Additionally, it does not solve the reliability problem inherent in a single server architecture. On the other hand, using multiple commodity-oriented server platforms, as suggested in the second approach, gives a much better performance/cost ratio and takes advantage of the whole PC industry’s effect to further improve performance/cost ratio down the road. However, how to implement the architectural structure of multiple media servers working together to produce the highest media service quality becomes a critical issue.

SUMMARY OF THE INVENTION

The disclosure hereof discloses a global media delivery system based on a hierarchical architecture of multiple servers to support a large number of streaming media sessions at any given time over data networks. The data networks may include the Internet, the Intranet or a network of other private networks. A media deliver system employing the present invention can advantageously provide streaming media over the data networks to unlimited number of devices coupled to the data networks at any given time without compromising the quality of service.

According to one aspect of the present invention, the global media delivery system comprises a plurality of video servers that may be located sparsely or remotely with respect to each other. Each of the video servers may provide different video titles that are to be shared among the video servers based on requests received regardless where the requests are from. According to another aspect of the present
invention, the distribution of the video titles is a function of the request frequency that controls appropriate caching of the most frequently accessed video titles. In other words, a video server receiving high number of requests to the same video title may cache a portion or an entire of the video title locally as such network traffic is reduced while the quality of service is ensured. According to still another aspect of the present invention, a server that provides a streaming service to a client device is selected according to a load balance manager. Based on capabilities of each of the servers and working load thereof being monitored, the load balance manager determines a most appropriate server to provide a requested media delivery. With the help of the load balance manager, the video servers in the global media delivery system will be neither overloaded nor idle. Further, in case one of the servers is disrupted, the load balance manager may immediately determine a next available server to continue the deliver service, which ensures a reliable, fault tolerance, media service provided by the global media delivery system.

In an exemplary embodiment, each server is connected to a local media depository, with some of the servers additionally connected to a shared media depository. A server can supply continuous media to the network from any of the media depositories to which it is connected. Service providers are assigned to servers within the system based upon their requirements as well as the relative abilities of the servers. A service provider request to use the system to stream media to users will include information such as the number of streams the provider wants, the amount of storage required, and any fault tolerance requirements. Using these parameters, the system assigns the service provider a set servers from which to stream content to a user of the provider’s service, the servers being selected based upon their relative abilities to meet the requirements. Once a service provider has been allocated to the servers, requests from a user is then decided based upon this allocation.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows the flow of streaming requests across a wide area network.
Figure 2A illustrates an exemplary configuration in which the present invention may be practiced.

Figure 2B illustrates another exemplary configuration in which the present invention may be practiced.

Figure 3A shows a block diagram of a global streaming gateway.

Figure 3B shows an exemplary functional block diagram of the global media delivery system.

Figure 4 shows a process flowchart of a gateway module that may be implemented as a server module to be installed in a server that is configured to function as a global streaming gateway.

Figure 5 shows a process flowchart in a selected video server.

Figure 6 shows a process flowchart in a second video server that supports the selected video server of Figure 5 when a selected video title is not in the local depository or cached.

Figure 7 presents an embodiment for global streaming with shared media repository storage in the media cluster.

Figure 8 presents an embodiment for global streaming with mixed shared and non-shared media repository storage.

Figure 9 shows a detail of Figure 8 to highlight the portions relevant to determining server capability.

Figure 10 is a flow chart showing an exemplary embodiment of a retailer allocation scheme.

Figure 11 shows an example of a retail allocation table construction.

Figure 12 is a media allocation table containing the information associated with a title stored.

Figure 13 shows the structure of a workload table.

Figure 14 is a block diagram showing a representative example logic device in which aspects of the present invention may be embodied.

The invention and various specific aspects and embodiments will be better understood with reference to the drawings and detailed descriptions. In the different
figures, similarly numbered items are intended to represent similar functions within
the scope of the teachings provided herein.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Although the media delivery system described below is based on video
streaming signals, those skilled in the art can appreciate that the description can be
equally applied to audio streaming signals or other media or multimedia signals as
well. The detailed description of the present invention here provides numerous
specific details in order to provide a thorough understanding of the present
invention. However, it will become obvious to those skilled in the art that the
present invention may be practiced without these specific details. In other instances,
well known methods, procedures, components, and circuitry have not been described
in detail to avoid unnecessarily obscuring aspects of the present invention. Before
the specifics of the embodiments are described, a few more embodiments are
outlined generally below.

The present invention presents a global media delivery system based on a
hierarchical architecture of multiple servers to support a large number of streaming
media sessions at any given time over data networks. The data networks may
include the Internet, the Intranet or a network of other private networks. A media
deliver system employing the present invention can advantageously provide
streaming media over the data networks to unlimited number of devices coupled to
the data networks at any given time without compromising the quality of service.

According to one aspect of the present invention, the global media delivery
system comprises a plurality of video or other continuous media servers that may be
located sparsely or remotely with respect to each other. Each of the servers may
provide different video titles that are to be shared among the servers based on
requests received regardless where the requests are from. According to another
aspect of the present invention, the distribution of the continuous media titles is a
function of the request frequency that controls appropriate caching of the most
frequently accessed titles. In other words, a server receiving high number of
requests to the same title may cache a portion of or an entire title locally so that network traffic is reduced while the quality of service is ensured. According to still another aspect of the present invention, a server that provides a streaming service to a client device is selected according to a load balance manager. Based on capabilities of each of the servers and working load thereof being monitored, the load balance manager determines a most appropriate server to provide a requested media delivery. With the help of the load balance manager, none of the video servers in the global media delivery system will be neither overloaded nor idle. Further, in case one of the servers is disrupted, the load balance manager may immediately determine a next available server to continue the deliver service, which ensures a reliable, fault tolerance, media service provided by the global media delivery system.

Regardless of user location, the global streaming technology enables users to watch the very best quality of the streaming media via the Internet. This technology integrates technologies from web technology, web proxy technology, and large-scale media streaming technology altogether to serve millions of on-line users with the streaming media capability no matter where they are.

The global streaming proposes a hierarchical architecture of multiple web servers, web proxy servers, and multiple media server architecture. Figure 1 illustrates the proposed technology via the following scenario: A user living in City A would like to access continuous media, for example view a recorded concert broadcasting in City B hosted by an Internet content or service provider over the Internet. The user could check the program listing from the home page of the content provider by specifying the appropriate URL. After browsing the web site, the user would find the required program to watch and click the corresponding streaming media link. The request from the user’s media playback station 10 will reach the web proxy server 23 in the City A. The request will then redirect to a special global streaming gateway 25 hosted in the same proxy server. If this is the first time the request to retrieve the concern has been made, or for other some other reason the media is not stored in proxy server 23, the request will then redirect up to one higher hierarchy of the web architecture via the Internet 30 until it reach the
main web server 41 in the City B. When the request reaches the main web server 41, the request is redirected to the global streaming gateway 45 hosted in the main web server 41. The global streaming gateway 45 then checks the status of collected data related to media server loads and geographical information about each media server that owns or caches the requested media. Based on the techniques described below, the global streaming gateway 45 will select the best suited media server nearby the user located in the City A. All the related information collected by the main global streaming gateway 45 will migrate to the down stream global streaming gateways in each web proxy server. This information is then passed back to the web server 41 or proxy server 43 in City B, and then through the Internet 30 back to City A. The web server 21 or proxy server 23 can then cache the streaming information about the response for later usage. The selected concert will then stream from the selected media server to stream to the user in the City A at the media playback station 10.

If a similar request for the same continuous media content in the City A is received at a later time in request handler 20, the nearby web proxy server 23 will intercept the request and redirect the request to its global streaming gateway 25 based on the streaming information cached during the previous request. Based on the local information gathered from nearby media servers, the global streaming gateway will select the most appropriate media server to serve the current streaming request. This scheme dramatically improves the performance and the consequent streaming accesses.

To achieve the above streaming service, the global streaming expands its streaming service to unlimited number of users to access any media content around the globe. The technology enables large-scale media service, fault-tolerance, automatic migration of media content, and 24-hour continuous media streaming by exploiting the web concept to enable the streaming media delivery. Users use the same web interface to select the media content.

Furthermore, it is well known in the art that logic or digital systems and/or methods can include a wide variety of different components and different functions
in a modular fashion. The following will be apparent to those of skill in the art from
the teachings provided herein. Different embodiments of the present invention can
include different combinations of elements and/or functions. Different embodiments
of the present invention can include actions or steps performed in a different order
than described in any specific example herein. Different embodiments of the present
invention can include groupings of parts or components into larger parts or
components different than described in any specific example herein. For purposes
of clarity, the invention is described in terms of systems that include many different
innovative components and innovative combinations of innovative components and
known components. No inference should be taken to limit the invention to
combinations containing all of the innovative components listed in any illustrative
embodiment in this specification. The functional aspects of the invention, as will be
understood from the teachings herein, may be implemented or accomplished using
any appropriate implementation environment or programming language, such as
C++, COBOL, Pascal, Java, Java-script, etc. All publications, patents, and patent
applications cited herein are hereby incorporated by reference in their entirety for all
purposes.

