

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
24 November 2005 (24.11.2005)

PCT

(10) International Publication Number
WO 2005/111837 A1

(51) International Patent Classification⁷: **G06F 15/16**

(21) International Application Number:
PCT/US2004/023936

(22) International Filing Date: 23 July 2004 (23.07.2004)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/838,465 3 May 2004 (03.05.2004) US

(71) Applicant (for all designated States except US): **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **KLEMETS, Anders, E.** [SE/US]; 10834 Redmond-Woodinville Road NE, Redmond, WA 98052 (US). **OLIVEIRA, Eduardo, P.** [BR/US]; 10239 186th Court NE, Redmond, WA 98052 (US). **BHATT, Sanjay** [IN/US]; 9402 225th Way NE, Redmond, WA 98053 (US). **WYNN, Tong, L.** [US/US]; 10828 180th Ct. NE, Redmond, WA 98052 (US).

(74) Agents: **LEE, Lewis, C.** et al.; Lee & Hayes, PLLC, 421 W. Riverside Avenue, Suite 500, Spokane, WA 99201 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

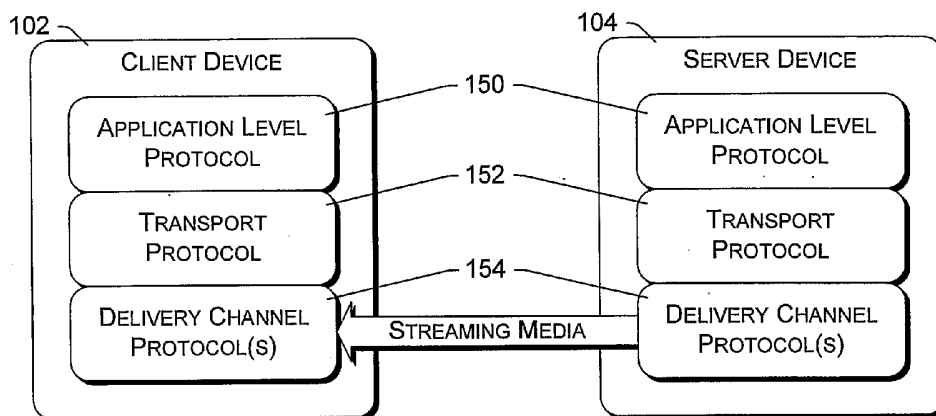
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

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

[Continued on next page]

(54) Title: FAST STARTUP FOR STREAMING MEDIA



(57) Abstract: Fast startup for streaming media includes one or both of a technique for predictive start and a technique for switching delivery channel(s) during streaming. Predictive start allows a media content source to predict which media stream(s) are desired by the requesting client device and begin streaming those predicted media stream(s) to the client device. The delivery channel switching technique allows the media content source to begin streaming the media stream(s) using one delivery channel (e.g., using the Transmission Control Protocol (TCP)), and then check whether another delivery channel (e.g., using the User Datagram Protocol (UDP)) will work and switch to that other delivery channel if it will work.



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.

FAST STARTUP FOR STREAMING MEDIA

TECHNICAL FIELD

This invention relates to streaming media and data transfers, and particularly to fast startup for streaming media.

BACKGROUND OF THE INVENTION

Content streaming, such as the streaming of audio, video, and/or text is becoming increasingly popular. The term “streaming” is typically used to indicate that the data representing the media is provided over a network to a client computer on an as-needed basis rather than being pre-delivered in its entirety before playback. Thus, the client computer renders streaming content as it is received from a network server, rather than waiting for an entire “file” to be delivered.

The widespread availability of streaming multimedia content enables a variety of informational content that was not previously available over the Internet or other computer networks. Live content is one significant example of such content. Using streaming multimedia, audio, video, or audio/visual coverage of noteworthy events can be broadcast over the Internet as the events unfold. Similarly, television and radio stations can transmit their live content over the Internet.

Current streaming technology, however, has its drawbacks. One such drawback is that there is typically a somewhat lengthy delay between the time that a playback of streaming media is requested and the time when the playback of the streaming media begins. Various communications are typically required between the client device where the playback of the media content will occur and the source

of the media content. From the perspective of the user of the client device, such delays reduce the user-friendliness of the client device as well as the user-friendliness of the streaming process. Thus, it would be beneficial to reduce the delay between the time the user requests playback of streaming media and the time when the playback of the streaming media begins.

SUMMARY OF THE INVENTION

Fast startup for streaming media is described herein. The fast startup for streaming media includes one or both of a technique for predictive start and a technique for switching delivery channel(s) during streaming.

In accordance with certain aspects, the technique for predictive start comprises a client device sending a predictive play request to a server device. In response to the predictive play request, the server device predicts one or more media streams desired by the client device and streams the predicted one or more media streams to the client device. The client device can select at least one media stream different from the predicted one or more media streams for streaming to the client device. If at least one such media stream is selected, then an indication of the selected at least one media stream is sent to the server device, and the server device streams the selected at least one media stream to the client device.

In accordance with certain aspects, the technique for switching delivery channel(s) during streaming includes a client device sending a request to a server device that one or more media streams be streamed to the client device. The server device receives the request and begins streaming the one or more media streams to the client device using TCP (Transmission Control Protocol). While streaming the one or more media streams from the server device to the client device using TCP, probing is performed to determine whether data can be streamed from the server

device to the client device using UDP (User Datagram Protocol). If the probing indicates that data can be streamed from the server device to the client device using UDP, then the streaming of the one or more media streams is switched from using TCP to using UDP. However, if the probing indicates that data cannot be streamed from the server device to the client device using UDP, then the streaming of the one or more media streams continues using TCP.

BRIEF DESCRIPTION OF THE DRAWINGS

The same numbers are used throughout the document to reference like components and/or features.

Fig. 1 illustrates an example network environment that supports the fast startup for streaming media described herein.

Fig. 2 illustrates example client and server devices that can stream media content using the fast startup for streaming media described herein.

Fig. 3 is a block diagram illustrating example client and server devices and the predictive start technique in additional detail.

Fig. 4 is a flowchart illustrating an example predictive start process.

Fig. 5 is a block diagram illustrating example client and server devices and the delivery channel switching process in additional detail.

Fig. 6 is a flowchart illustrating an example delivery channel switching process.

Fig. 7 is a flowchart illustrating an example fast startup process including both predictive start and delivery channel switching.

Fig. 8 illustrates a general computer environment, which can be used to implement the techniques described herein.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Fast startup for streaming media is described herein. The fast startup includes one or both of a technique for predictive start and a technique for switching delivery channel(s) during streaming. Predictive start allows a media content source to predict which media stream(s) are desired by the requesting client device and begin streaming those predicted media stream(s) to the client device. The delivery channel switching technique allows the media content source to begin streaming the media stream(s) using one delivery channel (e.g., using the Transmission Control Protocol (TCP)), and then check whether another delivery channel (e.g., using the User Datagram Protocol (UDP)) will work and switch to that other delivery channel if it will work.

Fig. 1 illustrates an example network environment 100 that supports the fast startup for streaming media described herein. In environment 100, multiple (*a*) client computing devices 102(1), 102(2), . . . , 102(*a*) are coupled to multiple (*b*) server computing devices 104(1), 104(2), . . . , 104(*b*) via a network 106. Network 106 is intended to represent any of a variety of conventional network topologies and types (including wired and/or wireless networks), employing any of a variety of conventional network protocols (including public and/or proprietary protocols). Network 106 may include, for example, the Internet as well as possibly at least portions of one or more local area networks (LANs).

Computing devices 102 and 104 can each be any of a variety of conventional computing devices, including desktop PCs, workstations, mainframe computers, Internet appliances, gaming consoles, handheld PCs, cellular telephones, personal digital assistants (PDAs), etc. One or more of devices 102 and 104 can be the same types of devices, or alternatively different types of devices.

Devices 104 can make any of a variety of data available for streaming to clients 102. The term “streaming” is used to indicate that the data representing the media is provided over a network to a client device and that playback of the content can begin prior to the content being delivered in its entirety (e.g., providing the data on an as-needed basis rather than pre-delivering the data in its entirety before playback). The data may be publicly available or alternatively restricted (e.g., restricted to only certain users, available only if the appropriate fee is paid, etc.). The data may be any of a variety of one or more types of content, such as audio, video, text, animation, etc. Additionally, the data may be pre-recorded or alternatively “live” (e.g., a digital representation of a concert being captured as the concert is performed and made available for streaming shortly after capture).

A client device 102 may receive streaming media from a server 104 that stores the streaming media content as a file, or alternatively from a server 104 that receives the streaming media from some other source. For example, server 104 may receive the streaming media from another server that stores the streaming media content as a file, or may receive the streaming media from some other source (e.g., an encoder that is encoding a “live” event).

As used herein, streaming media refers to streaming one or more media streams from one device to another (e.g., from a server device 104 to a client device 102). The media streams can include any of a variety of types of content, such as one or more of audio, video, text, and so forth.

Fig. 2 illustrates example client and server devices that can stream media content using the fast startup for streaming media described herein. Multiple different protocols are typically followed at both client device 102 and server device 104 in order to stream media content from server device 104 to client device 102. These different protocols can be responsible for different aspects of the

streaming process. Although not shown in Fig. 2, one or more additional devices (e.g., firewalls, routers, gateways, bridges, etc.) may be situated between client device 102 and server device 104.

In the example of Fig. 2, an application level protocol 150, a transport protocol 152, and one or more delivery channel protocols 154 are used as part of the streaming process. Additional protocols not shown in Fig. 2 may also be employed (e.g., there may be an additional protocol(s) between application level protocol 150 and transport protocol 152). Application level protocol 150 is a protocol at the application level for control of the delivery of data with real-time properties. Application level protocol 150 provides a framework, optionally extensible, to enable controlled, on-demand delivery of real-time data, such as streaming audio and video content. Application level protocol 150 is a control protocol for initiating and directing delivery of streaming multimedia from media servers. Examples of application level protocol 150 include the Real-Time Streaming Protocol (RTSP) as described in Network Working Group Request for Comments (RFC) 2326, April 1998, and the HyperText Transport Protocol (HTTP) as described in Network Working Group Request for Comments (RFC) 1945, May 1996 or Network Working Group Request for Comments (RFC) 2068, January 1997.

Application level protocol 150 uses transport protocol 152 for the delivery of real-time data, such as streaming audio and video. Transport protocol 152 defines a packet format for media streams. Transport protocol 152 provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio, video or simulation data, over multicast or unicast network services. Examples of transport protocol 152 include the Realtime Transport Protocol (RTP) and the Realtime Control Protocol (RTCP) as described in Network Working Group Request for Comments (RFC) 3550, July 2003. RTP does not address resource

reservation and does not guarantee quality-of-service for real-time services. The data transport is augmented by a control protocol (RTCP) to allow monitoring of the data delivery in a manner scalable to large multicast networks, and to provide some control and identification functionality.

