

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Google LLC,
Petitioner

v.

WAG Acquisition, L.L.C.
Patent Owner

IPR2022-01411

U.S. Patent No. 9,729,594 B2
Issue Date: August 8, 2017

Title: STREAMING MEDIA DELIVERY SYSTEM

DECLARATION OF NATHANIEL POLISH, Ph.D.

TABLE OF CONTENTS

	Page
I. INTRODUCTION AND QUALIFICATIONS	8
A. Qualifications and Experience	8
B. Materials Considered.....	12
II. PERSON OF ORDINARY SKILL IN THE ART	16
III. STATEMENT OF LEGAL PRINCIPLES.....	18
A. Claim Construction	18
B. Obviousness.....	20
IV. TECHNOLOGY BACKGROUND.....	26
A. Streaming Media over the Internet.....	26
B. Internet Transfer Protocols.....	29
C. Digital Audio and Video	30
D. Pull-Based Streaming Systems.....	32
V. THE '594 PATENT.....	34
A. Overview of the Specification.....	34
B. Prosecution History	37
C. Related <i>Inter Partes</i> Review Proceedings.....	38
D. The Challenged Claims	42
VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS	43
A. Brief Summary of Prior Art	44
1. Carmel [EX1003].....	44
2. Hill [EX1006]	47
3. Ravi [EX1004].....	53
4. Aharoni [EX1008].....	55
5. Galensky [EX1007].....	56

TABLE OF CONTENTS
(continued)

	Page
B. Ground 1: Claims 1-17 Are Obvious Over Hill.....	58
1. Claim 1 (Independent)	58
(a) “A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the method comprising:” (claim 1, preamble)	59
(b) “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,” (claim 1[a(i)])	61
(c) “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,” (claim 1[a(ii)])	68
(d) “each request specifying one or more serial identifiers of the media data elements requested;” (claim 1[a(iii)])	71
(e) “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and” (claim 1[b(i)])	73
(f) “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;” (claim 1[b(ii)])	75
(g) “storing the received media data elements in a memory of the media player;” (claim 1[c])	78
(h) “playing the received media data elements in series from the memory of the media player; and” (claim 1[d])	79

TABLE OF CONTENTS

(continued)

	Page
(i) “as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e])	80
2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”	87
3. Claim 3: “The method of claim 1 wherein the serial identifiers are sequential.”	89
4. Claim 4 (“The method of claim 1, wherein the receiving is via a reliable transmission protocol.”) and Claim 5 (“The method of claim 4, wherein the reliable transmission protocol is TCP.”).....	90
5. Claim 6 (Independent)	91
6. Dependent claims 7-10	96
7. Claim 11 (Independent)	97
8. Dependent claims 12, 13, 16, 17.....	100
9. Claim 14 (“The computer program product of claim 11, wherein the computer program product is provided as a software application for the media player.”) and Claim 15 (“The computer program product of claim 11, wherein the computer program product is incorporated in the media player.”).....	101
C. Ground 2: Claims 1-17 Are Obvious Over Hill In View Of Aharoni	102
D. Ground 3: Claims 1-17 Are Obvious Over Hill In View Of Ravi.....	107

TABLE OF CONTENTS
(continued)

	Page
E. Ground 4: Claims 1-17 Are Obvious Over Carmel in view of Hill.....	112
1. Claim 1 (Independent)	112
(a) “A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet” (claim 1, preamble).....	112
(b) “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,” (claim 1[a(i)]).....	116
(c) “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,” (claim 1[a(ii)])	120
(d) “each request specifying one or more serial identifiers of the media data elements requested;” (claim 1[a(iii)])	123
(e) “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements,” (claim 1[b(i)]).....	126
(f) “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;” (claim 1[b(ii)])	136
(g) “storing the received media data elements in a memory of the media player;” (claim 1[c]).....	141
(h) “playing the received media data elements in series from the memory of the media player; and” (claim 1[d])	142

TABLE OF CONTENTS

(continued)

	Page
<ul style="list-style-type: none"> (i) “as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e]) 	143
2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player. ”	146
3. Claim 3: “The method of claim 1 wherein the serial identifiers are sequential.”	148
4. Claim 4: “The method of claim 1, wherein the receiving is via a reliable transmission protocol.”	149
5. Claim 5: “The method of claim 4, wherein the reliable transmission protocol is TCP.”	150
6. Claim 6 (Independent)	151
7. Dependent claims 7-10	155
8. Claim 11 (Independent)	157
9. Dependent Claims 12, 13, 16, 17.....	159
10. Claim 14 (“The computer program product of claim 11, wherein the computer program product is provided as a software application for the media player.”) and Claim 15 (“The computer program product of claim 11, wherein the computer program product is incorporated in the media player.”).....	160
F. Ground 5: Claims 1-17 Are Obvious Over Carmel In View Of Galensky.....	162

TABLE OF CONTENTS
(continued)

	Page
1. “as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e])	162
2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”	166
G. Ground 6: Claims 1-17 Are Obvious Over Carmel In View Of Hill And Ravi.	167
H. Ground 7: Claims 1-17 Are Obvious Over Carmel In View Of Galensky And Ravi.	173
VII. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS.....	180
VIII. CONCLUSION.....	181

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

I, Nathaniel Polish, declare as follows:

I. INTRODUCTION AND QUALIFICATIONS

1. I have been retained by Google LLC (“Petitioner”) as an independent expert consultant in this proceeding before the United States Patent and Trademark Office (“USPTO”).

2. I am being compensated at a rate of \$700/hour for my services in this proceeding, which is my regular and customary rate.

3. My compensation is in no way contingent on the nature of my findings, the presentation of my findings in testimony or this declaration, or the outcome of this or any other proceeding. I have no other interest in this proceeding.

A. Qualifications and Experience

4. My Curriculum Vitae is attached as Exhibit 1067.

5. I reside in New York City, NY.

6. I am an independent consultant. All of my opinions stated in this declaration are based on my own personal knowledge and professional judgment. In forming my opinions, I have relied on my knowledge and experience in streaming media, signal processing, computer network design, and video processing.

7. I am over 18 years of age and, if I am called upon to do so, I would be competent to testify as to the matters set forth herein. A copy of my current

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

curriculum vitae, which details my education and professional and academic experience, is included as EX1067 in this proceeding. The following provides an overview of some of my experience that is relevant to the matters set forth in this declaration.

8. I expect to testify regarding my background, qualifications, and experience relevant to the issues in this litigation. I have a Ph.D. in Computer Science from Columbia University. I hold the following four degrees from Columbia, spanning the years 1980 to 1993:

- Ph.D. in Computer Science, May 1993, Thesis: Mixed Distance Measures for the Optimization of Concatenative Vocabularies in Speech Synthesis;
- M.Phil. in Computer Science, December 1989;
- M.S. in Computer Science, December 1987; and
- B.A. in Physics, Columbia College, May 1984.

9. For over thirty five years, I have run a computer technology development firm that I co-founded, called Daedalus Technology Group. My primary business activity is the development of computer-related products. This activity involves understanding the business objectives of customers, designing products to suit their needs, and supervising the building, testing, and deployment of these products. I develop hardware and software as well as supervise others who do

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

so.

10. Also, from time to time I found other companies in order to pursue particular product opportunities. I develop and ultimately sell these companies. Most of my business activity, however, is as a consulting product developer. From time to time I have also served as an expert witness on computer and software related cases. I am a named inventor on ten United States patents and am a member of several professional societies, including the IEEE and ACM.

11. I have extensive experience in several areas relevant to this case. My doctoral work was in the area of computer speech. As part of that work I developed signal processing systems and algorithms to filter and manipulate sound, especially speech. That work gave me strong practical insight into such topics as the management and transport of streaming media.

12. Much of my professional work over the past thirty five years has been as a computer technology product developer. Several of those products involved technologies closely related to the topics in the patent at issue in the current matter, including streaming audio and video systems and associated networking. One product in particular was for a company called Instant Video Technologies (IVT) in the mid-1990s. This company later changed its name to Burst.com. The products that I developed for IVT included a family of digital media servers to deliver high

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

quality media over a variety of network connection including Internet connections.

The work resulted in a patent (U.S. Patent No. 5,963,202) on our server technology.

13. In addition to this work for IVT, my firm also designed and developed, under my supervision, a number of products or systems for delivering digital audio and performing associated signal processing. Between 1982-1991, the firm designed and built, under my supervision, all parts of four generations of interactive voice editing systems for The Dun & Bradstreet Corporation. These editing systems produced output used in an extensive, high quality speech synthesis system that was part of a larger information delivery system. The systems were built on networks of computers in a variety of programming languages and systems. The systems involved interactive user interfaces, screen windows, signal processing, and voice file systems.

14. In 2000, I also supervised the development, design, and deployment of an industrial strength interactive voice response (IVR) system that allows users to access streaming media via their cell phones. This system supported up to 96 streams on a single chassis and was designed for a company called Savos. Under my supervision, the firm developed for this system the web-based content management system and interface for customizable user preferences.

15. In addition to the products and systems described above, I have also

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

developed many other systems that utilize various networking techniques for moving data around the Internet in a timely way taking into account spontaneous changes in network capacity and usage.

16. Through these and other experiences, I am very familiar with the technologies at issue in this matter.

B. Materials Considered

17. The analysis that I provide in this Declaration is based on my education, research, and experience, as well as the documents I have considered. In forming my opinions, I have read and considered U.S. Patent No. 9,729,594 B2 (“594 patent”) [EX1001] and its prosecution history. I have cited to or considered the following documents in my analysis below:

Exhibit No.	Description of Document
1001	U.S. Patent No. 9,729,594 B2 to Harold Edward Price (filed October 3, 2016, issued August 8, 2017)
1003	U.S. Patent No. 6,389,473 to Sharon Carmel et al. (filed March 24, 1999, issued May 14, 2002) (“Carmel”)
1004	U.S. Patent No. 6,292,834 to Hemanth Srinivas Ravi et al. (filed March 14, 1997, issued September 18, 2001) (“Ravi”)
1006	U.S. Patent No. 6,005,600 to Ralph D. Hill (filed October 18, 1996, issued December 21, 1999) (“Hill”)
1007	U.S. Patent No. 6,845,398 to Duane Galensky et al. (filed August 2, 1999, issued January 18, 2005) (“Galensky”)

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Exhibit No.	Description of Document
1008	U.S. Patent No. 6,014,694 to Amir Aharoni et al. (filed June 26, 1997, issued January 11, 2000) (“Aharoni”)
1009	U.S. Patent No. 5,867,230 to Feng Chi Wang et al. (filed June 30, 1997, issued February 2, 1999)
1010	U.S. Patent No. 6,637,031 to Phillip A. Chou (filed December 4, 1998, issued October 21, 2003)
1011	Shanwei Cen et al., Flow and Congestion Control for Internet Media Streaming Applications (1997)
1012	Jian Lu, Signal Processing for Internet Video Streaming: A Review (2000)
1013	H. Schulzrinne et al., Network Working Group Request for Comments: 2326, Real Time Streaming Protocol (RTSP) (1998)
1014	U.S. Patent No. 7,529,806 to Yevgeniy Eugene Shteyn (filed November 4, 1999, issued May 5, 2009)
1015	U.S. Patent No. 5,721,878 to Hal Hjalmar Ottesen et al. (filed June 7, 1995, issued February 24, 1998)
1016	R. Fielding et al., Hypertext Transfer Protocol -- HTTP/1.1 (1999)
1017	Sam Iren and Paul D. Amer, The Transport Layer: Tutorial and Survey (1999)
1018	U.S. Patent No. 5,793,980 to Robert D. Glaser et al. (filed November 30, 1994, issued August 11, 1998)
1019	M.H. Willebeek-Lemair et al., Bamba – Audio and video streaming over the Internet (1998)
1020	Excerpts from David Austerberry, <i>The Technology of Video and Audio Streaming</i> (2004)
1021	Cannon DV Format, https://web.archive.org/web/19991013131445/http://canondv.com:80/shared/dvinfo/dvinfo2.html (1999)