The invention therefore in specific aspects provides a streaming of
continuous media such as video/audio signals that can be played on various types of
video-capable terminal devices operating under any types of operating systems
regardless of what type of players are pre-installed in the terminal devices.

In specific embodiments, the present invention involves methods and systems
suitable for providing multimedia streaming over a communication data network
including a cable network, a local area network, a network of other private networks
and the Internet.

The present invention is presented largely in terms of procedures, steps, logic
blocks, processing, and other symbolic representations that resemble data processing
devices. These process descriptions and representations are the means used by those
experienced or skilled in the art to most effectively convey the substance of their
work to others skilled in the art. The method along with the system to be described
in detail below is a self-consistent sequence of processes or steps leading to a desired result. These steps or processes are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities may take the form of electrical signals capable of being stored, transferred, combined, compared, displayed and otherwise manipulated in a computer system or electronic computing devices. It proves convenient at times, principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, operations, messages, terms, numbers, or the like. It should be borne in mind that all of these similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise as apparent from the following description, it is appreciated that throughout the present invention, discussions utilizing terms such as “processing” or “computing” or “verifying” or “displaying” or the like, refer to the actions and processes of a computing device that manipulates and transforms data represented as physical quantities within the device’s registers and memories into analog output signals via resident transducers.

**Global Streaming**

The discussion of the exemplary functions designed for a media delivery system according to the present invention and for teaching and referring to the detailed design, features, use, advantages, configurations and characteristics of the present invention begins with Figures 2A and 2B, which provide a context for the invention. Details of a caching technique suitable for the present application can be found in copending U.S. patent application ______________, by Horng-Juing Lee, entitled “METHOD AND APPARATUS FOR CACHING FOR STREAMING DATA”, filed on September 8, 2000, and which is hereby included by this reference, which refers to an exemplary embodiment to cache those frequently requested video titles in a local video server.

Referring to the drawings, in which like numerals refer to like parts throughout the several views, Figure 2A illustrates an exemplary configuration in
which the present invention may be practiced. Global media delivery system
comprises a plurality of servers of which only three 102, 108 and 110 are shown.
Each of the servers stores or caches some of the video files in a video repository of
the global media delivery system. As used herein, video files or titles are referred to
any video footage, video films and/or video/audio clips or other continuous media
that typically are in a compressed format such as MPEG or MP3. It should be
noted, however, that the exact format of the video files do not affect the operations
of the present invention. As will be noted and appreciated, the present invention
applies to any formats of the video files.

Preferably data network 106 is a data network backbone, namely a larger
transmission line. At the local level, a backbone is a line or set of lines that local area
networks connect to for a wide area network connection or within a local area
network to span distances efficiently (for example, between buildings). On the
Internet or other wide area network, a backbone is a set of paths that local or
regional networks connect to for long-distance interconnection. Coupled to data
network A 106, there are two other networks 112 and 114 that are typically the
Internet, a local area network, or phone network through which terminal devices can
receive video files. The terminal devices may include, but not be limited to,
multimedia computers (e.g. 116 and 119), networked television sets or other
video/audio players (e.g. 117 and 118). Typically the terminal devices are equipped
with applications or capabilities to execute and display received video files. For
example, one of the popular applications is an MPEG player provided in WINDOWS
98 from Microsoft. When an MPEG video file is received in streaming from one of
the proxy servers, by executing the MPEG player in a multimedia computer, the
video file can be displayed on a display screen of the computer.

To receive a desired video, one of the terminal devices (e.g. 116) must send
in a request that may comprise the title of the desired video. Typically the request
is in a form of URL that may include a subscriber identification if the video services
allow only authorized access. Upon receiving the request, the server (e.g. 108), will
first check in its cache if the selected video is provided therein, meanwhile the
request is recorded by a request manager. The selected video will be provided as a streaming video to the terminal device if some or the entire video is in the cache. Otherwise server 108 proceeds to send a request to another server 102 for the selected video of the rest of the video if there are some units of the video in a cache memory of server 108 or the entire video if there is no any cached unit of the video.

Figure 1 B shows another exemplary configuration in which the present invention may be practiced. As shown in the figure, all the video servers 120 in the global media delivery system as well as the terminal devices (e.g. 122) are coupled to a network 126 (e.g. the Internet). When terminal device 122 sends in a request for a specific video title, the request is routed to one of the servers (e.g. 120-1). Server 120-1, referred to as a global streaming gateway, comprises a request handler that processes the received request. The processing includes authentication of the request and other administration control. Further server 120-1 comprises a load balancing manager that monitors the system performance of all the video servers in the global media delivery system. With the load balancing manager, an appropriate server is identified to provide the video server to terminal devices 122. The detail description of the request processing and the load balancing manager is provided in exemplary embodiments described below.

To better understand the present invention, Figure 3A shows a block diagram of global streaming gateway (i.e. a server).

According to one embodiment of the present invention, the gateway is loaded with a compiled and link version of the embodiment of the present invention, when executed by the processor, the compiled and link version will perform at least the following: receiving a request to a video title from a terminal device; retrieving system performance information on each of video servers in a global media delivery system; determining an appropriate video server in the global media delivery system with respect to the retrieved system performance information; and sending a response to the request, the response comprising an address identifier identifying the appropriate video server to service the video title selected by a user. Further, the compiled and link version, when executed, performs additionally: inserting
commercial information into the response so that the commercial information becomes available to the user when the video title is played.

Figure 3B shows an exemplary functional block diagram of the global media delivery system. After the user receives the response and activates the address identifier that is typically a universal resource identifier (URI) or universal resource locator (URL), one of the video servers (e.g., server 2) is linked and starts to provide the video streaming data. One of the important features in the present invention is that the video server that provides the video service is selected based on the system performance information on each of video servers in a global media delivery system. In other words, no single video server would be overloaded so as to affect the quality of services.

In the case that the selected video server does not have the desired video title in the local depository, a progressive cache technique is used to cache the video title. The detailed description of the progressive cache technique is provided in U.S. patent application ______ by Horng-Jueng Lee, entitled “METHOD AND APPARATUS FOR CACHING FOR STREAMING DATA”, filed on September 8, 2000, which was included by reference above.

To further understand the present invention, Figure 4 shows a process flowchart of the gateway module that may be implemented as a server module to be installed in a server that is configured to function as a global streaming gateway. Each of the process in the flowchart includes many sub-processes that are provided below. Figure 5 shows a process flowchart in the selected video server. The process flowchart may be implemented as a proxy server module to be installed in a server to function as one of the video servers in the global media delivery system. Figure 6 shows a process flowchart in a second video server that supports the selected video server of Figure 5 when a selected video title is not in the local depository or cached.

The global media delivery system as described in accordance with one aspect of the present invention is robust, operationally efficient and cost effective. The global streaming mechanism provides the best use of all proxy video servers and
permits seamless delivery of streaming video with highest quality of services possible. In addition, the present invention may be used in connection with presentations of any type, including sales presentations and product/service promotion, which provides the video service providers additional revenue resources.

While the embodiment discussed herein may appear to include some limitations as to the presentation of the cache units and the way of managing the units, in terms of the format and arrangement, the invention has applicability well beyond such embodiment, which can be appreciated by those skilled in the art.

While the foregoing and attached are illustrative of various aspects/embodiments of the present invention, the disclosure of specific sequence/steps and the inclusion of specifics with regard to broader methods and systems are not intended to limit the scope of the invention which finds itself in the various permutations of the features disclosed and described herein as conveyed to one of skill in the art.

In the following paragraphs the proposed architecture of the global streaming, described above with respect to Figure 1, is more fully developed. This shows the interactions within the architecture of web servers described in copending U. S. provisional patent application number 60/155,354, entitled “Method and System for Providing Real-Time Streaming Video Services”, filed September 22, 1999, and which is hereby included by this reference, web proxy servers, and associated global streaming gateways

System Components

Figures 7 and 8 show some of the system components in a two different embodiments. Figure 7 presents the detailed layout of the architecture for global streaming with shared media repository storage in the media cluster. The embodiment of Figure 8 has a more complicated storage arrangement with mixed shared and non-shared media repository storage. The elements are numbered similarly in this pair of figures and will initially be described with respect to Figure 7.
Request handler 720 is shown connected to media server 701, remote manager 703, media recording station 705, and media playback station 707 and uses one or more web servers and web proxy servers as front-end handler for all kinds of client streaming requests. The embodiment of Figure 7 shows a request handler with a pair of web servers 721 and 723. A client request, for example in URL format, for operations such as playback, recording or administration is entered the system through a request handler interface. This gives client the capability of accessing system resource through URL address. Therefore, the client can access system resource anywhere web access available.

Within the back-end of the web servers there are several components for handling client requests. These components are considered as common gateway interface (CGI) components. However, their implementations are not limited to CGI scripts and they can alternately be implemented as other CGI alternatives such as ISAPI or Servlets.

These components have a number of tasks. These can include authentication, admission control, load balancing, performance monitoring, the global streaming gateway function, and advertisement insertion.

Authentication block 731 responsible for verifying the identity of client with a user data base 741 before granting any operations. Admission control 739 is responsible for manage the concurrent number of streaming sessions and total streaming bandwidth. This limitation can be caused by machine capability and/or product license control. The purpose of restricting the session count is to ensure the playback quality for all admitted requests.

