US008122141B2

(12) **United States Patent**
Price

(10) **Patent No.: US 8,122,141 B2**
(45) **Date of Patent: *Feb. 21, 2012**

(54) **STREAMING MEDIA BUFFERING SYSTEM**

(75) Inventor: **Harold Edward Price**, Bethel Park, PA (US)

(73) Assignee: **WAG Acquisition, LLC**, Flanders, NJ (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **12/800,152**

(22) Filed: **May 10, 2010**

(65) **Prior Publication Data**

US 2010/0235536 A1 Sep. 16, 2010

Related U.S. Application Data

(63) Continuation of application No. 10/893,814, filed on Jul. 19, 2004, now Pat. No. 7,716,358, which is a continuation-in-part of application No. 09/819,337, filed on Mar. 28, 2001, now Pat. No. 6,766,376.

(60) Provisional application No. 60/231,997, filed on Sep. 12, 2000.

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(52) **U.S. Cl.** **709/231**

(58) **Field of Classification Search** 709/230, 709/231

See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,526,353	A	6/1996	Henley et al.	370/60.1
5,793,980	A	8/1998	Glaser et al.	395/200.61
5,822,537	A	10/1998	Katseff et al.	395/200.61
5,922,048	A	7/1999	Emura	709/219
5,923,655	A	7/1999	Veschi et al.	370/394
5,928,330	A *	7/1999	Goetz et al.	709/231
5,999,525	A	12/1999	Krishnaswamy et al.	370/352
6,002,720	A	12/1999	Yurt et al.	375/240
6,014,693	A	1/2000	Ito et al.	709/219
6,014,694	A	1/2000	Aharoni et al.	709/219
6,014,706	A	1/2000	Cannon et al.	709/231
6,029,194	A	2/2000	Tilt	709/219
6,047,317	A	4/2000	Bisdikian et al.	709/217
6,057,832	A	5/2000	Lev et al.	345/327
6,061,731	A	5/2000	Blakeslee	709/231
6,061,732	A	5/2000	Korst et al.	709/231
6,065,050	A	5/2000	DeMoney	709/219
6,085,221	A	7/2000	Graf	709/202
6,292,834	B1	9/2001	Ravi et al.	

(Continued)

OTHER PUBLICATIONS

Salehi et al., "Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing," IEEE/ACM Transactions on Networking, vol. 6, Issue 4, p. 397-410, Aug. 1998.

(Continued)

Primary Examiner — Joseph Avellino

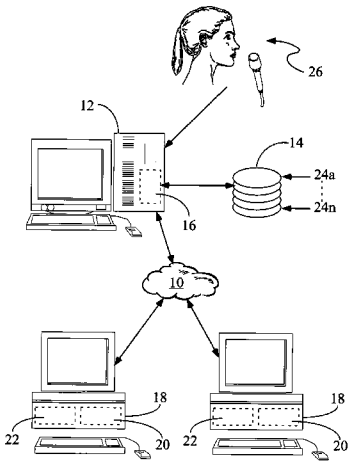
Assistant Examiner — Marshall McLeod

(74) *Attorney, Agent, or Firm* — Ernest D. Buff; Ernest D. Buff & Assoc. LLC; Gordon E. Fish

(57) **ABSTRACT**

Streaming media, such as audio or video files, is sent via the Internet. The media are immediately played on a user's computer. Audio/video data is transmitted from the server more rapidly than it is played out by the user system. The audio/video data in the user buffer accumulates; and interruptions in playback as well as temporary modem delays are avoided.

28 Claims, 3 Drawing Sheets



U.S. PATENT DOCUMENTS

6,588,015	B1	7/2003	Eyer et al.	
6,594,699	B1	7/2003	Sahai et al.	
6,598,228	B2	7/2003	Hejna	
6,637,031	B1 *	10/2003	Chou	725/87
6,665,751	B1	12/2003	Chen et al.	
6,708,213	B1	3/2004	Bommaiah et al.	
6,728,753	B1 *	4/2004	Parasnis et al.	709/203
6,829,368	B2	12/2004	Meyer et al.	
6,889,257	B1	5/2005	Patel	
6,907,481	B2	6/2005	Kovacevic	
7,054,500	B1	5/2006	Lillevold	
7,111,058	B1	9/2006	Nguyen et al.	
7,170,856	B1	1/2007	Ho et al.	
7,346,698	B2	3/2008	Hannaway	
7,373,413	B1	5/2008	Nguyen et al.	
2001/0047377	A1 *	11/2001	Sincaglia et al.	709/1

OTHER PUBLICATIONS

Rexford et al., "Smoothing variable-bit-rate video in an internetwork," IEEE/ACM Transactions on Networking, vol. 7, Issue 2, p. 202-215, Apr. 1999.

Bianchi, "Buffer sizing for high speed video information retrieval on ATM networks," Globecom—New York—, 1997, vol. 2, pp. 1057-1061.