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Exhibit No.	Description of Document
1022	Alan T. Wetzel and Michael R. Schell, Consumer Applications of the IEEE 1394 Serial Bus, and a 1394/DV Video Editing System (1996)
1023	U.S. Patent No. 5,568,192 to Eric C. Hannah (filed August 30, 1995, issued October 22, 1996)
1024	U.S. Patent No. 5,402,170 to Kenneth A. Parulski et al. (filed August 31, 1992, issued March 28, 1995)
1025	Jean-Phillipe Martin-Flatin, Push vs. Pull in Web-Based Network Management (1999)
1026	Lixin Gao et al., Catching and Selective Catching: Efficient Latency Reduction Techniques for Delivering Continuous Multimedia Streams (1999)
1027	U.S. Patent No. 5,822,524 to Huey-Shiang Chen et al. (filed July 21, 1995, issued October 13, 1998)
1028	Sriram S. Rao et al., Comparative Evaluation of Server-push and Client-pull Architectures for Multimedia Servers (1996)
1031	'594 Patent Prosecution History File
1033	WAG Acquisition, L.L.C.'s Proposed Claim Constructions from <i>WAG Acquisition, L.L.C. v. Google LLC</i> , No. 6:21-cv-00816-ADA (W.D. Tex.), dated February 18, 2022
1039	Brad Hansen, The Dictionary of Multimedia (1997)
1040	Defendants Google LLC and YouTube, LLC's Opening Claim Construction Brief from <i>WAG Acquisition, L.L.C. v. Google LLC</i> , No. 6:21-cv-00816-ADA (W.D. Tex.), Dkt. No. 37, filed March 14, 2022
1041	Declaration of Keith J. Teruya from <i>WAG Acquisition, L.L.C. v. Google LLC</i> , No. 6:21-cv-00816-ADA (W.D. Tex.), filed April 1, 2022
1042	Jonathan C. Soo, Live Multimedia over HTTP (1994)
1043	Plaintiff's Responsive Claim Construction Brief from <i>WAG Acquisition, L.L.C. v. Google LLC</i> , No. 6:21-cv-00816-ADA (W.D. Tex.), filed April 1, 2022

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Exhibit No.	Description of Document
1045	Phil Karn and Craig Partridge, Improving Round-Trip Time Estimates in Reliable Transport Protocols (1988)
1046	Hari Balakrishnan et al., Improving TCP/IP Performance over Wireless Networks (1995)
1047	WAG Acquisition, L.L.C.'s Preliminary Infringement Contentions from <i>WAG Acquisition, L.L.C. v. Google LLC</i> , No. 6:21-cv-00816-ADA (W.D. Tex.), dated November 15, 2021
1048	Nasir Ghani and Sudhir Dixit, TCP/IP Enhancements for Satellite Networks (1999)
1049	S.M. Bellovin, Security Problems in the TCP/IP Protocol Suite (1989)
1050	Ashish Ahuja et al., Performance of TCP over Different Routing Protocols in Mobile Ad-Hoc Networks (2000)
1051	Toshihiko Kato et al., Design of Protocol Monitor Emulating Behaviors of TCP/IP Protocols (1997)
1052	Osama Tolba et al., Pure Java-based Streaming MPEG Player (1999)
1053	Chaman Sabharwal, Java, Java, Java (1998)
1054	Koji Sato et al., Dynamic Multimedia Integration with the WWW (1999)
1055	Simon S.Y. Shim et al., Multimedia Presentation Components in E-commerce (2000)
1056	U.S. Patent No. 5,802,312 to Mihal Lazaridis et al. (filed September 27, 1994, issued September 1, 1998)
1058	Opening Claim Construction Brief of Amazon.com, Inc., Amazon Web Services, Inc., and Amazon.com Services, LLC from <i>WAG Acquisition, L.L.C. v. Amazon.com, Inc.</i> , No. 6:21-cv-00815-ADA (W.D. Tex.), Dkt. No. 37, filed March 11, 2022
1059	Plaintiff's Responsive Claim Construction Brief from <i>WAG Acquisition, L.L.C. v. Amazon.com, Inc.</i> , No. 6:21-cv-00815-ADA (W.D. Tex.), Dkt. No. 38, filed April 1, 2022

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Exhibit No.	Description of Document
1061	U.S. Patent No. 8,122,141 to Harold Edward Price (filed May 10, 2010, issued February 21, 2012)
1062	'594 YouTube Amended Contentions
1067	<i>Curriculum Vitae</i> of Dr. Nathaniel Polish

II. PERSON OF ORDINARY SKILL IN THE ART

18. I am informed by counsel that, under the patent laws in effect before the America Invents Act (“AIA”) of 2011, an assessment of claims of a patent filed before the AIA took effect should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date (i.e., the “time the invention was made”). I understand November 11, 1999 is the priority date asserted by Patent Owner in the concurrent district court litigation. (EX1047, 1.) I do not take a position as to whether any of the claims of the '594 patent are entitled to that priority date.

19. I have been advised that to determine the appropriate level of a person having ordinary skill in the art, I may consider the following factors: (1) the types of problems encountered by those working in the field and prior art solutions thereto; (2) the sophistication of the technology in question, and the rapidity with which innovations occur in the field; (3) the educational level of active workers in the field;

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

and (4) the educational level of the inventor. I am also informed that a person of ordinary skill in the art is presumed to be aware of all of the pertinent prior art.

20. The '594 patent purports to disclose methods and systems for distributing and playing live multimedia data over the Internet.

21. In my opinion and based on the subject matter of the '594 patent, a person of ordinary skill in the art at the time of the alleged invention would have had a bachelor's degree in computer science, computer engineering, or electrical engineering, or the equivalent, and at least two years of work experience in networking or streaming media systems, particularly audio and video, over the Internet. Additional education could have substituted for professional experience, and significant work experience could have substituted for formal education.

22. My opinions regarding the level of ordinary skill in the art are based on, among other things, my more than 40 years of experience in the field of computer science and data streaming technologies, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues, co-workers, students, and employees, both past and present.

23. Although my qualifications and experience exceed those of the

hypothetical person having ordinary skill in the art defined above, I qualify as a person of ordinary skill in the art as of the alleged priority date of November 11, 1999. In addition to being a person of ordinary skill in the art, I have worked closely with many such persons over the course of my career.

III. STATEMENT OF LEGAL PRINCIPLES

A. Claim Construction

24. I understand that a purpose of claim construction is to determine what a person of ordinary skill in the art would have understood the claim terms to mean. Claim terms are generally given their ordinary and customary meaning, which is the meaning that the term would have to a person of ordinary skill in the art in question as of the critical date (i.e., either the earliest claimed priority date (pre-AIA) or the effective filing date (AIA)).

25. I understand that the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification. I understand that the patent specification, under the legal principles, has been described as the single best guide to the meaning of a claim term, and is thus highly relevant to the interpretation of claim terms. Also, I understand for terms that do not have a customary meaning within the art, the specification usually supplies the best

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

context of understanding the meaning of those terms.

26. I further understand that other claims of the patent in question, both asserted and unasserted, can be valuable sources of information as to the meaning of a claim term. Because the claim terms are normally used consistently throughout the patent, the usage of a term in one claim can often illuminate the meaning of the same term in other claims. Differences among claims can also be a useful guide in understanding the meaning of particular claim terms.

27. I understand that the prosecution history can further inform the meaning of the claim language by demonstrating how the inventors understood the invention and whether the inventors limited the invention in the course of prosecution, making the claim scope narrower than it otherwise would be. Extrinsic evidence, such as dictionaries, may also be consulted in construing the claim terms.

28. I understand that, in *Inter Partes* Review (IPR) proceedings, a claim of a patent shall be construed using the same claim construction standard that would be used to construe the claim in a civil action filed in a U.S. district court (which I understand is called the “*Phillips*” claim construction standard), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.

29. I have been instructed by counsel to apply the “*Phillips*” claim construction standard for purposes of interpreting the claims in this proceeding, to the extent they require an explicit construction. The description of the legal principles set forth above thus provides my understanding of the “*Phillips*” standard as provided to me by counsel.

B. Obviousness

30. I understand that a patent claim is obvious if, as of the critical date (i.e., either the earliest claimed priority date (pre-AIA) or the effective filing date (AIA), it would have been obvious to a person having ordinary skill in the field of the technology (the “art”) to which the claimed subject matter belongs.

31. I understand that the following factors should be considered in analyzing obviousness: (1) the scope and content of the prior art; (2) the differences between the prior art and the claims; and (3) the level of ordinary skill in the pertinent art. I also understand that certain other facts known as “secondary considerations” such as commercial success, unexplained results, long felt but unsolved need, industry acclaim, simultaneous invention, copying by others, skepticism by experts in the field, and failure of others may be utilized as indicia of nonobviousness. I understand, however, that secondary considerations should be connected, or have a “nexus,” with the invention claimed in the patent at issue.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

32. I understand that a reference qualifies as prior art for obviousness purposes when it is analogous to the claimed invention. The test for determining what art is analogous is: (1) whether the art is from the same field of endeavor, regardless of the problem addressed, and (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

33. I understand that a person of ordinary skill in the art is assumed to have knowledge of all prior art. I understand that one skilled in the art can combine various prior art references based on the teachings of those prior art references, the general knowledge present in the art, or common sense. I understand that a motivation to combine references may be implicit in the prior art, and there is no requirement that there be an actual or explicit teaching to combine two references. Thus, one may take into account the inferences and creative steps that a person of ordinary skill in the art would employ to combine the known elements in the prior art in the manner claimed by the patent at issue. I understand that one should avoid "hindsight bias" and ex post reasoning in performing an obviousness analysis. But this does not mean that a person of ordinary skill in the art for purposes of the obviousness inquiry does not have recourse to common sense.

34. I understand that when determining whether a patent claim is obvious

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

in light of the prior art, neither the particular motivation for the patent nor the stated purpose of the patentee is controlling. The primary inquiry has to do with the objective reach of the claims, and that if those claims extend to something that is obvious, then the entire patent claim is invalid.

35. I understand one way that a patent can be found obvious is if there existed at the time of the invention a known problem for which there was an obvious solution encompassed by the patent's claims. I understand that a motivation to combine various prior art references to solve a particular problem may come from a variety of sources, including market demand or scientific literature. I understand that a need or problem known in the field at the time of the invention can also provide a reason to combine prior art references and render a patent claim invalid for obviousness. I understand that familiar items may have obvious uses beyond their primary purpose, and that a person of ordinary skill in the art will be able to fit the teachings of multiple prior art references together like the pieces of a puzzle. I understand that a person of ordinary skill is also a person of at least ordinary creativity. I understand when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this finite number of predictable solutions leads to the anticipated

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

success, I understand that the invention is likely the product of ordinary skill and common sense, and not of any sort of innovation. I understand that the fact that a combination was obvious to try might also show that it was obvious, and hence invalid, under the patent laws. I understand that if a patent claims a combination of familiar elements according to known methods, the combination is likely to be obvious when it does not more than yield predictable results. Thus, if a person of ordinary skill in the art can implement a predictable variation, an invention is likely obvious. I understand that combining embodiments disclosed near each other in a prior art reference would not ordinarily require a leap of inventiveness.

36. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from the knowledge of a person having ordinary skill in the art, or from sources such as explicit statements in the prior art. I understand that when there is a design need or market pressure, and there are a finite number of

predictable solutions, a person of ordinary skill may be motivated to apply common sense and his skill to combine the known options in order to solve the problem.

37. I understand the following are examples of approaches and rationales that may be considered in determining whether a piece of prior art could have been combined with other prior art or with other information within the knowledge of a person having ordinary skill in the art:

- (1) Some teaching, motivation, or suggestion in the prior art that would have led a person of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention;
- (2) Known work in one field of endeavor may prompt variations of it for use in the same field or a different field based on design incentives or other market forces if the variations would have been predictable to a person of ordinary skill in the art;
- (3) Combining prior art elements according to known methods to yield predictable results;
- (4) Applying a known technique to a known device, method, or product ready for improvement to yield predictable results;

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

- (5) Applying a technique or approach that would have been “obvious to try” (choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success);
- (6) Simple substitution of one known element for another to obtain predictable results; or
- (7) Use of a known technique to improve similar products, devices, or methods in the same way.

38. I understand that, when determining whether a claimed combination is obvious, the correct analysis is not whether one of ordinary skill in the art, writing on a blank slate, would have chosen the particular combination of elements described in the claim. Instead, I understand the correct analysis considers whether one of ordinary skill, facing the wide range of needs created by developments in the field of endeavor, would have seen a benefit to selecting the combination claimed.

39. I understand that the test for obviousness is not whether the features of a secondary reference may be bodily incorporated into the structure of the primary reference. The test for obviousness, in other words, is not whether the references could be physically combined but whether the claimed inventions are rendered obvious by the teachings of the prior art as a whole.

IV. TECHNOLOGY BACKGROUND

40. The concepts underlying the '594 patent have long been known to practitioners and scholars in the field of information retrieval.

A. Streaming Media over the Internet

41. Sending streaming media, such as real time audio, over computer networks has been known since the 1980s, at least with Xerox Parc's Etherphone. This technology quickly developed into the beginnings of efficient streaming media data from a source location to client devices, such as user computers in the early 1990s. Streaming media arose as an alternative to the "download-first-and-then-play" technique, which requires a client device to download a file in its entirety and then begin playing the file only after the entire file has been downloaded onto the client device. (*E.g.*, EX1009, 1:38-49.) In contrast, "[i]n streaming, audiovisual data is packetized, delivered over a network, and played as the packets are being received at the receiving end, as opposed to being played only after all the packets have been downloaded." (EX1010, 2:9-13.)