The transport protocol 152 uses delivery channel protocol(s) 154 for the transport connections. Delivery channel protocol(s) 154 include one or more channels for transporting packets of data from server device 104 to client device 102. Each channel is typically used to send data packets for a single media stream, although in alternate embodiments a single channel may be used to send data packets for multiple media streams. Examples of delivery channel protocols 154 include Transmission Control Protocol (TCP) packets and User Datagram Protocol (UDP) packets. TCP ensures the delivery of data packets, whereas UDP does not ensure the delivery of data packets. Typically, delivery of data packets using TCP is more reliable, but also more time-consuming, than delivery of data packets using UDP.

Fig. 3 is a block diagram illustrating example client and server devices and the predictive start technique in additional detail. In certain embodiments, the protocols 150, 152, and 154 of Fig. 2 are included in the client and server devices of Fig. 3, but have not been illustrated. Furthermore, although not shown in Fig. 3, one or more additional devices (e.g., firewalls, routers, gateways, bridges, etc.) may be situated between client device 102 and server device 104.

Client device 102 initiates the predictive start process by sending a predictive play request 182 to server device 104. Predictive play request 182 can include various information to assist server device 104 in predicting which media streams client device 102 desires. This information can include, for example, information about client device 102, information about a user(s) of client device

102, and/or information about the connection between server device 104 and client device 102. Server device 104 receives the predictive play request and predicts which media stream(s) client device 102 desires. This prediction is typically based at least in part on the information included in the predictive play request.

Server device 104 then begins to stream 184 the media stream(s) it predicted that client device 102 desires to client device 102. Thus, in the illustrated example of Fig. 3, streaming of the media content from server device 104 to client device 102 begins in response to the first request from client device 102 received at server device 104. As part of the streaming of the predicted stream(s) 184, server device 104 includes a description of the various media streams that are available to client device 102. These various media streams will typically include the streams predicted by server device 104 as well as one or more additional streams.

Client device 102 receives the streaming media from server device 104 and processes the streaming media. This processing can include, for example, buffering of the streaming media for presentation to a user at client device 102. Additionally, client device 102 checks the description, received from server device 104, of the various media streams that are available to client device 102, and determines whether the predicted streams currently being streamed to client device 102 are indeed the streams desired by client device 102. If they are the desired streams, then no further action need be taken by client device 102. However, if they are not the desired streams, then client device 102 sends a stream selection 186 to server device 104. Stream selection 186 identifies the streams desired by client device 102.

Server device 104 receives stream selection 186 and streams the selected stream(s) 188 to client device 102. Stream selection 186 may identify one or more streams, and may identify a stream that is to replace a predicted stream or that is to

be streamed in addition to a predicted stream. If an identified stream is to replace a predicted stream, then streaming of that predicted stream is stopped by server device 104. However, if an identified stream is to be in addition to a predicted stream, then streaming of that predicted stream is not stopped by server device 104.

Example embodiments of the predictive play request 182, response with predicted data stream(s) 184, stream selection 186, and response with selected data stream(s) 188 are discussed in additional detail in the following paragraphs.

Predictive play request 182 includes a resource locator, such as a uniform resource locator (URL). This resource locator identifies the media content that is desired by client device 102. However, there may be various options for that media content. For example, the media content may have multiple video streams encoded at different bit rates (for client devices having different connection speeds), multiple video streams for different languages, multiple audio streams for different languages, multiple sub-title media streams in different languages, and so forth. However, because client device 102 does not yet know what those various options are, client device 102 cannot yet select particular ones of the different available options for the media content.

Predictive play request 182 can also include one or more name-value pairs that include information used by server device 104 in predicting which media stream(s) are desired by client device 102. These name-value pairs include a name portion that indicates the type of information being identified, and a value portion that indicates the particular information being identified. Alternatively, the information may be identified in manners other than name-value pairs, such as a series of values that appear in a pre-defined order (thereby alleviating the need for the name portion). Additionally, in some situations predictive play request 182 may not include any such information, in name-value pairs or otherwise.

The information included in play request 182 can vary. In certain embodiments, play request 182 includes an identification of the connection speed or link bandwidth between client device 102 and server device 104, and also an identification of a language desired by client device 102. In other embodiments, other information may be included in addition to or in place of the link bandwidth and/or language, such as what types of media streams are desired (e.g., audio only, video only, video and audio, etc.), what codecs (e.g., audio codecs, video codecs, etc.) are available on client device 102, what language the video data should be in (e.g., so people's lips on the video are synchronized with the audio), and so forth.

Client device 102 can obtain the information to be included in predictive play request 182 in different manners. Typically, the resource locator is obtained by user-input (e.g., user-selection of particular media content) or by program-input (e.g., a program running on client device 102 selects particular media content). The information that is used by server device 104 in making its prediction may be pre-programmed (e.g., by a developer or designer, or by user-setting of preferences) into the application that is sending predictive play request 182. For example, if the application is a media player application, a user of the media player application may be able to set a preference or option of the media player application to identify the language that the user prefers. Information about connection speed or link bandwidth may be obtained, for example, by an application on client device 102 testing the bandwidth, based on a user-input of the available bandwidth, by monitoring the rate at which media was previously streamed to client device 102, and so forth. The information that is used by server device 104 in making its prediction may alternatively be pre-programmed into server device 104 as well (e.g., in situations where all clients are known or expected to be of the same type).

Predictive play request 182 can take a variety of different forms. In certain embodiments, predictive play request 182 is an HTTP get command. Alternatively, other formats can be used (e.g., different syntaxes, an HTTP post command, an RTSP command, and so forth). The predictive play request includes an identifier that identifies the request as a predictive play request. In certain embodiments, one or more headers are used to identify a request as a predictive play request, although the information identifying the request as a predictive play request could alternatively be included in the message body. Additionally, in certain implementations, the presence of one or more such headers in the request identifies the request as a predictive play request. An example of a predictive start header that can be used to identify a request as a predictive play request is "Pragma: PredStart=1", with the value "1" being used as a parameter to indicate that predictive play is being requested.

When server device 104 receives predictive play request 182, server device 104 identifies the media streams associated with request 182. Request 182 identifies a resource that may be one or more files that each include one or more media streams that can be streamed to client device 102. Associated with the resource is an identification of these media stream(s), and typically information describing what these media streams are (e.g., a bit rate that they are encoded in; whether they are audio, video, text, etc.; what language they are in; etc.). This identification can be part of the resource (e.g., it may be in a header file of the resource, such as an ASF (Advanced Streaming Format or Advanced Systems Format) header in an ASF file), or may alternatively be maintained separately.

Server device 104 uses the information included in predictive play request 182, as well as the information regarding what media stream(s) are available for the requested resource, to predict which media stream(s) are desired by client device

102. Server device 104 may also employ additional information, such as default values that are supplied by an administrator or user of server device 104, media stream(s) previously streamed to other client devices 102 (and/or to this client device 104), and so forth. The exact manner in which the stream(s) desired by client device 102 are predicted can vary based on the information available to server device 104.

By way of example, assume that client device 102 sends a predictive play request for a movie to be streamed to client device 102. Client device 102 does not know what media stream(s) are available for that movie, such as what different bit rates the video content has been encoded in, what languages the audio content is available in, what languages subtitles are available in, and so forth. However, assume that client device 102 does know that its default language is English and that it has a connection speed of 500 kilobits per second (kbps). So, client device 102 sends a predictive play request for the movie to server device 104, and includes in the predictive play request an indication that its default language is English and that its default connection speed is 500 kbps.

Server device 104 receives the predictive play request and accesses the information associated with the requested movie that describes the requested movie. Assume that the movie has three video streams, one encoded at 100 kbps, another at 300 kbps, and another at 500 kbps. Further assume that the movie has an audio stream in English encoded at 100 kbps, and an audio stream encoded in Spanish at 100 kbps, and further that the movie has a sub-title stream in French encoded at 10 kbps, and a sub-title stream in Japanese encoded at 10 kbps. Also assume that the information associated with the movie indicates that the video streams are mutually exclusive (only one of the three can be streamed at a time to client device 102), that

the audio streams are mutually exclusive (only one of the two can be streamed at a time to client device 102), and that audio is to have priority over video.

Since audio is to have priority over video, server device 104 will stream one of the two audio streams to client device 102. Further, since the predictive play request includes an indication that the default language of client device 102 is English, server device 104 predicts that client device 102 desires the English audio stream.

Given that the English audio stream is encoded at 100 kbps, and the default connection speed for client device 102 is 500 kbps, server device 104 predicts that client device 102 desires the video stream encoded at 300 kbps (since the audio stream is encoded at 100 kbps, server device 104 predicts that there is insufficient bandwidth to stream the 500 kbps stream).

The predictive play request does not include any information about subtitle streams, so server device 104 looks elsewhere to determine whether to predict that a subtitle stream is desired. Since there is leftover bandwidth (only 400 kbps of the 500 kbps have been used), server device 104 may predict that one of the subtitle streams is desired. If one of the subtitle streams is identified in the information associated with the movie as having priority over the other, then that subtitle stream is predicted as being the one desired by client device 102. Otherwise, one of the subtitle streams is predicted as being the one desired by client device 102 in some other way (e.g., randomly, in the order they are referenced in the information associated with the movie, etc.).

Alternatively, server device 104 may keep a record of how frequently the various media streams are desired by client devices. If this record indicates the subtitle streams are infrequently desired by client devices (e.g., less than 50% of the

time), then server device 104 may predict that no subtitle stream is desired by client device 102 and stream no subtitle stream to client device 102.

It should be noted that the media stream(s) predicted by server device 104 as being desired by client device 102 are just that: predictions. There is no guarantee that these media stream(s) are indeed the media stream(s) desired by client device 102. However, by making the prediction, server device 104 can immediately begin streaming the media stream(s) to client device 102, without having to wait for client device 102 to make a selection of which media streams it desires.

Server device 104 then streams the predicted media stream(s) to client device 102. Server device 104 also communicates information including an identification of what media streams are available for the requested resource, as well as what streams were predicted as being desired by client device 102. The information can be included with the predicted media stream(s) and thus streamed to client device 102, or alternatively may be communicated in some other manner, such as in a non-streaming manner (e.g., through an HTTP-based control protocol). In certain embodiments, this information is included in a session description message in accordance with the Session Description Protocol (SDP), Network Working Group Request for Comments (RFC) 2327, April 1998.

Client device 102 receives the predicted media streams, including the identification of what media streams are available for the requested resource, as well as what streams were predicted as being desired by client device 102. Client device 102 can then analyze the identification of what media streams are available for the requested resource and select a different media stream than the predicted media streams. The selection of a different stream is optional – if client device 102 desires the media streams predicted by server device 104, then no selection of other media streams need be made by client device 102.