Load balancer 735 responsible for determining which server with which the incoming request should make contact. The component interacts with performance monitor 733 to get the media server workload information 743. Together with the client and video clip information, the component evens out the amount of workload among all working servers within the media server cluster 740. Performance monitor 733 handles inquiries from the remote management block 703 as well monitoring the performance and operational status of the media server 701.
Global streaming gateway 730 is responsible for receiving the request and replying to the request with an HTML page. This component will interact with the rest of the components to get the required information and send it back to the client.

Ad inserter 737 is responsible for generating an HTML template with the advertisement postings. This component interacts with the user database 741 to get user profile (if possible) and uses it to determine the appropriate ad template from ad template database 747 for better targeting to the audience.

Media server cluster 740 comprises several media servers 741-i. Each media server provides its own streaming service. The media servers need not be capability-equivalent. The goal of media server cluster 740 is to ensure smooth and uninterrupted media playback/recording. Within the media server cluster, a two-level media depository architecture is used: local media depositories 743-i and a shared media depository 745. Each local media depository 743-i is attached to a single media server 741-i. Therefore, only that particular media server can access its contents directly. The concurrent number of streaming sessions on clips stored on a particular server then is limited by that server’s capability. On the other hand, every media server within the cluster can access the contents stored on the shared media depository. Hence, the aggregated server capability can support significantly more streaming sessions on clips stored on the shared media depository.

Figure 8 differs from Figure 7 in that not all of the media servers 841-i are connected to the shared media depository 845. In this generalization, media servers 841-1 and 841-2 are connected only to their respective local media depositories 843-1 and 843-2, while media servers 841-3 and 841-4 are connected to both their respective local media depositories 843-3 and 843-4 and the shared media depository 845. In a generic situation, each server will have its own local media depository, with some servers further connected to one or more shared media depositories which are in turn each connected to multiple servers.

Although the servers and the depositories are shown together as part of the media cluster 740 in Figures 7 and 8, they need not have any actually physical proximity, but may be separated from each other. Although they from a single block
conceptually in the present invention, generically they are distributed over an extended physical region as described above with respect to Figures 1, 2A, and 2B. The architecture of media cluster 740 differs from what is found in the prior art as described in the Background section above. The approach using multiple commodity-oriented server platforms would completely do away with the shared media depository 745 and rely solely upon the local media depositories. In this arrangement, only those servers which contain the particular continuous media requested in their own local media depository can supply it to the user. In order to insure a high quality of service, this requires a large amount of duplication and redundancy among the local depositories as the same title must be stored in many different locations, often with every local depository having a complete collection of titles.

The other approach found in the prior relies solely on the shared media depository 745, dispensing with the local depositories. In this approach, the total number of streams is then limited by the bandwidth of the interface 847. Additionally, as discussed in the Background section, it does not solve the reliability problem inherent in using only a single depository.

In this discussion, a client is a component asking for service. The service can be playback, recording, upload media clips, download media clips, or query of operation status. In this aspect, a client can be another media server system, a media proxy server, a remote management console, a media recording station or a media playback station. Thus, it is employed as a general term to cover both the service or content provider as well as the user requesting content from the provider.

Scenario of Media Streaming Service

Based on these concepts, several scenarios related to a streaming service can be presented in terms of the following steps.

Case 1: Playback client requests to play back pre-recorded video.
1. Client sends out the request to play back movie A by clicking on a URL link.

2. Request handler receives the request.

3. Request handler dispatches the request to global streaming gateway.

4. Request handler sends the request to the authentication component.

5. Authentication component verifies the client identity by consulting with user database and making sure the client has been granted the right to perform the request.

6. When the client gains the access right, the global streaming gateway passes the request to the admission control to make sure the new request will not interfere with existing service and also meet the product license agreement.

7. Admission control checks the existence of the movie A. If the content is not cached in the global streaming gateway, the request is pass to a nearby request handler for further process and repeat Step 2 – Step 6.

8. When the admission control component approves the request, the global streaming gateway passes the request to the load balancer component.

9. The load balancer component consults with the performance monitor component to understand the current server workload. Then based on the load balance scheme, it returns the address of the media server ready to serve the request.

10. The global streaming gateway then passes the user profile information (retrieved from user database) to the ad inserter component asking for the appropriate ad template.

11. The global streaming gateway combines the server address, video clip information, and ad template to create a response HTML page to the client. The response HTML page contains an object tag indicating the player component and its associated parameters.

12. The HTML page from the global streaming gateway will automatically invoke the Web browser to show the page content which including the invocation of media player when the Web browser sees the object tag.
13. The media player then communicates with the assigned media server to start the media streaming service. Any interactive playback control now is between the client and the assigned media server.

5  Case 2: Recording client requests to store pre-recorded media into media server.

1. Client sends out the request to record movie A by using client media recorder.
2. Request handler receives the request.
3. Request handler dispatches the request to global streaming gateway.
4. The global streaming gateway sends the request to the authentication component.
5. Authentication component verifies the client identity by consulting with user database and make sure the client has been granted the right to perform the request.
6. When the client gains the access right, the global streaming gateway passes the request to the admission control to make sure the new request will not interfere with existing service and meet the product license agreement as well.
7. Admission control checks if space has been allocated from storing movie A. If the space is not allocated in the global streaming gateway, the request will pass to a nearby request handler for further process and repeat Step 2 – Step 6.
8. When the admission control component accepts the request, the global streaming gateway passes the request to the load balancer component.
9. The load balancer component consults with the performance monitor component to understand the current server workload. Then based on the load balance scheme, it returns the address of the media server ready to serve the request.
10. The global streaming gateway returns an HTML page with the assigned media server information.
11. The media recorder then communicates with the assigned media server to start the media streaming service.
Case 3: Remote management client requests to monitoring server operational status.

1. Client sends out the request to observe a server’s status.
2. Request handler receives the request.
3. Request handler dispatches the request to global streaming gateway.
4. Global streaming gateway sends the request to the authentication component.
5. Authentication component verifies the client identity by consulting with user database and make sure the client has been granted the right to perform the request.
6. The global streaming gateway consults with the performance monitor component to understand the current server workload.
7. The performance monitor interacts with media servers listed on the configuration database and retrieves the performance data and passes it back to the global streaming gateway.
8. The global streaming gateway packages the performance data as an HTML page and returns it to the client.
9. The remote management client then displays the performance data into a graphic user interface (GUI) representation.

Implementation Approach:

In the following, methods are describes to determine a media server’s streaming capability, allocate a new subscriber media data to a particular media server, implement a load balancer component, and to implement an admission control component. These will be discussed in terms of a particular embodiment, all this readily extends to a more general case. For example, the amount of storage space in a local or shared media depository is often be referred to as disk space, although this could clearly be stored on an alternate media, the continuous media is often referred to as video, and the service or content provider is referred to as a retailer. For example, the system could be provided by a telecommunications company which would then provide this system for retails to store content for
supplying to users. Although these are particular examples of the more general situation, they will often be referred to in this way for simplicity of exposition.

The discussion begins with an approach for determination of a media server's streaming capability. Because maintaining streaming quality is an important requirement in any media streaming service, the system needs to control the number of streaming sessions to ensure every session is running at high quality. Therefore, determining server's streaming capacity is considered first. There are several configuration factors affecting capability of server machines including CPU power, main memory size ($M$), network bandwidth ($NB$), and storage access bandwidth for the local and shared repositories ($SBDL$, $SBDS$). In the determination method below, CPU power is excluded from the calculations since it is general found that streaming service is an input/output-intensive application and the CPU power is much less significant than other factors: if this may be a limiting factor in a particular application, it may be readily included with the other limiting factors.

Another parameter that affects the streaming server's operation is media data retrieval cycle time ($p$). Since media data is too large to be fit into the main memory space of a processor in one access, it is reasonable to retrieve media data one block at a time from the media repository and send it to the client. The retrieve-and-send pattern repeats until the whole media data is sent. The media data retrieval cycle time corresponds to the time required for one such block to be played back or consumed by the user. In this scenario, the system can use a simple two-buffer scheme: one for holding the media data being sent by networking module and the other one is storing next media data block from the disk access module. The role of the two buffers exchange roles at next cycle.

Based on the above discussion, an exemplary method to determine each media server's streaming capability can be defined. The streaming capability is abstracted into one single value, $SB_n$, the concurrent number of streams for streaming of video bit rate at $b$. This represents the upper limit of the streaming requests the server can accept without jeopardizing the streaming quality. To again simplify the discussion, the below uses a single media data retrieval cycle time, $p$, and a single bit
rate, b, for streaming video. In the more general situation, these are independent in each processor and would have distinct values \( p_i \) and \( b_i \). The following formulae all generalize to this case in a straightforward manner.

Figure 9 shows a detail of Figure 8 to highlight the portions relevant to this discussion. A list of n total servers \( S_1-S_n \) are shown, each connected to its respective local media depository \( LMD_1-LMD_n \). A single shared media depository \( SMD \) is shown, to which \( m \) of the servers are connected, where \( 0 \leq m \leq n \). These \( m \) servers are connected through block 847, which could be a fibre channel arbitration loop or some other switching mechanism.