* cited by examiner

Fig. 1

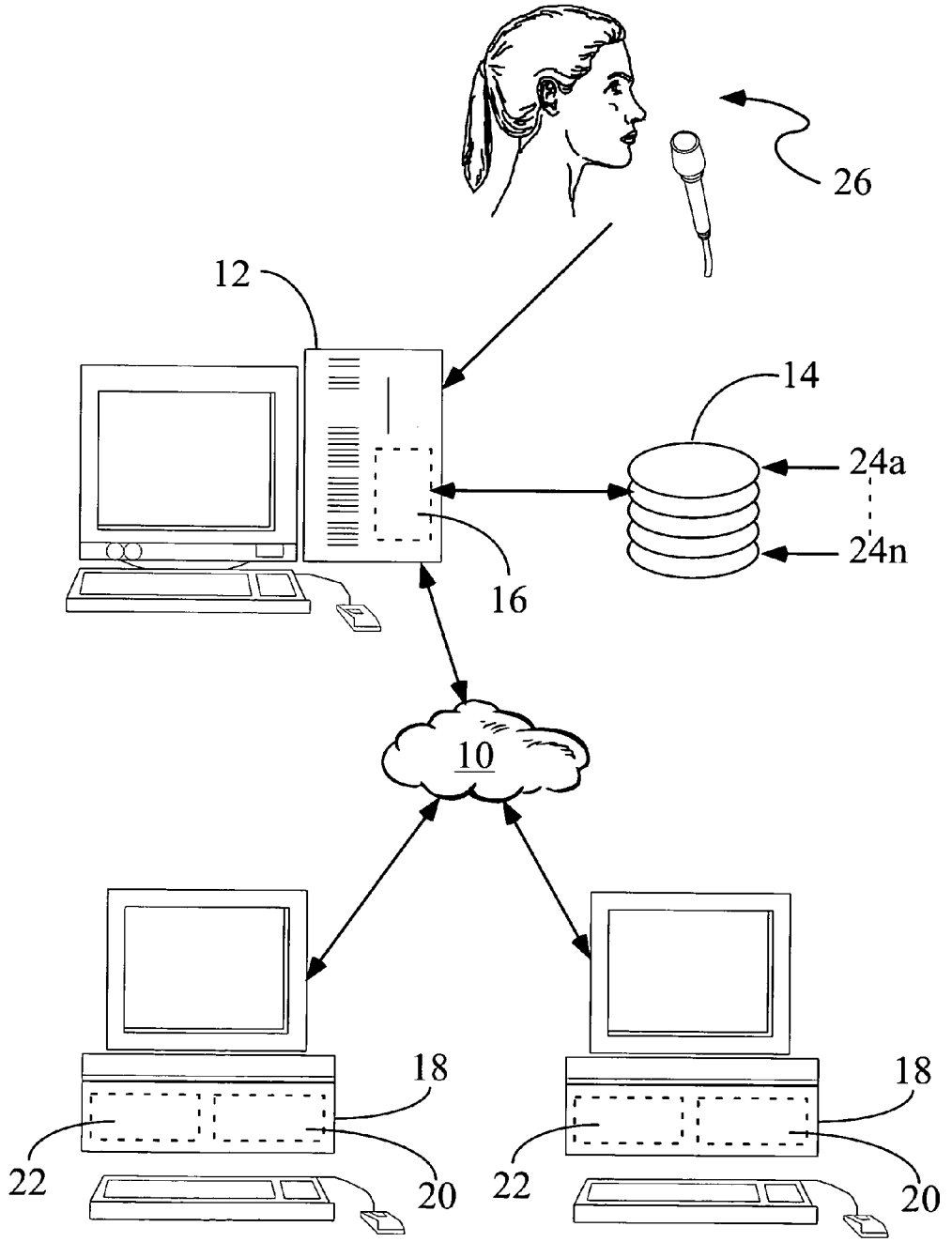


Fig. 2

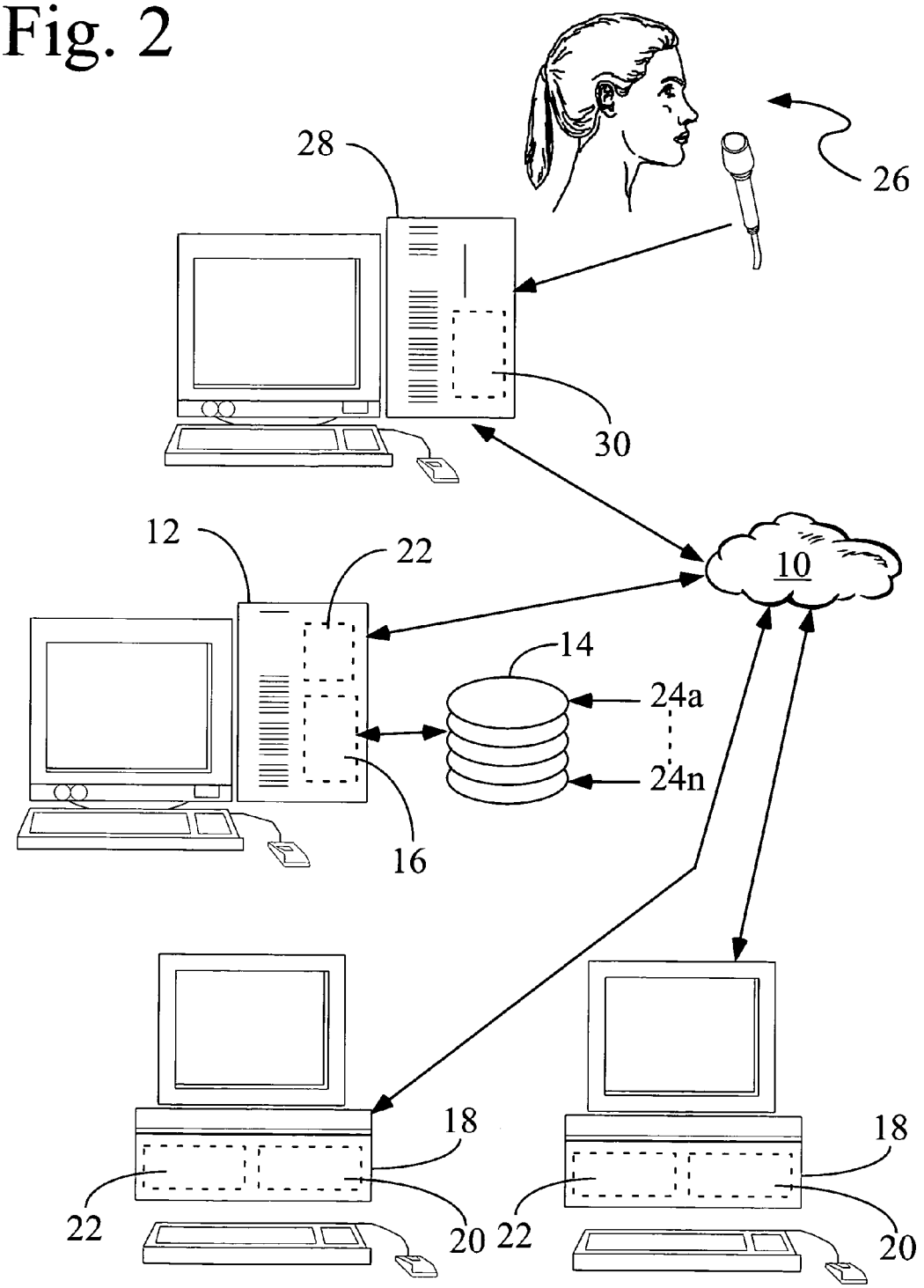
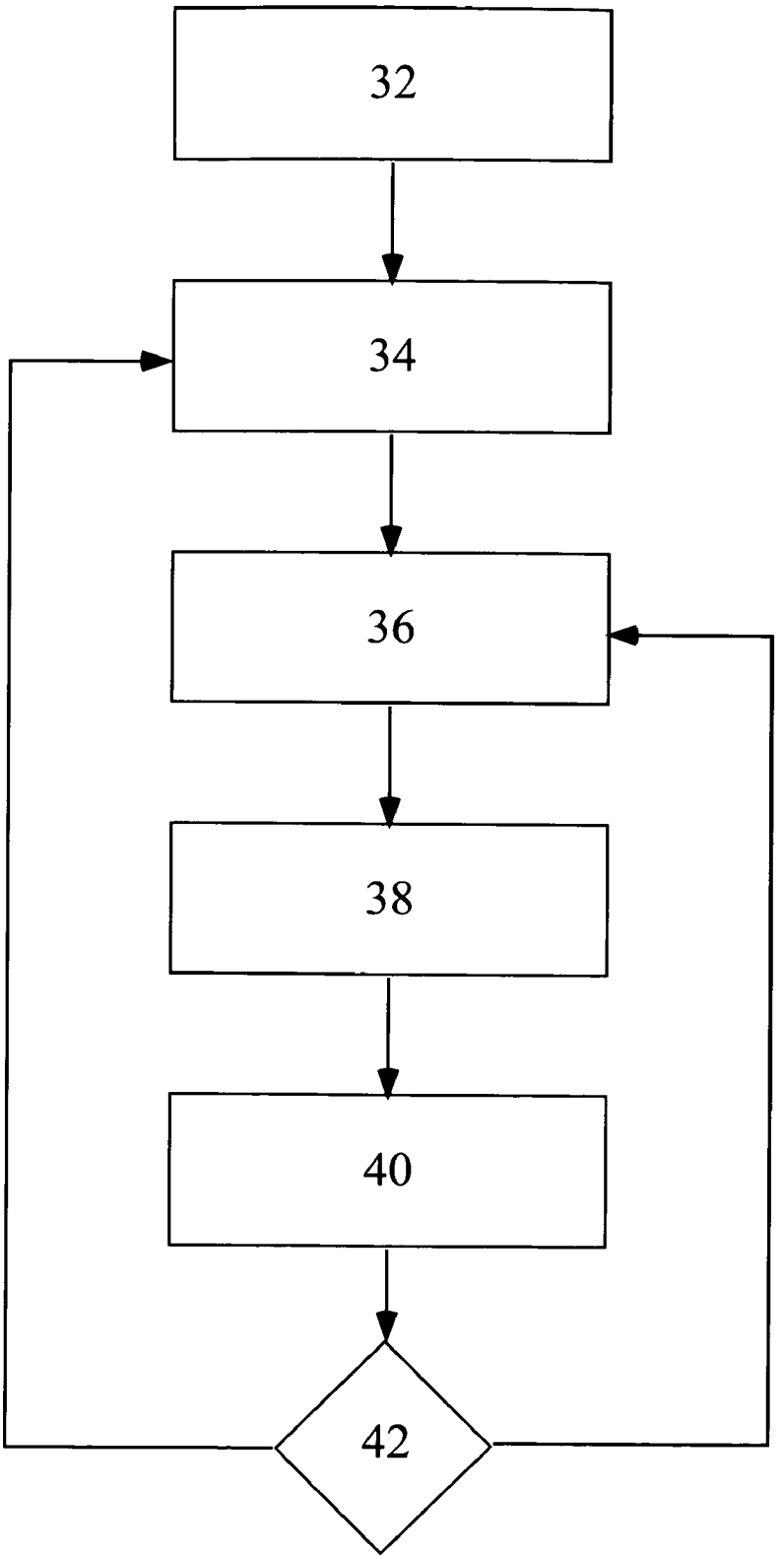