The selected media stream may be in addition to the predicted media streams (e.g., in the preceding example, no subtitle stream may be predicted by server device 104 as being desired by client device 102, but client device 102 may select a subtitle stream). The selected media stream may alternatively be in place of a predicted media stream (e.g., the client device 102 may select the Spanish audio stream rather than the English audio stream).

The selection at client device 102 can be performed in different manners. For example, an application (e.g., a media player application) may be programmed to make the selection. By way of another example, the various media streams available may be presented to a user of the client device and the user may be allowed to select which media stream(s) he or she desires. Any selection made by client device 102 is then sent to server device 104.

In certain embodiments, such as when using HTTP for streaming the media, an identifier of the streaming session is used. The identifier allows different requests to be associated with one another. For example, the predictive play request may be sent over one connection between the client device 102 and the server device 104, while the request that identifies the selected media streams is sent over a different connection between the client device 102 and the server device 104. In order for the server device 104 to know that the two requests are related to one another, the identifier of the streaming session is used. In certain embodiments, server device 104 returns, in the response to the predictive play request, an identifier. This identifier can then be used by client device 102 when sending a request that identifies the selected media streams to server device 104 in order to inform server device 104 that the request identifying the selected media streams is associated with the predictive play request.

The request that identifies the selected media streams can be in any of a variety of different formats. In certain embodiments, the request is a Select request which can be sent, for example, as an HTTP Post command or HTTP Get command to server device 104. Alternatively, the request can be delivered in other manners, such as using an RTSP command. The Select request includes an identifier that identifies the request as a Select request, zero or more headers, and optionally a message body. In certain embodiments, the information identifying the selected media streams is included in one or more headers of the Select request, although the information identifying the selected media streams could alternatively be included in the message body. Additionally, in certain implementations, the presence of one or more such headers in the request identify the request as a Select request. Table I illustrates an example of headers that can be included in the Select request.

Table I

| Header Name | Description |
|-----------------------------|---|
| Pragma: client-id | An identifier of the streaming media session. Allows the client device to indicate that the Select request is associated with a particular streaming session. |
| Pragma: stream-switch-entry | An identification of the media streams selected by the client device. The individual media streams that are available for the requested resource (e.g., as identified by the server device in the session description message) are assigned identifiers (which are typically included in the session description message). A list of one or more of these identifiers is included as parameters of this stream-switch-entry header. |

The information describing the selected media streams can describe the selected media streams in different ways. As part of the description of the selected

media streams, a “thinning level” may be included that indicates how much of the data of the media stream is to be streamed to client device 102 (e.g., all frames, only key frames, etc.). As an example, the description may be a list of the media stream(s) desired by client device 102, without regard for which streams were predicted by server device 104. By way of another example, the description may be a list of the media stream(s) desired by client device 102 that are in addition to the streams that were predicted by server device 104. By way of yet another example, the description may include one or more stream pair values, each pair identifying a predicted stream and a selected stream that is to replace the predicted stream. By way of yet another example, the description may be a list of all of the media stream(s) and an identification of whether the media stream has been selected for streaming to client device 102. By way of still another example, the description may be a “negative” description (e.g., listing all of the media stream(s) that the client device 102 does not desire, and thus inherently identifying the media stream(s) that the client device 102 does desire).

Server device 104 receives the Select request and begins streaming the selected streams to client device 102. Server device 104 stops streaming to client device 102 any predicted stream(s) that were replaced by a selected stream, but continues to stream any predicted stream(s) that was not replaced by a selected stream.

The resource identified by a predictive play request can be associated with a single media or multimedia presentation, or alternatively multiple such presentations. For example, the resource may identify a movie, which is a single multimedia presentation even though it may have multiple video, audio, and/or subtitle streams. By way of another example, the resource may identify a movie as well as a commercial and a trailer for a different movie. In this example, the movie

would be one multimedia presentation, the commercial a second multimedia presentation, and the trailer a third multimedia presentation.

When a resource is associated with multiple multimedia presentations, server device 104 can predict which presentations are desired by client device 102, as well as predict which media streams within those presentations are desired by client device 102. For example, assume that there are two different commercials associated with a resource, one commercial targeting a Hispanic audience and the other targeting a Chinese audience. If the predictive play request indicates that the default language is Spanish, then server device 104 can predict that the commercial targeting the Hispanic audience is desired by client device 102 rather than the commercial targeting the Chinese audience.

It should be noted that, in some situations, server device 104 may choose to not make a prediction. Such situations can arise, for example, when the likelihood of making an incorrect prediction is too great. In such situations, server device 104 sends the identification of what media streams are available for the requested resource but does not start streaming any predicted media stream(s). Rather, server device 104 waits until client device has made a selection of one or more of the available media streams, and starts streaming the selected media stream(s) when a request to play the selected media stream(s) is received.

Fig. 4 is a flowchart illustrating an example predictive start process 200. Fig. 4 shows acts performed by both a client device 102 (e.g., of Figs. 1, 2, and 3) and a server device 104 (e.g., of Figs. 1, 2, and 3). Acts performed by client device 102 are shown on the left side of Fig. 4, while acts performed by server device 104 are shown on the right side of Fig. 4.

Initially, client device 102 generates a predictive play request (act 202). The predictive play request identifies one or more media or multimedia presentations,

represented as a resource of server device 104, that is desired to be streamed to client device 102. Client device 102 sends the predictive play request to server device 104 (act 204), which receives the predictive play request (act 206).

In response to the predictive play request, server device 104 predicts one or more media stream(s) desired by the client device 102 (act 208). As discussed above, this prediction can be based on different factors, such as information included in the predictive play request, default settings at server device 104, information associated with the requested resource, previous requests for streaming media, and so forth. The predicted media stream(s) are then streamed to client device 102 (act 210). As part of this streaming in act 210, an identification of the predicted media stream(s) as well as the available media stream(s) for the requested resource is also sent to client device 102. Alternatively, as discussed above, the identification of the predicted media stream(s) as well as the available media stream(s) for the requested resource can be communicated to client device 102 in other manners, including non-streaming manners.

Client device 102 receives the predicted media stream(s) (act 212) and analyzes the received identification of the predicted media stream(s) as well as the available media stream(s) for the requested resource. Process 200 then proceeds based on whether client device 102 desires to have streamed to it one or more different media stream(s) than the predicted media stream(s). As discussed above, a selection of such a different media stream(s) can be made in different manners, such as by an application program on client device 102 or a user-input at client device 102. If a different media stream(s) is not selected (act 214), then client device 102 does desire the media stream(s) that server device 104 predicted and processing of those received media stream(s) continues (act 216). This processing

can include, for example, buffering the received media stream(s) and/or presenting (e.g., playing back) the received media stream(s) at client device 102.

However, if different media stream(s) are selected, then client device 102 sends an indication of those selected media stream(s) to server device 104 (act 218). Server device 104 receives the indication of the selected media stream(s) (act 220), and identifies the media stream(s) to send to client device 102 (act 222). As discussed above, the stream(s) selected by client device 102 can be in addition to the predicted media stream(s), or in place of one or more of the predicted media stream(s). The media stream(s) identified by server device 104 in act 222 include the streams indicated as being selected by client device 102 (received in act 220), and may also include one or more of the stream(s) predicted by server device 104 in act 208.

Server device 104 then streams the identified media stream(s) streamed to client device 102 (act 224). Typically, the identified media stream(s) are streamed from the current playback position (e.g., when the selection request is received, the temporal location of the media stream(s) currently being streamed is identified, and then the corresponding temporal location of the identified media stream(s) is determined and streaming of the identified media stream(s) begins at that determined location). Alternatively, other starting points may be used, such as the beginning of the identified media stream(s). Client device 102 receives the identified media stream(s) from server device 104 (act 226), and processes the received media stream(s) (act 216). This processing can include, for example, buffering the media stream(s) and presenting (e.g., playing back) the media stream(s) at client device 102.

Returning to Fig. 2, the fast startup discussed herein also includes a process for switching delivery channel(s) during streaming. Transport protocol 152 can use

a variety of different delivery channels. Typically, each media stream is sent from server device 104 to client device 102 using a different delivery channel. These delivery channels for different media streams are typically of the same type, although alternatively they can be of different types. Two common types of delivery channels are TCP channels and UDP channels. TCP channels transmit data in packets referred to as TCP packets, while UDP channels transmit data in packets referred to as UDP packets. Typically, delivery of data packets using TCP is more reliable, but also more time-consuming, than delivery of data packets using UDP. Additionally, situations can arise, such as where a firewall is located between client device 102 and server device 104, where UDP packets cannot be received by client device 102 but TCP packets can be received by client device 102.

Fig. 5 is a block diagram illustrating example client and server devices and the delivery channel switching process in additional detail. In certain embodiments, the protocols 150, 152, and 154 of Fig. 2 are included in the client and server devices of Fig. 5, but have not been illustrated. Furthermore, although not shown in Fig. 5, one or more additional devices (e.g., firewalls, routers, gateways, bridges, etc.) may be situated between client device 102 and server device 104. The delivery channel switching discussed herein allows the media content source to begin streaming the media stream(s) using one delivery channel (e.g., TCP), and then check whether another delivery channel (e.g., UDP) will work and switch to that other delivery channel if it will work.

Initially, client device 102 sends a play request 252 to server device 104, requesting that one or more media streams be streamed to client device 102. This play request 252 can be a predictive play request as discussed above, or alternatively any other request (or combination of multiple requests) that requests that one or more media streams be streamed to client device 102. For example, this

play request 252 could be an HTTP Post command with a Select request that identifies the desired media streams. In response to the play request, server device 104 streams 254 the requested media stream(s) to client device 102 using TCP. This response includes the data for the requested media stream(s) in multiple TCP packets.

While streaming the media from server device 104 to client device 102 using TCP, client device 102 sends a UDP probing packet 256 to server device 104. In certain embodiments, this is accomplished by server device 104 opening a UDP port and sending an identification of the opened UDP port to client device 102 along with the streaming media using TCP 254. Client device 102 receives the UDP port identification, and then can send a UDP packet to that port of server device 104. Alternatively, server device 104 may have a well-known UDP port to use for UDP probing packets, and client device 102 can send UDP probing packet 256 to this well-known UDP port without server device 104 first sending an identification of the opened UDP port to client device 102. Client device 102 also opens a UDP port that it can identify to server device 104, as part of UDP probing packet 256, as the UDP port on which it can receive UDP packets.

