IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

WEBGROUP CZECH REPUBLIC, A.S.,

Petitioners,

v.

DISH TECHNOLOGIES L.L.C.,

Patent Owner.

Case No. IPR2025-00468

U.S. Patent No. 10,951,680

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 10,951,680

TABLE OF CONTENTS

Page

EXHIBIT LIST

Ex.	Description
1001	U.S. Patent No. 10,951,680 ("'680 Patent")
1002	Prosecution History for 10,951,680
1003	Declaration of Dr. Rejaie
1004	U.S. Patent No. 7,447,791 ("Leaning")
1005	WO 2004/030310 ("Reme")
1006	Synchronized Multimedia Integration Language (SMIL 2.0) ("SMIL 2.0")
1007	U.S. Patent No. 6,553,413 ("Leighton")
1008	<RESERVED>
1009	337-TA-1265 ITC Investigation Initial Determination
1010	<RESERVED>
1011	<RESERVED>
1012	<RESERVED>
1013	<RESERVED>
1014	Article entitled "Video streaming: Concepts, algorithms, and systems"
1015	Reza Rejaie Dissertation entitled "An End-to-End Architecture for Quality Adaptive Streaming Applications in the Internet"
1016	U.S. Patent No. 6,389,473 ("Carmel")
1017	Article entitled "Design Issues for Layered Quality-Adaptive Internet Video Playback"

1018	Article entitled "Quality Adaptation for Congestion Controlled Video Playback over the Internet"
1019	Article entitled "PALS: Peer-to-Peer Adaptive Layered Streaming"

LIST OF CHALLENGED CLAIMS

Claim	Limitation
1[Pre]	A system for adaptive-rate content streaming of video playable on one or more end user stations over the Internet, the system comprising:
1[A]	at least one processor executing non-transitory executable instructions for generating at least one virtual timeline corresponding to the video;
1[B][1]	wherein the video encoded at a plurality of different bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream,
1[B][2]	the low quality stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of different bitrates,
1[B][3]	each group of streamlets comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;
1[C]	wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and
1[D][1]	wherein the first streamlet of each of the groups of streamlets has the same first duration and encodes the same first portion of the video in each of the low quality stream, the medium quality stream, and the high quality stream, and
1[D][2]	wherein the first streamlet of the low quality stream encodes the same first portion of the video at a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream.
2	The system of claim 1, wherein the processor is further for generating a plurality of virtual timelines wherein each virtual

Claim	Limitation
	timeline corresponds to each of the low quality stream, the medium quality stream, and the high quality stream.
3	The system of claim 1, wherein the video is a live event video.
4	The system of claim 1, wherein the video includes archived content.
5	The system of claim 1, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.
6	The system of claim 5, wherein the first and second durations are different.
7[A]	The system of claim 1, further comprising: a plurality of web servers located at different locations across the internet, each web server configured to:
7[B]	receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the

Claim	Limitation
	end user station to select a higher or lower bitrate version of the streams;
7[C]	retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and
7[D]	send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections.
8	The system of claim 1, further comprising: a first web server configured to: receive at least one virtual timeline request over the one or more internet connections from the one or more end user stations to retrieve a virtual timeline; and send the virtual timeline to the requesting one of the end user stations over the one or more network connections.
9[A]	The system of claim 8, wherein the first web server is further configured to:
9[B]	receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the video;
9[C]	retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

Claim	Limitation
9[D]	send the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.
10	The system of claim 1, wherein the at least one virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.
11	The system of claim 1, wherein the virtual timeline defines a playlist for a user to view.
12	The system of claim 1, wherein the virtual timeline comprises a file that is configured to define a playlist for a user to view.
14 [PRE]	An end user station to stream a video over a network from a server for playback of the video, the content player device comprising:
14[A]	a processor;
14[B]	a digital processing apparatus memory device comprising non-transitory machine-readable instructions that, when executed, cause the processor to:
14[C][1]	establish one or more network connections between the end user station and the server,
14[C][2]	wherein the server is configured to access at least one of a plurality of groups of streamlets;

Claim	Limitation
14[C][3]	wherein the video is encoded at a plurality of different bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream,
14[C][4]	each of the low quality stream, the medium quality stream, and the high quality stream comprising a group of streamlets encoded at the same respective one of the different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;
14[C][5]	wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; and
14[C][6]	wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets encodes the same portion of the video at a different one of the different bitrates;
14[D]	select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;
14[E]	place at least one virtual timeline request for at least one virtual times based on the selected one of the he low quality stream, the medium quality stream, and the high quality stream; and
14[F]	receive the at least one virtual timeline
15[A]	The end user station of claim 14, wherein the non-transitory machine-readable instructions that, when executed, further cause the processor to:

Claim	Limitation
15[B]	place one or more streamlet requests to the server over the one or more network connections for the first streamlet of the selected stream;
15[C]	receive the requested first streamlet from the server via the one or more network connections wherein the one or more streamlet requests are based on the at least one virtual timeline; and provide the received first streamlet for playback of the video.
16	The end user station of claim 14, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.
17	The end user station of claim 16, wherein the first and second durations are different.
18	The end user station of claim 17, wherein the virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.
19	The end user station of claim 18, wherein the virtual timeline defines a playlist for a user to view.
20	The end user station of claim 14, wherein the video is a live event video.

Claim	Limitation
21	The end user station of claim 14, wherein the video includes archived content.
22[Pre]	A process executable by one or more servers to stream a video for playback by one or more end user stations, the process comprising:
22[A]	storing, by the one or more servers, one or more virtual timelines corresponding to a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream,
22[B]	wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets encoded at a respective one of a plurality of different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;
22[C]	wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and
22[D]	wherein the first streamlet of each of the groups of streamlets has the same first duration and encodes the same first portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different one of the different bitrates than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;
22[E]	receiving at least one virtual timeline request over one or more internet connections from the one or more end user stations to retrieve a virtual timeline correspond to the first streamlet storing the first portion of the video,

Claim	Limitation
22[F]	wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the video;
22[G]	retrieving from the storage device the requested virtual timeline for the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and
22[H]	sending the retrieved virtual timeline to the requesting one of the end user stations over the one or more network connections.
23[A]	The process of claim 22, further comprising:
23[B]	storing, by the one or more servers, a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream; and
23[C]	receiving at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the video,
23[D]	wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the video,
23[E]	retrieving from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and
23[F]	wending the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and

Claim	Limitation
	the high quality stream to the requesting one of the end user stations over the one or more network connections.
24	The process of claim 22, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.
25	The process of claim 22, wherein the first and second durations are different.
26	The process of claim 22, wherein the video is a live event video.
27	The process of claim 22, wherein the video includes archived content.
28[PRE]	A process executable by a content player device to stream a video over a network from a server for playback of the video by the content player device, the process comprising:
28[A]	establishing one or more network connections between the content player device and the server,
28[B]	wherein the server accesses a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream,
28[C]	wherein the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets

Claim	Limitation
	encoded at a respective one of a plurality of different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;
28[D]	wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and
28[E]	wherein the first streamlet of each of the groups of streamlets has the same first duration and encodes the same first portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the first streamlet of the low quality stream having a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream;
28[F]	selecting, by the content player device, a currently selected one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the video;
28[G]	placing a virtual time request over one or more internet connections from the one or more end user stations to retrieve at least one virtual timeline corresponding to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and
28[H]	receiving the requested virtual timeline from the server via the one or more network connections
29[A]	The process of claim 28 further comprising: placing a streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the video, wherein the streamlet request is based, at least in part, on the received virtual timeline;

Claim	Limitation
29[B]	receiving the requested streamlet from the server via the one or more network connections; and
29[C]	rendering, by the content player device, the received streamlet for playback of the video.

I. INTRODUCTION

WebGroup Czech Republic A.S. and NKL Associates, S.R.O. ("Petitioners") petition under 35 U.S.C. § 312 and 37 C.F.R. § 42 for *inter partes* review of claims 1-12 and 14-29 ("Challenged Claims") of U.S. Patent Number 10,951,680 ("the '680 Patent") (EX1001). The '680 Patent is assigned to DISH Technologies L.L.C. ("DISH") and Sling TV L.L.C. is an exclusive licensee.

The '680 Patent relates to multi-bitrate streaming of streaming content, including a server storing streamlets at different bitrates, where the client determines which bitrate version to select. The Grounds advanced here were not relied upon during prosecution of the '680 Patent.

II. STANDING

Petitioners certify the '680 Patent is available for *inter partes* review and they are not estopped.

III. GROUNDS

Petitioners request cancellation of claims 1-12 and 14-29 based on the grounds below.

Ground	Basis	Reference(s)	Claims
1	§ 103	Leaning	1-6, 8-12, and 14-29
2	§ 103	Leaning in view of Leighton	7
3	§ 103	Leaning in view of Reme	1-6, 8-12, and 14-29
4	§ 103	Leaning in view of SMIL 2.0	1-6, 8-12, and 14-29

These grounds are supported by the Declaration of Dr. Reza Rejaie (EX1003) and other supporting evidence in the Exhibit List.

IV. POSITA

A POSITA at the time of the ’680 Patent had at least a bachelor’s in electrical engineering, computer engineering, computer science or equivalent, and two years of experience with networking or media streaming. EX1003, ¶¶ 79-80. Petitioners’ expert exceeded this level of experience by the priority date. EX1003, ¶¶ 81-82.

V. BACKGROUND

A. Technical Background

Streaming techniques have been known for over 30 years, and documented since at least 1994. EX1003, ¶¶ 42-68. Generally speaking, in a streaming system, one or more servers communicate multimedia content over a network connection to a client device that displays the multimedia to the user as it is received or shortly thereafter. The quality, or bitrate, of the media affects how quickly it can be received and displayed to the user, with greater quality requiring an increased bandwidth, relative to a lower quality, to avoid noticeable delays (*e.g.*, buffering) in playback. In an ideal user experience, the streaming system streams the highest quality version of the multimedia to the client device without causing playback delays.

B. ’680 Patent

The ’680 Patent is directed to automatically adjusting the quality of streaming content (*e.g.*, adaptive-rate shifting).

One embodiment of the system of the ’680 Patent includes “a content server 102” and “end user 104” coupled by a data communications network (e.g., the Internet 106) wherein the “end user station 104” may be a personal computer, entertainment system, or a portable electronic device configured to present content. *Id.*, FIG. 1, 6:36-50. The ’680 Patent describes encoding the same content file 200 into at least three different “quality” streams:

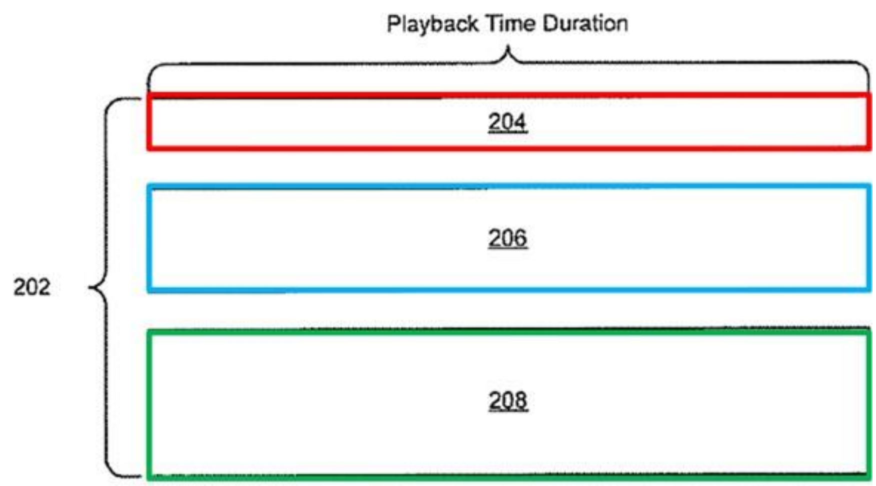


FIG. 2b

Id., FIG. 2b (annotated, low quality (red, 204), medium quality (blue, 206), and high quality (green, 208)), 7:25-36.

For example, “low quality stream 204 may be encoded and compressed to a bit rate of 100 kilobits per second (kbps), the medium quality stream 206 may be encoded and compressed to a bit rate of 200 kbps, and the high quality stream 208 may be encoded and compressed to 600 kbps.” *Id.*, 7:31-36.

Each stream (204, 206, and 208) is also "divided into a plurality of source streamlets 303" where "streamlet refers to any sized portion of the content file" and may be "an independent media object" where "streamlet 0 may have a time index of 00:00 representing the beginning of content playback, and streamlet 1 may have a time index of 00:002, and so on." *Id.*, 7:37-54; FIG. 3a (annotated).

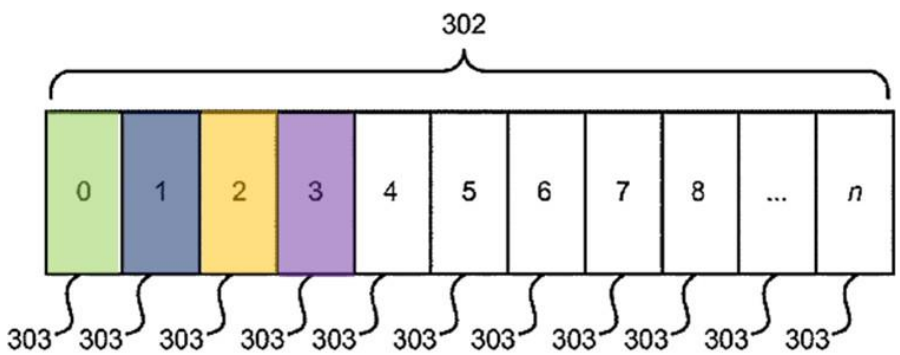


FIG. 3a

These streamlets form "sets" of streamlets 306 wherein a "set" is "a group of streamlets having identical time indices and durations but varying bitrates." *Id.*, 7:55-59; FIG. 3b (annotated, showing low, medium, and high quality streamlets in each "set" of streamlets" 306a-d).