Fig. 3



STREAMING MEDIA BUFFERING SYSTEM

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is a continuation of U.S. patent application Ser. No. 10/893,814, filed Jul. 19, 2004 (published on Dec. 9, 2004 as U.S. patent publication number 2004/0249969 A1, and now U.S. Pat. No. 7,716,358), which was a continuation-in-part of U.S. patent application Ser. No. 09/819,337, filed Mar. 28, 2001 (now U.S. Pat. No. 6,766,376), which claimed the benefit under 35 U.S.C. §119(e) of U.S. provisional patent application Ser. No. 60/231,997, filed Sep. 12, 2000; claims the benefit, under 35 U.S.C. §120, of the respective filing dates of said applications, as well as benefit of the filing date of copending U.S. patent application Ser. No. 10/825,869, filed Apr. 16, 2004 (published on Dec. 23, 2004 as U.S. patent publication number 2004/260828 A1), which was a continuation of said U.S. patent application Ser. No. 09/819,337, filed Mar. 28, 2001 (now U.S. Pat. No. 6,766,376), which claimed the benefit under 35 U.S.C. §119 (e) of said U.S. provisional patent application Ser. No. 60/231,997, filed Sep. 12, 2000; and hereby incorporates by reference the entire disclosure of each of said prior applications.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to multimedia computer communication systems; and more particularly, to a buffering system for streaming media, such as audio/video, on the Internet.

2. Description of the Related Art

Prior to the development of Internet streaming media technologies, audio and video were formatted into files, which users needed to download to their computer before the files could be heard or viewed. Real time, continuous media, as from a radio station, was not suitable for this arrangement in that a file of finite size must be created so it could be downloaded. The advent of streaming media technologies allowed users to listen or view the files as they were being downloaded, and allowed users to "tune-in" to a continuous media broadcast, or "stream", such as from a radio station. There are two fundamental types of streaming media: (i) material that originates from a source having a real-time nature, such as a radio or TV broadcast, and (ii) material that originates from a non-real-time source such as from a disk file. An example of non-real-time material might be a piece of music stored as a disk file, or a portion of a broadcast that originally was real-time, perhaps yesterday's TV evening news, and was recorded into a disk file. For purposes of clarity within this document, streaming media of type (i) will be referred to as "broadcast" media, and streaming media of type (ii) will be referred to as "file based" media. Broadcast streaming media has as its source a system or arrangement that by definition can only be transmitted to users as fast as the material is generated; for example, a disk jockey speaking into a microphone. Broadcast streaming media is the focus of this patent application.

Since audio and video media must play out over a period of time it is more appropriate to think of bandwidth requirements than file size. The bandwidth requirement of an audio or video media refers to the data rate in bits per second that must be transmitted and received in order to listen or view the material uninterrupted. Transmitting the audio or video material over a connection slower than the bandwidth requirement

results in unsatisfactory viewing or listening, if viewing or listening is possible at all. The connection available to most Internet users is by dial-up modem, which has a maximum receive data rate of 56,000 bits per second. Most audio and video available on the Internet has been compressed to be listenable or viewable within the 56,000 bits per second modem bandwidth. Requirements for achieving adequate audio and video over the Internet generally consume a considerable portion of the listener's available bandwidth.