42. Networks and compression have worked in tandem in enable different sorts of approaches to streaming. Over the years improvements in network performance has enabled better streaming. Improvements in compression, when enabled by sufficient processing power, has enhanced those improvements. None

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

of these improvements come without certain costs. For example, compression means that losing even a small amount of the transmitted data can make all of the data useless for a certain amount of time. Also, at higher bandwidths sometimes network latency can be highly variable. From the early 1990s to the present time, these issues have evolved until they have reached the current, highly functional, state of development.

43. Streaming increased in prevalence due to the proliferation of the Internet; users typically did not have access to fast Internet connections, which made downloading large media data files cumbersome. (*E.g.*, EX1010, 2:13-18.) Streaming media data allows users to begin playing out the media, whether that be video or audio, before the entire file is transmitted. (*Id.*) By the late 1990s, “[t]he real-time distribution of continuous audio and video data via streaming multimedia applications account[ed] for a significant, and expanding, portion of the Internet traffic.” (EX1011, 250.) Many commercial streaming media players were released around this time, including “RealAudio and RealVideo players, Microsoft Netshow, and Vxtreme.” (*Id.*)

44. Streaming media data is typically achieved by breaking the data into segments and sending each segment individually and sequentially to a client device, where each segment can be downloaded and the data played. Many forms of

compression start with an initial state and then encode differences to reduce the amount of data required.

45. Packets are a concept from the field of networking. Usually a packet is the smallest separately transmitted or routed unit of data within a network. In the Internet Protocol each packet contains an address and a fixed amount of data. Any larger amounts of data that need to be transmitted on the network are broken into multiple packets and separately transmitted.

46. The server that stores the media data “breaks the media into packets that are routed and delivered over the network.” (EX1012, 1.) “A series of time-stamped packets is called a stream.” (*Id.*) “Packetizing” media data was well known in the art before 1999. (*E.g.*, EX1013, 7 (“When using RTP, a stream consists of all RTP and RTCP packets created by a source...”); EX1014, 1:62-66 (“It is an object of the invention to provide an open architecture solution for content delivery in a download approach that allows for a low or negligible play-out latency. To this end the content file is split into multiple parts. Each part or segment requires a relatively short download time.”); (EX1010, 2:9-13 (“[i]n streaming, audiovisual data is packetized...”)).)

47. Data packets containing the media data are typically sequentially ordered so the segments can be properly reassembled and the stream can be played

in the proper order. For example, U.S. Patent No. 5,721,878 to Otteson teaches “an apparatus and method for effectuating local reception and processing of source program signals representative of a multimedia program received from a remote multimedia server. The multimedia server transmits a selected multimedia program as a **custom ordered series** of discrete, digitally compressed program segments...” (EX1015, 3:21-28 (emphasis added).) HTTP and RTSP, common internet protocols that were the backbone of streaming media data, likewise permitted requesting clients to request specific packets by identifying the packets, either by byte range or time range. (EX1016, 85-86 (HTTP); EX1013, 42 (RTSP).) Identifying packets of media data in order was therefore well known in the art by at least 1998.

B. Internet Transfer Protocols

48. Perhaps the most widely used and well-known transport protocol – through which different users or servers communicate over the Internet – is the Transmission Control Protocol, or TCP. (EX1017, 362 (“[T]he most widely used transport protocols today are TCP and its companion transport protocol, the User Datagram Protocol (UDP).”).) HTTP, which, as of 1999, is a common application layer protocol, uses TCP as its transport protocol. (*Id.*, 363.) TCP provides “a connection-oriented...byte-stream, no loss, no-duplicates, ordered transport service[.]” (*Id.*, 367.) TCP is thus a reliable transport protocol. (*E.g.*, EX1018, 7:9-

17 (“In operation, the TCP/IP protocol...is used to automatically regulate the flow of data between the server and the subscriber.”); EX1019, 271-72 (“TCP/IP...provides reliable end-to-end network connections...”)).

49. TCP transports streams from the source to their destination as fast as possible, taking into account bandwidth, congestion, and other factors. (*E.g., id.*, 272 (“If selected, the Bamba clip is transferred to the client (browser station)_ as fast as TCP/IP can move it...”); EX1020, 209 (“When a browser requests a web page, the files are delivered as fast as the network connection allows...TCP starts at a low rate then ramps up to the maximum that can be achieved.”), 211 (“TCP/IP is used as the transport layer over the Internet. The files are downloaded to the web browser cache as fast as the system allows.”)).

C. Digital Audio and Video

50. By the late 1990s, it was well known that video and audio data could be captured by a camcorder or other device and transmitted as digitally encoded data. In the mid-1990s, several commercial video camera manufacturers collaborated to create the DV, or digital video, standard, which was widely adopted by the camera and video industry. (*E.g., EX1021, 3.*) Subsequently, video cameras that recorded video in digital format were common in the late 1990s. For instance, “[i]n the fall of 1995 Sony introduced two DV camcorders, the DCR-VX100 and DCR-

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

VX700[,]” both of which “meet the HD-DVC requirements for SD...format digital camcorders.” (EX1022, 7.) As another example, U.S. Patent No. 5,568,192, entitled “Method and Apparatus for Processing Digital Video Camera Signals,” teaches a “relatively simple digital video [camera]” that creates “raw, unprocessed digital signals [that] are then transmitted as ‘high speed digital bit stream out’” to a computer. (EX1023, 4:1-16.)

51. Transmitting digital data from a device to a computer requires a specific interface. The SCSI or IEEE 1394 interfaces were common digital input interfaces, which enabled digital capture devices to transmit digital data to computers. (E.g., EX1024, 3:10-25 (“[A]n electronic camera 10 is coupled to a personal computer 12 through a computer interface 14....The interface 14 depends on the interface standard used....a digital interface standard may be used; for instance; if the Small Computer System Interface (SCSI) is used, the interface 14 is a SCSI interface board 14b.”); EX1022, 0002 (“Consumer DV products incorporating 1394 are now on the market....The 1394 interface used by DV equipment brings many benefits to a video editing system.”).) It was therefore known in the art that digital camcorders could transmit digital media data to a computer.

52. Digital cameras were desirable because they created superior media

quality and performance. As several commentators observed, “[t]he digital recording and playback, combined with the digital 1394 interface, completely eliminates the noise pick-up and signal losses that occur in analog systems and interconnects.” (EX1022, 0002.) Furthermore, “[t]he digital record/playback avoids the audio flutter and video picture jitter that is common in present analog systems.” (*Id.*) Finally, “[t]he 1394/DV combination also provides for virtually lossless audio/video editing and dubbing.” (*Id.*) The field thus recognized the advantages and benefits offered by digital media data.

D. Pull-Based Streaming Systems

53. Servers responsible for streaming media data to clients must be triggered in some manner to transmit the data stream. In the 1990s, there existed two general models: the pull model and the push model. The differences can be succinctly described with a “newspaper” metaphor. (EX1025, 6.) You can acquire a newspaper in two ways. You can travel to a newsstand or grocery store to buy the newspaper each morning; or you can subscribe to the newspaper once and expect a paper to be delivered each day thereafter. The former is analogous to the “pull” model of data streaming, while the latter is analogous to the “push” model. (*Id.*) In the “pull” model, “the client sends a request to the server, then the server answers....This is functionally equivalent to the client ‘pulling’ the data off the

server.” (*Id.*) In the “pull” method, “the data transfer is always initiated by the client,” in the same way that traveling to a newsstand to purchase a newspaper is always initiated by the purchaser. (*Id.*)

54. Pull-type streaming systems were well known in the art in the 1990s. (*E.g.*, EX1026, 3 (describing well-known “client-pull” and “server-push” techniques “for video delivery”); EX1027, Abstract (“A method in computer networks in which a client machine (playback client computer) requests multimedia files, such as compressed video clips, from a server (storage server computer).”).)

55. Pull-type streaming systems have several advantages over push-type systems. Pull-type systems provide better control to clients. By controlling when it receives data packets, the client avoids situations in which it receives too much data, leading to overload, or situations when it does not receive enough data, leading to playback lags. Another advantage of the pull-type system is it delegates responsibility of deciding when data should be sent to the client, and removes this responsibility from the server. The server is therefore “relatively simple and maintains very little state information.” (EX1028, 0003.) The benefits of the pull-type streaming system were thus well understood by skilled artisans as early as the 1990s.

V. THE '594 PATENT

A. Overview of the Specification

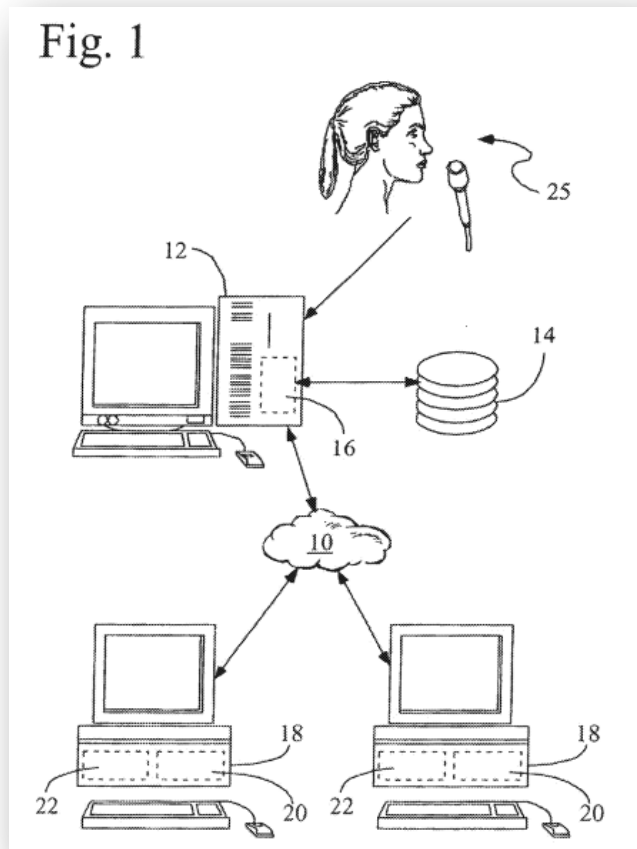
56. The '594 patent states it “relates to multimedia computer communications systems[,]” and in particular “to systems and methods for delivering streaming media, such as audio and video, on the Internet.” ('594 patent, 1:52-55.) According to the '594 patent, while streaming audio or video data, via the Internet, from a host server to a client computer “as a stream of [audio or] video frames” is known in the art, users nonetheless “often encounter interruptions, due to the frequency of unanticipated transmission delays and losses that are inherent in many Internet protocols.” (*Id.*, 2:16-18, 2:34-40.) One solution to these so-called “dropouts” is using “a pre-buffering technique to store up enough audio or video data in the user’s computer.” (*Id.*, 2:42-45.) Once the user’s buffer is sufficiently full of audio or video data, the user may begin playing the data. (*Id.*, 2:50-62.) As the user plays out the stored data, thus depleting the user’s buffer, the server supplies additional audio or video data to subsequently replenish the user’s buffer with data. (*Id.*, 2:62-3:4.)

57. According to the '594 patent, “audio or video data is [typically] delivered from the source [to the client computer] at the rate [the data] is to be played out.” ('594 patent, 2:63-3:1.) If dropouts occur, however, such that the server stops

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

sending data to the user's buffer while the user continues to play already stored data, the buffer will empty faster than it can be replenished. (*Id.*, 3:5-41.) If too many dropouts occur, "the user's media player buffer becomes exhausted" and "stops attempting to play" the audio or video data. (*Id.*)

58. The '594 purports to solve this problem with a system that sends "initial streaming media elements to the user system at a sending rate more rapid than the playback rate, to fill the user buffer[.]" ('594 patent, 3:59-64.) Specifically, the '594 patent discloses "a system for distributing via the Internet streaming media composed of a plurality of time-sequenced data elements." (*Id.*, 6:30-32.) This system includes "at least one user computer **18**...connected to the server **12** via the Internet **10**." (*Id.*, 6:48-51.) The server has a server buffer **14** and a buffer manager **16**, and each user computer **18** likewise includes a user buffer **20**. (*Id.*, 6:30-7:5.)



(*Id.*, Fig. 1.)