The ability of the servers can then be quantified by the quantities \( SB_i \), server \( i \)'s streaming bandwidth in number of streams, and \( SS_i \), server \( i \)'s space capacity. The first of these can be defined by

\[
SB_i = \min<SB_i, SBD_i, SBDM_i>,
\]

where this expression ignores the CPU as limiting factor. Similarly, if any of these factors is not found limiting, it can be ignored, or, conversely, other limiting factors can added. In this expression, \( SB_i \) is server \( i \)'s aggregated network interface card bandwidth in terms of number of streams,

\[
SB_i = \left\lfloor \frac{NB_i}{b} \right\rfloor,
\]

where \( \lfloor \rfloor \) around a quantity indicates the integer part and in the more general case \( b \) would also have the subscript \( i \) here and below. \( SBD_i \) is server \( i \)'s disk access bandwidth in terms of number of streams,

\[
SBD_i = \left\lfloor \frac{SBDL_i}{b} \right\rfloor.
\]

If server \( i \) is not connected to the shared media depository, or

\[
SBD_i = \left\lfloor \frac{SBDL_i}{b} \right\rfloor + \left\lfloor \frac{SBDS_i}{m \cdot b} \right\rfloor.
\]
if server \( i \) is connected to the shared media depository. In this last expression, \( SBDL \) denotes the concurrent number of streams limited by access module for the local depository and similarly \( SBDS \) is server disk access bandwidth for the shared media depository at time \( t \). This last term assumes that the access to the shared depository is evenly divided among the \( m \) servers. In a more general arrangement, block 847 in Figure 9 could divide up \( SBDS \) among the \( m \) servers asymmetrical, with a fraction other than \( 1/m \) for each. \( SBM_i \) is server \( i \)'s memory bandwidth in terms of number of streams. Here

\[
SBM_i = \left| \frac{M_i}{2 \cdot b \cdot p} \right|
\]

where \( b \) replaced by \( b_i \) in the more general case and the two is due to two-buffer scheme used for the memory in the exemplary embodiment.

The expression for the streaming bandwidth can be described with reference to Figure 9. For the example of a server not connected to the shared media depository, consider server \( S_1 \) 841-1. This can stream media from \( LMD_1 \) 843-1 through 851-1 into server \( S_1 \) 841-1. There it must pass through main memory \( MM \) 857-1 before passing on to the network along 855-1. The bottlenecks in this process can be the limit of the number of streams into the server from the depository along 851-1, the number of streams out of the server along 855-1, of the number of streams the server is able to concurrent handle through it main memory \( MM \): these three effects correspond to the three terms in the expression for \( SB_i \). If the server is connected to the shared depository, such as server \( S_n \) 841-n, then the number of streams from the shared media depository into the server along 853-n should also be included with those along 851-n.

The space capacity of server \( i \), \( SS_n \), can be express as

\[
SS_i = SSL_i
\]

if server \( i \) is not connected to the shared media depository, or

\[
SS_i = SSL_i + \langle 0, \ldots, SSS \rangle = SSL_i + SSS_i
\]
if server $i$ is connected to the shared media depository. Here $<0, \ldots, SSS>$ is the range of possible values, form zero to $SSS$, the total disk space of the shared media depository, and $SSS'$ is the total disk space of the shared media depository within this range at a time $t$.

When a new service provider, or retailer, joins the media streaming service, the provider can more effectively be assigned servers if a retailer profile is provided. The system can then allocate the resource to satisfy the new subscriber's requested requirements in light of the different servers relative abilities. In this way, the high-power server can service more requests than the one with less power. The resources mentioned here include disk space, disk access bandwidth, and network bandwidth. Although represented the same in the figures, each media server's configuration may be different. For instance, two media servers may have storage space of (10GB, 50GB), storage access bandwidth of (40 megabytes per second, 80 megabytes per second), and networking bandwidth of (240 megabits per second, 360 megabits per second).

A request from a new retailer, or for additional capability from a current retailer, can include a number of requirements. In this exemplary embodiment, a request is defined as

$$R_i = <RB_i, RS_i, RL_i>,\$$

where $RB_i$ is retailer $i$'s streaming requirements in terms of number of streams; $RS_i$ is retailer $i$'s space requirements in, for example, terms of number of streams; and $RF_i$ is retailer $i$'s fault tolerance requirements in fault tolerance degrees. The concept of fault tolerance can be defined in a number of ways. Here it is given in terms of the number of servers which the service provider can accept as non-functioning:

- if $RF_i = 0$, a fault tolerance of no servers down;
- if $RF_i = 1$, a fault tolerance of 1 server down and the service to the subscribers of retailer $i$ can still work;
- if $RF_i = j$, a fault tolerance of $j$ servers down and the service to the subscribers of retailer $i$ can still work.
Using this information, the system can then assign the new subscribers. In this allocation, the storage space usage must be within the system's limitations. For storage access bandwidth and networking bandwidth, the assigned resource may exceed the system limitation. This is possible since the allocation is dealing with the static scenario. In other words, the worst case is that every request is asking for streaming service on a particular server. The admission control component will and should recognize the possibility and deny the request at first place. This should avoid the system overloaded.

Figure 10 is a flow chart showing an exemplary embodiment of a retailer allocation scheme based on a request as defined above. This assumes that $k$ retailers with requests $R_i, i=1, \ldots, k$, are asking for new or additional allocations. If no retailers are currently allocated, the flow will use the initial capacities corresponding to the capabilities as defined above for the servers. Otherwise, it will start with the quantities as currently updated in the previous allocation determination corresponding to step 1013 in Figure 10.

If more than one retailer $i$ is making a request $R_i$, these requests are sorted in step 1001. This step sorts the $R_i$ based on the following criteria:

- first, sort $RB_i$ such that $RB_i \geq RB_{i+1}, 1 \leq i \leq k-1$
- if the same bandwidth requirement, the sort $RS_i$ such that $RS_i \leq RS_{i+1}$, $1 \leq i \leq k-1$
- if the same bandwidth and space requirements, the sort $RF_i$ such that $RF_i \geq RF_{i+1}, 1 \leq i \leq k-1$.

This particular scheme emphasizes the number of streams requested, $RB$, relative to the other two quantities. This order could be rearranged, for example using the fault tolerance as the first stage in this sieve if the system wished to stress this quantity, or deleted the storage space requirement comparison $RS$ if space limits are not a concern.

If only one request is presented, step 1001 would be skipped and the process would go directly to step 1005. If there are multiple requests, the process goes
through these one at a time in the order established in step 1001, beginning with \( R_i \) in step 1003.

Step 1005 finds a list of servers which satisfy the following two criteria for \( j = 1 \) to \( n \): first, that both \( SB_{j \geq R_i} \) and either

\[
\left| \frac{SB_{DL_j}}{b} \right| \geq R_B, \text{ or } \left| \frac{SB_{DS}}{m \cdot b} \right| \geq R_B ;
\]

and, second, that \( SS_j \geq RS \), and either \( SSL_j \geq RS \), or \( SSS \geq RS \). This restricts consideration only those servers which can supply both the requested number of streams from either its local or the shared media depository and the requested amount of storage in either its local or the shared media depository. This servers which meet this requirement form the server candidate list. In this simplified exemplary embodiment, a given server will, for a given request, consider storage on either its local or the shared media depository. In a more general arrangement, the media will be distributed over both, for example caching in its local media depository a portion of a title which is stored in full in the shared media depository, using a arrangement such as that described in U. S. patent application \( \ldots \) by Horng-Juing Lee, entitled "METHOD AND APPARATUS FOR CACHING FOR STREAMING DATA", filed on September 8, 2000, which was included by reference above.

Step 1007 determines whether the server candidate list is large enough. If the number of servers in the server candidate list for request \( R_i \) is less than \( (RF_i + 1) \), then the system will deny the retailer request \( i \) in block 1008 and return to step 2. Otherwise, there are enough servers for the request and the list is sorted in step 1009.

In step 1009, the servers in the server candidate list are ordered such that

\[
\frac{SB_i}{SS_i} \leq \frac{SB_{i+1}}{SS_{i+1}}, \quad \ell = 1, \ldots, |\text{servercandidate list}| - 1.
\]
Thus they are ranked in an order based on their relative abilities, varying inversely with their space capacity, as accessible on the local and shared depositories, and varying directly with their streaming bandwidth. Unless this is the initial assignment for the system, these quantities are the updated values from a previous step 1013. 

Step 1011 then picks the first \( p = (RF_i + 1) \) servers from the sorted server candidate list. This is the selected server list \( \{S_1^m, \ldots, S_p^m\} \).

From this selected server list, the actual allocation is determined in step 1013. In terms of a pseudo-code, this can be described as:

For \( j = 1 \) to \( p \)

\[
\{ \text{let } x = S_j^m \}
\]

case 1: Server \( x \) has enough local disk space and enough local disk access bandwidth from local media depository.

\( \{ \text{update } SD_x, SBD_x, SS_x, SSL_x \} \)

case 2: Server \( x \) has enough shared disk space, enough shared disk access bandwidth, and the space has not been allocated on the shared media repository before.

\( \{ \text{update } SB_x, SBD_S, SS_S, SSS \} \)

case 3: Server \( x \) has enough shared disk space, enough shared disk access bandwidth, and the space has on the shared media repository.