Internet connection quality can vary rapidly over time, with two primary factors responsible for degradation of the instantaneous bandwidth actually available to the user. These factors are the quality of the user's modem connection over telephone lines, which can have periods of interference causing reduced available bandwidth, and momentary Internet congestion at various points along the route over which the user's data flows. Each of these factors can cause delays and interruptions in the transmission of data to the user. Internet data communications devices such as routers are designed to drop data "packets" if they get overloaded. For material that is not time sensitive, these dropped packets will usually be resent, and the user will eventually be presented with the material. However, since streaming media is time sensitive, dropped packets can have a significant impact on the receipt and playback of an audio or video stream. These degradations in the receipt of Internet data are very common, and prevent most users from being able to listen to or view streaming media without interruption unless some special provisions have been incorporated into the user's computer software to accommodate data transmission interruptions.

These interruptions are commonly referred to as "dropouts", meaning that the data flow to the user has been interrupted (i.e., the audio "drops out"). Dropouts can be extremely annoying—for example, while listening to music. The current state-of-the-art solution to the problem uses a pre-buffering technique to store up enough audio or video data in the user's computer so that it can play the audio or video with a minimum of dropouts. This process requires the user to wait until enough of the media file is buffered in memory before listening or viewing can begin. The media data is delivered by a server computer which has available to it the source of the media data, such as by a connection to a radio station. When the user connects to the server via the Internet, audio/video output at the user's system is delayed while the user's buffer is filled to a predetermined level. Typical pre-buffering wait times range from 10 to 20 seconds or more, determined by the vendor providing the audio or video media. Even with this pre-buffering process, interruptions in playback still occur.

In this process, the user has a software application on the computer commonly called a "media player". Using the features built into the media player, the user starts the audio or video stream, typically by clicking on a "start" button, and waits 10-20 seconds or so before the material starts playing. During this time data is being received from the source and filling the media player's buffer. The audio or video data is delivered from the source at the rate it is to be played out. If, for example, the user is listening to an audio stream encoded to be played-out at 24,000 bits per second, the source sends the audio data at the rate of 24,000 bits per second. Provided that the user waits 10 seconds, and the receipt of the buffering data has not been interrupted, there is enough media data stored in the buffer to play for 10 seconds.

Gaps in the receipt of audio/video data, due to Internet slowdowns, cause the buffer to deplete. Because transmission of audio/video media data to the user takes place at the rate it is played out, the user's buffer level can never be increased or

replenished while it is playing. Thus, gaps in the receipt of audio/video media data inexorably cause the buffer level to decrease from its initial level. In time, extended or repeated occurrences of these gaps empty the user's buffer. The audio/video material stops playing, and the buffer must be refilled to its original predetermined level before playing of the media resumes.

By way of illustration in a 10 second pre-buffering scenario, if the data reception stopped the instant that the media started playing, it would play for exactly 10 seconds. Once it starts playing, the media data plays out of the buffer as new media data replenishes the buffer. The incoming data rate equals the rate at which the data is played out of the user's buffer, assuming the receipt of data across the Internet is unimpeded. If there are no interruptions in the receipt of the media data for the duration of the time the user listens to or watches the material, the buffer level remains constant and there will still be 10 seconds of data stored in the media player's buffer when the user stops the player. On the other hand, if the media player encounters interruptions totaling 6 seconds while playing the material, there would only be 4 seconds of media data remaining in the buffer when the user stopped it. If data reception interruptions at any time during the playing exceed 10 seconds, the user's media player buffer becomes exhausted. There is no media data to play, and the audio or video stops—a dropout has occurred. At this point a software mechanism in the media player stops attempting to play any more of the material, and starts the buffering process again. The media player remains silent until the buffer refills, at which time the media player will once again start playing the material.

There are two fundamental types of streaming media: (i) material that originates from a source having a real-time nature, such as a radio or TV broadcast, and (ii) material that originates from a non-real-time source such as from a disk file. An example of non-real-time material might be a piece of music stored as a disk file, or a portion of a broadcast that originally was real-time, perhaps yesterday's TV evening news, and was recorded into a disk file. For purposes of clarity within this document, streaming media of type (i) will be referred to as "broadcast" media, and streaming media of type (ii) will be referred to as "file based" media.

Both streaming media types are handled similarly in conventional systems, and both are handled similarly by the streaming media buffering system of the present invention. The two streaming media types are readily distinguished. Broadcast streaming media has as its source a system or arrangement that by definition can only be transmitted to users as fast as the material is generated; for example, a disk jockey speaking into a microphone. File based media, on the other hand, can be transmitted to users at any data rate, since there is no inherent time element to a file residing on a computer disk. With conventional Internet streaming media systems for streaming media of either type, media data is transmitted from the server to the user at the rate at which it will be played out, regardless of the data rate capabilities of the connection between the server and the user.

Conventional streaming media systems may incorporate buffering systems for programmatic purposes. For example, the system may buffer media data at the server for the purpose of packet assembly/disassembly. Media data may also be buffered at the server to permit programming conveniences such as dealing with chunks of data of a specific size. Such server buffering of media data is not used by conventional streaming media systems to mitigate long term Internet performance degradation as described hereinafter.

The sending of audio or video files via a network is known in the art. U.S. Pat. No. 6,029,194 to Tilt describes a media server for the distribution of audio/video over networks, in which retrieved media frames are transferred to a FIFO buffer. A clock rate for a local clock is adjusted according to the fullness of the buffer. The media frames from the buffer are sent in the form of data packets over the networks in response to interrupts generated by the local clock. In this manner, the timing for the media frames is controlled by the user to assure a continuous stream of video during editing. U.S. Pat. No. 6,014,706 to Cannon, et al. discloses an apparatus and method for displaying streamed digital video data on a client computer. The client computer is configured to receive the streamed digital video data from a server computer via a computer network. The streamed digital video data is transmitted from the server computer to the client computer as a stream of video frames. U.S. Pat. No. 6,002,720, to Yurt, et al. discloses a system of distributing video and/or audio information wherein digital signal processing is employed to achieve high rates of data compression. U.S. Pat. No. 5,923,655, to Veschi et al. discloses a system and method for communicating audio/video data in a packet-based computer network wherein transmission of data packets through the computer network requires variable periods of transmission time. U.S. Pat. No. 5,922,048 to Emura discloses a video server apparatus having a stream control section which determines a keyframe readout interval and a keyframe playback interval that satisfy a playback speed designated by a terminal apparatus. Finally, U.S. Pat. No. 6,014,694 to Aharoni, et al. discloses a system and method for adaptively transporting video over networks, including the Internet, wherein the available bandwidth varies with time.