59. According to the '594 patent, media data is provided by a “live source, shown as [25] in FIG. 1,” such as a microphone. ('594 patent, 6:42-44.) The media data, such as audio or video data, is broken up into sequential data elements. (*E.g.*, *id.*, 6:59-7:5, 7:15-35, 9:36-45, 10:5-17.) The data elements are stored within buffer 14 in server 12. (*Id.*, 6:35-41.) Once a user computer connects to server 12, “server 12 sends the media data to the user computer[.]” (*Id.*, 8:12-15.) The sending of media data elements may be accomplished by “[a] data communications transport

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

mechanism, such as the TCP protocol[.]” (*Id.*, 8:36-40.) The ’594 patent explains that “media data...may be sent at a higher than playback rate[.]” (*Id.*, 8:59-63.) This is “distinctly different from prior art,” the ’594 patent contends, “in which media data is only sent from the server **12** to the user computer **18** at the rate at which it is to be played out.” (*Id.*, 8:64-67.)

60. The user may play the audio/video data “as soon as the user connection is made to the audio server **12** and a minimal amount of data elements have been received and stored in the user’s buffer **20**.” (’594 patent, 9:40-45.) The ’594 patent explains the user’s buffer “is built up while the media is playing[.]” and each data element that is played is removed from the user’s buffer to make room for new data elements. (*Id.*)

B. Prosecution History

61. The ’594 patent was filed on October 3, 2016, and claims priority to a number of earlier filed patent applications. The earliest application to which the ’594 patent claims priority is provisional application 60/231,997, filed on September 12, 2000.

62. During prosecution of the ’594 patent, the examiner issued a non-final rejections of claims 1-17 as unpatentable over Bommaiah et al. (U.S. Pat. No. 6,708,213), Hooper et al. (U.S. Pat. No. 5,414,455), Hodgkinson et al. (U.S. Pat. No.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

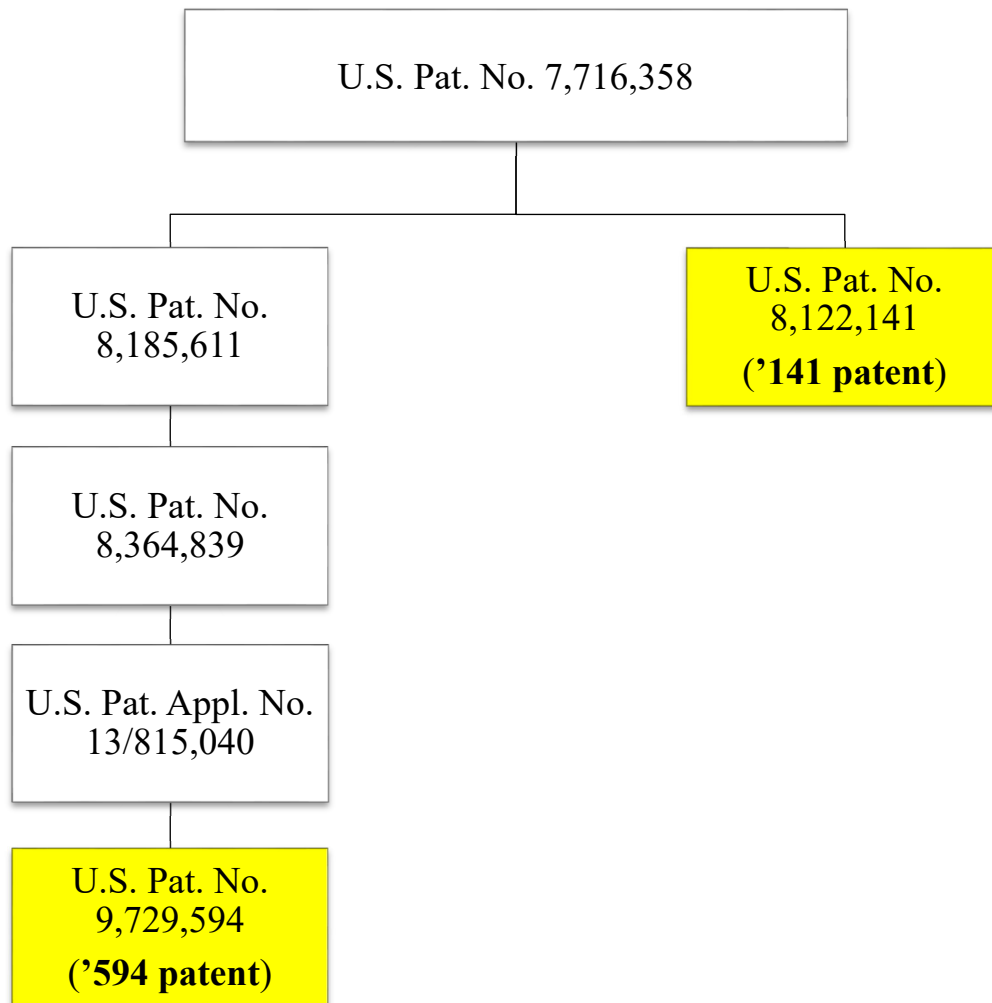
7,209,437), Shteyn (U.S. Pat. No. 7,529,806), and Chen et al. (U.S. Pat. No. 5,822,524). (EX1031, 0172-0182.)

63. In response, the applicant amended claims 1, 6, and 11 to recite the media player “automatically” sends requests for subsequent media data elements. (EX1031, 0140-0147.) The examiner subsequently issued a Notice of Allowability for claims 1-17. (*Id.*, 0024-0029.)

C. Related *Inter Partes* Review Proceedings

64. As I mentioned earlier, the '594 patent claims priority to, and is a continuation of, a number of earlier filed patent applications. One such patent application is No. 10/893,814 (“’814 application”), filed on July 19, 2004. The '814 application eventually issued as U.S. Pat. No. 7,716,358 (“’358 patent”). The '358 patent has several branches of continuation applications. One branch led to the '594 patent. Another branch resulted in U.S. Pat. No. 8,122,141 (EX1061, “’141 patent”). The relationship between the '594 patent and the '141 patent is illustrated below.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2



65. The '141 patent shares many similarities with the '594 patent. Its title is “Streaming Media Buffering System,” and it describes its alleged invention as “a buffering system for streaming media, such as audio/video, on the Internet.” ('141 patent, 1:30-33.) The process disclosed in the '141 patent is similar to that disclosed in the '594 patent. The '141 patent explains a server “transmits audio/video data as sequential data elements from its buffer” to the buffer of the client computer. (*Id.*, 11:31-34.) The '141 patent additionally explains that “audio/video data is

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

transmitted from the server faster than it is played out by the user system, thus building up audio/video data in the user buffer.” (*Id.*, 11:56-65.) The ’141 patent’s claims include many elements and limitations that are found in the ’594 patent’s claims, such as “sequential media data elements,” “digitally encoded audio or video program,” “serial identifiers,” and a sending rate “more rapid than the rate at which said streaming media is played back by a user.” (*Id.*, cl. 10.)

66. The ’141 patent was subject to an *inter partes* review. *See WebPower, Inc. v. WAG Acquisition, LLC*, IPR2016-01238 (P.T.A.B. June 21, 2016). The petition for IPR argued the ’141 patent’s claims 10-23 were anticipated or rendered obvious by U.S. Pat. No. 6,389,473 to Carmel et al. (“Carmel”). *WebPower, Inc. v. WAG Acquisition, LLC*, IPR2016-01238, Paper 1 at 4-5 (P.T.A.B. June 21, 2016). The Board instituted review, *id.*, Paper 7 (P.T.A.B. Jan. 4, 2017), and issued a Final Written Decision invalidating claims 10-23 as anticipated or rendered obvious by Carmel. (*Id.*, Paper 22 (P.T.A.B. Dec. 26, 2017).)

67. On appeal, the Federal Circuit reversed the Final Written Decision as to claims 10-18 and remanded for further consideration in light of the Federal Circuit’s construction of the term “rate” in claim 10. Claim 10 recites instructions that cause the server “to send media data elements to the user system...at a *rate* more rapid than the rate at which said streaming media is played back by a user.” (’141

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

patent, cl. 10 (emphasis added).) The Federal Circuit concluded the term “rate” refers to “the rate at which each requested media data element is transmitted from the server to the user computer.” *WAG Acquisition, LLC v. WebPower, Inc.*, 781 F. App’x 1007, 1011-12 (Fed. Cir. 2019).

68. On remand, the Board considered the Federal Circuit’s construction of “rate” and nonetheless found that claims 10-18 remained unpatentable over Carmel. The Board concluded Carmel disclosed multiple ways of sending media data elements “at a rate more rapid than the rate at which said streaming media is played back by a user.”

69. *First*, Carmel teaches that during normal operations, the system “may use a data rate that is ‘faster’ than the rate at which the data are generated by the transmitting computer.” *WebPower*, IPR2016-01238, Paper 28 at 18 (P.T.A.B. July 16, 2020) (referencing Carmel, 2:51-59). The Board concluded because Carmel teaches sending at least some media data elements at a more rapid rate than the playback rate, Carmel discloses “the last limitation of claim 10 under...normal operation.” *Id.* at 18-21.

70. *Second*, Carmel teaches a lag-recovery mechanism where the user can select a lower quality level for each media data element. *Id.* at 21. When a user selects a lower quality level in response to a lag, “the transmission of the media data

elements is faster than the media being played.” *Id.* Therefore, Carmel’s disclosure of a “lag-recovery mechanism” “meets the last limitation of claim 10[.]” *Id.* at 22-23.

71. The Board thus issued a Final Written Decision that claims 10-18 of the ’141 patent were invalid in light of Carmel.

D. The Challenged Claims

72. My declaration addresses claims 1-17, of which claims 1, 6, and 11 are independent. Representative claim 1 recites:

1. A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the method comprising:
 - [a(i)] sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,
 - [a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,
 - [a(iii)] each request specifying one or more serial identifiers of the media data elements requested;
 - [b(i)] receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and
 - [b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media

player allows;

- [c] storing the received media data elements in a memory of the media player;
- [d] playing the received media data elements in series from the memory of the media player; and
- [e] as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.

(’594, 16:30-57 (Claim 1; bracketed notations (e.g., [a]) added).) Additional details about the claims are provided in my analysis in **Part VI** below.

VI. APPLICATION OF THE PRIOR ART TO CHALLENGED CLAIMS

73. I have reviewed and analyzed the prior art references and materials listed in Part I.B above. In my opinion the claims of the ’594 patent are rendered obvious based on the following prior art.

Ground	Claims	Basis for Challenge under §103
1	1-17	Obviousness over Hill
2	1-17	Obviousness over Hill in view of Aharoni
3	1-17	Obviousness over Hill in view of Ravi
4	1-17	Obviousness over Carmel in view of Hill
5	1-17	Obviousness over Carmel in view of Galensky

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Ground	Claims	Basis for Challenge under §103
6	1-17	Obviousness over Carmel in view of Hill and Ravi
7	1-17	Obviousness over Carmel in view of Galensky and Ravi

74. I am informed by counsel that each of the references cited in the grounds above qualifies as prior art to the challenged claims because each reference was filed and/or published before the '594 patent critical date.

A. Brief Summary of Prior Art

1. Carmel [EX1003]

75. **Carmel**, U.S. Patent No. 6,389,473, is titled “Network Media Streaming.” Carmel “relates generally to network data communications, and specifically to real-time multimedia broadcasting over a network.” (Carmel, 1:10-12.) I am informed Carmel qualifies as prior art to the '594 patent because it was filed on March 24, 1999, which is several months before the '594 patent's alleged November 11, 1999 priority date.

76. Carmel discloses a method for generating a stream of media data and broadcasting the stream via a network (such as the Internet) to client devices. Carmel discloses a system **32** comprising “a transmitting computer **34**, which generates the sequence, a plurality of clients **30**, and a network server **36**, all which communicate over network **28**, preferably using the well-known Internet protocol[.]” (Carmel,

6:24-35.)