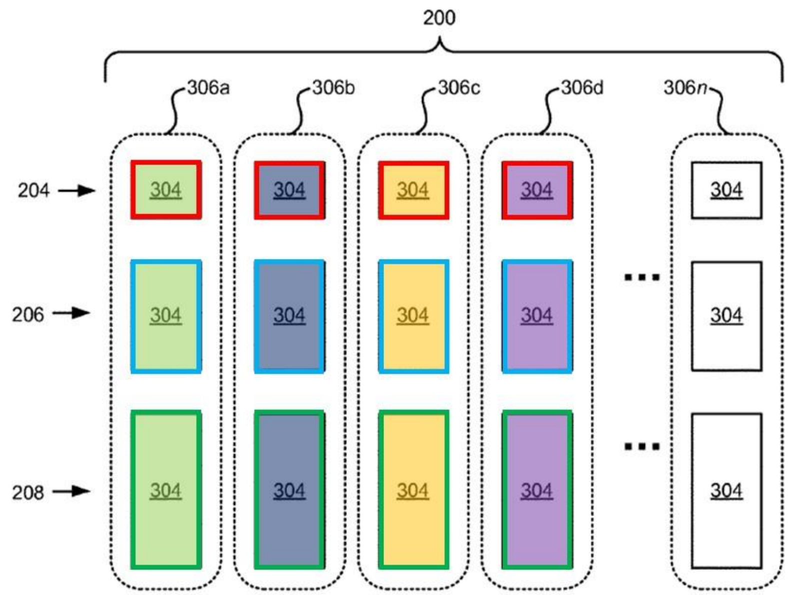


FIG. 3b

These sets of streamlets are stored in a streamlet database 408 and then a "client module 114 may request streamlets 304 using HTTP from the web server 116" or "a plurality of web servers 116." *Id.*, 9:1-4, 9:31-40. Further, the '680 Patent explains that the invention may be used to stream "live" content files on a short delay. *Id.*, 10:49-58.

To stream content, the client module's "agent controller module 702 is configured to select a quality level of streamlets to transmit to the viewer" and "requests lower or higher quality streams based upon continuous observation of time intervals between successive receive times of each requested streamlet." *Id.*, 13:8-26.

1. Prosecution History

The ’680 Patent’s application was filed on May 18, 2020, and claimed priority as a continuation of Application Serial No. 16/004,056, filed June 8, 2018. It further claimed priority from the following patent applications, which are a consecutive sequence of continuations: Application Serial No. 15/414,025, filed on January 24, 2017; Application Serial No. 14/719,122, filed on May 21, 2015; Application Serial No. 14/106,051, filed on December 13, 2013; Application Serial No. 13/617,114, filed on September 14, 2012; Application Serial No. 12/906,940, filed on October 18, 2010; and Application Serial No. 11/673,483, filed on February 9, 2007, which is a continuation-in-part of Application Serial No. 11/116,783. The ’680 Patent application further claimed the benefit of U.S. Provisional Application No. 60/566,831, filed April 30, 2004.

Petitioners assume for the purpose of this Petition a priority date of April 30, 2004 (the “Priority Date”) based on the claimed priority to the Provisional Application Serial No. 60/566,831, but reserves the right to contest this priority date (and priority chain) in other proceedings.

2. Claim Construction

For purposes of this Petition only Petitioners do not believe any terms require a formal construction. Every term carries its “ordinary and customary meaning,” which is “the meaning that the term would have to a person of ordinary skill in the art in question” at the time of the invention. *In re Translogic Tech., Inc.*, 504 F.3d

1249, 1257 (Fed. Cir. 2007) (citation omitted); EX1003, ¶¶ 83-84. However, Petitioners acknowledge that in the ITC Investigation involving Patent Owner and Respondent iFIT, Inc. asserting family members of the ’680 Patent, the Chief Administrative Law Judge (“ALJ”) construed “streamlet” as “any sized portion of the content file.” EX1009, 38-43. The ALJ also construed “low quality stream,” “medium quality stream,” and “high quality stream” under their plain and ordinary meaning. *Id*., 46-53. The ALJ also found that “end user station,” “content player device” required no construction. *Id*., 71.

Petitioners submit that “virtual timeline” does not require construction. The ’680 Patent describes a “virtual timeline” as “at least one quantum media extension 602” (“QMX”), and that a QMX describe[s] an entire content file.” *Id*. 12:35-36 (emphasis added). Thus, a “virtual timeline” may be “a file that is configured to define a playlist for a user to view” and that “may indicate that the publisher desires a user to watch a first show QMX 602a followed by QMX 602b and QMX 602c” such that “the publisher may define a broadcast schedule in a manner similar to a television station.” EX1001, 12:31-44, FIGs. 6a-6c. The ’680 Patent provides no other information regarding the nature of a “virtual timeline” envisioned by the Patent Owner. *Id*. Petitioners submit that “virtual timeline” should be construed with its ordinary meaning, which, according to the specification of the ’680 Patent, is “a playlist of entire content files.” EX1003, ¶¶ 101-102.

C. Prior Art

1. Leaning (EX1004)

Leaning was filed Dec. 14, 2001 and published April 1, 2004. Leaning is prior art under at least pre-AIA 35 U.S.C. § 102(b). While Leaning itself is not cited on the face of the '680 Patent, a patent application with a substantially identical specification does appear on the face of the '680 Patent. The ALJ did not consider Leaning in the ITC action.

Leaning is directed to techniques for streaming media content with different sets of files, called sub-files, that are successive temporal portions of the content being streamed. Each of the sets of files corresponds to a different quality version of the same content and a client can switch between quality versions each time it requests the sub-file for the next temporal portion of the content. Leaning teaches a client and server connected by a network wherein the client measures the actual data rate being received from the server and, based on that measurement, determines a "rate" directory (*e.g.*, a quality level) from which to request the next sequential sub-file in the media stream. EX1004, Abstract, 5:28-51. Leaning expressly teaches streams stored as three different quality versions (*e.g.*, high, medium, and low) with each version divided into sub-files that each represent the same portion of the media across the different versions. EX1004, 5:28-51, 6:1-33.

	Directory	Subdirectory	Filename
			000003.bin
			.
			.
			.
			000134.bin
Low Quality Stream	mp3_bwv565	018k_11_s	000000.bin
			000001.bin
			000002.bin
			000003.bin
			.
			.
			.
			000134.bin
Medium Quality Stream	mp3_bwv565	024k_11_s	000000.bin
			000001.bin
			000002.bin
			000003.bin
			.
			.
			.
			000134.bin
High Quality Stream	mp3_bwv565	032k_11_s	000000.bin
			000001.bin
			000002.bin
			000003.bin
			.
			.
			.
			000134.bin

EX1004, 6:1-29 (annotated, showing a portion of a table in Leaning with each of the subdirectories (indicated by the red, yellow, and green boxes) containing the sub-files for three different versions of the same audio media (mp3_bwv565, encoded at 18, 24, and 32 kbps) with each subdirectory containing sub-files with the same names (*e.g.*, 000000.bin-000134.bin) in the same order).

2. Reme (EX1005)

Reme was filed on September 15, 2003 and published on April 8, 2004. Reme is prior art under at least pre-AIA 35 U.S.C. § 102(e). Reme is not cited on the face of the ’680 Patent nor did the ALJ consider it in the ITC action.

Reme is directed to a technique for streaming content to a user via the internet and configured to switch among a plurality of pre-encoded versions of the content, where each version corresponds to a different encoding rate and hence to a different quality (*e.g.*, encoding video rates at 30 kbps, 300 kbps, and 5 Mbps). EX1005, Abstract, 3:15-17, 5:1-12. Reme discloses that its streaming system allows for automatically “selecting the version of the content which encoding rate best matches” the transmission rate of the network connection between the client and the server and that it may “switch from one version to another in order to take into account the modifications of the state of the transmission network.” EX1005, 5:7-12.

3. Leighton (EX1007)

Leighton was filed on June 28, 2000, and issued on April 22, 2003. Leighton is prior art under at least pre-AIA 35 U.S.C. § 102(b). Leighton is not cited on the face of the ’680 Patent nor did the ALJ consider it in the ITC action.

Leighton is directed to a network architecture for hosting and distributing content to clients across the globe. EX1007, Abstract. Leighton discloses a network with “a set of servers operating in a distributed manner,” including several “hosting servers" that are used to transmit content to clients using those hosting servers that are near the client machines. EX1007, Abstract. Additionally, Leighton's architecture includes a web server 12 that is "one of a plurality of [web] servers which are accessible by clients." EX1007, 5:2-4. In processing a client's HTTP request for content, Leighton's system "determine[s] where in the network [(*e.g.*, internet)] a user is located, and then [directs] the user to a … server 40 that is close-by." EX1007, 9:46-50.

4. SMIL 2.0 (EX1006)

Synchronized Multimedia Integration Language 2.0 ("SMIL 2.0") is an XML markup language for streaming presentation of multimedia and was publicly accessible on the World Wide Web Consortium's ("W3C") website at least as of November 9, 2001. *See* EX1006 (SMIL 2.0); EX1003, ¶¶ 93-94. Thus, SMIL 2.0 is prior art under at least (pre-AIA) 35 U.S.C. § 102(b). SMIL 2.0 is not cited on the face of the '680 Patent nor did the ALJ consider it in the ITC Investigation. Petitioners note that during prosecution of the '680 Patent, Applicants submitted an article entitled "Mobile Streaming Media CDN Enabled by Dynamic SMIL" by Yoshimura. Applicants did not submit the actual SMIL standard, and the article submitted by Applicants does not mention important elements of SMIL such as "<seq>" and "<switch>" elements or "systemBitrate" teaching three content quality levels (high, medium, and low) and switching to a particular quality level based upon measured system bitrate, which are pertinent to this Petition.

SMIL 2.0 discusses these elements. SMIL 2.0 discloses that, using the "<seq>" element, a client can request and play media elements (*e.g.*, successive temporal portions of a video) sequentially. EX1006, 123-126. SMIL 2.0 further teaches that a client can select different encoded files based on the client's available bandwidth. EX1006, 61-63. Specifically, the "<switch>" element can be used to list different media files (*e.g.*, videos), and depending on the client's bandwidth, the most suitable video file can be selected for streaming. *Id*.

VI. GROUNDS OF REJECTION

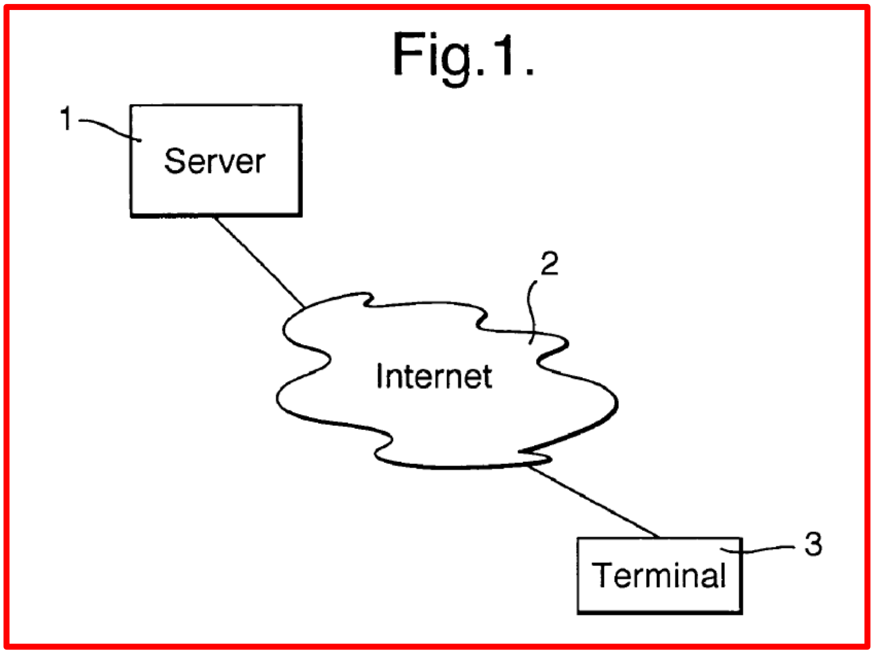
A. <u>Ground 1</u>: Claims 1-6, 8-12, and 14-29 are obvious over Leaning

1. Claim 1

(a) 1[Pre]: A system for adaptive-rate content streaming of video playable on one or more end user stations over the Internet, the system comprising:

Leaning (EX1004) discloses a streaming system, comprising a video that is streamed by dividing it into a sequence of sub-files, each of which is independently requested by a client. EX1004, Abstract. Specifically, the servers disclosed in Leaning store multiple "versions of the [content], recorded at different … data rates" (*e.g.*, bitrates) and a client terminal "is able to switch automatically between them" by using an "index file" that provides a list of the data rates available and the corresponding set of sub-files for each different rate. EX1004, 4:3-5:51 (emphasis added). And, while Leaning describes an exemplary system for streaming audio, Leaning also explains that "the system may be used to convey video signals instead of, ***or in addition to***, ***audio signals***." EX1004, 2:8-10 (emphasis added). The client in Leaning further makes repeated assessments of the network and determines whether to adjust the data rate, by choosing the next sub-file from the appropriate rate directory for a higher (or lower) data rate version of the content, based on those assessments. EX1004, 6:59-7:14.

Additionally, Leaning explains that "it is not necessary that the entire audio or video sequence should have been encoded—or even exist—before delivery is commenced. Thus a computer could be provided to receive a ***live feed***, to code it using the chosen coding scheme, and generate the sub-files 'on the fly' and upload them to the server, so that, once a few sub-files are present on the server, delivery may commence." EX1004, 14:66-15:5 (emphasis added); 15:17-22 ("The same [delivery] system can be used for a live audio (or video) feed.").