There remains a need in the art for a method and system that afford immediate and uninterrupted listening/viewing of streaming media by the user.

SUMMARY OF THE INVENTION

The present invention provides a system and method for sending streaming media, such as audio or video files, via the Internet. Immediate playing of the media on a user's computer is afforded while reducing interruptions in playback due to Internet congestion and temporary modem delays due to noisy lines. Nearly instantaneous playback is achieved, while maintaining protection against playback interruption. Delayed starts, heretofore required to provide protection against interruption, are avoided. Data loss due to interruptions in the receipt of media data by the media player can be recovered while the player continues to play out the audio or video material. If the interruptions are so severe as to deplete the user's buffer and stop the play out, the media player will begin to play out again as soon as the media player begins to receive media data without waiting to first build up the buffer.

Generally stated, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. The system has a server connected to the Internet for transmitting the data elements. Associated with the server are a buffer manager and a FIFO buffer for storing at least one of the data elements for transmission. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data elements in the buffer as new data

elements are received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

This invention presumes the existence of a data communications transport mechanism, such as the TCP protocol, for the reliable delivery of data in an ordered sequence from the source of the media data to the server, or from the server to the media player software of the user computer. Thus, the delivery of data in the proper sequence is outside the scope of this invention.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

There are two types of encoding schemes used for audio and video material—"Variable Bit Rate"—VBR, and "Constant Bit Rate"—CBR. CBR encoding represents the encoded media with a constant bit rate per second, regardless of the complexity of the material being encoded, for example, if an audio source is encoded at 20 kilobits per second at a Constant Bit Rate, the media data being produced from the encoding is at 20 kilobits per second whether the audio material is symphonic music or silence. Variable Bit Rate encoding uses a variable number of bits to represent sounds, with more bits required for complex (symphonic) sounds than for simple sounds or silence. The standard encoding scheme used for streaming media is CBR because the resulting data rate is more predictable than for VBR.

The server stores a predetermined amount of media data in a First-In First-Out (FIFO) buffer in an arrangement that receives media data directly or indirectly from a real-time source, such as a radio station. For example, the server buffer might be set to store up 30 seconds of media data. Because the source produces media data in real time, the media data is delivered to the server approximately at the rate it is generated. Of course there can be variability's in this data delivery process due to networking, disk accesses, and so on, causing the delivery rate of the media data to be variable over short periods of time, typically measured in seconds. But over a longer period of time measured in minutes or tens of minutes or longer, the media data is delivered from source to server at the rate it is generated, and the server in turn provides that media data to the FIFO buffer at that same rate. Since CBR encoding is normally used for streaming media, the media data is generated, received by the server, and provided to the buffer approximately at a fixed rate. Once the buffer is full, for each new data element received into the buffer the oldest data element is deleted from the buffer. Once a connection is made to a user's computer, the server sends the media data to the user computer's buffer in the following manner. First, media data is sent to the user at the highest rate that the data connection between the server and the user computer will support until the predetermined amount of data that had been stored in the server buffer has been transferred to the user's computer. Once the buffer has been transferred a steady state condition is reached wherein as each media data element arrives at the server, it is immediately sent out to the user computer. In this steady state condition, the media data is sent at a rate that matches the constant fill rate of the server buffer, and is received at the same rate by the user computer if there are no interruptions in the transmission of media data between the server and the user's computer. If interruptions have interfered with the arrival of sent media data to the user's computer, that data may have been "dropped" by routers in the Internet and needs to be resent. This causes data to "back up" into the server FIFO for that user.

In one method of operation, the resending of missing data is the responsibility of a reliable transport mechanism, such as TCP. The server buffer "sends" data by delivering it to the transport mechanism. The transport mechanism actually "sends" the data across the communications medium, and has processes which determine if all the data that has been sent has been received by the destination. If not, missing pieces of data are automatically resent to the destination, and are arranged to be delivered to the target software on the destination system in an ordered fashion. In the circumstance of this invention, the destination is the user computer, and the target software on the destination system is the media player. If the transport mechanism determines that data is missing, it retransmits that data to the destination as fast as the connection between the server and destination will allow. The net effect of this invention is that all media data to be delivered to a user computer is always sent as fast as the communications medium will support, either by the server buffer passing media data to the transport mechanism, or by the transport mechanism delivering or redelivering the media data to the user computer. This is enabled by buffering data at the server, and is distinctly different from prior art in which media data is only sent from the server to the user computer at the rate at which it is to be played out.

In another method of operation, the server can use an unreliable transport mechanism, such as UDP, and rely on a streaming software process to manage data delivery and the resending of data elements not received by the media player.

As an example of the preceding description, if the server had been set to store 30 seconds of audio in its buffer, when a user connects that 30 seconds worth of media, data is transferred to the user's media player buffer as fast as the data connection between the two will allow. The media player can begin playing as soon as it has received a very minimum amount of data, perhaps comprising only a single packet of media data. For ease of understanding, consider the server buffer and the media player buffer to be an elastic system that between the two stores up 30 seconds of audio data. The server starts with 30 seconds of buffered audio data which it transfers to the media player until the server has no buffered media data and the media player has 30 seconds of buffered media data. Regardless of how much of the buffered media data has been transmitted to the media player, there always is 30 seconds of media data being buffered between the two locations. Consequently, the audio being played out by the media player will always be 30 seconds behind the audio at the source. If there were a media player in the radio station studio, an announcer would hear themselves through the media player with a 30 second delay.

Routinely, once a steady state has been achieved, the next data element to be sent is the next sequential data element from that which has already been received by the user's computer buffer. However, if there is more data to be sent than at the routine constant fill rate, such as in the condition where some media data has been resent by the reliable transport layer, the server transport mechanism will again send the buffered media data as fast as the connection between the server and the user's computer will support. Similarly, if the media player buffer begins to deplete or becomes depleted due to networking interruptions, the server will attempt to send as much data as is necessary to rebuild the user comput-

er's buffer to the proper level. This allows for rebuilding the user's computer buffer under circumstances wherein Internet interruptions have blocked the normal flow of data. When compared to conventional systems, which provide no capability to rebuild the user's computer buffer when data is lost, the streaming media buffering system of the present invention provides for recovery of lost data elements and the restoration of the user's buffer, even while the user media player continues to play.