Server device 104 receives the UDP probing packet 256 and generates and sends a UDP probing response packet 258 to client device 102. UDP probing response packet 258 is an indication to client device 102 that UDP probing packet 256 was received by server device 104. UDP probing response packet 258 optionally includes some sort of identifier of UDP probing packet 256, such as an identifier included in UDP probing packet 256. Alternatively, the mere existence of a response over the same connection may itself associate the response packet 258 with the probing packet 256. For example, if a single request and associated

response are the only two messages sent over a connection, then the response 258 will be viewed as the response to probing packet 256.

It should be noted that while in some embodiments a single UDP probing packet 256 is sent by client device 102, in other embodiments multiple UDP probing packets 256 are sent by client device 102 and a UDP probing response packet 258 is returned by server device 104 for each such UDP probing packet 256. The multiple UDP probing packets 256 may be sent by client device 102 concurrently, or alternatively at different times (e.g., client device 102 may wait for a UDP probing response packet 258 to be returned for one UDP probing packet 256 before sending another UDP probing packet 256). For example, in embodiments where transport protocol 152 includes the RTP and RTCP protocols, the RTP data is sent using one delivery channel or flow, and the RTCP data is sent using a different delivery channel or flow. A UDP probing packet 256 can be sent for each of these two delivery channels or flows. By way of another example, a different UDP probing packet 256 may be sent by client device 102 for each of multiple media streams (e.g., for each media stream that client device 102 would like to switch to streaming using UDP, for each media stream currently being streamed to client device 102 using TCP, and so forth). These two examples may also be combined (e.g., two UDP probing packets 256 (one for the RTP flow and the other for the RTCP flow) may be sent by client device 102 for each of multiple media streams).

When client device 102 receives the UDP probing response packet 258, client device 102 knows that UDP packets can be sent from server device 104 to client device 102. So, client device 102 sends a request to server device 104 to switch to streaming the media stream(s) from server device 104 to client device 102 using UDP 260. Typically, client device 102 sends a request to server device 104

that all of the media stream(s) be switched to being streamed using UDP, although alternatively client device 102 may identify in the request only particular media stream(s) that are to be switched to being streamed using UDP (with the remaining media stream(s) being still streamed using TCP). Alternatively, multiple requests may be sent from client device 102 to server device 104, each request identifying a particular one or more media streams that should be streamed to client device 102 using UDP.

In response to this request, server device 104 begins streaming the media stream(s) requested in play request 252 to client device 102 using UDP 262. In certain embodiments, because streaming of the media stream(s) using TCP has already begun, server device 104 does not start streaming using UDP at the beginning of the media stream(s). Rather, server device 104 stops streaming the data using TCP and starts streaming using UDP at whatever point the streaming using TCP was stopped. Alternatively, there may be some overlap between packets (e.g., at the point where the switch to UDP occurs, some packets already sent using TCP may be re-sent using UDP).

The request to switch to streaming the media stream(s) from server device 104 to client device 102 using UDP 260 can be in any of a variety of different formats. In certain embodiments, the request is a Select request, which can be sent as an HTTP Post command to server device 104. Alternatively, the request can be delivered in other manners, such as using an RTSP command. The Select request includes an identifier that identifies the request as a Select request, zero or more headers, and optionally a message body. In certain embodiments, the information requesting that the server device switch to streaming the media stream(s) using UDP is included in one or more headers of the Select request, although the information requesting that the server device switch to streaming the media

stream(s) using UDP could alternatively be included in the message body, or alternatively the request to switch to streaming the media stream(s) using UDP could be inherent in the request. In certain implementations, the presence of a Pragma: stream-switch-entry header (discussed with reference to Table I) and/or an X-Transport header (discussed with reference to Table II) serves as the identifier that identifies the request as a Select request. Table II illustrates an example of headers that can be included in the Select request.

Table II

| Header Name | Description |
|-------------------|---|
| Pragma: client-id | An identifier of the streaming media session. Allows the client device to indicate that the Select request is associated with a particular streaming session. |
| X-Transport | Used to indicate that the media stream(s) should be switched to being streamed using UDP. A parameter of "RTP/UDP transport" is used in certain implementations to indicate that the media stream(s) should be switched to being streamed using UDP. This header also typically includes a parameter that identifies which UDP port(s) that the client is using for receiving RTP and RTCP packets. |

In certain embodiments, control information (e.g., RTCP packets) are sent from server device 104 to client device 102 using TCP packets even if the data representing the actual content of the media streams (e.g., the data representing the video or audio that is played back at client device 102) is sent from server device 104 to client device 102 using UDP packets. In other embodiments, all data, including control information (e.g., RTCP packets) is sent from server device 104 to client device 102 using UDP packets.

It should be noted that if UDP packets cannot be used to stream data from server device 104 to client device 102, UDP probing packet 256 will not arrive at

server device 104 (so server device 104 will never send UDP probing response packet 258), or UDP probing response packet 258 will not arrive at client device 102. So, if client device 102 does not receive a response to the UDP probing packet 256, client device 102 knows that UDP packets cannot be used to stream data from server device 104 to client device 102. Optionally, if client device 102 does not receive a response to the UDP probing packet 256, client device 102 may re-send UDP probing packet(s) one or more times to verify that UDP packets cannot be used to stream data from server device 104 to client device 102.

Additionally, it should be noted that in situations where multiple UDP probing packets 256 are sent (e.g., for multiple media streams or multiple flows), UDP probing response packets 258 may be received in response to some of the probing packets 256 but not others, indicating that UDP packets can be streamed from server device 104 to client device 102 for some streams or flows (those for which the UDP probing response packet 258 was received) but not for other streams or flows (those for which the UDP probing response packet was not received).

It should also be noted that the probing of whether UDP packets can be used to stream data from server device 104 to client device 102 occurs while the requested media stream(s) are being streamed to client device 102 using TCP 254. Thus, the client device 102 begins receiving the requested media stream(s) without having to wait for a determination of whether the data can be streamed to client device 102 using UDP packets.

It should be further noted that the UDP probing packet 256 and UDP probing response packet 258 can include any of a variety of information. Information about the media stream(s) need not be sent between client device 102 and server device 104 using the UDP probing packet 256 and UDP probing response packet 258. The

importance of the UDP probing and UDP probing response packets is simply whether the packets arrive at their intended targets, rather than any particular information about the media stream(s) that is included in the packets.

Fig. 6 is a flowchart illustrating an example delivery channel switching process 300. Fig. 6 shows acts performed by both a client device 102 (e.g., of Figs. 1, 2, and 5) and a server device 104 (e.g., of Figs. 1, 2, and 5).

Initially, client device 102 sends a request for streaming media to a server device 104 (act 302). Server device 104 receives the request and begins streaming the requested media stream(s) to client device 102 using TCP (act 304). Server device 104 continues to stream the requested media stream(s) using TCP, and client device 102 receives and processes the media stream(s) (act 306). This processing can include, for example, buffering and/or presenting (e.g., playing back) the media stream(s) at client device 102.

Concurrent with the streaming of the requested media stream(s) using TCP in act 306, the link between client device 102 and server device 104 is probed to determine whether data can be streamed from server device 104 to client device 102 using UDP (act 308). As discussed above, this probing can involve information and packets sent between client device 102 and server device 104.

Process 300 then proceeds based on whether the probing indicates that data can be streamed from server device 104 to client device 102 using UDP (act 310). If data cannot be streamed from server device 104 to client device 102 using UDP, then server device 104 continues to stream the requested media stream(s) using TCP and client device 102 receives and processes the media stream(s) received using TCP (act 306). However, if data can be streamed from server device 104 to client device 102 using UDP, then server device 104 switches to streaming the requested media stream(s) using UDP rather than TCP, and client device 102

receives and processes the media stream(s) received using UDP (act 312). Typically, server device 104 switches to streaming the requested media stream(s) using UDP rather than TCP in response to a request (e.g., a Select request) sent from client device 102 requesting the switch to UDP. Additionally, it should be noted that not all streams need to be switched to streaming using UDP. Rather, client device 102 may request that only a particular one or more media streams be switched to streaming using UDP, and that one or more other media streams remain being streamed using TCP.

The fast startup described herein can include the predictive start technique and/or the delivery channel switching technique. Figs. 3 and 4 above discuss example embodiments of the predictive start technique. Figs. 5 and 6 above discuss example embodiments of the delivery channel switching technique.

Fig. 7 is a flowchart illustrating an example fast startup process 350 including both predictive start and delivery channel switching. Fig. 7 shows acts performed by both a client device 102 (e.g., of Figs. 1, 2, and 5) and a server device 104 (e.g., of Figs. 1, 2, and 5).

Initially, client device 102 sends a predictive play request to server device 104 (act 352). Server device 104 receives the predictive play request and predicts the media stream(s) desired by client device 102 (act 354). Server device 104 then begins streaming the predicted media stream(s) to client device 102 using TCP (act 356).

The link between client device 102 and server device 104 is then probed to determine whether data can be streamed from server device 104 to client device 102 using UDP (act 358). If the probing indicates that data can be streamed from server device 104 to client device 102 using UDP, then server device 104 switches to streaming the media stream(s) using UDP rather than TCP (act 360); otherwise,

server device 104 continues streaming the media stream(s) using TCP. Typically, server device 104 switches to streaming the requested media stream(s) using UDP rather than TCP in response to a request (e.g., a Select request) sent from client device 102 requesting the switch to UDP. Additionally, it should be noted that not all streams need to be switched to streaming using UDP. Rather, client device 102 may request that only a particular one or more media streams be switched to streaming using UDP, and that one or more other media streams remain being streamed using TCP.

Additionally, after streaming of the predicted media stream(s) using TCP has begun in act 356, client device 102 determines whether it desires media stream(s) other than the predicted media stream(s) (act 362). If so, then one or more media streams other than the predicted media stream(s) are selected (in addition to and/or in place of the predicted media streams), and client device 102 sends server device 104 an indication of the selected media stream(s) (act 364). Server device 104 receives this indication from client device 102 and streams the selected media stream(s) to client device 102 (act 366).

It should be noted that the media stream(s) being streamed in act 360 can include media stream(s) predicted in act 354 and/or media stream(s) selected in act 364. Whether the media stream(s) being streamed in act 360 are media stream(s) predicted in act 354 and/or media stream(s) selected in act 364 will depend on whether any media stream(s) are selected in act 364, as well as whether server 104 begins streaming any selected media stream(s) in act 366 prior to or subsequent to the switch of act 360.

In certain embodiments, the delivery channel switching of acts 358 and 360 is performed concurrently with the media stream selection of acts 362, 364, and 366. In other embodiments, the delivery channel switching of acts 358 and 360 can

be performed subsequent to, or alternatively prior to, the media stream selection of acts 362, 364, and 366.

Fig. 8 illustrates a general computer environment 400, which can be used to implement the techniques described herein. The computer environment 400 is only one example of a computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the computer and network architectures. Neither should the computer environment 400 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary computer environment 400.