Accordingly, the system in Leaning's Figure 1 above is ***a system for adaptive-rate content streaming of live event video playable on one or more end user stations over the Internet***. EX1003, ¶¶ 96-99.

(b) 1[A]: At least one processor executing non-transitory executable instructions for generating at least one virtual timeline corresponding to the video;

Leaning discloses that the server (*e.g.*, server 1) includes a processor that is configured to execute non-transitory executable instructions (*e.g.*, a program) to generate a virtual timeline (*e.g.*, a menu of available recordings or "menu.htm"). EX1004, 3:38-41 ("It is also convenient that the web server stores one or more ***(html) menu pages (e.g. menu.htm)*** containing a ***list of recordings available, with hyperlinks to the corresponding link pages***.") (emphasis added), 4:3-22 ("1. The user uses the browser to ***retrieve and display the menu page menu.htm*** from the server 1. 2. The ***user selects one of the hyperlinks*** within the menu page which causes the browser to retrieve from the server, and display, the link page for the desired recording") (emphases added), 7:40-8:65 (flowchart showing ***request and retrieval*** of virtual timeline including "[Terminal] ***Request*** http:\\server1.com/***menu.htm*** . . . [Server] ***Send*** http:\\server1.com/***menu.htm***") (emphasis added); EX1003, ¶¶ 100-101.

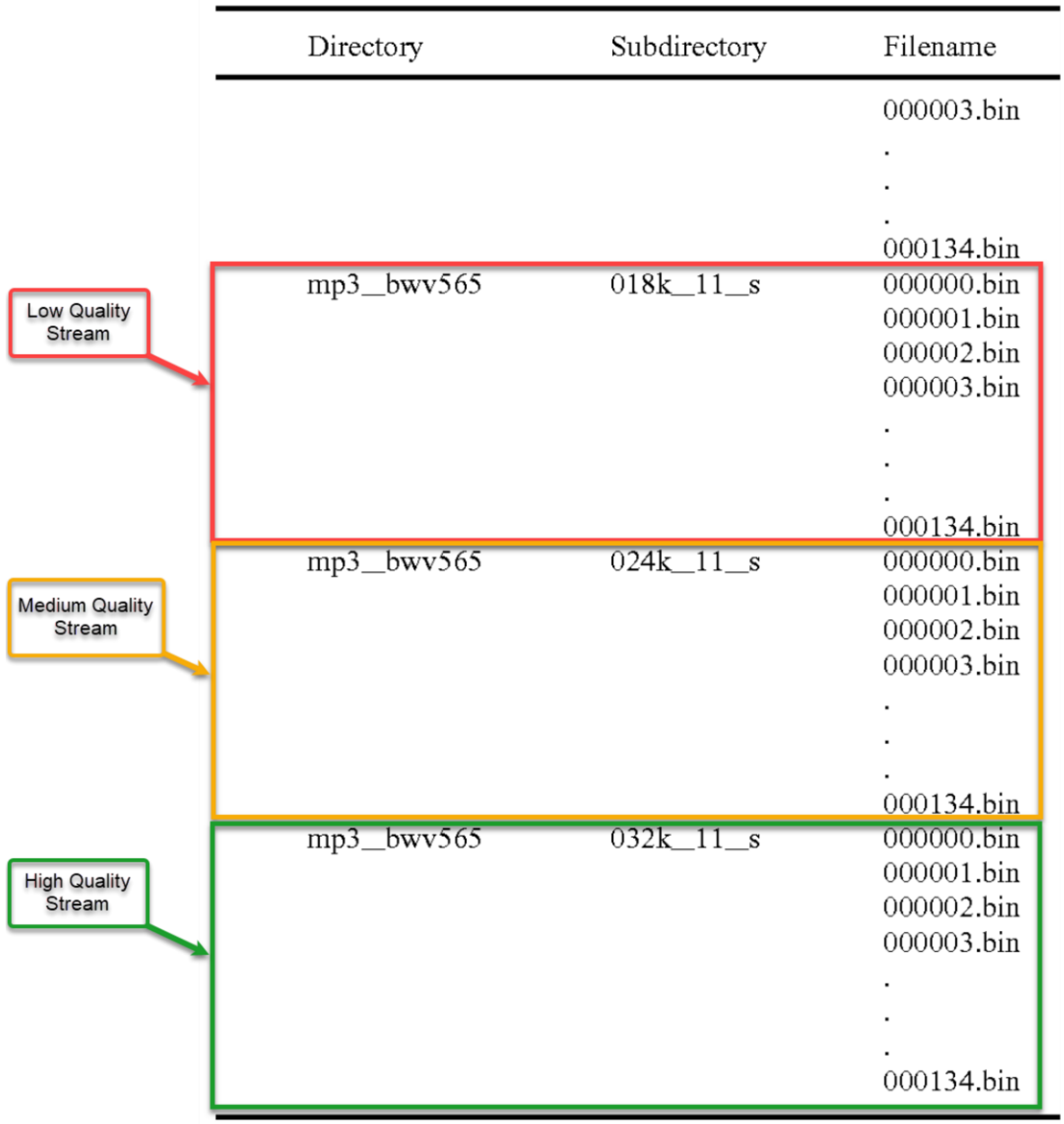
According to the specification of the '680 Patent, a "virtual timeline" may comprise a file (such as Leaning's "menu.htm") configured to present a playlist to the end user that is sent by the server to a client terminal (*e.g.*, an end user station). *See* EX1001, 12:33-38 ("In one embodiment, the virtual timeline 600 comprises at least one quantum media extension 602. The quantum media extension (hereinafter 'QMX') 602 describes an entire content file 200. Therefore, the virtual timeline (hereinafter 'VT') 600 may comprise a file that is configured to define a playlist for a user to view."); EX1003, ¶ 102.

Additionally, Leaning teaches use of a second "virtual timeline," referred to as an "index file," that "provide[s] a list of the data rates that are available" and the sequentially-named files that correspond to each data rate. EX1004, 5:49-6:29; EX1003, ¶¶ 103-105. Leaning's index file, therefore, is also a "virtual timeline" because it is a content file that is configured to define a playlist for a user to view. Leaning expressly discloses that the client terminal's player program requests from the server "the index file, and stores locally a list of available data rates for future reference[,]" which allows it to "begin[] to request" sub-files encoded a particular data rate and play them out automatically and sequentially. EX1004, 6:30-58.

Leaning further describes arranging the sub-files requested from the server using the virtual timeline (*e.g.*, the "menu.htm" and/or "index file") because Leaning describes arranging the sub-files for playback in sequential order. See EX1004, Abstract ("Delivery of recorded audio or video material over a telecommunications link from a server is accomplished by dividing the material into ***a sequence of sub-files*** each of which is independently requested by the terminal, which thereby has control of the rate of delivery."), 5:14-16 ("The sub-file naming convention used here, of a simple fixed length ***sequence of numbers*** starting with zero, is preferred as it is simple to implement") (emphases added); EX1003, ¶¶ 100-105.

(c) 1[B][1]: Wherein the video encoded at a plurality of different bitrates creating a plurality of streams including a low quality stream, a medium quality stream, and a high quality stream,

Leaning describes that recorded content may be digitally encoded at a plurality of different bitrates (different compression rates) creating a plurality of streams including a low quality stream (*e.g.*, 18 kbit/s), a medium quality stream (*e.g.*, 24 kbit/s), and a high quality stream (*e.g.*, 32 kbit/s). EX1004, 5:28-32, 5:41-48 (disclosing that the server stores multiple versions of the recording, "recorded at different compression rates ... loaded onto the server in separate directories corresponding to the different rate, as in the following example structure, where '008k', '024k' in the directory name indicates a rate of 8 kbit/s or 24 kbit/s and so on."). As shown in the annotated excerpt below, Leaning discloses at least a low quality stream encoded at 18kbps ("018k_11_s" (indicated in red)), a medium quality stream encoded at 24kbps ("024k_11_s" (indicated in orange)), and a high quality stream encoded at 32 kbps ("032k_11_s" (indicated in green)).



Directory	Subdirectory	Filename
		000003.bin . . . 000134.bin
mp3__bwv565	018k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin
mp3__bwv565	024k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin
mp3__bwv565	032k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin

As noted previously, although the annotated excerpt above specifically describes streaming audio content (*i.e.*, an mp3 stream), Leaning's system is not limited to audio streams and the teachings described with reference to an audio stream are equally applicable to streams that include video. EX1004, 2:9-11, 12:48-57 ("The same principle may be applied to the delivery of video recordings. … the file is a video file (e.g. in H.261 or MPEG format) and the player program incorporates a video decoder. The manner of partitioning the file into sub-files is unchanged. As in the audio case, there may be two ***or more*** recordings ***corresponding to different data rates***, selected by the control mechanism already described.") (emphasis added), claim 4; EX1003, ¶¶ 106-108.

(d) 1[B][2]: The low quality stream, the medium quality stream, and the high quality stream each comprising a group of streamlets encoded at a respective one of the plurality of different bitrates,

Leaning discloses the low quality stream, the medium quality stream, and the high quality stream each comprise a group of streamlets (called a set of "sub-files" in Leaning) encoded at a respective one of the plurality of different bitrates (*e.g.*, the sub-files of each set are encoded, or compressed, at a bitrate of¸ *e.g.*, 18kbps, 24kbps, or 32kbps, respectively). EX1003, ¶¶ 109-110. The '680 Patent defines "a streamlet" as "any sized portion of a portion of the content file" (EX1001, 7:39-40). Similarly, Leaning's "sub-files" are portions of "the original file containing the whole recording." EX1004, 3:13-19, Abstract (content is "divid[ed] [] into a sequence of sub-files").

As shown in the annotated portion of Leaning below, each of the directories (*e.g.*, for each the different quality versions of the stream) contains a group of streamlets, or a set of sub-files (*e.g.*, "000000.bin" to "000134.bin"), encoded at a bitrate that corresponds to the directory into which they are loaded.

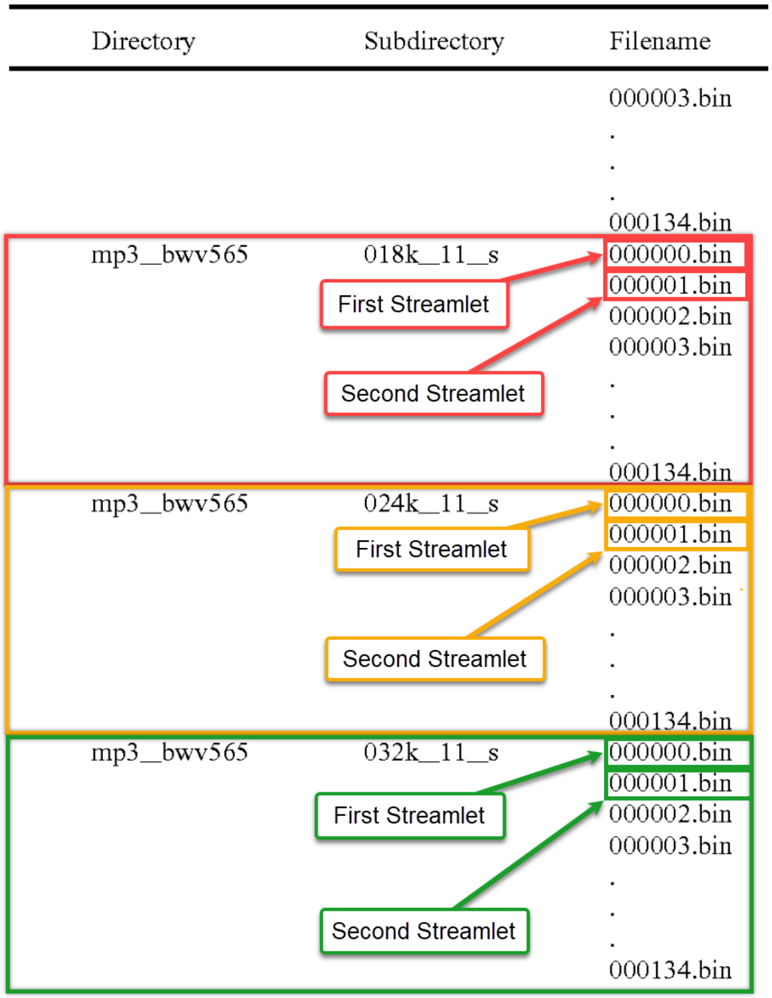
Directory	Subdirectory	Filename
		000003.bin
		.
		.
		.
		000134.bin
mp3__bwv565	018k__11__s	000000.bin
		000001.bin
	Low Quality Group of Streamlets	000002.bin
		000003.bin
		.
		.
		.
		000134.bin
mp3__bwv565	024k__11__s	000000.bin
		000001.bin
	Medium Quality Group of Streamlets	000002.bin
		000003.bin
		.
		.
		.
		000134.bin
mp3__bwv565	032k__11__s	000000.bin
		000001.bin
	High Quality Group of Streamlets	000002.bin
		000003.bin
		.
		.
		.
		000134.bin

EX1004, 6:1-29 (annotated).

The above sets of "sub-files" are created by "encoding the same [media] file several times at different [bit]rates" and portioning the encoded files into sets of sub-files that are "loaded onto the server in [the] separate directories corresponding to the different [bit] rate[s]" shown in the annotated excerpt above. EX1004, 5:41-48, 2:57-3:19 ("the file is ***divided into smaller files*** before being stored on the server 1. … [Preferably,] each of these smaller files is of a size corresponding to a fixed playing time, perhaps four seconds") (emphasis added); *see also* EX1004, claims 1 and 11 and EX1003, ¶¶ 109-111.