\( \{ \text{update } SB_x, SBD_S \} \)

Here, case 1 is used if its condition is met; otherwise case 2 is used, unless its condition is not met, and then case 3 is used. The resultant updated values are the values which will used in the next time through the allocation process.

In step 1015, the current value of \( i \) is checked to see if all the requests have been dealt with. If not, it continues back to step 1005, incrementing \( i \) and repeating until all \( k \) requests are treated. Once they have been, a retailer allocation table is then created in step 1017.
Figure 11 shows one example of how such a retail allocation table can be constructed. This will list a retailer by its identifier and present the information from the request: the total number of streams, total storage (or disk) space, and the fault tolerance requirement. It will also give the associated information assigned in the allocation scheme, namely the locations assigned to the retailer in both the shared and the local media depositories.

Figure 12 is a media allocation table containing the information associated with a title stored in the media server cluster. Associated with a title will be its owner, given by the retailer's identifier, the locations within the local media depositories where it is stored, whether it is stored in the shared media depository, and a title license specifying how many copies of the title the retailer is entitled to use. When a media file from the retailer is ready to be put into a server, it is to be copied into the shared media depository, if the retailer has allocated it, or into the local media repositories if local media depository has been allocated. The file may have multiple local media repositories containing it to allow multiple servers to provide the title from their respective local media repositories.

The admission control function determines if a request for a particular title will be accepted. When a subscriber's request comes in, if accepting the request would exceed either the retailer's stream license, in terms of the number of streams allocated for the retailer, or the title's stream license, in terms of the number of streams allocated for the title, the request is rejected. If this is not the case, the admission control accepts the request and determines the server to serve the request.

Determining the server that will serve the request (for a read operation) is part of the load balancing function. When a request comes in it will be in the form

\[ \text{<subscriber ID, media title>} \]

In a string of pseudo-code, this determination process can be implemented as:

if (the title is in local media depository (maybe multiple servers) only)

then
{assign the request to the servers with the lowest \textit{SB}, value within the group of servers within which the title resides }

else if (the title is in the shared media depository only)

then

{assign the request to the servers with the lowest \textit{SB}, value }

The workload table is subsequently updated to reflect the assignment. Figure 13 shows the structure of a workload table. This will list the retailer by its identifier and the title associated with the request. The server filling this request is then listed along with whether the continuous media is being supplied from the identified servers local media depository or from the shared media depository.

Determine the server that will serve a request for a write operation when a retailer wants to input a file with a title license is a similar process similar to that for the read operation. As a string of pseudo-code, this determination process can be implemented as:

if (the subscriber’s allocation is shared media depository)

then

{copy it into shared media depository and update the media allocation table }

else if (the subscriber’s allocation is in multiple local media depositaries)

then

{copy it into multiple server locations and update the media allocation table }

Once this information is stored in the media allocation table, it can then be accessed for the media supplier to provide to a user upon request according to the previous procedure.
Architectural Features

The architecture of the present invention provides a number of features which improve upon the prior art as described in the Background section. Three aspects of this architectural structure are described in some detail below: scalability and reliability, load balancing, and value-added E-commerce aspects.

The aspect of server scalability and reliability provides access transparency on locating media server. From a client's point of view, there is no need to know where a particular piece of continuous media is really stored and which one of the working media servers should be contacted to provide it. By exploiting web server(s) as the front-end request handler, any client requests always go to the web server first. A component within the server called “global streaming gateway” will, based on the request and the media server workloads, redirect the request to the appropriate working media server. From that point on, the streaming service is between client and the assigned media server. The client need not know the exact media server location. The only thing client needs to know is the request handler’s URL link, nothing more.

This design provides easily scalable deployment. A system can start from a single media server, adding more media servers as the streaming demand increases. Adding/removing media servers from a media server cluster will not affect the client’s connection practices. All client requests still go to the front-end web site.

The described structure also provides scalability and availability for request handler(s). Since all client requests go through a web front end first, the availability of the request handlers should be assured. In addition, as the system expands, the number of requests increases accordingly. The request handlers preferably scales up as well to match up the increase in requests. By using web server(s) as the front-end component, the design can take advantage of all kinds of web server(s) scalability/reliability products, such as Microsoft’s WLBS (Windows NT Load Balancing Service). This kind of product can detect a faulty web server within a web server cluster and automatically redistribute the requests to other working web servers. In the meantime, it is easy to add an extra web server into the system with-
out interfere the operation. This design can ensure the request handler can receive all client requests.

Using the media server cluster concept also supports server scalability. Since it uses multiple media servers to service streaming requests, when the streaming demand increases, the system administrator can simply add one more media server into the media server cluster. A client need not know about the newly installed media server at all.

Using a de-coupled design among media servers enhances the scalability and reliability. In this architecture, each media server handles multiple streaming sessions. When one streaming session is established between a client and a media server, the server handles all media streaming from start to end. This design cuts down the dependency between media servers. Therefore, when one server goes down, it will not bring down the other servers. When it is time to add a new server, it can simply plug into the media server cluster without affecting the rest. Server scalability is thus easy to reach and deploy.

In addition to the above benefit, the de-coupling design does not restrict the system to using a collection of equivalent media servers. For instance, in the beginning, all of the servers within media server cluster can be equipped with, say, Ultra disk having a speed of 40 Mega Bytes Per Second (MBPS). The system may later be expanded by adding a newer server configured differently, such as with Ultra 2 disk having a speed of 80 MBPS, resulting in a heterogeneous server environment. Although the original media servers, having a 40 MBPS capability, have relatively loss capability, lacking any advantage of newer technology found in the newer server, all these servers of differing abilities may be retained in the system. This saves replacing all old servers with new ones each time a new server is introduced to the system, which would cost a lot of money and be impractical as the technology may again improve within a short period of time. With the help from the described load balancer, the de-coupling design can utilize the full capability of each individual media servers despite the differences in performance.
There are several aspects of load balancing. Load balancing is provided among the request handlers. Using web servers as the front-end request handlers enhances the architecture. By using existing web servers balancing product such as Microsoft's Windows NT Load Balancing Service (WLBS), it can distribute out the request to participating web servers and achieve the goal of load balancing. It also provides load balancing within the media server cluster. As we discussed above, each media server may not have the same capabilities. Our Load Balancer recognizes this and tries to balance out the streaming requests. The load balancing scheme maintains the equal utilization ratio in each media server.

The design can also incorporate valued-added E-commerce service by creating reply web page based on user profile: When a playback request comes in, it is represented as a URL link. In the present invention, the URL link represents a request rather than accessing an existing web page. After interpreting the URL link, the "global streaming gateway" component knows information such as user name and the requested media clip. By dynamically generating a response web page, the client will be redirected to the appropriate media server to start media playback. Based upon the dynamic feature and the known user name (and its profile information on user database), the "ad inserter" component can select a template (filled of presentation style and ad postings) from a template database and pass it to the "global streaming gateway" module. By combining the template with the object tag (indicating the playback information), the "global streaming gateway" can send back the reply HTML page to the client.

Application Domains

As already noted, the described structures and methods are suitable for continuous media besides video related service, and deliverable over both the Internet and Intranet environments. This includes (but not limit to) video on demand service, video mail service, movie on demand service, etc.

Additionally, although this discussion has focused on streaming continuous media, these techniques extend to the non-continuous data. This is particularly so
where the amount of data being transmitted for a single media data title is, although not continuous, very large. An example is the transmission of an image, for example a high-resolution X-ray. Here the amount of data may be of sufficient size that it is more practical to transmit the particular media data title broken up into blocks as is done for the continuous case. The limits on transmitting this data then become the same as for the continuous case, with similar storage and transmission bandwidth concerns.

The media delivery system as described herein is robust, operationally efficient and cost-effective. In addition, the present invention may be used in connection with presentations of any type, including sales presentations and product/service promotion, which provides the video service providers additional revenue resources.

The processes, sequences or steps and features discussed herein are related to each other and each are believed independently novel in the art. The disclosed processes and sequences may be performed alone or in any combination to provide a novel and nonobvious file structure system suitable for media delivery system. It should be understood that the processes and sequences in combination yield an equally independently novel combination as well, even if combined in their broadest sense.

Other Embodiments

The invention has now been described with reference to specific embodiments. Other embodiments will be apparent to those of skill in the art. In particular, a user digital information appliance has generally been illustrated as a personal computer. However, the digital computing device is meant to be any device for interacting with a remote data application, and could include such devices as a digitally enabled television, cell phone, personal digital assistant, etc.

Furthermore, while the invention has in some instances been described in terms of client/server application environments, this is not intended to limit the invention to only those logic environments described as client/server. As used
herein, "client" is intended to be understood broadly to comprise any logic used to access data from a remote system and "server" is intended to be understood broadly to comprise any logic used to provide data to a remote system.

It is understood that the examples and embodiments described herein are for illustrative purposes only and that various modifications or changes in light thereof will be suggested by the teachings herein to persons skilled in the art and are to be included within the spirit and purview of this application and scope of the claims and their equivalents.