Under conditions in which interruptions have interfered with the arrival of sent media data to the user's computer, data loss exceeding certain levels will cause the transport mechanism software to stop accepting data for transmission from the application software, namely the streaming media server software. Although other arrangements are possible within the scope of this invention, in the preferred embodiment, the streaming media server software keeps track of the last data element in the FIFO buffer that has been "sent" to each user using a software pointer. An interruption in the ability to send media data to a user results in this pointer "backing up" in the FIFO in such a way that the server knows from what point in the buffer to restart sending data when the transport mechanism again requests data to send. When the server software receives that notification, it will begin sending data to the user starting from the next data element to send as indicated by the pointer, and sending as much data as the transport mechanism will accept. The transport mechanism will again send this data as fast as it can to the user. This process continues until the steady state condition is again reached wherein each data element is sent to the user as soon as it arrives from the media source.

In another embodiment, the server is connected to the Internet, and to a broadcast media source, such as a radio station. A radio station computer is provided with a means for receiving media data elements as they are generated by the audio and/or video source, and for transmitting those media data elements to the server as they are generated. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the broadcast media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: supplying the FIFO buffer with a predetermined number of data elements; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Importantly, the buffer manager is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet or other data communications medium.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In another embodiment, the server is connected to the Internet and provisioned as initially described, and has available to it file based media data as the source material. The file based media data can be read by the server which can deliver media data elements to the server FIFO buffer at a constant time-sequenced rate, as if the data were arriving from a broadcast media source. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequentially arranged media data elements from the file based media source and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements at a constant time-sequenced fill rate; maintaining a pointer into the buffer for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. The server buffer manager, or a separate process on the server, or a process on another computer having access to the file based media data, provides for reading the media data file and making available to the FIFO buffer sequentially arranged media data elements. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, which are received sequentially by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are received from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer.

In another embodiment, the server is connected to the Internet and provisioned as initially described. The server buffer manager, or the media source, provides for sequentially numbering the media data elements. The server buffer manager does not maintain a pointer into the server buffer for each user. Instead, the media player buffer manager in the user computer maintains a record of the serial number of the last data element that has been received. Via the use of standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements, specifying the serial numbers of the data elements. The server responds by sending the requested data elements, and depends upon the reliable transmission protocol to assure delivery. The user computer then continues with additional data requests for the duration of playing the audio/video material. In this manner, the user computer, not the server, maintains the record of the highest data element number stored in the user computer buffer. The media data will be transmitted to the user computer as fast as the data connection between the user computer and the server will allow. As before, the server provides a buffer manager and a FIFO buffer, and provides a means for receiving the sequential media data elements from a broadcast media source or a file based media source, and storing those data elements in the FIFO buffer. The buffer manager comprises means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the FIFO buffer with a predetermined number of data elements; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received. Such means is arranged to maintain the pre-determined number of data elements in the FIFO buffer. At least one user computer is connected to the server via the Internet.

The user computer is associated with a media player software incorporating a user buffer and comprises means for receiving and storing a predetermined number of media data elements, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out. As data is played out, the next sequential data elements are requested from the server in such a fashion as to approximately maintain the predetermined number of data elements in the user's buffer. It should be understood that data might arrive at the media player out-of-sequence and that processes in the media player or the media player buffer manager are responsible for properly arranging this data.

In yet another embodiment, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. A predetermined number of the data elements are sequentially loaded into a FIFO buffer. Additional data elements continue to be received. As each new data element is input to the buffer, the oldest data element is deleted from the buffer, maintaining in the buffer the same predetermined number of data elements. At the request of a user computer for connection to a media stream, a group of the data elements is sequentially sent via the Internet from the FIFO buffer to the user computer connected to the Internet. Upon being received by the user computer, the sent group of data elements is loaded into a user's buffer associated with the user computer. The users computer immediately begins to play the audio/video streaming media material. The server continues to send the next data elements in sequence until the contents of the FIFO buffer have been sent. The data elements are sent by the server as fast as the connection between the server and user computer will allow. Once the contents of the FIFO buffer have been sent to a user computer, as each new data element is received into the FIFO buffer it is immediately sent to the user computer in such a manner as to keep the user computer buffer full. The process repeats for substantially the entire time that the audio/video material is played.

Unlike conventional buffering systems, audio begins to play on the user system as soon as the user connection to the audio server is effected and a small amount of data has been transferred-conventional systems required many seconds of data. Audio/video media data is initially transmitted from the server more rapidly than it is played out by the user system, until the server buffer has been transferred to the user computer. The user's buffer is built up while the audio is playing, and can be restored if it is diminished by data transmission interruptions. Advantageously, the system and method of this invention afford faster data transmissions than the playback data rate of the media data. Audio/video data is transmitted from the server more rapidly than it is played out by the user system under conditions wherein the user's computer buffer is not full. The audio/video data in the user buffer accumulates; interruptions in playback due to temporary Internet and modem delays are avoided. It should be realized that, although the invention has been described hereinabove in connection with a process wherein the server sends buffered media data to the user "as fast as the network connection will permit", it is adequate, as mentioned in this paragraph, that the buffered data be transferred from the server to the user at a rate faster than the playback rate.

Although the preferred embodiment utilizes a reliable transport mechanism to move data between the server and the user, alternative embodiments could incorporate this invention's buffering system in combination with an unreliable datagram-based transport mechanism.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be more fully understood and further advantages will become apparent when reference is had to the following detailed description and the accompanying drawings, in which:

FIG. **1** is a schematic diagram illustrating the elements of a streaming media buffering system in accordance with the present invention;

FIG. **2** is a schematic diagram of an alternative embodiment of the system shown by FIG. **1**; and

FIG. **3** is a block diagram illustrating the method of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention relates to a buffering system for streaming media, such as audio/video, on the Internet.