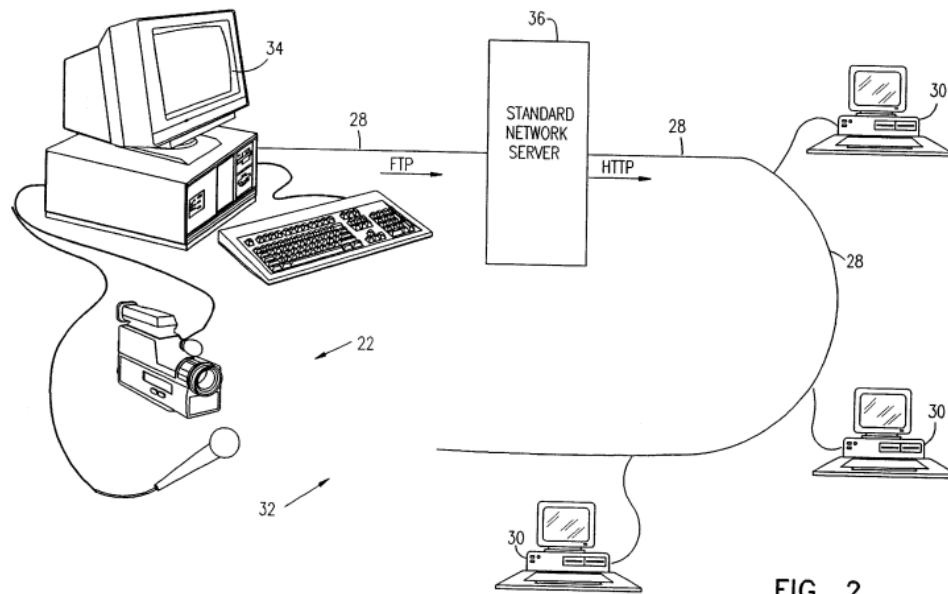
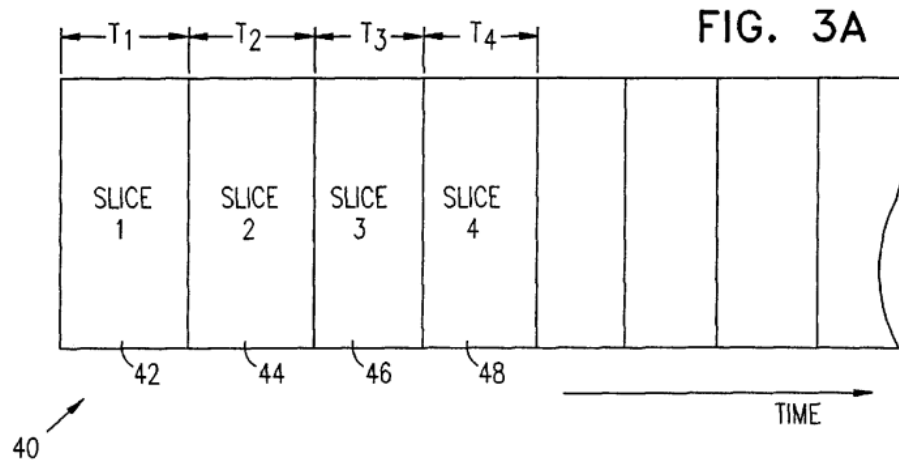


FIG. 2

(*Id.*, Fig. 2.)

77. The streaming system disclosed by Carmel operates in the following exemplary manner. Carmel teaches computer 34 “receives audiovisual input from input devices 22,” such as a microphone or video camera. (Carmel, 6:32-34.) Computer 34 produces “a stream of broadcast data 40...typically corresponding to a multimedia data sequence[.]” (*Id.*, 7:18-22.) The data stream 40 “comprises a series of data slices 42, 44, 46, 48, etc.” (*Id.*, 7:22-23.) Each slice has a corresponding slice index and is stored in computer 34’s “running slice index 1, 2, 3...N.” (*Id.*, 7:27-28.)



(*Id.*, Fig. 3A.)

78. The data slices of data stream **40** are stored in server **36** in a “first-in-first-out” sequence. (Carmel, 7:50-58.) Client computers connect to server **36** to “begin[] to download the data stream...to start receiving the data stream substantially in real time[.]” (*Id.*, 8:1-6.) The client can select where in the stream to begin receiving the data: the client can read the index file to determine “the file in data stream **40** that was most recently uploaded by computer **34**[;]” or, alternatively, the client can “choose to begin downloading data stream **40** from an earlier point in time[.]” (*Id.*, 7:59-8:11.)

79. Clients connect to server **36** and begin downloading and consuming data stream **40** using a well-known Internet protocol, such as HTTP, UDP, or other Internet protocols. (Carmel, 7:4-17, 10:24-54.) The client “monitors the rate of data coming in over...its links with server **36**.” (*Id.*, 10:55-56.) Carmel teaches multiple

means of maintaining a consistent and sufficient transmission rate between server **36** and clients **30**. Carmel explains the data rate between server **36** and clients **30** “should be generally equal to or faster than the rate at which the data are generated at the transmitting computer.” (*Id.*, 2:57-59.) Additionally, Carmel teaches clients can “open[] additional links with server **36** in order to increase the overall data rate.” (*Id.*, 10:55-63.) Finally, Carmel discloses users may select lower quality levels for the slices, which permits faster transmission speed. (*Id.*, 11:1-22.)

80. Carmel teaches each client **30** includes a media player through which the media data can be played. Clients **30** “will typically need only add a Java applet or plug-in to their existing Web browsers...in order to receive and play back the broadcast.” (Carmel, 7:14-17.) “The operation of client is controlled by a Java applet[.]” (*Id.*, 10:28-29.) Carmel provides software for a “sample applet” that can be used to download and play back data stream **40**. (*Id.*, 10:33-34, 13:53-14:16.)

81. Additional details about Carmel are provided beginning in the analysis of the '594 patent claim limitations, in Part **VI.C** below.

2. Hill [EX1006]

82. **Hill**, U.S. Patent No. 6,005,600, is titled “High-Performance Player for Distributed, Time-Based Media.” Hill “relates generally to audio/video playback devices, and more particularly to playing movie data[.]” (Hill, 1:7-9.) I am informed

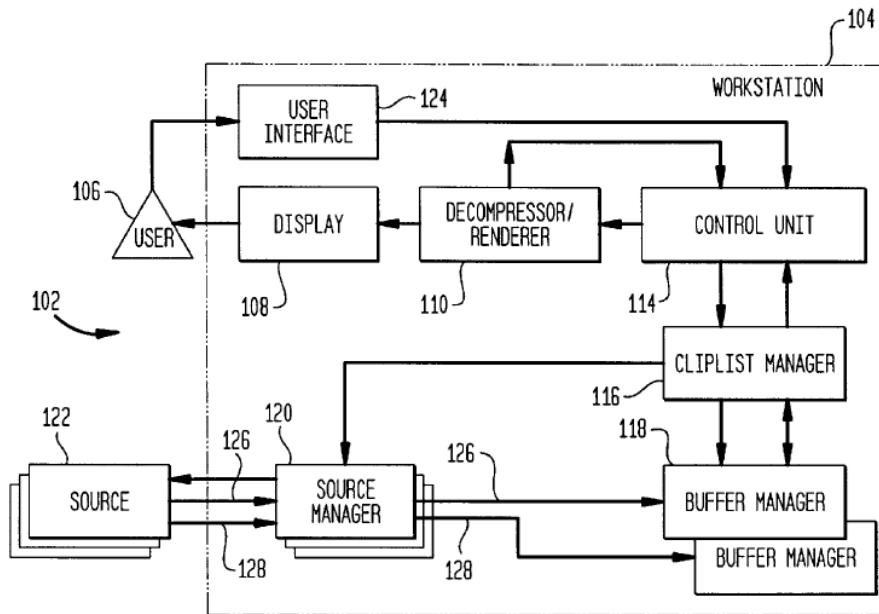
Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Hill qualifies as prior art to the '594 patent because it was filed on October 18, 1996, which is well before the '594 patent's alleged November 11, 1999 priority date.

83. Hill discloses a “system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network.” (Hill, 3:15-18.) The system comprises a workstation **104** that has a media player “to play movie data” via display **108**. (*Id.*, 3:27-33.) Movie data “is broken into fundamental units called frames. For each video frame there is an associated audio frame. A video frame contains the video information to display a single image.” (*Id.*, 3:37-41.) Furthermore, a “clip is a sequence of frames,” so “movie data of interest to a user will be composed of one or more clips.” (*Id.*, 3:46-47.)

84. The media player taught by Hill is comprised of several components. Control unit **114** is “responsible for the overall control of the system” and “receives control information from the user[,]” such as whether to start playing movie data. (Hill, 3:57-64.) Decompressor/renderer **110** decompresses the movie data and presents them to the user via display **108** “in correct synchronization.” (*Id.*, 3:67-4:2.) Control unit **114** feeds frames to decompressor/renderer **110** in the proper sequence. (*Id.*, 4:2-4.)

FIG. 1



(*Id.*, Fig. 1.)

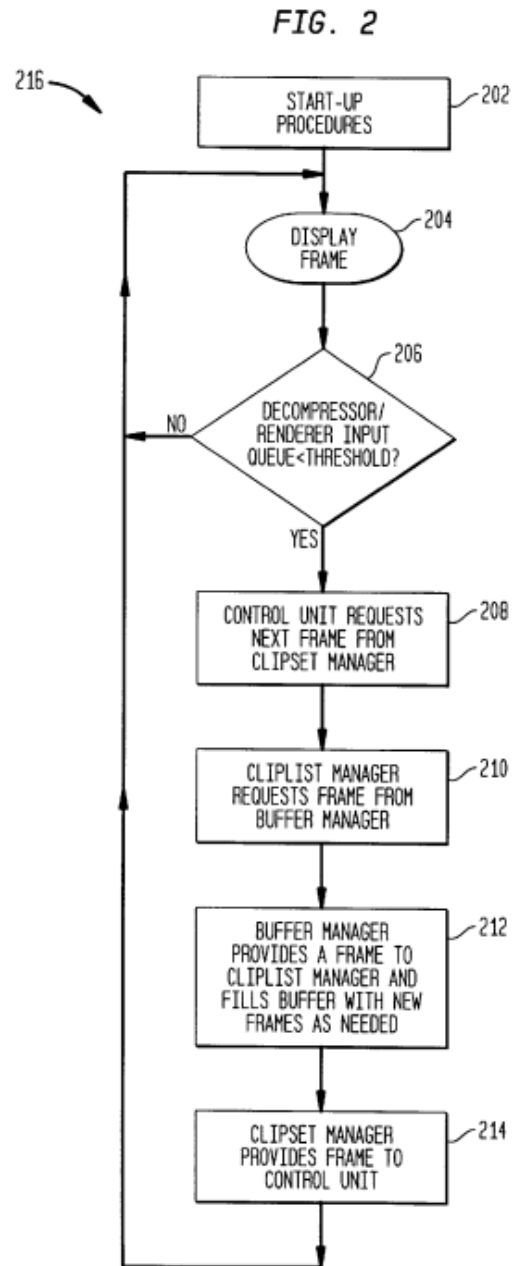
85. Buffer manager **118** “stores a limited number of frames in local digital memory” in order to “always have available the next frame required by the control unit **114** when that frame is requested.” (*Hill*, 4:7-16.) Buffer manager **118** accomplishes this by “prefetching” future frames. (*Id.*) In response to a request from control unit **114**, “buffer manager **118** transfers frames to the control unit **114** via the cliplist manager **116**.” (*Id.*, 4:21-22.) The cliplist manager is responsible for “keep[ing] track of where within the digital network environment **102** each clip is stored.” (*Id.*, 4:22-24.)

86. To replenish its store of frames after providing frames to control unit **114** for playback, buffer manager **118** requests future frames via cliplist manager

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

116. (Hill, 4:24-26.) Cliplist manager **116** then “requests frames from a particular source **122** via the appropriate source manager **120**.” (*Id.*, 4:27-28.) Once the correct frame is located in source **122**, the frame is sent from source **122** to source manager **120** and then to buffer manager **118** to store. (*Id.*, 4:27-37.)

87. Movie data is played in the follow exemplary manner. Control unit **114** “initiates start-up procedures when the user first requests playback of movie data.” (Hill, 4:59-61.) Start-up procedures involves filling the buffer manager with frames and “passing the first frame to the control unit **114**,” which “causes the first frame to be displayed to the user[.]” (*Id.*, 4:61-5:3.) When the decompressor/renderer **110** is ready for the next frame, control unit requests the frame from cliplist manager **116**, which in turn requests the next frame from buffer manager **118**. (*Id.*, 5:4-10.) Buffer manager **118** provides the frame to cliplist manager **116**, which provides the frame to control unit **114**. The process then repeats. (*Id.*, 5:10-21.) This process is generally depicted in Figure 2.



88. As buffer manager **118** provides frames to control unit **114** via cliplist manager **116**, it must replenish its store of frames. The buffer manager **118** tracks

the buffer fill level, which is “the number of future frames currently being stored in the buffer manager **118**,” and the request threshold, which “represents the optimum number of future frames stored in the buffer at any given time[.]” (Hill, 6:58-67.) If the buffer fill level drops below the request threshold, buffer manager requests new frames. (*Id.*, 7:4-10.)

89. The buffer manager **118** “sends requests for new frames to the cliplist manager **116**.” (Hill, 5:66-6:8.) The buffer manager **118** specifies the desired next frame (the next frame in the sequence) by reference to the frame’s “global frame number[.]” which is the frame’s “position within the global playback sequence[.]” (*Id.*) The cliplist manager **116** “converts the global frame number to a clip number and local frame number[.]” where the “local frame number” is the frame’s “position within the clip from which they originate[.]” (*Id.*, 6:1-12.) The cliplist manager **116** requests the next frame from source manager **120**, specifying the desired frame by clip number and local frame number. (*Id.*, 6:13-16.) The source manager **120** retrieves the desired frame from source **122** and passes the frame to buffer manager **118**. (*Id.*, 6:14-34.) In this way, the buffer manager **118** maintains a cache of future frames. (*See generally id.*, 6:58-7:56, 12:7-44.)