Computer environment 400 includes a general-purpose computing device in the form of a computer 402. Computer 402 can be, for example, a client device 102 or server device 104 of Figs. 1, 2, 3, and 5. Computer 402 can also be an encoder device that is the source of a multimedia presentation. The components of computer 402 can include, but are not limited to, one or more processors or processing units 404, a system memory 406, and a system bus 408 that couples various system components including the processor 404 to the system memory 406.

The system bus 408 represents one or more of any of several types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, and a processor or local bus using any of a variety of bus architectures. By way of example, such architectures can include an Industry Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA) local bus, and a Peripheral Component Interconnects (PCI) bus also known as a Mezzanine bus.

Computer 402 typically includes a variety of computer readable media. Such media can be any available media that is accessible by computer 402 and includes both volatile and non-volatile media, removable and non-removable media.

The system memory 406 includes computer readable media in the form of volatile memory, such as random access memory (RAM) 410, and/or non-volatile memory, such as read only memory (ROM) 412. A basic input/output system (BIOS) 414, containing the basic routines that help to transfer information between elements within computer 402, such as during start-up, is stored in ROM 412. RAM 410 typically contains data and/or program modules that are immediately accessible to and/or presently operated on by the processing unit 404.

Computer 402 may also include other removable/non-removable, volatile/non-volatile computer storage media. By way of example, Fig. 8 illustrates a hard disk drive 416 for reading from and writing to a non-removable, non-volatile magnetic media (not shown), a magnetic disk drive 418 for reading from and writing to a removable, non-volatile magnetic disk 420 (e.g., a "floppy disk"), and an optical disk drive 422 for reading from and/or writing to a removable, non-volatile optical disk 424 such as a CD-ROM, DVD-ROM, or other optical media. The hard disk drive 416, magnetic disk drive 418, and optical disk drive 422 are each connected to the system bus 408 by one or more data media interfaces 426. Alternatively, the hard disk drive 416, magnetic disk drive 418, and optical disk drive 422 can be connected to the system bus 408 by one or more interfaces (not shown).

The disk drives and their associated computer-readable media provide non-volatile storage of computer readable instructions, data structures, program modules, and other data for computer 402. Although the example illustrates a hard disk 416, a removable magnetic disk 420, and a removable optical disk 424, it is to

be appreciated that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like, can also be utilized to implement the exemplary computing system and environment.

Any number of program modules can be stored on the hard disk 416, magnetic disk 420, optical disk 424, ROM 412, and/or RAM 410, including by way of example, an operating system 426, one or more application programs 428, other program modules 430, and program data 432. Each of such operating system 426, one or more application programs 428, other program modules 430, and program data 432 (or some combination thereof) may implement all or part of the resident components that support the distributed file system.

A user can enter commands and information into computer 402 via input devices such as a keyboard 434 and a pointing device 436 (e.g., a "mouse"). Other input devices 438 (not shown specifically) may include a microphone, joystick, game pad, satellite dish, serial port, scanner, and/or the like. These and other input devices are connected to the processing unit 404 via input/output interfaces 440 that are coupled to the system bus 408, but may be connected by other interface and bus structures, such as a parallel port, game port, or a universal serial bus (USB).

A monitor 442 or other type of display device can also be connected to the system bus 408 via an interface, such as a video adapter 444. In addition to the monitor 442, other output peripheral devices can include components such as speakers (not shown) and a printer 446 which can be connected to computer 402 via the input/output interfaces 440.

Computer 402 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computing device 448. By way of example, the remote computing device 448 can be a personal computer, portable computer, a server, a router, a network computer, a peer device or other common network node, and the like. The remote computing device 448 is illustrated as a portable computer that can include many or all of the elements and features described herein relative to computer 402.

Logical connections between computer 402 and the remote computer 448 are depicted as a local area network (LAN) 450 and a general wide area network (WAN) 452. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets, and the Internet.

When implemented in a LAN networking environment, the computer 402 is connected to a local network 450 via a network interface or adapter 454. When implemented in a WAN networking environment, the computer 402 typically includes a modem 456 or other means for establishing communications over the wide network 452. The modem 456, which can be internal or external to computer 402, can be connected to the system bus 408 via the input/output interfaces 440 or other appropriate mechanisms. It is to be appreciated that the illustrated network connections are exemplary and that other means of establishing communication link(s) between the computers 402 and 448 can be employed.

In a networked environment, such as that illustrated with computing environment 400, program modules depicted relative to the computer 402, or portions thereof, may be stored in a remote memory storage device. By way of example, remote application programs 458 reside on a memory device of remote computer 448. For purposes of illustration, application programs and other executable program components such as the operating system are illustrated herein

as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computing device 402, and are executed by the data processor(s) of the computer.

Various modules and techniques may be described herein in the general context of computer-executable instructions, such as program modules, executed by one or more computers or other devices. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically, the functionality of the program modules may be combined or distributed as desired in various embodiments.

An implementation of these modules and techniques may be stored on or transmitted across some form of computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example, and not limitation, computer readable media may comprise “computer storage media” and “communications media.”

“Computer storage media” includes volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

“Communication media” typically embodies computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as

carrier wave or other transport mechanism. Communication media also includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared, and other wireless media. Combinations of any of the above are also included within the scope of computer readable media.

Although the description above uses language that is specific to structural features and/or methodological acts, it is to be understood that the invention defined in the appended claims is not limited to the specific features or acts described. Rather, the specific features and acts are disclosed as exemplary forms of implementing the invention.

CLAIMS**1.** A method comprising:

sending, from a client device, a request that one or more media streams be streamed to the client device;

receiving, at a server device, the request;

beginning streaming the one or more media streams from the server device to the client device using TCP (Transmission Control Protocol);

probing, while streaming the one or more media streams from the server device to the client device, whether data can be streamed from the server device to the client device using UDP (User Datagram Protocol); and

switching to streaming the one or more media streams from the server device to the client device using UDP if the probing indicates that data can be streamed from the server device to the client device using UDP, otherwise continuing to stream the one or more media streams from the server device to the client device using TCP.

2. A method as recited in claim 1, the probing comprising:

sending, from the client device to the server device, one or more UDP probing packets;

sending, in response to each of the one or more UDP probing packets, a UDP response packet from the server device to the client device; and

determining, if at least one of the UDP response packets is received by the client device, that at least some data can be streamed from the server device to the client device using UDP, and otherwise determining that data cannot be streamed from the server device to the client device using UDP.

3. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors, causes the one or more processors to:

receive, from a requestor, a request for media content to be streamed to the requestor;

stream at least a portion of the media content to the requestor using Transmission Control Protocol (TCP) packets;

receive, concurrent with the at least a portion of the content being streamed to the requestor, a request to change to using User Datagram Protocol (UDP) packets; and

stream one or more additional portions of the media content to the requestor using UDP packets rather than TCP packets.

4. One or more computer readable media as recited in claim 3, the request to change to using UDP packets being received after probing has been performed to determine whether media content can be streamed to the requestor using UDP packets and a determination has been made that media content can be streamed to the requestor using UDP packets, the probing occurring while the at least a portion of the media content is being streamed to the requestor using TCP packets.

5. One or more computer readable media as recited in claim 3, the one or more additional portions of the media content streamed to the requestor using UDP packets including none of the at least a portion of the media content streamed to the requestor using TCP packets.

6. One or more computer readable media as recited in claim 3, wherein the instructions further cause the one or more processors to:

send, after receipt of the request to change to using UDP packets, data representing control information to the requestor using TCP packets.

7. One or more computer readable media as recited in claim 3, wherein the instructions that cause the one or more processors to stream the one or more additional portions of the media content to the requestor using UDP packets rather than TCP packets comprise instructions that cause the one or more processors to stream one or more additional portions of only selected media streams of the media content to the requestor using UDP packets rather than TCP packets.

8. One or more computer readable media as recited in claim 3, wherein the request to change to using UDP packets comprises a Select request having one or more headers to indicate that the media content is to be streamed to the requestor using UDP packets.

9. One or more computer readable media as recited in claim 8, wherein the one or more headers comprises an X-Transport header with an RTP/UDP transport parameter.

10. One or more computer readable media as recited in claim 3, the requestor comprising a client device.

11. One or more computer readable media as recited in claim 3, the one or more processors being part of a server device.

12. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a device, causes the one or more processors to:

receive, from a media content source, at least a portion of media content using Transmission Control Protocol (TCP) packets;

determine, during receipt of the at least a portion of the media content using TCP packets, whether the media content can be streamed to the device using User Datagram Protocol (UDP) packets; and

send, if the media content can be streamed to the device using UDP packets, a request to the media content source to stream the media content to the device using UDP packets.

13. One or more computer readable media as recited in claim 12, wherein the instructions further cause the one or more processors to:

receive, after having sent the request, data representing the media content using UDP packets and data representing control information using TCP packets.

14. One or more computer readable media as recited in claim 12, wherein the instructions that cause the one or more processors to send the request to the media content source comprise instructions that cause the one or more processors to send a request to the media content source to stream only selected media streams of the media content to the device using UDP packets.

15. One or more computer readable media as recited in claim 12, wherein the instructions that cause the one or more processors to determine whether the media content can be streamed to the device using UDP packets comprise instructions that cause the one or more processors to:

send, to the media content source, a UDP probing packet; and

determine, if a UDP response packet is received from the media content source in response to the UDP probing packet, that the media content can be streamed to the device using UDP packets, and otherwise determine that the media content cannot be streamed to the device using UDP packets.

16. One or more computer readable media as recited in claim 12, wherein the media content source comprises a server device.

17. One or more computer readable media as recited in claim 12, wherein the request comprises a Select request having an X-Transport header with an RTP/UDP transport parameter to indicate that the media content is to be streamed to the device using UDP packets.

18. One or more computer readable media as recited in claim 12, wherein the request comprises a HyperText Transport Protocol (HTTP) Post command.

19. One or more computer readable media as recited in claim 12, wherein the request comprises a HyperText Transport Protocol (HTTP) Get command.

20. A method comprising:

sending, from a client device to a server device, a predictive play request;

predicting, at the server device and in response to the predictive play request, one or more media streams desired by the client device;

streaming at least the predicted one or more media streams from the server device to the client device;

determining, at the client device, whether at least one media stream different from the predicted one or more media streams are selected for streaming to the client device; and

if the at least one media stream different from the predicted one or more media streams are selected for streaming to the client device, then sending, to the server device, an indication of the selected at least one media stream, and streaming from the server device to the client device the selected at least one media stream.

21. A method as recited in claim 20, wherein the at least one media stream different from the predicted one or more media streams is selected to replace at least one of the predicted one or more streams.

22. A method as recited in claim 20, wherein the streaming from the server device to the client device the selected at least one media stream comprises streaming from the server device to the client device the selected at least one media stream in addition to the predicted one or more media streams.

23. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a device, causes the one or more processors to:

receive a predictive play request from a requestor;

predict, in response to the predictive play request, one or more media streams to be streamed to the requestor; and

stream the predicted one or more media streams to the requestor.

24. One or more computer readable media as recited in claim 23, wherein the predictive play request comprises a request having a "Pragma: PredStart=1" header.

25. One or more computer readable media as recited in claim 23, wherein the instructions further cause the one or more processors to:

receive, from the requestor, a request to stream a different media stream other than one of the predicted one or more media streams;

stop streaming the one of the predicted one or more media streams to the requestor; and

begin streaming the different media stream to the requestor.

26. One or more computer readable media as recited in claim 25, wherein the request to stream a different media stream other than one of the predicted one or more media streams comprises a Select request having a "pragma: stream-switch-entry" header with one or more parameters to indicate the different media stream.

27. One or more computer readable media as recited in claim 23, wherein the instructions further cause the one or more processors to:

communicate, to the requestor, an identification of the predicted one or more streams as well as an identification of one or more additional streams that may be streamed to the requestor.

28. One or more computer readable media as recited in claim 23, wherein the requestor comprises a client device.

29. One or more computer readable media as recited in claim 23, wherein the instructions that cause the one or more processors to predict one or more media streams to be streamed to the requestor comprise instructions that cause the one or more processors to predict the one or more media streams to be streamed to the requestor based at least in part on information included in the predictive play request.

30. One or more computer readable media as recited in claim 29, wherein the information included in the predictive play request comprises an identification of a connection speed between the requestor and the device.

31. One or more computer readable media as recited in claim 29, wherein the information included in the predictive play request comprises an identification of a language desired by the requestor.

32. One or more computer readable media as recited in claim 23, wherein the instructions that cause the one or more processors to predict one or more media streams to be streamed to the requestor comprise instructions that cause the one or more processors to predict the one or more media streams to be streamed to the requestor based at least in part on information, other than information included in the predictive play request, available to the device.

33. One or more computer readable media having stored thereon a plurality of instructions that, when executed by one or more processors of a device, causes the one or more processors to:

send a predictive play request to a media content source; and

receive, in response to the predictive play request, one or more media streams predicted by the media content source as being desired by the device.

34. One or more computer readable media as recited in claim 33, wherein the predictive play request comprises a request having a "Pragma: PredStart=1" header.

35. One or more computer readable media as recited in claim 33, wherein the instructions that cause the one or more processors to receive one or more media streams predicted by the media content source as being desired by the device cause the one or more processors to receive the one or more media streams predicted by the media content source as being desired by the device without the device having selected any of the one or more media streams.

36. One or more computer readable media as recited in claim 33, wherein the instructions further cause the one or more processors to:

select a different media stream than one of the one or more media streams predicted by the media content source as being desired by the device;

send, to the media content source, an indication of the different media stream; and

receive, in response to the indication and from the media content source, the different media stream in place of the one of the one or more media streams predicted by the media content source as being desired by the device.

37. One or more computer readable media as recited in claim 36, wherein the indication of the different media stream comprises a Select request having a “pragma: stream-switch-entry” header with one or more parameters to indicate the different media stream.

38. One or more computer readable media as recited in claim 33, wherein the instructions further cause the one or more processors to:

select a different media stream than the one or more media streams predicted by the media content source as being desired by the device;

send, to the media content source, an indication of the different media stream; and

receive, in response to the indication and from the media content source, the different media stream in addition to the one or more media streams predicted by the media content source as being desired by the device.

39. One or more computer readable media as recited in claim 38, wherein the indication of the different media stream comprises a Select request having a “pragma: stream-switch-entry” header with one or more parameters to indicate the different media stream.

40. One or more computer readable media as recited in claim 33, wherein the instructions further cause the one or more processors to include, in the predictive play request, information to be used by the media content source in predicting the one or more media streams.

41. One or more computer readable media as recited in claim 40, wherein the information included in the predictive play request comprises an identification of a connection speed between the device and the media content source.

42. One or more computer readable media as recited in claim 40, wherein the information included in the predictive play request comprises an identification of a language desired by the device.

43. A method comprising:
sending, from a client device to a server device, a predictive play request;
predicting, at the server device and in response to the predictive play request, one or more media streams desired by the client device;
streaming at least the predicted one or more media streams from the server device to the client device;

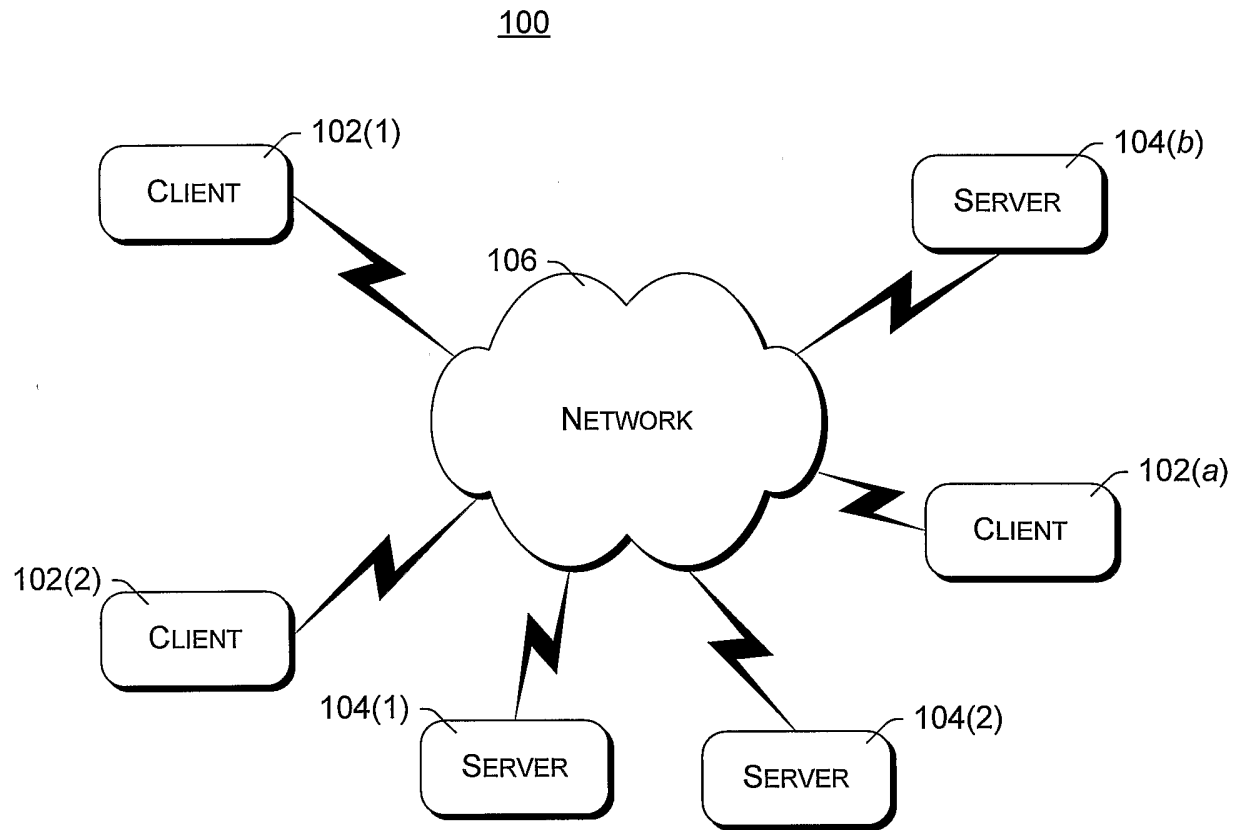
determining, at the client device, whether at least one media stream different from the predicted one or more media streams are selected for streaming to the client device;

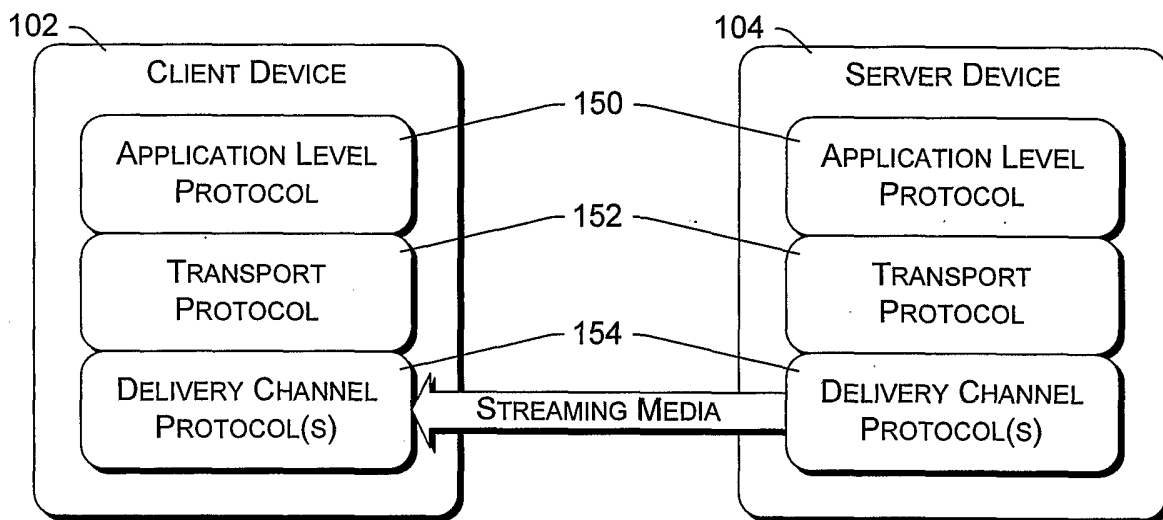
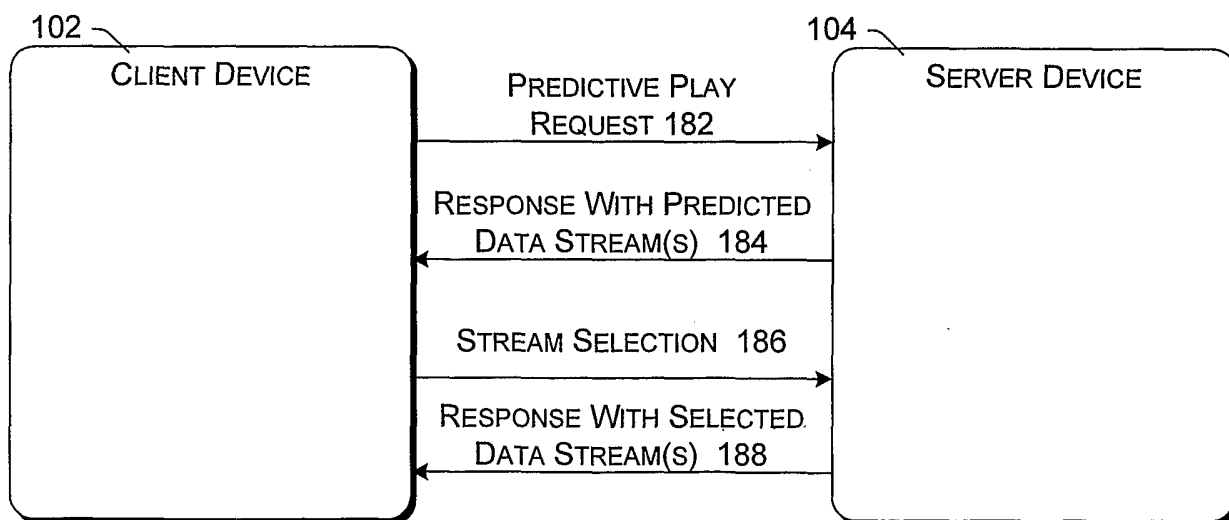
if the at least one media stream different from the predicted one or more media streams are selected for streaming to the client device, then sending, to the server device, an indication of the selected at least one media stream, and streaming from the server device to the client device the selected at least one media stream rather than at least one of the predicted one or more media streams;