(e) 1[B][3]: Each group of streamlets comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;

Leaning discloses the claimed groups of streamlets wherein each group of streamlets comprises at least first (*e.g.*, 000000.bin) and second (*e.g.*, 000001.bin) streamlets corresponding to a portion of the live event video. The specific sub-files that correspond to the first and second streamlets are shown in the annotated portion of Leaning below.



EX1004, 6:1-29 (annotated). Further, the first and second sub-files in Leaning each correspond to a portion of the live event video that is portioned to create each set of sub-files. EX1004, 2:57-3:19 (the original video file "is divided into smaller files … [preferably] of a size ***corresponding to a fixed playing time***, perhaps four seconds. ... [Thus, the original video file is] divided into 135 smaller files ***each representing four seconds' playing time***.") (emphasis added). Indeed, each of the ".bin" files correspond to a fixed playing time and an order in the sequence indicated by the file names of the sub-files. 2:57-3:19 (the sub-files of the original video file "are given file names … indicative of their sequence in the original file, for example: ***000000.bin*** [and] ***000001.bin*** … [Where the term] 'sub-files' is used here to distinguish them from the original file containing the whole recording: it should however be emphasized that, as far as the server is concerned, each 'sub-file' is just a file like any other file.") (emphasis added); EX1003, ¶¶ 112-115.

(f) 1[C]: Wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bitrate of no less than 600 kbps; and

Leaning renders obvious this limitation.[1] As explained above, Leaning teaches multiple streams encoded at different bitrates including one example with at least three streams encoded at 16, 24, and 32 kbit/s, respectively. *See* 1[PRE]-1[B][3]. Further, Leaning's example bitrates of 16, 24, and 32 kbit/s, respectively, are for audio encodings—which a POSITA would have understood generally require less data to stream than video encodings and thus typically have lower bitrates. EX1003, ¶¶ 117-118. As described previously, Leaning also teaches that alternative embodiments could include video encodings. EX1004, 2:9-11. Additionally, the '680 Patent admits that content (*e.g.*, video encodings) "may be compressed using ***standard or proprietary encoding schemes***[]" including "DivX, Windows Media

[1] To the extent Patent Owner argues that a stream encoded at a bitrate of no less than 600 kbps is not within the ordinary knowledge of a POSITA, ground 3 combines Reme (EX1005) with Leaning for its express teaching of this limitation.

Ivdeo, Quicktime Sorenson 3, On2, OGG Vorbis, MP3 or Quicktime 6.5/MPEG-4 encoded content." EX1001, 7:19-24 (emphasis added). While Leaning discloses further encoding schemes known to a POSITA at the presumed time of invention, the H.261 and MPEG video coding formats, which Leaning teaches can be used to encode the video streams transmitted as streamlets (*e.g.*, sub-files) to an end user. EX1004, 12:48-54. Further, it was well known in the art to use at least the H.261 video format to stream video content at a bitrate equal to, or greater than, 600 kbps. EX1003, ¶¶ 118-119[2].

A POSITA would have understood that a user streaming video encodings at the time of the alleged priority date would generally prefer higher bitrate encoding if supported by the user's network requirements, such as 1,920 kbps, to enjoy high quality viewing. EX1004, 13:42-14:64. (describing a video file ("mpg_name") that is encoded at 96 kbps and at 128 kbps and partitioned into two corresponding sets of sub-files (stored in directories "mpg_name/096k_x1/" and "mpg_name/0128k_x1/", respectively)); EX1003, ¶¶ 119-120.

A POSITA would have been motivated to encode a video embodiment of Leaning (including the corresponding sets of sub-files) into each of a low quality, medium quality, and high quality level bitrate stream (low at 30 kbps, medium at

[2] The references cited to corroborate the opinions of Dr. Rejaie (*e.g.*, EX1014) are prior art printed publications, as testified to by Dr. Rejaie. EX1003, ¶ 118-120 and 255.

300 kbps, and high at 1,920 kbps), using the H.261 video codec disclosed by Leaning, because it would enable client terminals with a wide variety of network bandwidth requirements to stream the video and would have allowed for those with user terminals with high bandwidth connections to stream at 1,920 kbps and enjoy higher resolution viewing and a better user experience. EX1003, ¶ 120. A POSITA would have had a reasonable expectation of success with such an embodiment of Leaning's disclosure because Leaning expressly teaches the use of the H.261 video codec, which was known to support bitrates of up to 1,920 kbps at the presumed time of invention and because simply encoding one of the video streams at a particular bitrate, such as the 1,920 kbps of the H.261 video format, would have been a simple and straightforward implementation of Leaning's teachings for a POSITA to make when switching from the audio to video embodiments of Leaning. EX1003, ¶¶ 116-120.

(g) 1[D][1]: Wherein the first streamlet of each of the groups of streamlets has the same first duration and encodes the same first portion of the video in each of the low quality stream, the medium quality stream, and the high quality stream, and

Leaning's first sub-files (*e.g.*, 000000.bin) in each set "correspond[s] to a ***fixed playing time***" so that they "represent the same fixed time period" within the sequence of the live event video file that is encoded and portioned to create the sets of sub-files. EX1004, 2:57-3:19, 10:1-4 ("To arrange that every sub-file should represent the same fixed time period (e.g. the 4 seconds mentioned above) is not the only way of achieving this, but it is certainly the most convenient.")

For example, a media file "mp3_BWV565," which has a duration of 9 minutes, can be "divided into 135 smaller files ***each representing four seconds' playing time***." *Id*. at 2:62-64 (emphasis added). Indeed, each of the ".bin" files created from this media file would correspond to a fixed playing time and an order in the sequence indicated by the file names of the sub-files. *Id*. at 2:57-3:19 (The sub-files "are given file names … indicative of their sequence in the original file, for example: ***000000***.bin ***000001***.bin ***000002***.bin" and so on.) (emphases added); EX1003, ¶ 123. Accordingly, for a video file encoded and partitioned into several sets of sub-files, the first sub-file (000000.bin) in each of the sets of sub-files represents the same ***fixed playing time*** (*e.g*., the first four seconds) of the content, with the second sub-file (000001.bin) in each of the sets representing the next fixed segment (*e.g.,* the second four seconds) of the content.

Additionally, to facilitate rate switching, "it is, if not actually essential, ***highly desirable*** that the sub-file boundaries ***are the same for each rate***, so that the first sub-file received for a new rate ***continues from the same point in the recording that the last sub-file at the old rate ended***." EX1004, 9:64-10:4 (emphases added); EX1003, ¶¶ 121-124.

(h) 1[D][2]: wherein the first streamlet of the low quality stream encodes the same first portion of the video at a different bitrate than the first streamlet of the high quality stream and the first streamlet of the medium quality stream.

Leaning's first sub-files (*e.g*., "000000.bin") each correspond to a set of sub-files that are each encoded with a different bitrate or stream quality, with each sub-file corresponding to the same portion of the video. EX1004, 2:57-3:19.

For example, a media file "mp3_BWV565," which has a duration of 9 minutes, can be "divided into 135 smaller files ***each representing four seconds' playing time***." *Id*. at 2:62-64 (emphasis added), 5:41-51 (describing the same set of 135 sub-files (*e.g.*, 000000.bin (first sub-file), 000001.bin (second sub-file), 000002.bin. . . 000134.bin) each set of sub-files encoded at a different bitrate including low, medium, and high quality (*e.g.*, 18 kbps, 24 kbps, and 32 kbps) and listed in an index file and saved in the directory appropriate for their bitrates (*e.g.*, "018_11_s", "024k_11_s", and "032k_11_s")).

Directory	Subdirectory	Filename
		000003.bin . . . 000134.bin
mp3__bwv565	018k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin
mp3__bwv565	024k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin
mp3__bwv565	032k__11__s	000000.bin 000001.bin 000002.bin 000003.bin . . . 000134.bin

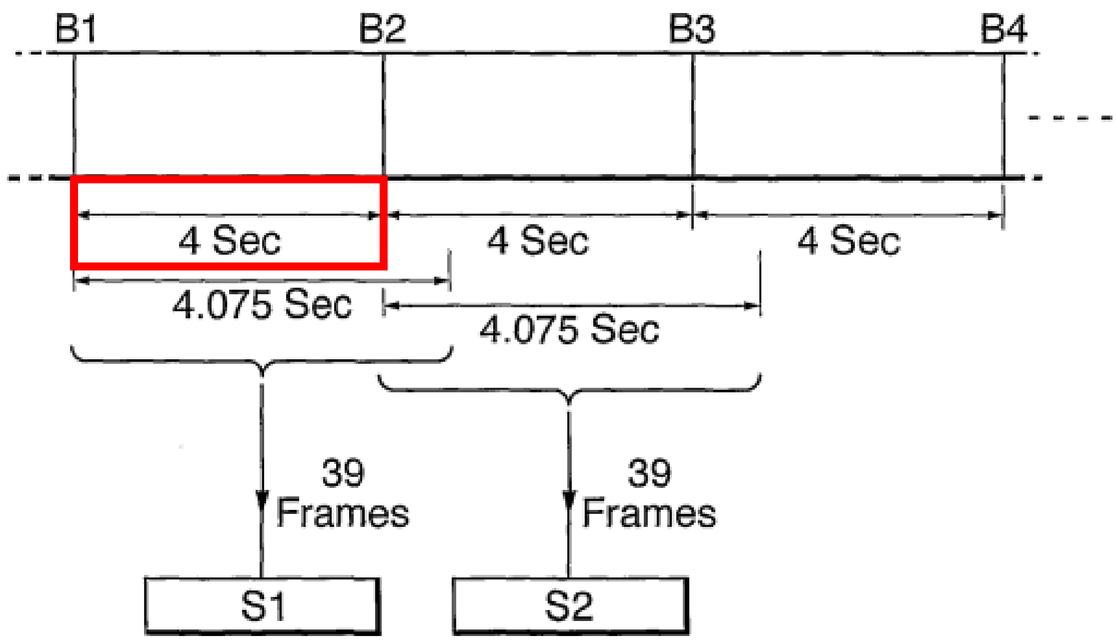
EX1004, 6:1-29 (annotated, showing each of the directories containing the different versions of the same recording (encoded at 18, 24, and 32 kbit/s) and each containing the same set of subfiles (000000.bin-000134.bin) in the same order).

Indeed, each of the ".bin" files created from this media file would correspond to a fixed playing time and an order in the sequence indicated by the file names of the sub-files. *Id*. at 2:57-3:19 (The sub-files "are given file names … indicative of their sequence in the original file, for example: 000000.bin 000001.bin 000002.bin" and so on.); EX1003, ¶¶ 125-128. Accordingly, for a video file encoded and partitioned into several sets of sub-files, the first sub-file (000000.bin) in each of the sets of sub-files represents the same ***fixed playing time*** (*e.g.*, the first four seconds) of the content, with the second sub-file (000001.bin) in each of the sets representing the next fixed segment (*e.g.*, the second four seconds) of the content.

Additionally, to facilitate rate switching, "it is, if not actually essential, ***highly desirable*** that the sub-file boundaries ***are the same for each rate***, so that the first sub-file received for a new rate ***continues from the same point in the recording that the last sub-file at the old rate ended***." EX1004, 9:64-10:4 (emphases added), FIG. 4 (annotated); EX1003, ¶¶ 129-130.

.

Fig.4.



2. **Claim 2: The system of claim 1, wherein the processor is further for generating a plurality of virtual timelines wherein each virtual timeline corresponds to each of the low quality stream, the medium quality stream, and the high quality stream.**

Leaning teaches that the virtual timeline (*e.g.*, "menu.htm") includes a list of available recordings (EX1004, 3:38-41, 4:5-8) each of which has multiple quality versions (*e.g.*, low, medium, and high) with playlists of subfiles as described above, which corresponds to each stream. EX1003, ¶¶ 131-134.

Leaning teaches use of a second "virtual timeline," referred to as an "index file," that "provide[s] a list of the data rates that are available" and the sequentially-named files that correspond to each data rate. EX1004, 5:49-6:30. Leaning's index file, therefore, is also a "virtual timeline" because it is a content file that is configured to define a playlist for a user to view.

Leaning discloses that the end user device's player program requests from the server "the index file, and stores locally a list of available data rates for future reference[,]" which allows it to "begin[] to request" sub-files encoded a particular data rate and play them out automatically and sequentially. EX1004, 6:30-58. Leaning goes on to disclose arranging the streamlets (sub-files) requested from the server using the virtual timeline (*e.g.*, the "menu.htm" and/or "index file") by arranging the sub-files for playback in sequential order. See EX1001, Abstract ("Delivery of recorded audio or video material over a telecommunications link from a server is accomplished by dividing the material into ***a sequence of sub-files*** each of which is independently requested by the terminal"), 5:14-16 ("The sub-file naming convention used here, of a simple fixed length ***sequence of numbers*** starting with zero, is preferred as it is simple to implement"). (emphases added).

Further, Leaning discloses "a person preparing the sub-files for loading onto the server." EX1004, 3:32-41. Leaning discloses similar preparation of the index.htm file. *See* 5:49-51 (the person preparing sub-files "also creates an index file (e.g. index.htm) the primary purpose of which is to provide a list of the data rates that are available."), 10:28-32 ("The actual process of converting the sub-files can be implemented by means of a computer programmed in accordance with the criteria discussed. Probably it will be convenient to do this on a separate computer, from which the sub-files can be uploaded to the server.”).

3. Claim 3: The system of claim 1, wherein the video is a live event video.

Leaning discloses that the video encoded by the sub-files includes videos captured and transmitted in real time or near real time (*e.g.*, “on the fly”). EX1004, 6:46-47 (“‘Mode’ indicates ‘recorded’ (as here) ***or ‘live’*** (to be discussed below).”) (emphasis added), 14:65-15:5 (disclosing that “it is not necessary that the entire audio or video sequence should have been encoded—or even exist—before delivery is commenced” and “a computer could be provided to receive a live feed, to code it using the chosen coding scheme, and generate the sub-files ‘on the fly’ and upload them to the server, so that, ***once a few sub-files are present on the server, delivery may commence***.”) (emphasis added). Leaning further discloses that a live feed, as it uses that term, “is in a sense still ‘recorded’ – the difference being primarily that delivery and replay commence before recording has finished.” EX1004, 15:17-22; EX1003, ¶¶ 135-136.