90. Additional details about Carmel are provided beginning in the analysis of the ’594 patent claim limitations, in Part **VI.B** below.

3. Ravi [EX1004]

91. **Ravi**, U.S. Patent No. 6,292,834, is entitled “Dynamic Bandwidth Selection for Efficient Transmission of Multimedia Streams in a Computer Network.” Ravi “relates to multimedia communications[,]” and more particularly, “to the efficient and reliable delivery of multimedia streams over a diverse computer network with dynamically variable bandwidth capacity.” (Ravi, 1:56-60.) I am informed Ravi qualifies as prior art to the ’594 patent because it was filed on March 14, 1997, which is over two years before the ’594 patent’s alleged November 11, 1999 priority date.

92. Ravi recognizes that given the proliferation of the internet to connect various users and devices, “the ability to efficiently deliver multimedia data over a diverse computer network such as the internet is severely limited by the reliability and bandwidth capacity of the network connection.” (Ravi, 2:4-8.) Ravi addresses this problem by providing “efficient transmission of multimedia streams from a server to a client computer over a diverse computer network...such as the internet.” (*Id.*, 3:3-7.) “Examples of multimedia streams provided to the client computer include a compressed video stream [and/or] a compressed audio stream[.]” (*Id.*, 3:7-10.)

93. Ravi discloses a system with “a production station **210**, a stream server

220, at least one web server **230** and at least one client computer **240**[.]” (Ravi, 5:46-49.) “Stream server **220** and web server **230** are coupled to client computer **240** via a computer network **290**, e.g., the internet.” (*Id.*, 5:49-51.) A multimedia source is created, for example by a video camera **312**, and stored on stream server **220**. (*Id.*, 6:6-17; Fig. 3.)

94. Ravi teaches an “efficient and reliable streaming of data packets from stream server **220** to client computer **240**, accomplished by optimally utilizing the bandwidth of the connection provided by computer network **290** while minimizing the loss of packets.” (Ravi, 6:32-36.) Ravi discloses how “the transmission rate of the data stream is dynamically adjusted in response to changes in bandwidth made available by computer network **290** for the network connection between server **220** and client computer **240**.” (*Id.*, 6:36-41.) Server **220**, “in response to feedback from client computer **240**, dynamically selects transmission rates in order to better match the varying bandwidth capacity of the network connection.” (*Id.*, 6:41-44.)

95. Ravi teaches several ways the transmission rate can be dynamically updated to match the network’s bandwidth capacity. In one embodiment, Ravi teaches that performance variables are computed and used “to determine if it is desirable” to increase or decrease the transmission rate. (*E.g.*, Ravi, 7:16-11:7.) Additionally, Ravi explains “an average client packet computational rate” is

computed to determine if “the data packets [are] arriving at client computer **240** at a faster, equal or slower rate than client computer **240** is able to decompress and render the data packets.” (*Id.*, 11:8-41.)

96. Ravi thus discloses a streaming method where the transmission rate is adjusted based on the available bandwidth of the connection between the server and the client devices.

4. Aharoni [EX1008]

97. **Aharoni**, U.S. Patent No. 6,014,694, is entitled “System for Adaptive Video/Audio Transport over a Network.” Aharoni “relates generally to transport of video and audio information over networks[,]” and more particularly to “adapting the transport of video and audio information over IP networks having varying bandwidth capacities.” (Aharoni, 1:5-9.) I am informed Aharoni qualifies as prior art to the ’594 patent because it was filed on June 26, 1997, which is over two years before the ’594 patent’s alleged November 11, 1999 priority date.

98. Aharoni recognizes that while the value video content adds to the Internet is tremendous, it is challenging to deliver streaming real time audio and video due to the high bandwidth requirements of video data and variations in the bandwidth available over a connection that can cause packet loss. (Aharoni, 1:21-34.) Aharoni addresses this problem by disclosing a system that functions to

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

“compress, code, decode and decompress video streams that are transmitted over networks having available bandwidths that vary with time and location.” (*Id.*, 2:16-20.) The system can account for varying bandwidths by offering a tradeoff between video resolution, frame rate, and individual frame quality. (*Id.*, 2:23-26.) Aharoni teaches “compressing a raw video source into a plurality of frames” and sending the frames “over the network channel[.]” (*Id.*, 2:54-65.) A video server **18** transmits the audio/video data over network **20** to clients **22**, who decompress the audio/video data for display. (*See generally id.*, 6:33-8:40.)

99. Aharoni teaches its system “has application to any type of network including those that utilize the Internet Protocol (IP) such as the Internet or other TCP/IP networks.” (Aharoni, 2:12-15; *see also id.*, 6:37-40, 6:52-54, 7:31-34 (“The system, utilizing ‘best effort’ protocols such as those found on TCP/IP networks...”), 7:39-43.) Aharoni discloses some frames of video/audio data “are sent using a reliable mechanism. Such a reliable mechanism includes using a network protocol such as TCP[.]” (*Id.*, 9:64-66.) Aharoni thus teaches sending video data from a server to a client device using TCP, which is a reliable transmission protocol.

5. Galensky [EX1007]

100. **Galensky**, U.S. Patent No. 6,845,398, is entitled “Wireless Multimedia

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Player.” Galensky “relates generally to the field of multimedia communications and, more particularly, to a method and device for streaming and/or storing high quality, real time multimedia information over a wireless telecommunications network.” (Galensky, 1:6-11.) I am informed Galensky qualifies as prior art to the ’594 patent because it was filed on August 2, 1999, which is before the ’594 patent’s alleged November 11, 1999 priority date.

101. Galensky recognizes and strives to solve the problem of bandwidth capacity when transmitting media data across a network. (*See generally* Galensky, 6:1-27.) Galensky teaches a system and method “for receiving, playing, and storing streamed multimedia files over a wireless telecommunications network from a multimedia server.” (*Id.*, 3:13-16.) Galensky teaches multimedia server **10** is connected, via telecommunications network **40**, to multiple wireless multimedia devices **80**. (*Id.*, 3:20-63.) Multimedia device **80** is “capable of decoding streamed, compressed data for playback to the user in real time[.]” (*Id.*, 3:52-54.) Galensky teaches “multimedia server **10** [] streams [] selected multimedia files (e.g., digital MP3 audio files, video data files, etc.) [] to the wireless device **80** over link **20** [and] network **40**[.]” (Galensky, 5:37-45.) The multimedia file “is preferably transmitted...to the network **40** in TCP/IP packets[.]” (*Id.*, 5:41-44.)

102. Galensky further discloses that device **80** receives the multimedia file

and “creates a temporary buffer” to ensure “continuous transmission of data so that the multimedia file (e.g., digital MP3 audio file) sounds of high quality.” (Galensky, 5:46-53.) Galensky teaches the buffer helps conserve bandwidth by creating “5-10 seconds of buffer[.]” (*Id.*, 6:2-10.) This buffer also has a “minimum threshold level” of data, such as 2-3 seconds – if the amount of data in the buffer falls below the minimum threshold level, the system increases the rate at which data is sent to the device **80** to build up the data in the buffer to the upper threshold. (*Id.*, 6:10-27.)

103. Aharoni thus teaches a buffering method to provide a continuous transmission of media data to maintain a high quality stream.

B. Ground 1: Claims 1-17 Are Obvious Over Hill.

104. As explained in greater detail below, Claims 1-17 of the '594 patent are rendered obvious by Hill.

1. Claim 1 (Independent)

105. As is explained below, claim 1 is obvious over the prior art. For reference, claim 1 recites (with bracketed notations to delineate its limitations):

1. A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the method comprising:
[a(i)] sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,

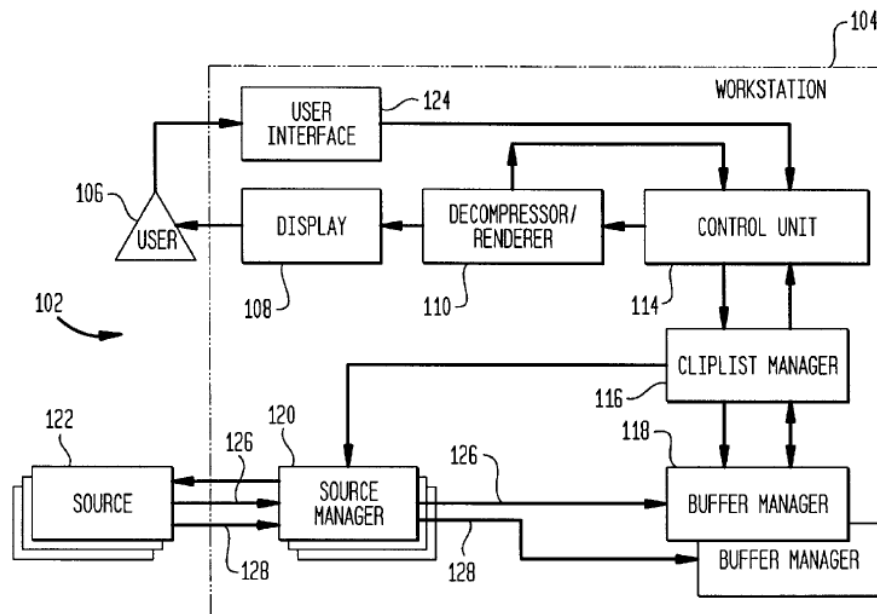
- [a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,
- [a(iii)] each request specifying one or more serial identifiers of the media data elements requested;
- [b(i)] receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and
- [b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;
- [c] storing the received media data elements in a memory of the media player;
- [d] playing the received media data elements in series from the memory of the media player; and
- [e] as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.

(a) **“A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the method comprising:” (claim 1, preamble)**

106. The preamble is disclosed by Hill.

107. Hill discloses its invention “is directed to a system and method for user-controlled playback of synchronized audio and video information where the data is accessed via sources of varying bandwidth capabilities.” (Hill, 2:25-28.) Another way, Hill’s disclosure “is directed to a system and method for enabling a user to view a moving picture with synchronized audio using a workstation connected to a digital network.” (*Id.*, 3:15-18.) Hill’s system comprises a “digital network environment **102**” including a “workstation **104** to play movie data[,]” where the “workstation **104** may be a conventional personal computer, a high performance workstation, or dedicated hardware specifically suited for audio/video playback.” (*Id.*, 3:27-56; *see also id.*, 3:29-32 (“A user **106** utilizes the workstation **104** to play movie data.”).) The audio/video data is “stored in a separate source **122**[,]” and each source “may be any type of digital memory, and may be accessed either locally or via a network server.” (*Id.*, 3:48-56; *see also id.*, 4:35-37 (explaining “[m]ovie data flows between the source **122**” and the workstation **104**).) The source **122** “may be a site on the Internet.” (*Id.*, 3:48-56.) Hill thus discloses a method for operating a media player (*i.e.*, on workstation **104**) to receive and play audio or video program (such as a movie) from a remote media source (*i.e.*, source **122**) via a data connection over the Internet.

FIG. 1



(*Id.*, Fig. 1.)

- (b) “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,” (claim 1[a(i)])

108. Hill discloses “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program.”

109. Hill teaches “media data elements representing the audio or video program.” Hill teaches movie data “is composed of synchronized audio and video information. This information is broken into fundamental units called frames.”

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

(Hill, 3:37-41.) Furthermore, “[a] clip is a sequence of frames. The movie data of interest to a user will be composed of one or more clips.” (*Id.*, 3:46-47.) A person of ordinary skill in the art would understand each “frame” of a movie is a “media data element.” The ’594 patent explains that media data elements are segments of the relevant media data. (*E.g.*, ’594 patent, 6:30-32 (“In one embodiment, the invention provides a system for distributing via the Internet **streaming media composed of a plurality of time-sequenced data elements.**”) (emphasis added), 7:16-35, 8:4-6 (“This buffer may, for example, hold enough data elements for about one minute of play.”), 9:36-38 (“...the server **12** transmits **audio/video data as sequential data elements...**”) (emphasis added), 9:44-45 (“As each data element is played, it is deleted or displaced from the user’s buffer[.]”), 10:9-12 (“In one embodiment, the number of data elements in the server buffer **14** actually never changes, it always maintains one minute of audio/video data.”).) Similarly, Hill teaches each frame is a segment of media data: “Movie data is composed of synchronized audio and video information. This information is broken into fundamental units called frames.” (Hill, 3:37-39.) Furthermore, “[a] video frame contains the video information to display a single image.” (*Id.*, 3:40-41.) The “frames” of movie data described in Hill thus are “media data elements.”