10 Embodiment in a Programmed Information Appliance

As shown in Figure 14, the invention can be implemented in hardware and/or software. In some embodiments of the invention, different aspects of the invention can be implemented in either client-side logic or a server-side logic. As will be understood in the art, the invention or components thereof may be embodied in a fixed media program component containing logic instructions and/or data that when loaded into an appropriately configured computing device cause that device to perform according to the invention. As will be understood in the art, a fixed media program may be delivered to a user on a fixed media for loading in a users computer or a fixed media program can reside on a remote server that a user accesses through a communication medium in order to download a program component.

Figure 14 shows an information appliance (or digital device) 1400 that may be understood as a logical apparatus that can read instructions from media 1417 and/or network port 1419. Apparatus 1400 can thereafter use those instructions to direct server or client logic, as understood in the art, to embody aspects of the invention. One type of logical apparatus that may embody the invention is a computer system as illustrated in 1400, containing CPU 1407, optional input devices 1409 and 1411, disk drives 1415 and optional monitor 1405. Fixed media 1417 may be used to program such a system and may represent a disk-type optical or magnetic media, magnetic tape, solid state memory, etc. The invention may be embodied in whole or in part as software recorded on this fixed media. Communication port
1419 may also be used to initially receive instructions that are used to program such a system and may represent any type of communication connection.

The invention also may be embodied in whole or in part within the circuitry of an application specific integrated circuit (ASIC) or a programmable logic device (PLD). In such a case, the invention may be embodied in a computer understandable descriptor language which may be used to create an ASIC or PLD that operates as herein described.
WHAT IS CLAIMED IS:

1. A method of operating a system to supply media titles over a network, wherein each of said media titles is divided into a plurality of blocks, comprising:

   providing a media deliver system comprising:

   a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;

   a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;

   receiving a request for a new service provider to use the media delivery system to provide media titles to users; and

   allocating the new service provider to one or more media servers based upon the relative ability of the media servers to provide media titles stored on the local and shared depositories to the network at the time of receiving the request.

2. The method of claim 1, wherein said media titles include continuous media.

3. The method of claim 2, wherein the relative ability of a media server varies directly with the number of streams the media server can provide from the local and shared depositories to the network.

4. The method of claim 3, wherein the relative ability of a media server is proportional to the ratio of the number of streams the media server can provide from the local and shared depositories to the network to the storage capacity on the respective local media depository and shared media depository that the server is able to access.
5. The method of claim 2, wherein the request includes the number of streams of stored media to the network that the new service provider requires, and wherein media servers are allocated only from among those servers that can provide at least as many such streams as requested.

6. The method of claim 5, wherein each of the servers has a maximum number of streams between itself and its respective local media depository and a maximum number of streams between itself and the shared media depository, and wherein the media servers are allocated only from among those servers that can provide at least as many streams between itself and one of either its respective local media depository or the shared media depository as the number of streams that the new service provider requires.

7. The method of claim 6, wherein the maximum number of streams between a server and its respective local media depository is the server's local storage access bandwidth divided by the server's streaming bit rate, and wherein the maximum number of streams between a server and the shared media depository is the server's shared storage access bandwidth at the time of the request divided by the product of the server's streaming bit rate and the number of servers connected to the shared media depository.

8. The method of claim 2, wherein the request includes the amount of storage the service provider requires, and wherein media servers are allocated only from among those servers that can provide at least as much storage as requested.

9. The method of claim 8, wherein at the time of the request each of the servers has a maximum amount of accessible storage space on its respective local media depository and a maximum amount of accessible storage space on the shared media depository, and wherein the media servers are allocated
only from those servers for which either the maximum amount of accessible storage space on its respective local media depository or the maximum amount of accessible storage space on the shared media depository exceeds the amount of storage requested at the time of the request.

10. The method of claim 2, wherein the request includes the number of server failures that the service provider can tolerate, and wherein the number of media server's allocated exceeds the number of server failures that the service provider can tolerate.

11. The method of claim 10, wherein the media servers allocated are those with the lowest ratio of the number of streams the media server can provide from the local and shared depositories to the network at the time of receiving the request to the storage capacity on the respective local media depository and shared media depository that it is able to access network at the time of receiving the request.

12. The method of claim 2, wherein the request includes the number of streams of stored media to the network that the new service provider requires and the amount of storage the new service provider requires, and wherein if an allocated server can provide the requested number of stream and requested amount of storage from the server's respective media depository, the server stores media from the service provider on the allocated server's local media depository.

13. The method of claim 2, wherein the request includes the number of streams of stored media to the network that the new service provider requires and the amount of storage the new service provider requires, and wherein if an allocated server can provide both the requested number of stream and requested amount of storage from the shared media depository but not from the server's
respective local media depository, the server stores media from the service provider on the shared media depository.

14. A method of operating a system to supply media titles over a network, wherein each of said media titles is divided into a plurality of blocks, comprising:

- providing a media deliver system comprising:
  - a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;
  - a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;
  - receiving a plurality of requests for new service providers to use the media delivery system to provide media titles to users, wherein each of the requests includes the number of streams to the network that the new service provider requires; and
  - allocating each of the new service providers to one or more media servers based upon the relative ability of the media servers to provide media titles stored on the local and shared depositories to the network at the time of receiving the requests, wherein service providers are allocated servers based upon the number of streams from the local and shared depositories to the network that the new service provider requires, service providers requiring more such streams being given higher priority.

15. The method of claim 14, wherein said media titles include continuous media.
16. The method of claim 15, wherein the relative ability of a media server varies directly with the number of streams the media server can provide from the local and shared depositories to the network.

17. The method of claim 16, wherein the relative ability of a media server is proportional to the ratio of the number of streams the media server can provide from the local and shared depositories to the network to the storage capacity on the respective local media depository and shared media depository that the server is able to access.

18. The method of claim 15, wherein media servers are allocated to a respective one of the new service providers only from among those servers that can provide at least as many such streams as requested by the respective one of the new service providers.

19. The method of claim 18, wherein each of the servers has a maximum number of streams between itself and its respective local media depository and a maximum number of streams between itself and the shared media depository, and wherein the media servers are allocated to the respective one of the new service providers only from among those servers that can provide at least as many streams between itself and one of either its respective local media depository or the shared media depository as the number of streams that the respective new service provider requires.

20. The method of claim 19, wherein the maximum number of streams between a server and its respective local media depository is the servers local storage access bandwidth divided by the server's streaming bit rate, and wherein the maximum number of streams between a server and the shared media depository is the server's shared storage access bandwidth at the time of the request.
divided by the product of the server's streaming bit rate and the number of servers connected to the shared media depository.

21. The method of claim 15, wherein each of the requests additionally includes the amount of storage the service provider requires, and wherein for service providers requiring the same number of streams from the local and shared depositories to the network, service providers are additionally allocated servers based upon the amount of storage the service provider requires, service providers requiring lesser amounts of storage being given higher priority.

22. The method of claim 21, wherein media servers are allocated to a respective one of the new service providers only from among those servers that can provide at least as much storage as requested by the respective one of the new service providers.

23. The method of claim 22, wherein at the time of allocating servers each of the servers has a maximum amount of accessible storage space on its respective local media depository and a maximum amount of accessible storage space on the shared media depository, and wherein the media servers are allocated to a respective one of the new service providers only from those servers for which either the maximum amount of accessible storage space on its respective local media depository or the maximum amount of accessible storage space on the shared media depository exceeds the amount of storage requested at the time of the respective server's request.

24. The method of any of claims 3, 5, 8, 16, 18 or 22, wherein each of the servers has a first maximum number of streams to the network, has a second maximum number of streams from the combination of the server's respective local media depository and the shared media depository, and can process a third maximum number of streams through the server's memory, wherein the number of
combined streams a server can provide from the server's respective local media
depository and the shared media depository to the network is the minimum of the
first, second, and third maximum numbers.

25. The method of any of 24, wherein the maximum number of
streams a server can through the server's memory is the main memory size divided
by twice the product of the server's streaming bit rate times the server's media data
retrieval cycle time, the maximum number of streams to the network the server can
provide is the aggregated networking bandwidth divided by the server's streaming
bit rate, and the maximum number of streams to its respective local media depository
and the shared media depository the server can provide is sum of the respective local
media depository storage access bandwidth divided by the server's streaming bit rate
plus the shared media depository shared storage access bandwidth accessible at the
time of the request divided by the product of the server's bit rate and the number of
servers connected to the shared media depository.

26. The method of claim 21, wherein each of the requests
additional includes the number of server failures that the service provider can
tolerate, and wherein for service providers requiring the same number of streams
from the local and shared depositories to the network and the same amount of
storage space, service providers are additionally allocated servers based upon the
number of server failures that the service provider can tolerate, service providers that
can tolerate more server failures being given higher priority.

27. The method of claim 26, wherein the number of media server's
allocated to a given service provider exceeds the number of server failures that the
given service provider can tolerate.

28. The method of claim 27, wherein the media servers allocated
are those with the lowest ratio of the number of streams the media server can
provide from the local and shared depositories to the network at the time of allocating servers to the storage capacity on the respective local media depository and shared media depository that it is able to access network at the time of allocating servers.

30. The method of claim 16, wherein if an allocated server can provide the requested number of stream and requested amount of storage from the server's respective media depository, the server stores media from the service provider on the allocated server's local media depository.