In one embodiment, the invention provides a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements. As shown in FIG. **1**, the system is provided with a server **12** connected to the Internet **10** for transmitting the streaming media data elements. Associated with the server **12** is a FIFO buffer **14** for storing at least one of the data elements for transmission, and a buffer manager **16**. Buffer **14** is a conventional computer storage mechanism such as a hard disk, as shown for convenience of illustration, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM). The buffer manager **16** is in the form of software or firmware that provides means for: receiving the media data; supplying media data in order to the FIFO buffer; supplying the buffer **14** with a predetermined number of data elements; maintaining a pointer **24***a* through **24***n* into the buffer, one for each user computer indicating the last media data element that has been sent to that user, thus indicating the next element or elements to be sent; and, once the FIFO buffer is full, deleting the oldest data element in the buffer as each new data element is received, said means arranged to maintain the pre-determined number of data elements in the FIFO buffer. Buffer Manager **16** may also comprise means for: formatting media data according to the requirements of buffer **14**, and for digitizing, encoding, and packetizing the media data.

There is at least one user computer **18** connected to the server **12** via the Internet **10**. A user buffer **20** is associated with the user computer **18**. The user buffer **20** is provided with means for storing a predetermined number of the data elements. User buffer **20** is a conventional computer storage mechanism such as a hard disk, or, preferably, an electronic storage arrangement such as Random Access Memory (RAM) as suggested by the illustration. A buffer manager **22** is associated with the user computer **18**. The buffer manager **22**, having the form of software or firmware, is provided with means for receiving and storing a predetermined number of media data elements which are received by the media player, playing the data out sequentially as audio and/or video, and deleting media data elements from the buffer as they are played out.

The media may come from a live source, shown as **26** in FIG. **1**, or from a stored file on the server **12**, or another storage device, such as a hard drive. For broadcast media, as the term is used herein, such as an announcer speaking into a microphone, or playing a CD, the media source **26** can only transmit audio/video data as fast as it is generated. If the media source is file based, such as a music clip stored as a disk file, and if that disk file is stored on the server or an associated

server computer, this connection could be considered to be near instantaneous. In this case, rather than audio/video data filling and depleting the buffer **14**, an amount of audio/video data equivalent to the desired buffer size is logically constituted as a FIFO buffer. Such a construct is commonly called a data window. The data window moves on a time-sequenced basis through the media data file, thus defining the contents of the buffer on a moment-by-moment basis and performing the equivalent functions to receiving a new data element and deleting the oldest data element.

The server **12** maintains a buffer of audio/video data comprising an amount adequate to bridge gaps typical of Internet and modem delays to the user. Preferably, this buffer holds enough data elements for about one minute of play. The server buffer **14** is filled the first time the media source connection is established or a disk file is read.

Connections from the server **12** through the Internet **10** commonly are much faster than the data rate required for audio or video playback. This fact is insignificant for conventional servers because, not having a FIFO buffer or a buffer pointer for each user, audio/video data can only be sent as fast as it becomes available, or as fast as the pace at which it must be delivered to the user in order to be properly replayed. The user, typically interacting with "media player" software on their computer, selects an audio source requiring a data rate slower than that available by the user's connection to the Internet. For example, if the user's connection to the Internet is made via a 56,000 bits per second modem, the user might select a media source encoded for playback at 24,000 bits per second.

With the present invention, as soon as a user connects to the server **12**, the server **12** transmits audio/video data as sequential data elements from its buffer **14** to the buffer **20** of the user as fast as the data connection will allow. Unlike the prior art, media begins to play on the user computer **18** as soon as the user connection is made to the audio server **12** and a minimal amount of data elements have been received and stored in the user's buffer **20**. The user's buffer **20** is built up while the media is playing. As each data element is played, it is deleted from the user's buffer **20**. Initially, the user buffer manager **22** requests the server **12** to send media data elements to start the playback stream, such as by selecting a radio station from a list. The server **12** responds by sending data elements to the user computer **18** as fast as it can, until the entire FIFO buffer **14** has been sent to the user computer. Upon receipt of the initial data elements, the user buffer manager **22** begins playback. Because this is a synchronous system with the source, server, and user computer operating by the same playback clock rate as determined by the encoding rate of the media, as each data element is played out and is deleted from the user buffer **20**, another data element has been deposited into the server buffer **14** and is available to be sent to the user computer. Server **12** sends the newly available data elements as fast as the data rate of the connection between server **12** and user computer **18** will allow.

Since the connection from the Internet to the user is faster than that required for media playback, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer. For example, if the user's connection to the Internet is at 56,000 bits per second, and the data rate encoded for the media to be played is 24,000 bits per second, the buffer level of the user buffer **20** will fill at the rate of 32,000 bits per second (56,000 bits per second receive rate, minus 24,000 bits per second playout depletion rate).

If, for example, the server buffer **14** held one minute of audio/video data, eventually the user buffer **20** will hold one minute of audio/video data. The effect is that, over a brief period of time, the server buffer **14**, or a designated portion of it, is transferred to the user buffer **20**. The number of data elements in the server buffer **14** actually never changes, it always maintains one minute of audio/video data. However, for the particular user under discussion, a copy of all the data held in the buffer has been sent to the user. Since the user buffer **20** now holds one minute of audio/video data, it can play continuously despite data reception interruptions of less than a minute, and as soon as the interruption ceases the user buffer **20** can begin to rebuild, which will take place as fast as the connection between the user computer **18** and the server **12** will allow. The media player can continue to play out the audio/video material while the user buffer **20** rebuilds.

The predetermined buffer level in the user buffer **20** may be set at less than the predetermined buffer level of the server buffer **14** if desired. For example, the server buffer **14** might be set to hold one minute of media data, and the user buffer **20** might be set to hold thirty seconds of media data.

Alternatively, the user computer is replaced by an Internet radio or Internet Appliance, which is comprised of a dedicated processor for receiving Internet radio or audio/video material. Examples of such devices might range from familiar computing devices such as palmtops, PDAs (Portable Digital Assistants), and wireless phones, to devices that appear and operate similarly to conventional consumer electronic devices such as radios and televisions, but with the additional capability of Internet access.

In another embodiment, as shown in FIG. **2**, the media source may be remote from the server **12**, such as a computer system **28** in a radio station studio. This computer includes a source manager **30** which may be implemented in software or firmware. The source manager **30** comprises means for: formatting media data according to the requirements of server **12**, buffer **14**, and buffer manager **16**; and, for transmitting that media data to server **12**. Source manager **30** may further include means for digitizing, encoding, and packetizing the media data. Media data typically is generated in real time such as by a speaker talking into a microphone or by playing a CD. Generally, computer system **28** transmits media data to server **12** in real time as the media data is generated. Buffering of media data might occur at computer system **28** for convenience of programming, but such buffering is incidental to the operation of the end-to-end system being described. Computer system **28** connects via the Internet **10**, or other suitable data communications medium, to a server **12**, wherein server buffer manager **16** receives the media data for input into the FIFO buffer **14** as described previously. Server **12**, in turn, transmits the media data to one or more user computers **18**, also as previously described. Network connections between the source computer and the server may cause media data to be delayed in arrival at the server, causing the server FIFO buffer occasionally to be less than full. In this circumstance, the server buffer transfers the media data that it has available in its buffer to the media player, and when more data arrives from the source, the server sends it out to the media player.