4. Claim 4: The system of claim 1, wherein the video includes archived content.

Leaning discloses that the video encoded by the sub-files may be archived content (recordings). EX1004, Abstract (describing the delivery “of ***recorded*** audio or ***video material*** over a telecommunications link for a server”), 5:28-29 (“the server stores two or more versions of ***the recording, recorded at*** different compression rates") (emphasis added), 6:46-47 ("'Mode' indicates '***recorded'*** (as here) or 'live' (to be discussed below).") (emphasis added); EX1003, ¶ 137.

5. Claim 5: The system of claim 1, wherein the second streamlet of each of the groups of streamlets each has the same second duration and corresponds to the same second portion of the video in the low quality stream, the medium quality stream, and the high quality stream, the second streamlet of the low quality stream having the same bitrate as the first streamlet of the low quality stream.

Leaning discloses that the second streamlet (*e.g*., sub-file "000001.bin") of each of the groups of streamlets (sets of sub-files encoded at different bitrates) has the same second duration (*e.g*., 4 seconds) and encodes the same second portion of the live event video in each of the low quality stream, medium quality stream, and high quality stream, and wherein the second streamlet of the low quality stream encodes the same second portion of the video at a lower bitrate than the second streamlet of the high quality stream and the second streamlet of the medium quality stream. *See* claim 1[B][2]-[D][1].

Further, Leaning discloses that the second streamlet (*e.g*., sub-file "000001.bin") of the low quality stream has the same bitrate as the first streamlet (*e.g*., sub-file "000000.bin") of the low quality stream because every sub-file in each quality level has the same (continuous) bitrate. EX1004, 5:28-31, 5:43-48 ("…encoding the same PCM file several times at different rates. He then partitions each source file into sub-files, as before. These can be loaded onto the server in separate directories corresponding to the different rate, as in the following example structure, where '008k', '024k' in the directory name indicates a rate of 8 kbit/s or 24 kbit/s and so on."); EX1003, ¶¶ 138-140.

6. Claim 6: The system of claim 5, wherein the first and second durations are different.

Leaning discloses that, while it is preferred that all sub-files are the same duration, Leaning also teaches that making each sub-file (including the first and second sub-files) the same duration "is not the only way of achieving [that the first sub-file received for a new rate continues from the same point in the recording that the last sub-file at the old rate ended.] " EX1004, 9:64-10:4. For example, each sub-file "can be a fixed number of bits, or a fixed playing time length (or neither of these)." *Id.* at 9:57-59. Leaning further teaches that, due to compression, sub-files with fixed playing time lengths may "be of different sizes in in terms of the number of bits they actually contain." *Id.* at 2:60-62. Additionally, Leaning acknowledges that in some embodiments "the playing duration of the sub-files does vary slightly." *Id*. at 10:7-8.

Accordingly, Leaning discloses that sub-files (*e.g.*, 000000.bin and 000001.bin) that have the same size (*i.e.*, a fixed number of bits) may have different durations because the sub-files would not also have a fixed time length; for example, sub-file 1 (000000.bin) may be a different time duration than sub-file 2 (000001.bin.). EX1003, ¶¶ 141-143.

7. Claim 8: The system of claim 1, further comprising: a first web server configured to: receive at least one virtual timeline request over the one or more internet connections from the one or more end user stations to retrieve a virtual timeline; and send the virtual timeline to the requesting one of the end user stations over the one or more network connections.

Leaning discloses that the server is further configured to: receive a request for a virtual timeline (*e.g.*, a request for a menu of available recordings or "menu.htm") from the one or more end user stations (*e.g.*, the client terminal) to retrieve a virtual timeline (*e.g.*, the "menu.htm" file) and, in response, send the virtual timeline (*e.g.*, the "menu.htm" file corresponding to the request) to the requesting end user station(s) using the internet connection. EX1003, ¶ 144. Leaning discloses the web server receiving a virtual timeline request because it explains that "the web server stores one or more ***(html) menu pages (e.g. menu.htm)*** containing a ***list of recordings available, with hyperlinks to the corresponding link pages***" followed by a flowchart showing the request and retrieval of the virtual timeline. *See* EX1004, 3:38-41, 4:3-22, 7:40-8:65 (flowchart showing request and retrieval of virtual timeline including "[Terminal] ***Request*** http:\\server1.com/***menu.htm*** . . . [Server] ***Send*** http:\\server1.com/***menu.htm***") (emphases added); EX1003, ¶ 144.

According to the specification of the '680 Patent, a "virtual timeline" may comprise a file (such as Leaning's "menu.htm") configured to present a playlist to the end user that is sent by the server to a client terminal (*e.g.*, an end user station). *See* EX1001, 12:36-38 ("In one embodiment, the virtual timeline 600 comprises at least one quantum media extension 602. The quantum media extension (hereinafter 'QMX') 602 describes an entire content file 200. Therefore, the virtual timeline (hereinafter 'VT') 600 may comprise a file that is configured to define a playlist for a user to view."); EX1003, ¶ 144.

Additionally, Leaning teaches use of a second "virtual timeline," referred to as an "index file," that "provide[s] a list of the data rates that are available" and the sequentially-named files that correspond to each data rate. EX1004, 5:49-6:29; EX1003, ¶ 144. Leaning's index file, therefore, is also a "virtual timeline" because it is a content file that is configured to define a playlist for a user to view. Leaning expressly discloses that the client terminal's player program requests from the server "the index file, and stores locally a list of available data rates for future reference[,]" which allows it to "begin[] to request" sub-files encoded a particular data rate and play them out automatically and sequentially. EX1004, 6:30-58.

Leaning further describes arranging the sub-files requested from the server using the virtual timeline (*e.g.*, the "menu.htm" and/or "index file") because Leaning describes arranging the sub-files for playback in sequential order. See EX1001, Abstract ("Delivery of recorded audio or video material over a telecommunications link from a server is accomplished by dividing the material into ***a sequence of sub-***

files each of which is independently requested by the terminal, which thereby has control of the rate of delivery."), 5:14-16; EX1003, ¶¶ 104, 133, and 144.

8. Claim 9:

(a) 9[A]: The system of claim 8, wherein the first web server is further configured to:

Leaning discloses a first web server (*e.g.*, server 1). *See* EX1004, 2:40-49 (explaining that the server of Leaning is "merely an ordinary 'web server'"), 3:19-4:3 (describing loading subfiles "onto a web server"); EX1003, ¶ 145.

(b) 9[B]: Receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing the first portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the video;

Leaning teaches that the client terminal is configured to request at least one streamlet (*e.g.*, sub-file) over an Internet connection with one of the one or more servers (*e.g.*, server 1) to retrieve the first streamlet (sub-file) storing a portion of the video. EX1004, Abstract ("Delivery of recorded audio or video material over a telecommunications link from a server [to a terminal] is accomplished by dividing the material into a sequence of sub-files each of which is independently requested by the terminal, which thereby has control of the rate of delivery. Provision may be made for switching between alternative sub-file sets representing alternative delivery modes or data rates.”), 1:28-40 (“According to one aspect of the invention there is provided a terminal for playing audio or video material which is stored on a remote server as a set of files representing successive temporal portions of the said material, the terminal comprising: a telecommunications interface for communication with the server; a buffer for receiving the files from the telecommunications interface; means for playing the contents of the buffer; and control means responsive to the state of the buffer to generate request messages for further files for replenishment of the buffer.”).

For example, Leaning discloses that the “function of the server 1 is to store data files, to receive from a user terminal a request for delivery of a desired data file and, in response to such a request, to transmit the file to the user terminal via the network[,]” where the “data files” requested by the client terminal are the sub-files discussed above. EX1004, 2:23-26. Leaning explains that the client produces a complete address for the sub-file of a request (*e.g.*, www.serverl.com/mp3_bwv565/000000.bin), which it uses to “construct[] a request message for the file having [the] URL and transmits it to the server 1 via the communications interface 35 and the internet 2.” EX1004, 4:34-36. The server responds by transmitting the required sub-file, the player program increments the filename (*e.g.*, to 000001.bin) and requests, receives, decodes and stores this second sub-file. EX1004, 4:43-44, 57-59; EX1003, ¶¶ 146-148.

Leaning further teaches that the at least one streamlet (sub-file) request from the one or more end user stations (terminals) includes a request for a currently selected one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the client to request a higher or lower bitrate (*e.g.*, quality) version of the streams. *See* 1[B]; EX1004, Abstract, 5:24-6:32 (describing creating and storing on a server sub-file sets at different bitrates to be requested by a terminal). For example, Leaning discloses that "[i]nitially the player program will begin [streaming] by requesting, from the directory specified in the link file, the index file, and stores locally a list of available data rates for future reference." EX1004, 650:55. The client then begins to request the audio sub-files as described earlier, from the first-mentioned 'rate' directory in the index file (*e.g.*, 024k_ll_s) from then on the player program measures the actual data rate being received from the server, and, based on that measured data rate, effects a rate change in the bitrate of the stream by changing the relevant part of the sub-file address. *See* EX1004, 6:55-7:34. As a result, the next request to the server becomes a request for the higher (or lower) rate, and the sub-file from the new directory is received, decoded and entered into the buffer. EX1004, 6:55-7:34, 17:12-30 (claim 1); EX1003, ¶ 149.

Additionally, Leaning discloses that selecting a sub-file (*e.g.*, the sub-file of the request sent to the server) occurs according to a determination, made by the client, of whether its current bandwidth is insufficient to stream the current bitrate without interruption or, instead, is sufficient to request an increased bitrate. EX1004, 18:27-35 (claim 9) ("monitoring the received data rate at the second station; ***and in the event that the measured rate is below that needed*** for the set to which the currently requested field belongs, ***performing mode switching*** to provide that subsequent said request messages shall ***request files from a set corresponding to a lower data rate***.") (emphases added); 18:10-26 (claim 8) ("storing a plurality of sets of files, which sets correspond to respective different delivery modes, and including, at the second station, ***effecting mode switching by providing that subsequent request messages shall request files from a set different from the set to which the immediately preceding request related***.") (emphasis added); 18:36-45 (claim 10); 6:59-7:14; EX1003, ¶¶ 146-150.

(c) 9[C]: Retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

Leaning discloses that the server retrieves from a storage device (*e.g.*, a hard disk) the requested sub-file (*e.g.*, "000000.bin") from the currently selected set of sub-files (*e.g.*, a ".bin" file from the selected directory for one of the low quality stream, the medium quality stream, and the high quality stream). *See* EX1004, 2:23-26 ("The function of the server 1 is to ***store*** data files [(*i.e.*, sub-files)], to receive from a user terminal a request for delivery of a desired data file and, in response to such a request, to transmit the file to the user terminal via the network.") (emphasis added); *see also* claim; EX1003, ¶¶ 151-152.

(d) 9[D]: Send the retrieved first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream to the requesting one of the end user stations over the one or more network connections.

Leaning discloses that the server sends the retrieved first sub-file (*e.g.*, streamlet) to the client terminal from which the request was received (*e.g.*, the respective one of the one or more end user stations) over the internet connection between server and client terminal. *See* claim 9[B]-9[C]; EX1004, 2:23-26 ("The function of the server 1 is to store data files, to receive from a user terminal a request for delivery of a desired data file and, in response to such a request, to transmit the file to the user terminal via the network."); EX1003, ¶¶ 153-154.

9. Claim 10: The system of claim 1, wherein the at least one virtual timeline corresponds to the currently selected one of the low quality stream, the medium quality stream, and the high quality stream.

Leaning explains that the virtual timeline "menu.htm" includes a list of available recordings (EX1004, 3:38-41) each of which has multiple quality versions (*e.g.*, low, medium, and high) as described above (*see supra* at 1[A][2]), which corresponds to the currently selected stream. *See* EX1004, 9:29-50 ("Alternatively, the user may be presented with a list of text labels or indices ***which may be selected*** (*e.g*. by means of a mouse) to initiate a jump. … The player program reads the index.htm file and if it recognizes an index command it generates a message ('an event') to the link.htm page which contains commands (typically written in Javascript) to handle such an event by displaying the labels at a desired place on the displayed page ***and respond to selection of those commands*** by generating a jump (containing the corresponding time) to the player program.") (emphases added); EX1003, ¶¶ 155-157.

10. Claim 11: The system of claim 1, wherein the virtual timeline defines a playlist for a user to view.

Leaning discloses the limitations of this claim. *See* claims 8, 10.

Leaning teaches the "menu.htm" includes a webpage "containing a list of recordings available, with hyperlinks to the corresponding link pages." EX1004, 3:38-41, 4:3-4 ("The user uses the browser to retrieve ***and display*** the menu page menu.htm from the server.") (emphasis added), 9:29-50; EX1003, ¶¶ 158-159.

11. Claim 12: The system of claim 1, wherein the virtual timeline comprises a file that is configured to define a playlist for a user to view.

Leaning discloses the limitations of this claim. *See* claims 1, 8, 10, and 11. EX1003, ¶¶ 160-161.

12. Claim 14:

(a) 14[PRE]: An end user station to stream a video over a network from a server for playback of the video, the content player device comprising:

Regardless of whether the preamble is limiting, Leaning discloses this limitation. Leaning discloses an end user station (client terminal 3), to stream a video over a network (Internet 2) from a server (server 1) for playback of the video. *See* 1[PRE]; EX1003, ¶¶ 162-163.