110. Hill discloses the frames (*i.e.*, media data elements) are “serially

identified.” “Frames may either be referenced according to their position within the global playback sequence (i.e., global frame number), or according to their position within the clip from which they originate (i.e., local frame number).” (Hill, 5:66-6:2.) The control unit **114** and buffer manager **118** within the media player “reference frames according to the [global frame number].” (*Id.*, 6:2-4, 5:60-62.) The cliplist manager **116** converts the global frame number to a clip number and local frame number and uses that identifier (clip number and local frame number) to request the frame from the source. (*Id.*, 6:7-16.) The global frame number and local frame number thus identify the frames serially: the frame number increases sequentially as the movie data progresses. The 20th frame occurs later in time than the 10th frame, and so on, exemplarily reflected in Figure 5A, below. Likewise, Hill Figure 7 shows that both the global frame number and local frame number for each clip proceed in a serial fashion from 1 to N. Hill thus teaches “serially identified media data elements.”

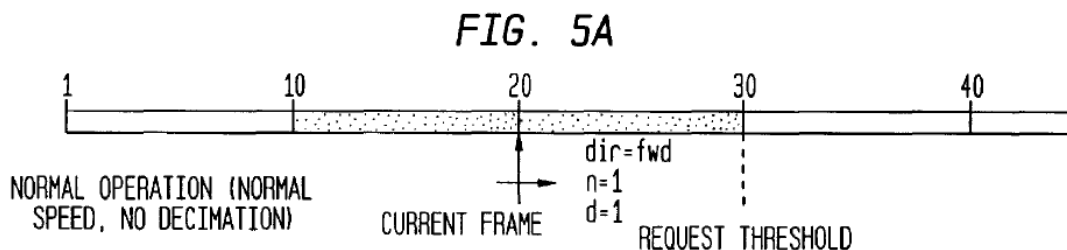
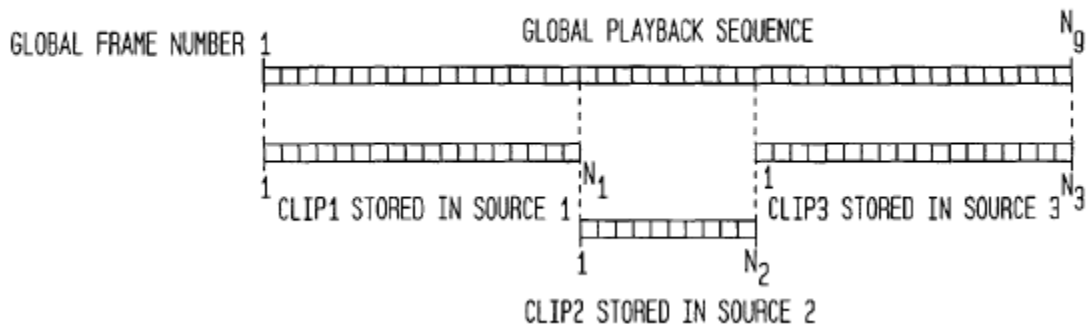


FIG. 7



(*Id.*, Figs. 5A, 7.)

111. Hill further teaches the frames are initially stored in the “media source” or source(s) **122**. A clip is a “sequence of frames[,]” and clips may be stored in the same source or may be “stored in different locations within the digital network environment **102**.” (*Id.*, 3:46-49.) “Each source **122** may be any type of digital memory, and may be accessed either locally or via a network server.” (*Id.*, 3:48-56.) Hill thus teaches the frames of movie data (*i.e.*, media data elements) are stored in source **122** (*i.e.*, media source).¹

¹ I understand in the district court litigation, Petitioner and Patent Owner disagree on the construction of “media source.” Petitioner contends the term means “the storage device or live source device from which the streaming material originates,” while Patent Owner argues the term should be given its plain and ordinary meaning. (EX1040, 8.) I understand the distinction to be Petitioner’s construction excludes

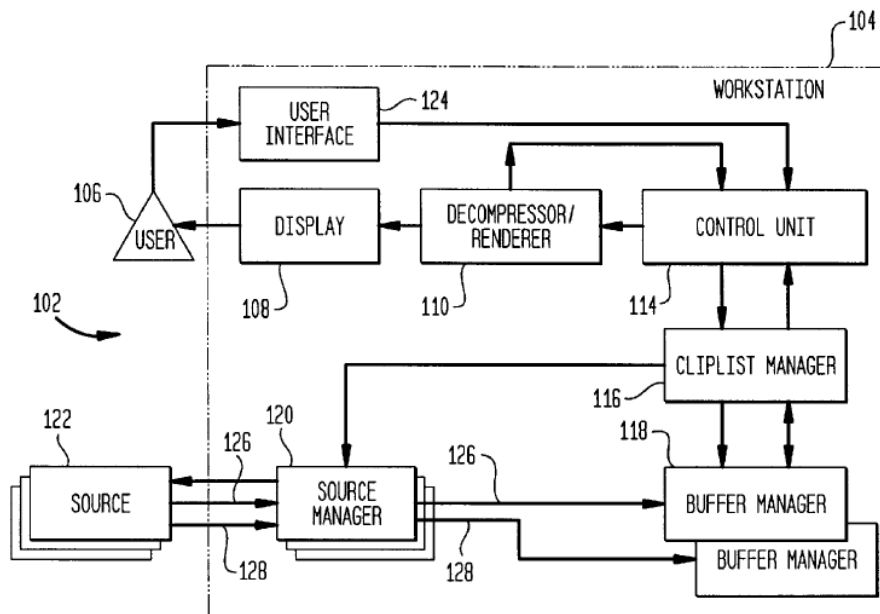
112. Hill finally teaches “sending requests from the media player to the media source via the data connection for one or more” frames. The media player in workstation **104** “contains a control unit **114** responsible for the overall control of the system.” (Hill, 3:57-58.) The control unit **114** “receives control information from the user **106** via the user interface **124**.” (*Id.*, 3:58-60.) If a user wishes to begin viewing the movie data, the control unit **114** “initiates start-up procedures[.]” (*Id.*, 4:59-61.) Frames are provided to control unit **114** from buffer manager **118** via cliplist manager **116**. (*Id.*, 4:7-8.) The buffer manager **118** “stores a limited number of frames in local digital memory, such as random-access memory (RAM) or a local

from the scope of “media source” any intermediate network servers between the media source (which could be a live recording apparatus or a host server that stores the media data) and the user’s computer, while Patent Owner argues the “media source” is “whatever server the user computer connects to so as to obtain the underlying media data.” (EX1058, 3-5; EX1059, 6-7.) This dispute does not affect my analysis. Hill unambiguously discloses the source **122** “may be any type of digital memory, and may be accessed either locally or via a network server. Note that a source may be a site on the Internet.” (Hill, 3:51-56.) Hill thus discloses each parties’ interpretation of “media source.”

disk drive.” (*Id.*, 4:8-11.) “The goal is to always have available the next frame required by the control unit **114** when that frame is requested” by the user. (*Id.*, 4:11-13.)

113. The buffer manager **118** “accomplishes this goal [of always having the next frame ready] by fetching frames likely to be required before they are requested.” (*Hill*, 4:12-15.) To replenish its cache of frames after sending frames to control unit **114** for playback, the buffer manager **118** “must request [additional frames] via the cliplist manager **116**.” (*Id.*, 4:24-26.) The cliplist manager “keep[s] track of where within the digital network environment **102** each clip is stored.” (*Id.*, 4:22-24.) In response to the buffer manager **118**’s request for additional frames, cliplist manager **116** “requests frames from a particular source **122** via the appropriate source manager **120**.” (*Id.*, 4:27-29; *see also id.*, 6:13-16 (“In step **314**, the cliplist manager **116** *requests the frames* from the appropriate source manager **120**. The source manager **120** in step **316** *retrieves the requested frame data from the source 122.*”), *Fig. 3* (steps 308 through 316) (emphasis added).)

FIG. 1



(*Id.*, Fig. 1.)

114. Thus, the media player's cliplist manager **116**, in response to a signal from buffer manager **118**, sends a request for additional frames (*i.e.*, media data elements) to the source **122** (*i.e.*, media source) via the source manager **120**. And as discussed above, the request for additional frames is sent "via a data connection," in this case the Internet. (*E.g.*, Hill, 3:53-56 ("Note that a source may be a site on the Internet."), 3:20-22 (referencing sources can be "network servers").)

115. Hill thus teaches "sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program."

(c) “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,” (claim 1[a(ii)])

116. Hill discloses “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate.”

117. As discussed above, Hill teaches each frame (*i.e.*, media data element) comprises a “portion of the audio or video program.” (*E.g.*, Hill, 3:37-56.) Each frame represents a segment of the media, such as movie data, stored in source(s) **122**. Hill further teaches each frame is “digitally encoded.” Hill consistently refers to the system as a “digital network.” (*E.g.*, Hill, 3:17-18, 3:27-28, 3:48-56, 4:22-24, 4:31-34.) Further, Hill discloses the buffer manager **118** “stores a limited number of frames in local digital memory[.]” (*Id.*, 4:8-11.) A person of ordinary skill in the art would understand that frames of a movie sourced from a digital network and stored in digital memory would be “digitally encoded.” If any doubt remained, Hill’s claim 1 recites “[a] system for playing *digital movie data*....” (*Id.*, cl. 1.) Hill thus discloses the frames are “digitally encoded portion[s] of the audio or video program.”

118. Hill further discloses each frame has a “playback rate.” A person of ordinary skill in the art would understand that the “playback rate” of a media file is the rate at which the media is played at normal speed, from beginning to end, through

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

the computer. (*E.g.*, '594 patent, 5:60-65 (“In conventional systems for streaming media over the Internet, media data (whether real-time or file based) is simply transmitted from the server to the user at **the rate at which it will be played out** (the ‘playback rate’)...”)) (emphasis added).) Hill teaches storing and transmitting video on a frame-by-frame basis. In this context, each frame has a playback rate, providing an example wherein “the play rate is 24 frames per second[.]” (Hill, 7:20-21.) The “play rate” in Hill’s frame-by-frame system is analogous to the “playback rate.” In this scenario, the “playback rate” of each frame is 1/24th of a second. In addition, it would be obvious that each frame would have a playback rate; a person of ordinary skill in the art would understand a playback rate is an inherent property of media data; media data, such as a move, is a function of time, and the speed at which the media plays through time is the playback rate. Hill thus teaches each

frame (*i.e.*, media data element) has a playback rate.^{2,3}

² I understand Patent Owner submitted the expert declaration of Keith J. Teruya in support of its claim construction arguments in the district court litigation. (EX1041.)

I read Mr. Teruya's declaration and noted his opinion that "[a]s used in the specification, a POSITA would understand 'playback rate' to refer to the bitrate of the media data at a normal (1.0) rendition." (EX1041, ¶ 26.) Mr. Teruya's understanding is consistent with my analysis, with the caveat that I would clarify that a "playback rate" need not be defined in terms of bitrate. A person of ordinary skill in the art would understand the rate at which media data is played could be measured in a number of ways, using a number of different metrics well known in the art. As Mr. Teruya explained, "playback rate" is "described by a metric, which is not a data rate itself, but rather a characteristic of how the data is to be played back." (*Id.*, ¶ 24.) The "playback rate" *may* be described in terms of bitrate, but it need not *always* be described in terms of bitrate.

³ I further understand in a separate litigation between Patent Owner and Amazon.com, the parties dispute the construction of "playback rate." *WAG Acquisition, L.L.C. v. Amazon.com, Inc. et al.*, No. 6:21-cv-00815 (W.D. Tex. 2021). Amazon.com argues the term means "a rate at which the data is encoded for playback

119. Hill thus discloses “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate.”

(d) “each request specifying one or more serial identifiers of the media data elements requested;” (claim 1[a(iii)])

120. Hill discloses “each request specifying one or more serial identifiers of the media data elements requested.”

121. As discussed above, the serial identifier for each frame is the global frame number or, when converted by the cliplist manager, the clip number and local frame number. Hill teaches each request to source **122** for a new frame includes the clip number and local frame number of the desired frame. For example, Hill discloses in step 310, “the buffer manager **118** sends requests for new frames to the cliplist manager **116**.” (Hill, 6:7-8.) The cliplist manager **116** “converts the global frame number to a clip number and local frame number” and then, at step 314, “requests the frames from the appropriate source manager **120**.” (*Id.*, 6:8-15.) The source manager **120** then “retrieves the requested frame data from the source **122**.” (*Id.*, 6:14-16.) A person of ordinary skill in the art would understand from this

to a user,” while Patent Owner argues the term should be given its plain and ordinary meaning. (EX1058, 6-8.) These respective arguments do not change my opinion that Hill teaches limitation **1[a(ii)]**.

process that the request includes the clip number and/or at least the local frame number: the cliplist manager **116** calculates the “clip number and local frame number” and requests that specific frame from source manager **120**. Indeed, the goal of prefetching frames to fill buffer manager **118** is “to always have available *the next frame* required by the control unit **114** when *that frame* is requested.” (*Id.*, 4:11-13 (emphasis added), *see also* 4:27-28.) The phrase “next frame” suggests a specific frame that buffer manager **118** needs. It would be obvious to a person of ordinary skill in the art that requests to source **122** for a new frame would specify the desired frame by frame number.