beginning streaming the one or more media streams from the server device to the client device using TCP (Transmission Control Protocol);

probing, while streaming the one or more media streams from the server device to the client device, whether data can be streamed from the server device to the client device using UDP (User Datagram Protocol); and

switching to streaming the one or more media streams from the server device to the client device using UDP if the probing indicates that data can be streamed from the server device to the client device using UDP, otherwise continuing to stream the one or more media streams from the server device to the client device using TCP.

*Fig. 1*

*Fig. 2**Fig. 3*

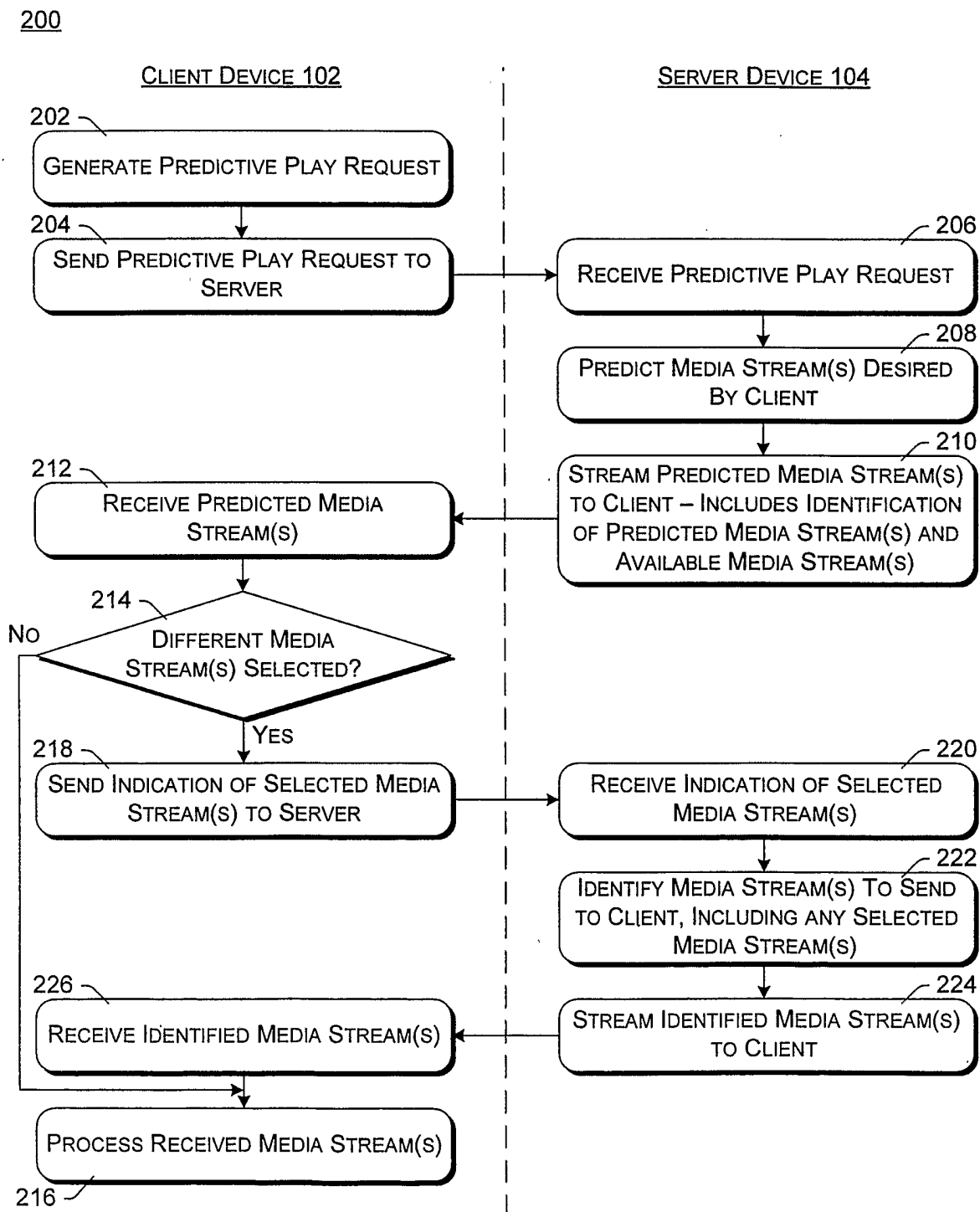
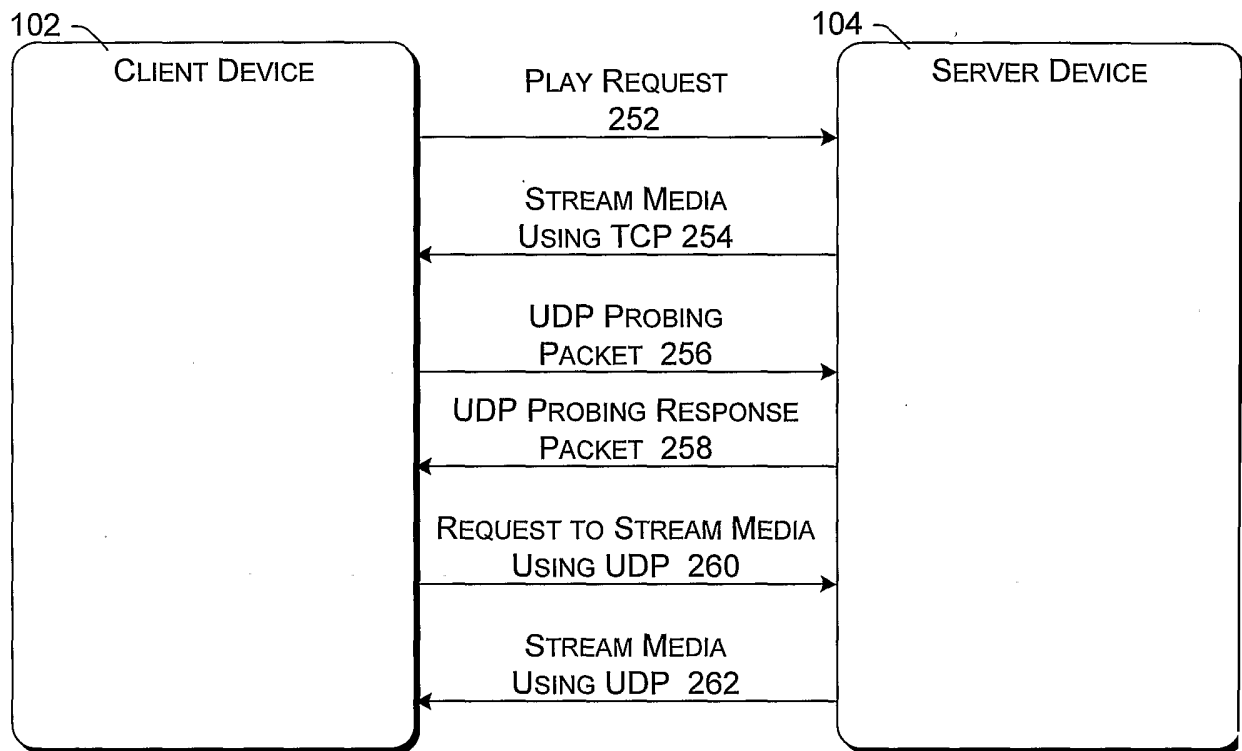
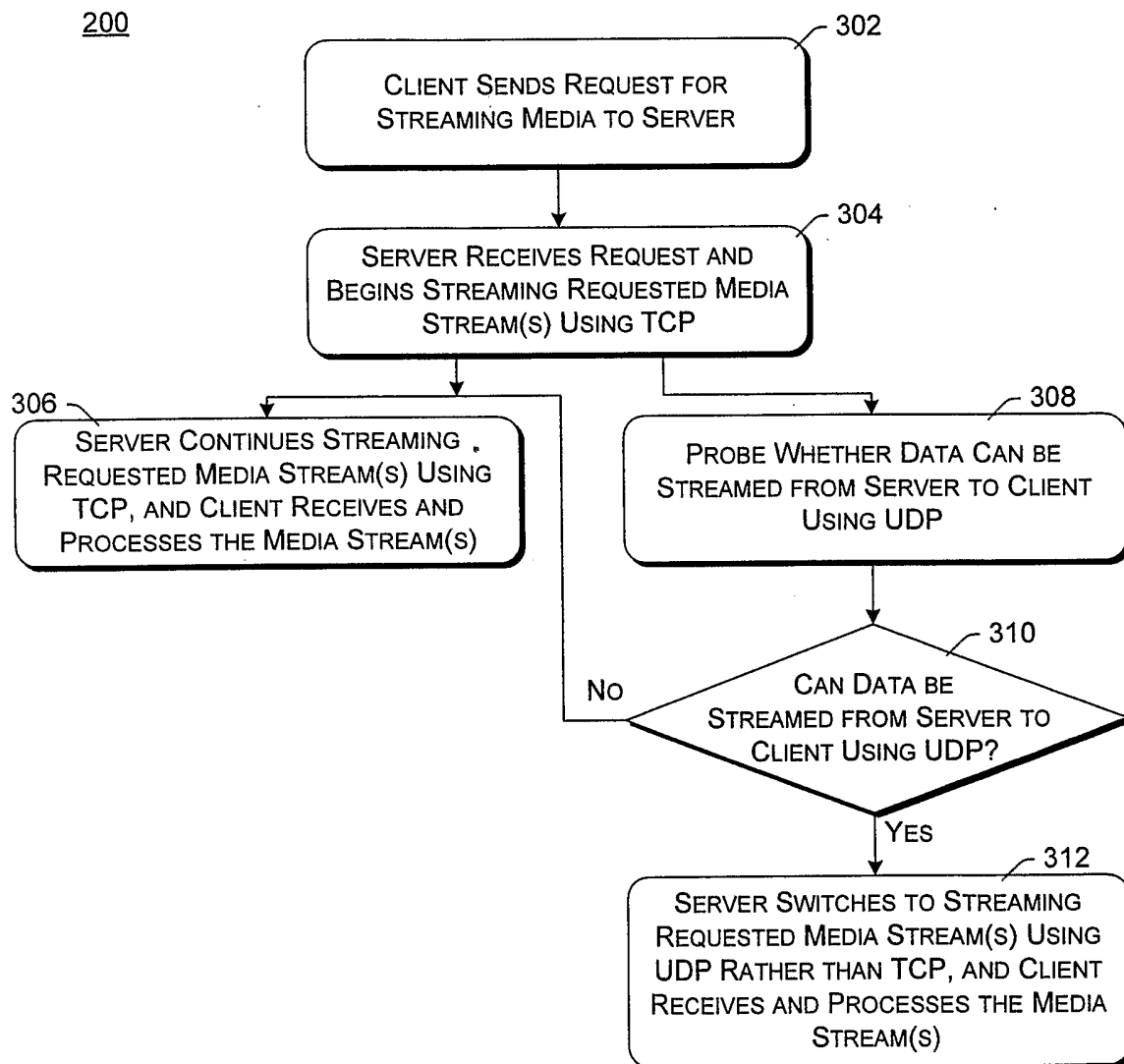
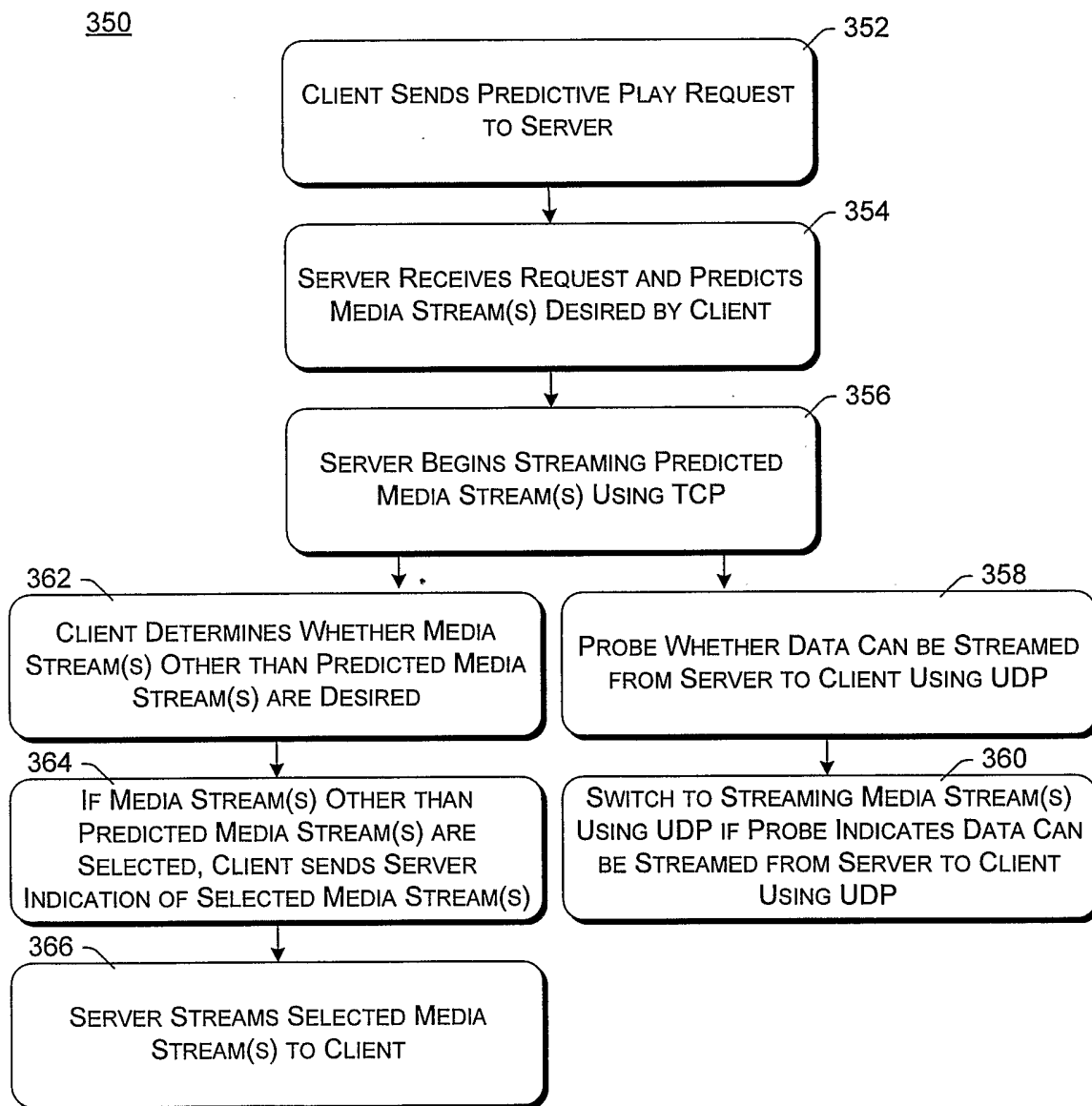