For purposes of this Petition only, Petitioners assume that the phrase "the content player device" in the preamble is intended to refer to "An end user station," otherwise the phrase "the content player device" lacks antecedent basis.

(b) 14[A]: A processor;

Leaning discloses a processor (*e.g.*, CPU 30) as part of the client terminal 3. EX1004 3:47-51 ("Thus Figure 2 shows such a terminal with a central processor 30, memory 31, a disk store 32, a keyboard 33, video display 34, communications interface 35, and audio interface ('sound card') 36. For video delivery, a video card would be fitted in place of, or in addition to, the card 36."), FIG. 2 (depicting components of terminal 3, including CPU 30); EX1003, ¶¶ 164-165.

(c) 14[B]: a digital processing apparatus memory device comprising non-transitory machine-readable instructions that, when executed, cause the processor to:

Leaning discloses a digital processing apparatus memory device (disk 32 and/or memory 31) comprising non-transitory machine-readable instructions (*e.g.*, software program for player program and browser) that, when executed, causes the processor (*e.g.*, CPU 30) to perform functions, such as receiving video presentation information. EX1004, 3:47-60 ("Thus Figure 2 shows such a terminal with a ***central processor 30***, ***memory 31***, . . . In the disk store are programs which may be retrieved into the memory 31 for ***execution by the processor 30, in the usual manner***. These programs include a communications program 37 for call-up and display of html pages - that is, ***a 'web browser' program*** such as Netscape Navigator or Microsoft Explorer, and a further program 38 which will be referred to here as 'the player program' ***which provides the functionality*** necessary for the playing of audio files in accordance with this embodiment of the invention.") (emphases added), 2:10-11 ("the system may be used to convey video signals instead of, or in addition to, audio signals"), FIG. 2 (showing CPU 30 on user terminals) (emphases added); EX1003, ¶¶ 166-167.

(d) 14[C][1]: Establish one or more network connections between the end user station and the server,

Leaning discloses a processor establishing one or more Internet network connections between the end user station (terminal 3) and the server (server 1). EX1004, 2:21-22 ("server 1 is connected via the internet 2 to user terminals [sic] 3"), 3:47-49 ("Thus Figure 2 shows such a terminal with a central processor 30, memory 31, a disk store 32, a keyboard 33, video display 34, ***communications interface 35***) (emphasis added), FIGS. 1-2; EX1003, ¶¶ 168-169.

(e) 14[C][2]: Wherein the server is configured to access at least one of a plurality of groups of streamlets;

Leaning discloses that the server is configured to access at least one of a plurality of groups (sets) of streamlets (sub-files). *See* claim 1[B][1]-[D][2]; EX1004, 5:28-48 ("the server stores two or more versions of the recording, recorded at different compression rates (for example at compressions corresponding to (continuous) data rates of 8, 16, 24 and 32 kbit/s respectively)"). Leaning explains that, to enable bitrate switching, the person preparing the content file for loading onto the server prepares several source files—by encoding the same content file several times at different rates and partitioning each source file into sub-files, as described above. EX1004, 5:28-48; EX1003, ¶¶ 170-171.

(f) 14[C][3]: Wherein the video is encoded at a plurality of different bitrates to create a plurality of streams including at least a low quality stream, a medium quality stream, and a high quality stream,

Leaning discloses this limitation. *See* claim 1[B][1]; EX1003, ¶ 172.

(g) 14[C][4]: Each of the low quality stream, the medium quality stream, and the high quality stream comprising a group of streamlets encoded at the same respective one of the different bitrates, each group comprising at least first and second streamlets, each of the streamlets corresponding to a portion of the video;

Leaning discloses this limitation. *See* claim 1[B][2]-[B][3]; EX1003, ¶ 173.

(h) 14[C][5]: Wherein at least one of the low quality stream, the medium quality stream, and the high quality stream is encoded at a bit rate of no less than 600 kbps; and

Leaning renders obvious this limitation. *See* claim 1[C]; EX1003, ¶ 174.

(i) 14[C][6]: Wherein the first streamlets of each of the low quality stream, the medium quality stream and the high quality stream each has an equal playback duration and each of the first streamlets encodes the same portion of the video at a different one of the different bitrates;

Leaning discloses this limitation. *See* claim 1[D][1]; EX1003, ¶ 175.

(j) 14[D]: Select a specific one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

Leaning further discloses selection, by a processor (*e.g.*, CPU 30), of a specific one of the quality streams based upon a determination by the end user station (*e.g.*, at the client terminal 3) to select a higher or lower bitrate version of the streams. *See* EX1004, Abstract, 5:28-5:51 (describing creating sub-file sets at different bitrates for a requesting terminal), 6:50-7:34 ("Initially the player program will begin by requesting, from the directory specified in the link file, the index file, and stores locally a list of available data rates for future reference. ... It then begins to request the audio sub-files as described earlier, from the first-mentioned 'rate' directory in the index file - viz. 024k_ll_s ... [The client] ***measures the actual data rate being received from the server***, averaged over a period of time … [and] ***the***

transfer rate achieved (number of bits per second) between the client and server is determined. ... The actual ***rate change is effected simply by the player program changing the relevant part of the sub-file address*** for example, changing '008k' to '024k' to increase the data rate from 8 to 24 kbit/s, and changing the current rate parameter to match. ***As a result, the next request to the server becomes a request for the higher (or lower) rate***, and the sub-file from the new directory is received, decoded and entered into the buffer.") (emphases added), 18:27-35 (claim 9) ("***monitoring the received data rate*** at the second station; and in ***the event that the measured rate is below that needed*** for the set to which the currently requested field belongs, ***performing mode switching*** to provide that subsequent said request messages shall ***request files from a set corresponding to a lower data rate***.") (emphases added), 18:10-26, 36-45 (claims 8 and 10); EX1003, ¶¶ 176-177.

(k) 14[E]: Place at least one virtual timeline request for at least one virtual times based on the selected one of the he low quality stream, the medium quality stream, and the high quality stream; and

Leaning discloses a processor placing a streamlet (sub-file) request to the server over the internet connection for the first streamlet of the selected stream. *See* claims 8 and 14[D]; EX1004, 4:28-5:10 ("The player program, having received the URL, adds to this the filename of the first sub-file, to produce a complete address for the sub-file – *i.e.* www.serverl.com/mp3_bwv565/000000.bin. ... The program ***constructs a request message for the file having this URL and transmits it to the***

server 1 via the communications interface 35 and the internet 2. ... We envisage that ***the player program would send the requests*** directly to the communications interface, rather than via the browser. The server responds by transmitting the required sub-file. ... The player program increments the filename to 000001.bin and requests, receives, decodes and stores this second sub-file as described in (4) and (5) above. ... This process is repeated until a 'file not found error' is returned.") (emphases added); EX1003, ¶¶ 178-179.

(l) 14[F]: Receive the at least one virtual timeline.

Leaning discloses this limitation. *See* claim 8.

13. Claim 15:

(a) 15[A]: The end user station of claim 14, wherein the non-transitory machine-readable instructions that, when executed, further cause the processor to:

Leaning discloses and/or renders obvious this limitation. *See* claim 14[PRE]-[F]; EX1003, ¶ 181.

(b) 15[B]: Place one or more streamlet requests to the server over the one or more network connections for the first streamlet of the selected stream;

Leaning discloses a processor placing a streamlet (sub-file) request to the server over the internet connection for the first streamlet of the selected stream. *See* EX1004, 4:28-5:10 ("The player program, having received the URL, adds to this the filename of the first sub-file, to produce a complete address for the sub-file – *i.e.* www.serverl.com/mp3_bwv565/000000.bin. ... The program ***constructs a request***

message for the file having this URL and transmits it to the server 1 via the communications interface 35 and the internet 2. ... We envisage that ***the player program would send the requests*** directly to the communications interface, rather than via the browser. The server responds by transmitting the required sub-file. ... The player program increments the filename to 000001.bin and requests, receives, decodes and stores this second sub-file as described in (4) and (5) above. ... This process is repeated until a 'file not found error' is returned.") (emphases added); EX1003, ¶¶ 182-183.

(c) 15[C]: Receive the requested first streamlet from the server via the one or more network connections wherein the one or more streamlet requests are based on the at least one virtual timeline; and provide the received first streamlet for playback of the video.

Leaning discloses receiving the requested first streamlet (*e.g.*, sub-file "000000.bin") from the server via the one or more Internet network connections and providing the received first streamlet for playback of the live event video. *See* claim 1, EX1004, 1:50-51 (" b) ***receiving*** the files; and c) decoding the files ***for replay*** of the material"); EX1003, ¶¶ 184-185.

14. Claim 16

Leaning discloses the limitations of this claim. *See* claim 5; EX1003, ¶ 186.

15. Claim 17

Leaning discloses the limitations of this claim. *See* claim 6; EX1003, ¶ 187.

16. Claim 18

Leaning discloses the limitations of this claim. *See* claim 10; EX1003, ¶ 188.

17. Claim 19

Leaning discloses the limitations of this claim. *See* claim 11; EX1003, ¶ 189.

18. Claim 20

Leaning discloses the limitations of this claim. *See* claim 3; EX1003, ¶ 190.

19. Claim 21

Leaning discloses the limitations of this claim. *See* claim 4; EX1003, ¶ 191.

20. Claim 22:

(a) 22[PRE]

Leaning discloses this limitation. *See* claim 1[PRE]; EX1003, ¶ 192.

(b) 22[A]

Leaning discloses this limitation. *See* claim 1[A]; EX1003; ¶ 193.

(c) 22[B]

Leaning discloses this limitation. *See* claim 1[B][2]-[B][3]; EX1003, ¶ 194.

(d) 22[C]

Leaning renders obvious this limitation. *See* claim 1[C]; EX1003, ¶ 195.

(e) 22[D]

Leaning discloses this limitation. *See* claim 1[D][1]-[D][2]; EX1003, ¶ 196.

(f) 22[E]

Leaning discloses this limitation. *See* claim 8; EX1003, ¶ 197.

(g) 22[F]

Leaning discloses this limitation. *See* claim 9[B]; EX1003, ¶¶ 198-199.

For purposes of this Petition only, Petitioners assume that the phrase "the at least one streamlet request" in the preamble is intended to refer to "at least one virtual timeline request," otherwise the phrase "the at least one streamlet request" lacks antecedent basis.

(h) 22[G]

Leaning discloses this limitation. *See* claim 9[C]; EX1003, ¶ 200.

(i) 22[H]

Leaning discloses this limitation. *See* claim 9[D]; EX1003, ¶ 201.

21. Claim 23:

(a) 23[A]

Leaning discloses and/or renders obvious this limitation. *See* claim 22; EX1003, ¶ 202.

(b) 23[B]

Leaning discloses a server storing "two or more versions" of a recording (including live event video) "at different compression rates[,]" such as at compressions corresponding to bit rates of 8, 16, 24, and 32 kbit/s respectively. EX1004, 5:28-32. Additionally, Leaning explains how the videos stored on the server can include live video, which is recorded and delivered in real time (*e.g.*, "on the fly"). EX1004, 15:1-5 ("Thus a computer could be provided to receive a live feed, to code it using the chosen coding scheme, and generate the sub-files "on the fly" and upload them to the server, so that, once a few sub-files are present on the server, delivery may commence."); 15:17-22. And, as described above, Leaning describes preparing several source files at different bit rates, which are each partitioned into sub-files and "loaded onto the server in separate directories corresponding to the different rate[.]" EX1004, 5:41-48; EX1003, 203-204.

(c) 23[C]

Leaning discloses this limitation. *See* claim 9[B]; EX1003, ¶ 205.

(d) 23[D]

Leaning discloses this limitation. *See* claim 9[B]; EX1003, ¶ 206.

(e) 23[E]

Leaning discloses this limitation. *See* claim 9[C]; EX1003, ¶ 207.

(f) 23[F]

Leaning discloses this limitation. *See* claim 9[D]; EX1003, ¶ 208.

22. Claim 24

Leaning discloses this limitation. *See* claim 5; EX1003, ¶ 209.

23. Claim 25

Leaning discloses this limitation. *See* claim 6; EX1003, ¶ 210.

24. Claim 26

Leaning discloses this limitation. *See* claim 3; EX1003, ¶ 211.

25. Claim 27

Leaning discloses this limitation. *See* claim 4; EX1003, ¶ 212.

26. Claim 28

(a) 28[PRE]

Leaning discloses this limitation. *See* claim 14[PRE]; EX1003, ¶ 213.

(b) 28[A]

Leaning discloses this limitation. *See* claim 14[C][1]; EX1003, ¶ 214.

(c) 28[B]

Leaning discloses this limitation. *See* claim 14[C][2]-[C][3]; EX1003, ¶ 215.

(d) 28[C]

Leaning discloses this limitation. *See* claim 14[C][4]-[C][6]; EX1003, ¶ 216.

(e) 28[D]

Leaning renders obvious this limitation. *See* claim 14[C][5]; EX1003, ¶ 217.

(f) 28[E]

Leaning discloses this limitation. *See* claim 14[C][6]; EX1003, ¶ 218.

(g) 28[F]

Leaning discloses this limitation. *See* claim 14[D]; EX1003, ¶ 219.

(h) 28[G]

Leaning discloses this limitation. *See* claim 14[E]; EX1003, ¶ 220.

(i) 28[H]

Leaning discloses this limitation. *See* claim 14[F]; EX1003, ¶ 221.

27. Claim 29:

(a) 29[A]

Leaning discloses this limitation. *See* claim 15[B]-[C]; EX1003, ¶ 222.

(b) 29[B]

Leaning discloses this limitation. *See* claim 15[C]; EX1003, ¶ 223.

(c) 29[C]

Leaning discloses this limitation. *See* claim 15[C]; EX1003, ¶ 224.