31. The method of claim 16, wherein if an allocated server can provide both the requested number of stream and requested amount of storage from the shared media depository but not from the server's respective local media depository, the server stores media from the service provider on the shared media depository.

32. A method of operating a system to supply continuous media over a network, comprising:
   providing a media deliver system comprising:
   a plurality of local media depositories connected to the network to supply media stored therein to the network;
   a shared media depository connected to the network to supply media stored therein to the network;
   receiving a request for a new service provider to use the media delivery system to provide continuous media to users, wherein the request includes information on the service provider's performance requirements; and
   allocating the new service provider a respective maximum number of streams from each of the local media depositories to the network and a maximum number of streams from the shared media depository to the network based upon the service provider's performance requirements.
33. The method of claim 32, wherein the service provider's performance requirements includes the number of streams of stored media to the network that the new service provider requires.

34. The method of claim 33, wherein the service provider's performance requirements further includes the amount of storage the service provider requires.

35. The method of claim 33, wherein one of either the maximum number of streams from each of the local media depositories to the network or the maximum number of streams from the shared media depository to the network are zero.

36. A method of supplying continuous media from a service provider over a network to one or more users, comprising:

   providing a media deliver system comprising:
   
   a plurality of local media depositories connected to the network to supply media identified by title stored therein to the network, wherein the service provider is allocated a respective maximum number of streams from each of the local media depositories;

   a shared media depository connected to the network to supply media identified by title stored therein to the network, wherein the service provider is allocated a maximum number of streams from the shared media depository;

   receiving a request from one of the users to receive one or more new streams of continuous media identified by title from the service provider; and

   accepting the request only if the resultant number of streams that the service provider will provide to the users if the request is accepted does not exceed
any of said maximum number of streams allocated to the service provider within the media deliver system.

37. The method of claim 36, wherein the titles stored on the shared media depository are distinct from those on the local media depositories.

38. The method of claim 36, wherein for each of the titles stored therein a respective maximum number of streams is allocated, and further comprising accepting the request only if the resultant number of streams that the service provider will provide if the request is accepted for each of the titles does not exceed any of the respective maximum number of streams for each of the titles.

39. The method of claim 36, further comprising:
assigning an accepted request to the server connected to supply the title which has the lowest the maximum number of combined streams at the time of receiving the request.

40. A method of operating a system to supply continuous media over a network to a user, comprising:

providing a media deliver system comprising:

a plurality of media servers connected to the network, each connected to a respective local media depository to supply media identified by title stored therein to the network;

a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media identified by title stored therein to the network, and wherein each of the servers has a maximum number of combined streams to the network from the server's respective local media depository and the shared media depository;

receiving a request from a user to receive a new stream of continuous media identified by title stored on the media deliver system; and
assigning the request to the server connected to supply the title which has the lowest the maximum number of combined streams at the time of receiving the request.

41. The method of claim 40, wherein the titles stored on the shared media depository are distinct form those on the local media depositories.

42. The method of claim 40, wherein each of the servers has a first maximum number of streams to the network, has a second maximum number of streams from the combination of the server's respective local media depository and the shared media depository, and can process a third maximum number of streams through the server's memory, wherein the maximum number of combined streams to the network from the server's respective local media depository and the shared media depository is the minimum of the first, second, and third maximum numbers.

43. A method of operating a system to supply continuous media over a network, comprising:

providing a media deliver system comprising:

a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;

a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;

receiving a request for a new service provider to use the media delivery system to provide continuous media to users, wherein the request includes a fault tolerance requirement; and

allocating the new service provider to one or more media servers based upon the fault tolerance requirement.
44. The method of claim 43, wherein the fault tolerance requirement includes the number of server failures that the service provider can tolerate, and wherein the number of media server's allocated exceeds the number of server failures that the service provider can tolerate.

45. A media deliver system for providing continuous media over a network, comprising:

a plurality $N$ of local media depositories to store continuous media;
a plurality $N$ of media servers connected to the network, each connected to a respective one of said plurality of $N$ local media depositories to stream continuous media stored therein to the network; and

a shared media depository to store continuous media, wherein a plurality of the media servers are connected to the shared media depository to stream continuous media stored therein to the network.

46. The media deliver system of claim 45, wherein said media is identified by title, and wherein a given title is stored in one of either the shared media depository or in a set of one or more of the local media depositories.

47. The media deliver system of claim 45, wherein not all of said media servers are of equivalent capability.

48. The media deliver system of claim 45, wherein each of the servers has a first maximum number of streams to the network, has a second maximum number of streams from the combination of the server's respective local media depository and the shared media depository, and can process a third maximum number of streams through the server's memory, wherein the number of combined streams a server can stream from the server's respective local media depository and the shared media depository to the network is the minimum of the first, second, and third maximum numbers.
49. The media deliver system of claim 48, wherein the maximum number of streams a server can through the server's memory is the main memory size divided by twice the product of the server's streaming bit rate times the server's media data retrieval cycle time, the maximum number of streams to the network the server can provide is the aggregated networking bandwidth divided by the server's streaming bit rate, and the maximum number of streams to its respective local media depository and the shared media depository the server can provide is sum of the respective local media depository storage access bandwidth divided by the server's streaming bit rate plus the shared media depository shared storage access bandwidth accessible at the time of the request divided by the product of the server's bit rate and the number of servers connected to the shared media depository.

50. The media deliver system of claim 45, wherein a plurality of service providers each store continuous media in the shared and local media depositories for streaming to one or more users over the network.

51. The media deliver system of claim 50, wherein said continuous media is identified by title, and wherein each of the service providers is allocated a respective maximum number of streams from the shared and local media depositories to the network for a given title.

52. The media deliver system of claim 50, wherein said continuous media is identified by title, and wherein for each of the service providers each title stored is stored in one of either the shared media depository or in a set of one or more of the local media depositories.

53. The media deliver system of claim 50, wherein each of the service providers is allocated a respective maximum number of streams from the shared and local media depositories to the network.
54. The media deliver system of claim 53, wherein each of said respective maximum number of streams is distributed between the media servers based upon the relative abilities of the media servers to provide continuous media stored on the local and shared media depositories to the network.

55. The media deliver system of claim 53, wherein the relative ability of each of said media servers is proportional to the ratio of the number of streams the media server can provide from the local and shared depositories to the network to the storage capacity on the respective local media depository and shared media depository that the server is able to access.

56. The media deliver system of claim 53, wherein each of said respective maximum number of streams is distributed between the media servers based upon the service provider's requirements for streaming continuous media in the shared and local media depositories to one or more users over the network.

57. A computer readable storage device embodying a program of instructions executable by a computer to perform a method of operating a system to supply continuous media over a network, said method comprising:

provide a media deliver system comprising:

a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;

a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;

receiving a request for a new service provider to use the media delivery system to provide continuous media to users; and
allocating the new service provider to one or more media servers based upon the relative ability of the media servers to provide continuous media stored on the local and shared depositories to the network at the time of receiving the request.

58. The method of claim 57, wherein the relative ability of a media server is proportional to the ratio of the number of streams the media server can provide from the local and shared depositories to the network to the storage capacity on the respective local media depository and shared media depository that the server is able to access.

59. The method of claim 57, wherein the request includes the number of streams of stored media to the network that the new service provider requires, and wherein media servers are allocated only from among those servers that can provide at least as many such streams as requested.

60. The method of claim 57, wherein the request includes the amount of storage the service provider requires, and wherein media servers are allocated only from among those servers that can provide at least as much storage as requested.

61. The method of claim 57, wherein the request includes the number of server failures that the service provider can tolerate, and wherein the number of media server's allocated exceeds the number of server failures that the service provider can tolerate.

62. A method for transmitting a program of instructions executable by a computer to perform a process of operating a system to supply continuous media over a network, said method comprising:
causing the transmission to a client device a program of instructions, thereby enabling the client device to perform, by means of such program, the following process:

providing a media deliver system comprising:

- a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;
- a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;

receiving a request for a new service provider to use the media delivery system to provide continuous media to users; and

allocating the new service provider to one or more media servers based upon the relative ability of the media servers to provide continuous media stored on the local and shared depositories to the network at the time of receiving the request.

63. The process of claim 62, wherein the relative ability of a media server is proportional to the ratio of the number of streams the media server can provide from the local and shared depositories to the network to the storage capacity on the respective local media depository and shared media depository that the server is able to access.

64. The process of claim 62, wherein the request includes the number of streams of stored media to the network that the new service provider requires, and wherein media servers are allocated only from among those servers that can provide at least as many such streams as requested.

65. The process of claim 62, wherein the request includes the amount of storage the service provider requires, and wherein media servers are
allocated only from among those servers that can provide at least as much storage
as requested.

66. The process of claim 62, wherein the request includes the
number of server failures that the service provider can tolerate, and wherein the
number of media server's allocated exceeds the number of server failures that the
service provider can tolerate.