In another embodiment, shown in FIG. **3**, the invention provides a method for distributing from a server via the Internet streaming media composed of a plurality of time-sequenced data elements. Time-sequenced data elements are generated or received **32**. Next, a predetermined number of the data elements are sequentially loaded **34** into a server buffer, which process of **32** and **34** continues indefinitely as long as there is media data available. Next, a group of the data elements is sequentially sent **36** via the Internet from the server buffer to a user computer connected to the Internet. Upon receipt by the user computer, the sent group of data

elements is loaded **38** into a user buffer associated with the user computer. The user computer immediately plays **40** the received portion of the media on the user computer. At **42**, if the user buffer is not full, then additional data elements are sent to the user computer **36**. And also at **42**, if the user buffer is full, the system waits until new media data is delivered to the server buffer **34**. This process is repeated until the entire media file is played at the user computer.

Unlike conventional buffer arrangements, audio begins to play on the user system as soon as the user connection is made to the audio server. The user's buffer is built up while the audio is playing. Advantageously, the system and method of this invention create a faster than real time connection. That is to say, audio/video data is transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer.

Having thus described the invention in rather full detail, it will be understood that such detail need not be strictly adhered to, but that additional changes and modifications may suggest themselves to one skilled in the art, all falling within the scope of the invention as defined by the subjoined claims.

I claim:

1. A method for distributing streaming media via a data communications medium such as the Internet to at least one user system of at least one user, the streaming media comprising a plurality of sequential media data elements for a digitally encoded audio or video program, comprising

providing a server programmed to receive requests from the user system for media data elements corresponding to specified serial identifiers and to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user; and

providing a machine-readable medium accessible to said user, on which there has been recorded software for implementing a media player for receiving and playing the streaming media on said user system, said software being programmed to cause the media player to maintain a record of the identifier of the last data element that has been received; and to transmit requests to the server to send one or more data elements, specifying the identifiers of the data elements, as said media player requires in order to maintain a sufficient number of media data elements in the media player for uninterrupted playback.

2. The method of claim **1**, wherein said serial identifiers are sequential.

3. The method of claim **1**, wherein said media is encoded at a constant bit rate.

4. The method of claim **1**, wherein said media is encoded at a variable bit rate.

5. The method of claim **1**, further comprising distributing said streaming media to a plurality of users.

6. The method of claim **5**, wherein said server does not maintain a pointer into a buffer established within said server, for each said user.

7. The method of claim **1**, wherein said server is adapted to receive said requests from said user system via standard communications protocol techniques such as TCP.

8. The method of claim **1**, wherein said server obtains said streaming media from a live source.

9. The method of claim **1**, wherein said server obtains said streaming media from a disk file.

10. A server for distributing streaming media via a data communications medium such as the Internet to at least one user system of at least one user, the streaming media comprising a plurality of sequential media data elements for a digitally encoded audio or video program, said user system being assumed to have a media player for receiving and playing the streaming media on said user system, which is operable to obtain media data elements from said server by transmitting requests to said server to send one or more specified media data elements, said server comprising

at least one data storage device, memory for storing machine-readable executable routines and for providing a working memory area for routines executing on the server, a central processing unit for executing the machine-readable executable routines, an operating system, at least one connection to the communications medium, and a communications system providing a set of communications protocols for communicating through said at least one connection;

a machine-readable, executable routine containing instructions to cause the server to assign serial identifiers to the sequential media data elements comprising the program;

a machine-readable, executable routine containing instructions to cause the server to receive requests from the user system for one or more media data elements specifying the identifiers of the requested data elements; and

a machine-readable, executable routine containing instructions to cause the server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user.

11. The server of claim **10**, wherein said serial identifiers are sequential.

12. The server of claim **10**, wherein said media is encoded at a constant bit rate.

13. The server of claim **10**, wherein said media is encoded at a variable bit rate.

14. The server of claim **10**, wherein said server is adapted to distribute said streaming media to a plurality of simultaneous users.

15. The server of claim **10**, wherein said server does not maintain a pointer into a buffer established within said server, for each said user.

16. The server of claim **10**, wherein said operating system further comprises a protocol to receive said requests from said user system via standard communications protocol techniques such as TCP.

17. The server of claim **10**, wherein said server is adapted to obtain said streaming media from a live source.

18. The server of claim **10**, wherein said server obtains said streaming media from a disk file.

19. A non-transitory machine-readable medium on which there has been recorded a computer program for use in operating a computer to prepare streaming media content for transmission by a server wherein said server responds to user requests for media data elements identified by a serial identifier, said program recorded on said non-transitory machine readable medium comprising a routine to store and serially identify sequential data elements comprising said streaming media content, in a format capable of being served to users by said server.

20. The non-transitory machine readable medium of claim **19**, wherein said streaming media content is obtained by said computer from a disk file.

21. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is obtained from a live source.

22. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is encoded at a constant bit rate.

23. The non-transitory machine-readable medium of claim **19**, wherein said streaming media content is encoded at a variable bit rate.

24. A non-transitory machine-readable medium on which there has been recorded a computer program for use in operating a media player for receiving and playing streaming media comprising a plurality of sequential media data elements, said elements being available on request by said player via a data communications medium such as the Internet, from a server assumed to be capable of sending streaming media elements at a rate more rapid than the rate at which said streaming media is played back by a user, each said data element having a serial identifier, said program recorded on said machine readable medium comprising:

a routine that maintains a record of the identifier of the last sequential media data element that has been received by said player;

a routine that requests transmission of the next sequential media data elements following said last sequential media data element, as said media player requires in order to maintain a sufficient number of media data elements in the media player for uninterrupted playback.

25. The non-transitory machine-readable medium of claim **24**, wherein said streaming media content is encoded at a constant bit rate.

26. The non-transitory machine-readable medium of claim **24**, wherein said streaming media content is encoded at a variable bit rate.

27. The non-transitory machine-readable medium of claim **24**, further comprising routines to send said requests to said server via standard communications protocol techniques such as TCP.

28. The method of claim **1**, wherein said server assigns serial identifiers to the sequential media data elements comprising the program.

* * * * *