B. <u>Ground 2</u>: Claim 7 is obvious over Leaning in view of Reme and Leighton

The sections of Ground 1 are incorporated into this section by reference. Leaning in view of Leighton renders obvious claim 7. For example, it would have been obvious to combine the network architecture in Leighton, which includes a plurality of web servers, with the streaming system of Leaning. EX1003, ¶ 225.

1. Claim 7

(a) 7[A]: The system of claim 1, further comprising: a plurality of web servers located at different locations across the internet, each web server configured to:

Leaning in view of Leighton renders obvious this limitation. Leaning discloses a first web server. EX1004, 2:40-49 (explaining that the server of Leaning is "merely an ordinary 'web server'"), 3:19-4:3 (describing loading sub-files "onto a web server"); EX1003, ¶¶ 226-227.

To the extent that Leaning only teaches a single web server containing the sub-files, Leighton teaches providing a plurality of web servers located at different locations across the Internet to more effectively and efficiently deliver streams to clients based on physical location. EX1007, 2:51-56 (disclosing a "network architecture [that] is used to speed-up the delivery of richer Web pages, and [] allow[] Content Providers with large audiences to serve them reliably and economically, ***preferably from servers located close to end users***.") (emphasis added). In Leighton's network architecture, content is "served from particular hosting ***servers*** identified by the top- and low-level DNS servers. ***To locate the appropriate hosting servers to use***, the top-level DNS server ***determines the user's location in the network*** to identify a given low-level DNS server to respond to the request" for content. EX1007, 3:36-41 (emphases added), 4:67-5:4 (disclosing that "[w]eb server 12 ***is one of a plurality of servers which are accessible by clients***, one of which is illustrated by machine 10.") (emphasis added).

A POSITA would have been motivated to combine Leighton's teaching of a network architecture with multiple web servers at different locations on the internet, into Leaning's streaming system. EX1003, ¶¶ 228-230.

The system in Leaning is already in the context of improved client-server streaming, including to reduce the delay perceived by the user before requested content can be displayed. EX1004, Abstract, 10:21-24; EX1003, ¶ 230. A POSITA would have been motivated to further improve Leaning by applying Leighton's teachings regarding the use of multiple web servers to transmit content to users from the web server on the network (*e.g.*, the internet) that is closest to that user. EX1003, ¶¶ 230-232.

As described above, Leighton expressly teaches storing multiple copies of the same content on several different web servers, which are each located at different addresses on the network:

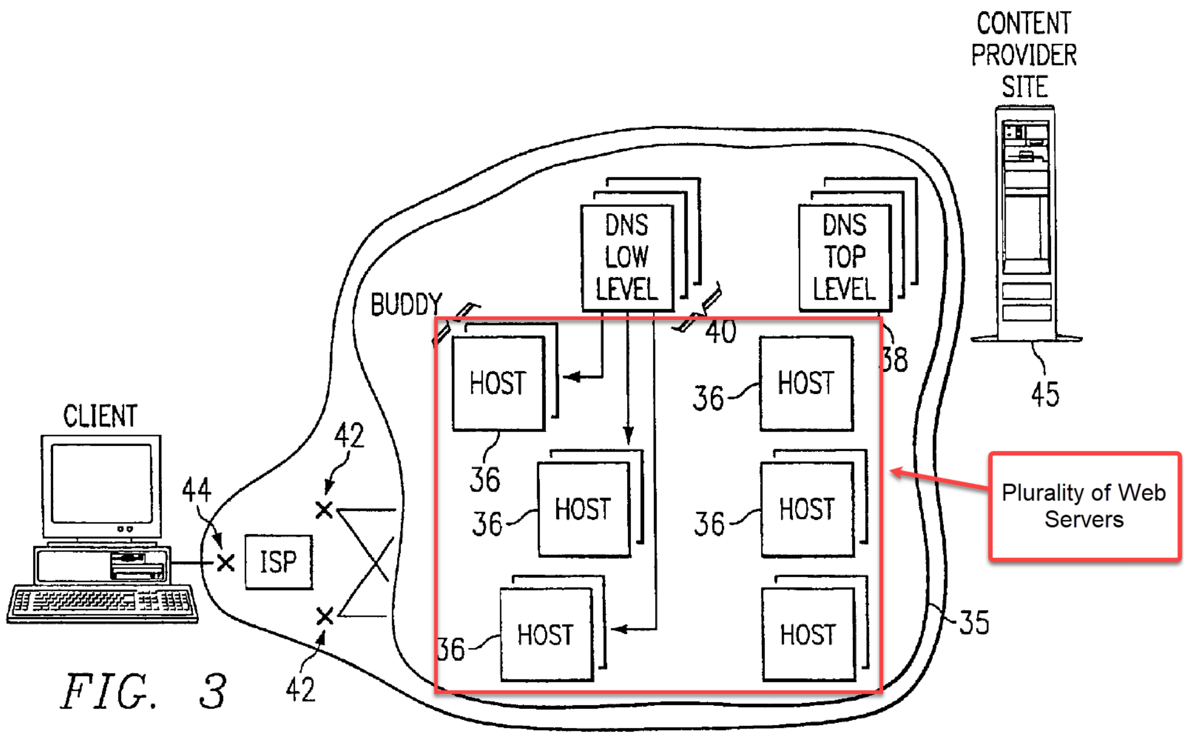


FIG. 3

EX1007, FIG. 3 (annotated). To improve the overall user streaming experience, a POSITA would have been motivated to create a multi-web server network architecture, as taught by Leighton, in a system for client-server streaming of content files, as taught by Leaning, in order to provide an improved download time and minimize delays perceived by the user at the client (*e.g.*, using several web servers at different locations on the network and requesting content from the web server closest to the user on the network). EX1003, ¶ 232. A POSITA would have had a reasonable expectation of success with such a combination because both Leaning and Leighton are drawn to similar systems of client-server communication of content via HTTP requests sent, via the internet, to a web server. EX1003, ¶ 232.

Finally, Leaning, Leighton, and the ’680 Patent are analogous art because all three are directed to systems and techniques to improve performance in client-server based streaming over the internet. EX1001, 6:33-7:67; EX1004, Abstract; EX1007, 2:51-56 and 3:33-45 (Disclosing a “network architecture [that] is used to speed-up the delivery of richer Web pages” by servicing clients’ HTTP requests with “servers located close to end users.”); EX1003, ¶¶ 226-233.

(b) 7[B]: Receive at least one streamlet request over one or more internet connections from the one or more end user stations to retrieve the first streamlet storing a portion of the video, wherein the at least one streamlet request from the one or more end user stations includes a request for a currently selected first streamlet from one of the low quality stream, the medium quality stream, and the high quality stream based upon a determination by the end user station to select a higher or lower bitrate version of the streams;

Leaning teaches that the conventional web servers are each configured to receive at least one streamlet (sub-file) request over one or more Internet connections from a respective one of the one or more end user stations (*e.g.*, terminal 3) to retrieve the first streamlet (*e.g.*, sub-file “000000.bin”) storing a portion of the video. *See supra* claim 9[B]; EX1004, Abstract, 1:28-19, 2:23-26 (“The function of the server 1 is to store data files, to receive from a user terminal a request for delivery of a desired data file and, in response to such a request, to transmit the file to the user terminal via the network."), 4:27-5:10; EX1003, ¶¶ 234-235.

Additionally, Leaning discloses that the terminal's request for a streamlet (*e.g.*, sub-file "000000.bin") includes a request for a currently selected one of the low quality stream, the medium quality stream, and the high quality stream, which is based upon a determination by the client (*e.g.*, at the terminal 3) to select a higher or lower bitrate version of the stream. *See* claim 9[B]; EX1004, Abstract, 5:28-6:32 (describing creating sub-file sets at different bitrates to be requested by a terminal). As described above, Leaning discloses that "[i]nitially the player program will begin [streaming] by requesting, from the directory specified in the link file, the index file, and stores locally a list of available data rates for future reference." EX1004, 6:50-55.

The client then begins to request the audio sub-files as described earlier, from the first-mentioned 'rate' directory in the index file (*e.g.*, 024k_ll_s) and from then on the player program measures the actual data rate being received from the server, and, based on that measured data rate, effects a change in the bitrate of the stream by "changing the relevant part of the sub-file address for example, changing '008k' to '024k' to increase the data rate from 8 to 24 kbit/s, and changing the current rate parameter to match." *See* EX1004, 6:55-7:34 17:12-30 (claim 1) ("A terminal for playing audio or video material which is stored on a remote server as a set of files representing successive temporal portions of the said material ...”), 18:27-35 (claim 9), 18:10-26 (claim 8), 18:36-45 (claim 10); EX1003, ¶¶ 234-236.

(c) 7[C]: Retrieve from the storage device the requested first streamlet from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream; and

Leaning discloses that the server retrieves from the storage device the requested streamlet (sub-file) from the currently selected one of the low quality stream, the medium quality stream, and the high quality stream. *See* claim 9[C]; EX1004, 2:23-26; EX1003, ¶¶ 237-238.

(d) 7[D]: Send the retrieved first streamlet from the currently selected one of the different copies to the requesting one of the end user stations over the one or more network connections.

Leaning discloses that the server sends the retrieved first streamlet (*e.g.*, the first sub-file “000000.bin”) to the respective one of the one or more end user stations (*e.g.*, terminal 3) over the internet connection (*e.g.*, between server 1, Internet 2, and terminal 3). *See* claim 7[B]-4[C]; EX1004, 2:23-26 (“The function of the server 1 is to store data files, to receive from a user terminal a request for delivery of a desired data file and, in response to such a request, to transmit the file to the user terminal via the network.”); EX1003, ¶¶ 239-240.

C. <u>Ground 3</u>: Claims 1-6, 8-12 and 14-29 are obvious over Leaning and Reme

1. To the extent that claims 1-6, 8-12, and 14-29 are not obvious over Leaning—they are, as explained in Ground 1, *supra*—these claims are obvious over the combination of Leaning in view of Reme.

1. Claims 1, 14, 22, and 28

(a) 1[C]; 14[C][5]; 22[C]; and 28[D]

As explained above in Ground 1, Leaning teaches 1[C], 14[C][5], 22[C], and 28[D]. To the extent it is argued that Leaning does not explicitly disclose or render obvious a stream of at least 600 kbps, Leaning in view of Reme teaches this limitation. For example, Reme teaches that it was known to encode video content into at least three different quality levels (or bitrates) for streaming, for example, at 30 kbps, at 300 kbps, and at 5 Mbps, to allow streaming to various clients with different network conditions. For example, Reme proposes "a plurality of pre-encoded versions of the [video] content, each version corresponding to a different encoding rate (and hence to a different quality)." EX1005, 3:15-17. Indeed, Reme explains that "Internet streaming applications should be quality adaptive" and that "streaming applications should adjust the quality of the delivered stream such that the bandwidth required for transmitting the stream ***matches*** the available bandwidth." 1005, 3:11-14. To achieve this, Reme proposes that streaming clients "switch among [the] plurality of pre-encoded versions of the content" each with a different encoding rate (**bit rate**) or quality level." EX1005, 3:15-17.

	Version $V_{i,1}$ (encoding rate 30kbps)	Version $V_{i,2}$ (encoding rate 300kbps)	Version $V_{i,3}$ (encoding rate 5Mbps)

EX1004, 5:1-4 (annotated). A POSITA would have been motivated to encode a video embodiment of Leaning (including the corresponding sets of sub-files) into each of a low quality, medium quality, and high quality level bitrate stream (low at 30 kbps, medium at 300 kbps, and high at 5 Mbps), as taught by Reme, because it would enable client terminals with a wide variety of network bandwidth requirements to stream the video and would have allowed for those with user terminals with high bandwidth connections to stream at 5 Mbps and enjoy higher resolution viewing and a better user experience. EX1003, ¶¶ 241-243. A POSITA would have had a reasonable expectation of success with such a combination because both **(1)** Leaning and Reme are drawn to similar systems of client-server based streaming of video encoded into multiple different bitrate copies; and **(2)** because simply encoding one of the video streams at a particular bitrate, such as the 5 Mbps taught by Reme, would have been a simple and straightforward implementation of Leaning for a POSITA to make when switching from the audio to video embodiments of Leaning, which would not require any modification to Leanings teachings. EX1003, ¶¶ 241-246.

D. <u>Ground 4</u>: Claims 1-6, 8-12 and 14-29 are obvious over Leaning and SMIL 2.0

2. To the extent that claims 1-6, 8-12, and 14-29 are not obvious over Leaning—they are, as explained in Ground 1, *supra*—these claims are obvious over the combination of Leaning in view of SMIL 2.0.

1. Claims 1, 14, 22, and 28

3. SMIL 2.0 discloses the "virtual timeline" recited in the Challenged Claims. SMIL 2.0 discloses the use of certain web elements (*e.g.*, the <seq> container) to "define[] a sequence of elements [(*e.g.*, videos)] in which elements play one after the other". EX1006, 158. The <seq> container disclosed by SMIL 2.0 defines the videos played at the client and it would have been obvious to a POSITA to use the <seq> container disclosed in SMIL 2.0 as a virtual timeline that defines a playlist of videos to stream to an end user, each video comprising a plurality of sub-files, as disclosed by Leaning. EX1003, ¶¶ 247-248. SMIL 2.0's defined sequence of elements that successively play is a "virtual timeline," according to the plain and ordinary meaning of that term and how it is used in the '680 patent. *See* EX1001, 12:36-38 ("the virtual timeline (hereinafter 'VT') 600 may comprise a file that is configured to define a playlist for a user to view").