67. A computer readable storage device embodying a program of
instructions executable by a computer to perform a method of operating a system to
supply continuous media over a network, said method comprising:

providing a media deliver system comprising:

a plurality of media servers connected to the network, each
connected to a respective local media depository to supply media stored
therein to the network;

a shared media depository, wherein two or more of the media
servers are connected to the shared media depository to supply media stored
therein to the network;

receiving a plurality of requests for new service providers to use the
media delivery system to provide continuous media to users, wherein each of the
requests includes the number of streams to the network that the new service provider
requires; and

allocating each of the new service providers to one or more media
servers based upon the relative ability of the media servers to provide continuous
media stored on the local and shared depositories to the network at the time of
receiving the requests, wherein service providers are allocated servers based upon
the number of streams from the local and shared depositories to the network that the
new service provider requires, service providers requiring more such streams being
given higher priority.
68. The method of claim 67, wherein each of the requests additionally includes the amount of storage the service provider requires, and wherein for service providers requiring the same number of streams from the local and shared depositories to the network, service providers are additionally allocated servers based upon the amount of storage the service provider requires, service providers requiring lesser amounts of storage being given higher priority.

69. The method of claim 68, wherein each of the requests additionally includes the number of server failures that the service provider can tolerate, and wherein for service providers requiring the same number of streams from the local and shared depositories to the network and the same amount of storage space, service providers are additionally allocated servers based upon the number of server failures that the service provider can tolerate, service providers that can tolerate more server failures being given higher priority.

70. A method for transmitting a program of instructions executable by a computer to perform a of operating a system to supply continuous media over a network, said method comprising:

causing the transmission to a client device a program of instructions, thereby enabling the client device to perform, by means of such program, the following process:

providing a media deliver system comprising:

a plurality of media servers connected to the network, each connected to a respective local media depository to supply media stored therein to the network;

a shared media depository, wherein two or more of the media servers are connected to the shared media depository to supply media stored therein to the network;

receiving a plurality of requests for new service providers to use the media delivery system to provide continuous media to users, wherein each of the
requests includes the number of streams to the network that the new service provider requires; and

allocating each of the new service providers to one or more media servers based upon the relative ability of the media servers to provide continuous media stored on the local and shared depositories to the network at the time of receiving the requests, wherein service providers are allocated servers based upon the number of streams from the local and shared depositories to the network that the new service provider requires, service providers requiring more such streams being given higher priority.

71. The process of claim 70, wherein each of the requests additionally includes the amount of storage the service provider requires, and wherein service providers requiring the same number of streams from the local and shared depositories to the network, service providers are additionally allocated servers based upon the amount of storage the service provider requires, service providers requiring lesser amounts of storage being given higher priority.

72. The process of claim 71, wherein each of the requests additionally includes the number of server failures that the service provider can tolerate, and wherein service providers requiring the same number of streams from the local and shared depositories to the network and the same amount of storage space, service providers are additionally allocated servers based upon the number of server failures that the service provider can tolerate, service providers that can tolerate more server failures being given higher priority.

73. A computer readable storage device embodying a program of instructions executable by a computer to perform a method of operating a system to supply continuous media over a network, said method comprising:

providing a media deliver system comprising:
55

a plurality of local media depositories connected to the network to supply media stored therein to the network;
a shared media depository connected to the network to supply media stored therein to the network;
receIVING a request for a new service provider to use the media delivery system to provide continuous media to users, wherein the request includes information on the service provider's performance requirements; and
allocating the new service provider a respective maximum number of streams from each of the local media depositories to the network and a maximum number of streams from the shared media depository to the network based upon the service provider's performance requirements.

74. The method of claim 73, wherein upon the service provider's performance requirements include a fault tolerance requirement.

75. A method for transmitting a program of instructions executable by a computer to perform a process of operating a system to supply continuous media over a network, said method comprising:
causing the transmission to a client device a program of instructions,
thereby enabling the client device to perform, by means of such program, the following process:
providing a media deliver system comprising:
a plurality of local media depositories connected to the network to supply media stored therein to the network;
a shared media depository connected to the network to supply media stored therein to the network;
receIVING a request for a new service provider to use the media delivery system to provide continuous media to users, wherein the request includes information on the service provider's performance requirements; and
allocating the new service provider a respective maximum number of streams from each of the local media repositories to the network and a maximum number of streams from the shared media repository to the network based upon the service provider's performance requirements.

76. The process of claim 75, wherein upon the service provider's performance requirements include a fault tolerance requirement.
START

NO

RECEIVE ANY QUEST?

YES

ADMINISTRATION PROCESS

RETRIEVE SYSTEM PERFORMANCE INFORMATION ON EACH OF VIDEO SERVERS

DETERMINE AN APPROPRIATE VIDEO SERVER TO PROVIDE VIDEO SERVICE TO THE REQUEST

SEND OUT A RESPONSE IN RESPONDING TO THE REQUEST, RESPONSE INCLUDING AN ADDRESS IDENTIFIER IDENTIFYING THE APPROPRIATE VIDEO SERVER

END

GLOBAL STREAMING GATEWAY

Figure 4
START

NO

RECEIVE ANY QUEST?

YES

PROXY SERVER

EXTRACT A TITLE

LOOK UP IN CACHE IF THERE IS ANY PORTION OF VIDEO CORRESPONDING TO THE TITLE?

YES

COMPLETE?

NO

NO

COMPLETE?

YES

DETERMINE NUMBER OF CACHE UNITS

GENERATE AND SEND A REQUEST TO CENTRAL VIDEO SERVER FOR THE VIDEO TITLE

INCREMENT NUMBER OF REQUESTS

DOWNLOADING THE VIDEO INTO CACHE

PROVIDING THE VIDEO IN RESPONDING TO THE REQUEST

CACHE UPDATE

FIGURE 5
START

2ND VIDEO SERVER

NO RECEIVED ANY QUEST?

YES

LOOK UP IN USER DB
IF THE REQUEST IS FROM AN AUTHORIZED PROXY SERVER/USER

NO

REQUEST IS DENIED

YES

EXTRACT A TITLE FROM THE REQUEST

DETERMINE NUMBER OF CACHE UNITS OF THE VIDEO TO BE DELIVERED

PROVIDING THE NUMBER OF CACHE UNITS IN RESPONDING THE REQUEST

END

Figure 6
SORT \( R_i \), \( i = 1 \to n \)

1003

\( i = 1 \)

1005

FORM A SERVER CANDIDATE LIST

1007

SIZE OF CANDIDATE LIST \( \neq \emptyset \) ?

NO

DENY REQUEST \( R_i \)

1008

YES

SORT THE CANDIDATE LIST

1009

PACK FIRST \( (\lfloor \log_2 R \rfloor + 1) \) SERVERS FROM LIST

1011

DETERMINE ALLOCATION

1013

\( i < k \) ?

YES

CREATE RETAILER ALLOCATION TABLE

1015

1017

FIGURE 10
<table>
<thead>
<tr>
<th>RETAILER ID</th>
<th>MEDIA TITLE</th>
<th>SERVER ID</th>
<th>LOCAL OR SHARED</th>
</tr>
</thead>
</table>

**FIGURE 13**
**INTERNATIONAL SEARCH REPORT**

**A. CLASSIFICATION OF SUBJECT MATTER**

IPC 7  H04L29/06  H04N7/24

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)

IPC 7  H04L

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 practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC, COMPENDEX, IBM-TDB

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>WO 99 06913 A (ARROWPOINT COMMUNICATIONS, INC.) 11 February 1999 (1999-02-11)</td>
<td>1,14,32, 36,40, 43,45, 57,62, 67,70, 73,75</td>
</tr>
<tr>
<td></td>
<td>page 2, line 24 - line 31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 10, line 3 - page 13, line 34</td>
<td></td>
</tr>
<tr>
<td></td>
<td>page 27, line 23 - page 28, line 23</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 5 568 181 A (GREENWOOD DAVID G ET AL) 22 October 1996 (1996-10-22)</td>
<td>1,14,32, 36,40, 43,45, 57,62, 67,70, 73,75</td>
</tr>
<tr>
<td></td>
<td>column 1, line 34 - line 60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>column 3, line 7 - line 48</td>
<td></td>
</tr>
</tbody>
</table>

☐ Further documents are listed in the continuation of box C.  
☒ Patent family members are listed in 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 but published on or after the international filing date
  *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  *O* document referred to in 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.

*8* document member of the same patent family

**Date of the actual completion of the international search**

15 February 2001

**Date of mailing of the international search report**

22/02/2001

**Name and mailing address of the ISA**

European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2400, Tx. 31 651 epo nl,  
Fax. (+31-70) 340-3016

**Authorized officer**

Ströbeck, A.
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AU 8373298 A</td>
<td>22-02-1999</td>
</tr>
<tr>
<td>US 5568181 A</td>
<td>22-10-1996</td>
<td>AT 184442 T</td>
<td>15-09-1999</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BR 9404912 A</td>
<td>08-08-1995</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2130395 A,C</td>
<td>10-06-1995</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1113623 A</td>
<td>20-12-1995</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 69420526 D</td>
<td>14-10-1999</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DE 69420526 T</td>
<td>27-04-2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 0658055 A</td>
<td>14-06-1995</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ES 2135554 T</td>
<td>01-11-1999</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2579445 B</td>
<td>05-02-1997</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 7200380 A</td>
<td>04-08-1995</td>
</tr>
</tbody>
</table>