Fig. 4

*Fig. 5*

*Fig. 6*

*Fig. 7*

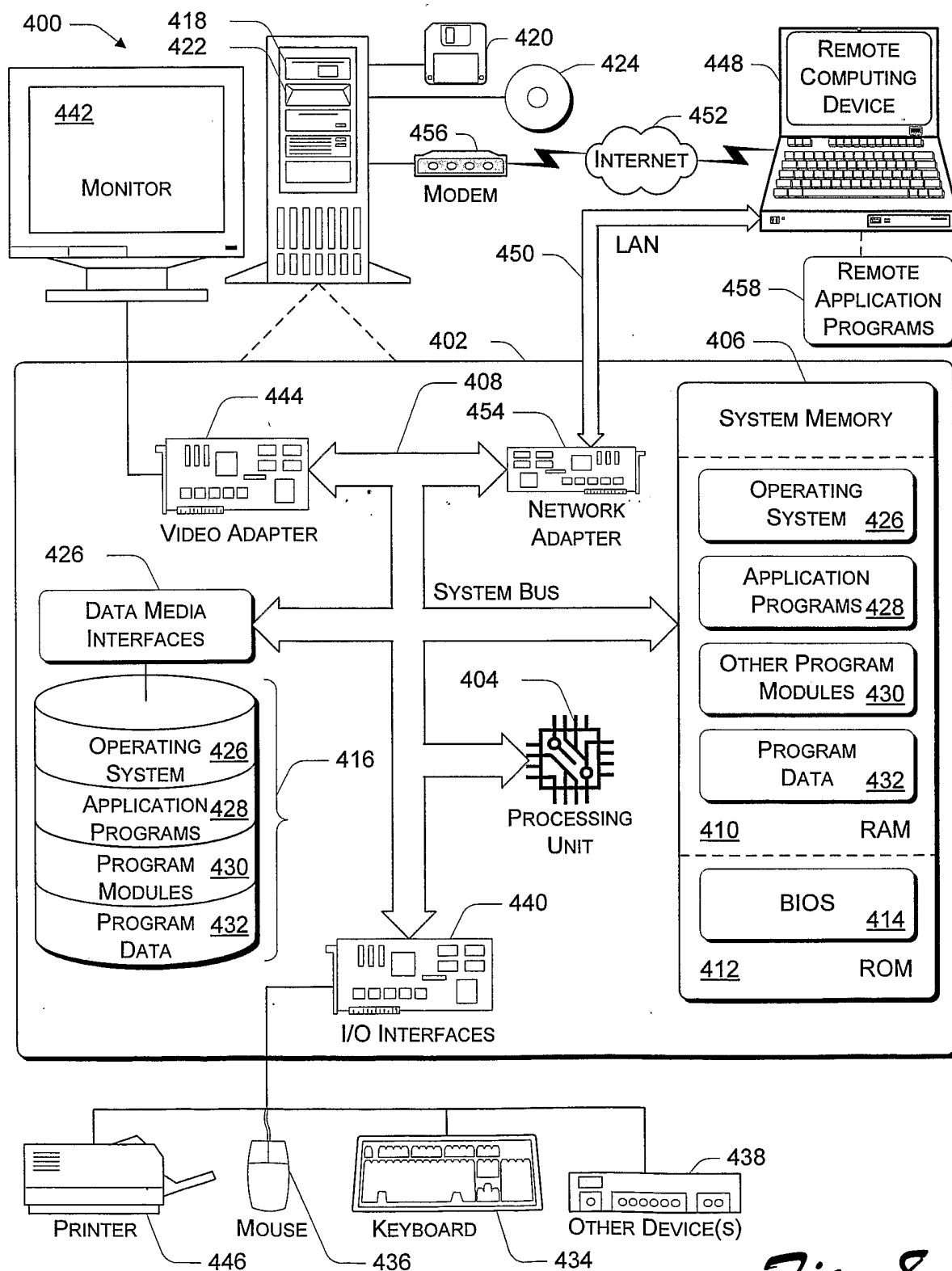


Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US04/23936

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) : G06F 15/16
US CL : 709/230, 231

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 709/230, 231

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category * | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|------------|--|-----------------------|
| Y, P | US PGPUB 2005/0002337 A1 (WANG ET AL.) 06 JANUARY 2005, ABSTRACT. | 1-43 |
| Y, P | US PGPUB 2004/0252701 A1 (ANADAKUMAR ET AL.) 16 DECEMBER 2004, ABSTRACT. | 1-43 |

☐

Further documents are listed in the continuation of Box C.

☐

See patent family annex.

| * Special categories of cited documents: | |
|---|--|
| "A" document defining the general state of the art which is not considered to be of particular relevance | "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention |
| "E" earlier application or patent published on or after the international filing date | "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone |
| "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) | "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art |
| "O" document referring to an oral disclosure, use, exhibition or other means | "&" document member of the same patent family |
| "P" document published prior to the international filing date but later than the priority date claimed | |

Date of the actual completion of the international search

06 September 2005 (06.09.2005)

Date of mailing of the international search report

15 SEP 2005

Name and mailing address of the ISA/US

Mail Stop PCT. Attn: ISA/US
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

Facsimile No. (703) 305-3230

Authorized officer

William C. Vaughn, Jr.
William C. Vaughn, Jr.

Telephone No. (703) 305-9700