SMIL 2.0 expressly teaches generating a playlist of entire content files – for example, the "vid2.mpg" file is followed by the "vid3.mpg" file using the "<seq>" element:

```
<seq>
   <img src="img1.jpg" />
   <video src="vid2.mpg" />
   <video src="vid3.mpg" />
</seq>
```

Playlist of entire content files

EX1006, 207, 158 (annotated, the seq container, shown by the <seq> and </seq> tags, "defines a sequence of elements in which ***elements play one after the other***") (emphasis added). A POSITA would have been motivated to create a virtual timeline playlist in Leaning of entire content files (*e.g.*, the .bin files streamed to the client in Leaning's system) that play one after the other, as taught by SMIL 2.0 (*e.g.*, using the <seq> element, shown above), in order to set out a video schedule of similar videos for clients to view in succession. EX1003, ¶¶ 250-251. A POSITA would have had a reasonable expectation of success with such a combination because both Leaning and SMIL 2.0 are drawn to similar systems of client-server based streaming and generating a playlist of entire content files would have been a straightforward modification for a POSITA. EX1003, ¶ 250.

Finally, Leaning, SMIL 2.0, and the ’680 Patent are analogous art because all three are directed to systems and techniques to improve performance in client-server based streaming over the internet. EX1001, 6:33-7:67; EX1004, Abstract; EX1006, 104-107 (describing “The SMIL 2.0 Media Object Models,” including video files that SMIL 2.0 enables clients to download using several HTTP links associated with different “temporal subparts” of the video); EX1003, ¶¶ 247-251.

VII. SECONDARY CONSIDERATIONS DO NOT SUPPORT A CONCLUSION OF NONOBVIOUSNESS

The first three *Graham* factors were discussed for each ground, including the scope of the prior art, the prior art as compared to the claims, and the level of ordinary skill in the pertinent art. *See* Sections V-VI.

For the last factor, secondary considerations, the Patent Owner cannot show a nexus between previously alleged secondary considerations and the claim language. Petitioners acknowledge that the ALJ in the ITC Investigation, while discussing secondary considerations of non-obviousness, accepted Patent Owner’s argument that there was a presumption of a nexus between the claims that were asserted in the ITC Investigation (including from the ‘680 Patent) and the Move Media Player (which was acquired by DISH). EX1009, 212-213. The ALJ found that DISH’s expert demonstrated that Move’s success was tied to the supposedly unique features of the claimed invention, namely “offering ‘a smooth end user experience as the Move Media Player up-shifts and down-shifts in response to network and client CPU availability" and streaming using "simple HTTP protocol transfer of media files from standard Web servers rather than deployment of expensive media servers." EX1009, 213.

However, because Leaning discloses those same features, the secondary factors, such as commercial success, "are irrelevant." *Magseis FF LLC v. Seabed Geosolutions (US) Inc.*, 860 F. App'x 746, 752 (Fed. Cir. 2021) (citing *Ormco Corp. v. Align Technology, Inc.*, 463 F.3d 1299, 1312 (Fed. Cir. 2006)). Indeed, these characteristics cannot support a nexus to the claim language "because the evidence of secondary considerations is not tied to the claimed invention's ***unique*** characteristics." *Magseis* , 860 F. App'x at 752 (emphasis added) (quoting *Ormco*, 463 F.3d at 1312 ("[I]f the feature that creates the commercial success was known in the prior art, the success is not pertinent."). Accordingly, the purportedly unique and successful aspects of the Move system were already in existence in the prior art and thus cannot form the basis of any nexus to the claimed invention. *See Yita LLC v. MacNeil IP LLC*, 69 F.4th 1356, 1364 (Fed. Cir. 2023) ("[O]ur case law makes clear that objective evidence of nonobviousness lacks a nexus if it exclusively relates to a feature that was *known* in the prior art—not necessarily well-known") (emphasis in original) (internal citations and quotations omitted).

Accordingly, the secondary considerations relied upon by DISH in previous proceedings cannot support a finding of nonobviousness in light of those same features being disclosed in Leaning.

VIII. *FINTIV*

The Board balances six factors in considering discretionary denial under 35 U.S.C. § 314(a) when parallel litigation exists. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential). Here, the factors ("*Fintiv* factors") favor institution.

Factor 1—potential stay of the district court litigation—is neutral because no party has requested a stay. *VMWare, Inc. v. Intellectual Ventures I LLC*, IPR2020-00470, Paper 13 (P.T.A.B. Aug. 18, 2020), at 17 (finding in the absence of a stay motion that this factor "does not weigh for or against discretionary denial").

Factor 2—district court trial date and the Board's statutory deadline—weighs heavily against discretionary denial. The District Court has not yet issued a schedule or set a trial date. Accordingly, earliest estimated trial date for the district court litigation is far in the future and is likely to occur years after the estimated date for the Board's Final Written Decision.

Factor 3—investment in the district court proceedings—weighs heavily against discretionary denial. The associated district court case is still in the very early stages of litigation: Petitioners' Motion to Dismiss Patent Owner's complaint is pending, Petitioners have yet to file their answer, a schedule has not been set, discovery has not yet begun, Petitioners have not yet served their invalidity contentions, and a Claim Construction hearing is not yet scheduled. Moreover, a claim construction order will not issue prior to the PTAB's projected institution decision date. Accordingly, the district court will not invest significant resources or issue substantive orders related to the challenged patent prior to the issuance of an institution decision. *See Fintiv*, IPR2020-00019, Paper 11 at 9-12.

Factor 4—overlap in the parallel proceedings—is neutral as all the claims asserted in the district court are challenged here. .

Factor 5—overlapping parties—is neutral as it is "far from an unusual circumstance that a petitioner in *inter partes* review and a defendant in a parallel district court proceeding are the same." *Sand*, IPR2019-01393, Paper 24, at 12-13.

Factor 6—other considerations—weighs against discretionary denial. As explained, the merits of the Petition are strong, and the Challenged Claims are invalid. For example, none of the particular combinations of prior art relied on were previously considered by either the Office or the district courts in evaluating the validity of the '680 Patent. *Cf. Comcast Cable Commn's, LLC v. Rovi Guides, Inc.*, IPR2019-00231, Paper 14 at 11 (PTAB May 20, 2019) (obviousness challenges not "previously considered by the Office or any court" weigh in favor of not denying institution).

Additionally, the Office recently granted *Inter Partes* Review for two ancestor patents to the '680 Patent, U.S. Patent Nos. 8,868,772 and 9,407,564, which was in response to a petition filed by fuboTV Media Inc. and Yanka Industries, Inc., also based on Leaning. *See* IPR2024-00919, Paper 14 (decision granting institution) and IPR2024-00918, Paper 14 (same). Moreover, the '680 Patent is currently asserted in several district court cases identified below. Institution of this IPR provides the opportunity for narrowing and simplifying the litigations for the district court. Accordingly, this factor weighs heavily in favor of institution.

Compelling Merits – Finally, discretionary denial is not warranted because this petition presents compelling evidence of unpatentability. The '680 Patent purports to have invented video streaming using groups of streamlets that encode the same portions of a video at different qualities. As discussed above, Leaning clearly teaches this.

Accordingly, the Board should decline to exercise its discretion under *Fintiv* and instead institute trial.

IX. ***ADVANCED BIONICS***

Denial under § 325(d) is not warranted under *Advanced Bionics'* two-factor test. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6, at 8 (PTAB Feb. 13, 2020).

Factor 1—whether the "same or substantially the same prior art or arguments previously were presented to the Office" (*id*.) weighs in favor of institution because the specific grounds here were not considered during prosecution of the '680 Patent. None of Leaning, Reme, Leighton, or SMIL 2.0 are cited on the face of the '680 Patent. Additionally, the Examiner never considered the combination of Leaning and Reme during prosecution of the '680 Patent.

As noted above, while the '680 Patent does cite on its face US2004/0030547 (also to Leaning), that application was buried in an IDS containing 204 references, and never specifically discussed during prosecution. EX1002, 232; EX1001, 1-7. And other than the Examiner stamping that every reference in the IDS was considered (EX1002, 627), there is no evidence that the Examiner considered the material disclosures of US2004/0030547—it was never used in a rejection and was never substantively addressed during prosecution of the '680 Patent.

Factor 2—Because the first *General Plastics* factor is not satisfied and there is no analysis of Leaning in view of Reme, Leighton, and/or SMIL 2.0 by the Office, the second factor—whether the petitioner has demonstrated that the Office erred in a manner material to the patentability of challenged claims—is neutral.

X. MANDATORY NOTICES AND FEES

A. Real Party-In-Interest

The real parties-in-interest for Petitioners are WebGroup Czech Republic A.S. and NKL Associates, S.R.O.

B. Related Matters

The ’680 Patent is currently asserted in the following pending proceedings.

Table 1: Pending DISH Litigations Asserting the ’680 Patent

No.	Case Caption	Court
	DISH Technologies L.L.C. et al v. WebGroup Czech Republic A.S. et al 2:23-cv-00553	District of Utah
	DISH Technologies L.L.C. et al v. MG Premium Limited et al, 2:23-cv-00552	District of Utah
	DISH Technologies L.L.C. et al v. fuboTV Media Inc. f/k/a fuboTV Inc., 1:23-cv-00986	District of Delaware
	DISH Technologies L.L.C. et al v. Vidgo Inc., 2:23-cv-00624	District of Utah
	DISH Technologies L.L.C. et al v. BritBox, LLC, 1:23-cv-08971	Southern District of New York
	DISH Technologies L.L.C. et al v. Yanka Industries, Inc. d/b/a MasterClass, 1:23-cv-01305	District of Delaware
	DISH Technologies L.L.C. et al v. Aylo Freesites Ltd. f/k/a/ MG Freesites Ltd., 2:24-cv-00066	District of Utah

Table 2: Concluded DISH Litigations Asserting the ’680 Patent

No.	Case Caption	Court
	iFIT Inc. v. ITC, 23-1965, which is an appeal from Investigation No. 337-TA-1265 (ITC)	Federal Circuit
	DISH Technologies L.L.C. et al v. ICON Health & Fitness, Inc., 1:21-cv-00531	District of Delaware
	DISH Technologies L.L.C. et al v. Lululemon Athletica Inc. et al, 1:21-cv-00532	District of Delaware

	DISH Technologies L.L.C. et al v. Peloton Interactive, Inc., 2:21-cv-00132	Eastern District of Texas
	MG Freesites ltd. v. DISH Technologies L.L.C. et al, 3:23-cv-03674	Northern District of California
	DISH Technologies L.L.C. et al v. Beachbody, LLC d/b/a/ BODi, 1:23-cv-00987	District of Delaware
	DISH Technologies L.L.C. et al v. A Parent Media Co. Inc. et al, 1:23-cv-01000	District of Delaware

Table 3: *Inter Partes* Reviews as to the ’680 Patent

No.	Case Caption	Status
	Aylo Freesites Ltd. f/k/a/ MG Freesites Ltd. v. DISH Technologies L.L.C. et al., IPR2024-00046	Instituted; Final Written Decision Pending
	fuboTV Media Inc. f/k/a fuboTV Inc. v. DISH Technologies LLC, IPR2024-00904	Instituted; Joined with IPR2024-00046
	Aylo Freesites Ltd. f/k/a/ MG Freesites Ltd. v. DISH Technologies L.L.C. et al., IPR2024-00516	Not Instituted

C. Counsel and Service Information

Petitioners provide the following counsel and service information. Petitioners consent to electronic service the email addresses listed in the table below. Pursuant to 37 C.F.R. § 42.10(b), a Power of Attorney accompanies this Petition.

LEAD COUNSEL	**BACK-UP COUNSEL**
James M. Glass (Reg. No. 46729) jimglass@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 51 Madison Avenue, 22nd Floor New York, NY 10010 Tel: (212) 849-7000 Fax: (212) 849-7100	Quincy Lu (Reg. No. 76,954) quincylu@quinnemanuel.com QUINN EMANUEL URQUHART & SULLIVAN, LLP 1109 First Avenue, Suite 210 Seattle, WA 98101 Tel: (206) 905-7000 Fax: (206) 905-7100

D. Payment of Fees

The undersigned authorizes the Office to charge the fee required for this Petition for *Inter Partes* Review to Deposit Account No. 50-5708. Any additional fees that might be due are also authorized.

XI. CONCLUSION

For the reasons above, *inter partes* review is requested.

Date: January 17, 2025

Respectfully submitted,

By: /s/ *James M. Glass*

Counsel for Petitioners

CERTIFICATE OF COMPLIANCE WITH TYPE-VOLUME LIMITATION, TYPEFACE REQUIREMENTS, AND TYPE STYLE REQUIREMENTS

1. This Petition complies with the type-volume limitation of 14,000 words, comprising 13,995 words, as counted using the Microsoft Word software that was used to prepare this paper, excluding the parts exempted by 37 C.F.R. § 42.24(a).

2. This Petition complies with the general format requirements of 37 C.F.R. § 42.6(a) and has been prepared using Microsoft® Word 2016 in 14-point Times New Roman.

Date: January 17, 2025

By: */s/ James M. Glass*
James M. Glass
QUINN EMANUEL URQUHART
& SULLIVAN, LLP
51 Madison Avenue, 22nd Floor
New York, NY 10010
Tel: (212) 849-7000
Fax: (212) 849-7100

Counsel for Petitioners

CERTIFICATE OF SERVICE

The undersigned hereby certifies that on January 17, 2025, true and correct copies of the foregoing document and supporting materials were served in its entirety on the Patent Owner at the following address of record as listed in Patent Center via Priority Mail Express® or Express Mail:

James Hanft
Max Gratton
Spencer Thevenin
DISH Technologies L.L.C.
9601 S. Meridian Blvd
Englewood, CO 80112

Courtesy copies were also sent via electronic mail to Patent Owner's counsel of record in the related district court proceeding:

hatch@hatchpc.com
pace@hatchpc.com
ali.dhanani@bakerbotts.com
hop.guy@bakerbotts.com
kurt.pankratz@bakerbotts.com

DATED: January 17, 2025 Respectfully submitted,

By: *<u>/s/ James M. Glass</u>*