122. Hill supplies an example explicitly demonstrating that requests for additional frames specify the frames by clip number and local frame number. Hill states that “[w]hen the control unit **114** requests global frame 20 (via the cliplist manager **116**), the buffer manager **118** replies by providing global frame 20 to control unit **114**[.]” (Hill, 7:39-41.) Once it has sent frame 20, the buffer manager **118** checks how many frames it now has and compares this number against the request threshold. (*Id.*, 7:41-43.) As will be discussed in greater detail below, the request threshold “represents the optimum number of future frames stored in the buffer at any given time[.]” (*Id.*, 6:63-65.) If the buffer manager removed frame 20, and the request threshold is 10 frames, buffer manager **118** would need to have

frames 21-31 prefetched and ready to provide to control unit **114**. (*Id.*, 7:39-45.) Assuming frame 31 has not yet been requested, “the buffer manager **118** will then request global frame 31. This request is sent to the cliplist manager [**116**], which maps the global frame number to a clip number and local frame number and forwards the request to the appropriate source.” (*Id.*, 7:46-50.) Hill thus expressly teaches each request to a source for a particular frame (*e.g.*, frame 31) specifies the clip number and local frame number (*i.e.*, serial identifier) of the desired frame (*i.e.*, media data element). (*See also id.*, 12:7-44.)

123. Hill thus teaches “each request specifying one or more serial identifiers of the media data elements requested.”

(e) **“receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and” (claim 1[b(i)])**

124. Hill discloses “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements.”

125. Hill teaches “receiving each of the requested media data elements via the data connection.” After cliplist manager **116** requests the desired frame from source manager **120**, source manager **120** “retrieves the requested frame data from the source **122**.” (Hill, 6:13-16.) After receiving the desired frame, “the source

manager **120** transfers the requested frame to the buffer manager [**118**].” (*Id.*, 6:30-31.) As discussed above, the frame is received over a data connection, such as the Internet. (*E.g., id.*, 3:48-56.)

126. Hill further teaches “the data connection has a data rate more rapid than the playback rate of the media data elements.” A person of ordinary skill in the art would understand a “data rate” is the rate at which data is carried across a data connection. (EX1039, 0006.) Hill teaches the data rate of the connection between source **122** and workstation **104** has a rate “more rapid than the playback rate.” For example, Hill discloses that when the buffer manager requests additional frames, steps are taken, such as never requesting more than two new frames for each request received by the cliplist manager **116** “to avoid overloading the sources” with excessive frame requests. (Hill, 7:15-20.) “Hence, if the play rate is 24 frames per second, the sources will not be asked to supply more than 48 frames per second.” (*Id.*, 7:20-22.) In this scenario, the playback rate is 24 frames per second, and the data rate is 48 frames per second; a person of ordinary skill in the art would thus understand the data rate (48 frames/second) is “more rapid than” the playback rate (24 frames/second).

127. Hill thus discloses “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than

the playback rate of the media data elements.”⁴

(f) “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;” (claim 1[b(ii)])

128. Hill discloses “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.”

129. A person of ordinary skill in the art would have found it obvious that Hill would send frames from source **122** to the workstation **104** as fast as the data connection allows. As I have discussed at length, Hill teaches sources **122** may be remote servers connected to workstations 104 via the Internet. (*E.g.*, Hill, 3:48-56.) The most common communication protocol is TCP. It would be obvious to a person of ordinary skill in the art that a source sending data via Internet protocol, such as

⁴ I understand in the separate litigation between Patent Owner and Amazon.com, the parties dispute the construction of “data rate.” *WAG Acquisition, L.L.C. v. Amazon.com, Inc. et al.*, No. 6:21-cv-00815, Dkt. No. 38 at 10 (W.D. Tex. 2021). Amazon.com argues the term means “the actual rate at which the data connection delivers data to the requesting user system at any given time,” while Patent Owner argues the term should be given its plain and ordinary meaning. *Id.* These respective arguments do not change my opinion that Hill teaches limitation **1[b(i)]**.

TCP, would send the data as fast as the data connection allows. (EX1019, 271-72 (explaining file “is transferred to the client (browser station) as fast as TCP/IP can move it”); EX1042, 0002 (“The server sends data to the client as fast as the TCP connection will accept[.]”).) Patent Owner’s expert, Keith J. Teruya, opined that the “[a]s fast as the data connection...allows” limitation “requires nothing more than simply handing the data to the transport mechanism.” (EX1041, ¶ 30.) It would be obvious that, given the disclosure of the Internet as a network to transfer data, Hill uses a transport mechanism, such as TCP, to send frames from sources **122** to workstation **104** via the Internet.⁵

⁵ Hill contains some statements regarding frames being transmitted from sources **122** as fast as the play rate. (E.g., Hill, 8:21-22 (“The preceding discussion assumed that the sources **122** will always be able to provide frames at the normal play rate.”).) In my opinion, this statement does not mean each frame is transmitted from sources **122** *only* at the play (playback) rate in every instance or any instance. Rather, I understand this statement to mean that during normal operations, the sources will always be able to send frames *at least as fast as* the play rate. This statement occurs in the context of Hill’s discussion of “Inadequate Source Bandwidth,” which is a condition Hill recognizes may occur. (E.g., *id.*, 8:20-30.) Hill’s disclosures

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

130. Furthermore, as I discuss in Ground 2, I understand Petitioner argued during claim construction that a similar limitation in a related patent is indefinite. (EX1040, 8-11.) In particular, I understand Petitioner argues in the district court litigation that a person of ordinary skill in the art would understand this limitation to encompass three possible transmission rates: “(1) the maximum possible capacity of the data connection, (2) the available bandwidth on a data connection, and/or (3) the permitted bandwidth for uses on a data connection.” (*Id.*, 8.) I understand Petitioner argues these three plausible interpretations render the limitation indefinite because the claim “does not provide an objective basis for those of skill in the art to make an informed and confident choice among the contending definitions.” (*Id.*) But, just as in Ground 2, Petitioner’s argument does not disturb my opinion that Hill renders obvious sending data as fast as the data connection allows. It would have been obvious to a person of ordinary skill in the art that Hill renders obvious sending data at either (1) the maximum possible speed supported by the protocol or (2) the maximum available speed of the protocol, taking into consideration external drains on the connection’s bandwidth. For example, as discussed above, Hill teaches using

explicitly state that its system transmits frames from sources **122** more rapidly than the playback rate.

an internet connection, which is commonly a TCP/IP based connection, where data is sent as fast as the protocol allows (in case of no competing drains) or as fast as the protocol allows based on currently available bandwidth given competing external drains. (*See, e.g.*, EX1019, 271-72 (explaining file “is transferred to the client (browser station) as fast as TCP/IP can move it”); EX1042, 0002 (“The server sends data to the client as fast as the TCP connection will accept[.]”).) In addition, it would have been well-known to a POSITA reading Hill to send data at (3) an artificially enforced maximum permitted data rate. (*See, e.g.*, Hill, 3:22-26 (recognizing artificial bandwidth limits of various sources and teaching that the purpose of Hill is to adapt to these artificial bandwidth limits).) Therefore, Petitioner’s indefiniteness argument does not affect my opinion that Hill renders obvious sending frames as fast as the data connection allows.

131. Hill thus discloses “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.”

(g) “storing the received media data elements in a memory of the media player;” (claim 1[c])

132. Hill discloses “storing the received media data elements in a memory of the media player.”

133. As discussed above, after requesting a particular frame from source

122, the source manager **120** “retrieves the requested frame data from the source **122**.” (Hill, 6:14-16.) The source manager **120** then “transfers the requested frame to the buffer manager [**118**].” (*Id.*, 6:30-34.) The buffer manager **118** “stores [the] frames in local digital memory, such as random-access memory (RAM) or a local disk drive.” (*Id.*, 4:8-11; *see also id.*, 6:58-67 (referencing frames being stored in buffer manager **118**), 11:17-28, 12:7-44.) Because buffer manager **118** is part of the media player on workstation **104**, a person of ordinary skill in the art would understand the frames are stored “in a memory of the media player.”

134. Hill thus discloses “storing the received media data elements in a memory of the media player.”

(h) “**playing the received media data elements in series from the memory of the media player; and**” (claim 1[d])

135. Hill discloses “playing the received media data elements in series from the memory of the media player.”

136. Hill teaches that buffer manager **118** sends frames to control unit **114** when the user indicates s/he wants to view the movie data. (Hill, 4:7-8.) The control unit **114** forwards the frames to decompressor/renderer **110**, which “is responsible for decompressing the video and audio and presenting them to the display **108** *in correct synchronization*.” (*Id.*, 3:67-4:2 (emphasis added).) Furthermore, “control

unit **114** is responsible for feeding frames to the decompressor/renderer **110** in the proper sequence.” (*Id.*, 4:2-4; *see also id.*, 4:66-5:1 (“In step 204, the control unit **114** causes the first frame to be displayed to the user by passing the data to the decompressor/renderer **110**.”).) Hill thus provides the frames “in series”: a person of ordinary skill in the art would understand displaying frames “in correct synchronization” after receiving the frames “in the proper sequence” means the frames are displayed “in series,” *i.e.*, one after the other in the correct order. Hill thus teaches displaying the frames to the user via display **108** (*i.e.*, playing the received media data elements) in series from buffer manager **118** (*i.e.*, from the memory of the media player). (*See generally* Hill, 5:9-15, 3:27-45.)

137. Hill thus discloses “playing the received media data elements in series from the memory of the media player.”

- (i) **“as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e])**

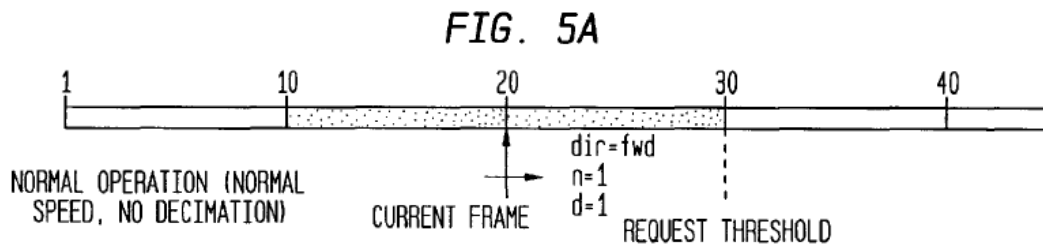
138. Hill discloses “as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about

a predetermined number of media data elements in the memory of the media player during playing.”

139. Hill teaches a technique called “prefetching,” the goal of which is “to always have available the next frame required by the control unit **114** when that frame is requested.” (Hill, 4:7-15.) In other words, while frames are being played out to the user, the buffer manager **118** is continuously replenishing its store of frames. The goal of the buffer manager is “to get frames that will soon be shown to the user if the current speed and direction are maintained (i.e., future frames).” (*Id.*, 6:53-57.)

140. The buffer manager **118** keeps track of the “buffer fill level,” which “refers to the number of future frames currently being stored in the buffer manager **118**[.]” (Hill, 6:60-62.) The buffer manager **118** also has a “request threshold,” which “represents the optimum number of future frames stored in the buffer at any given time[.]” (*Id.*, 6:63-65.) The status quo is the number of frames stored in buffer manager **118** equals the request threshold. The buffer manager **118** “determines whether the buffer fill level has dropped below the request threshold[.]” (*Id.*, 6:58-60.) If the buffer fill level is less than the request threshold – indicating the buffer manager **118** is storing fewer than the optimum number of future frames – an additional frame will be requested. (*Id.*, 7:4-10.) “The buffer manager **118** will

request” additional frames until the request threshold exceeds the fill level. (*E.g.*, *id.*, 7:11-24.) The buffer manager **118** requests future frames according to the process I described in my analysis of claim limitations **1[a(i)]** through **1[a(iii)]**. (*See generally* Hill, 4:57-5:21, Fig. 2, 7:11-56, 9:36-44 (“FIG. 6 graphically summarizes the various thresholds applied to the buffer fill level.”), 12:21-44.)



(*Id.*, Fig. 5A.)

141. A person of ordinary skill in the art would understand the “request threshold” corresponds to “about a predetermined number of media data elements in the memory of the media player during playing.” The request threshold represents the optimum number of future frames that should be stored in buffer manager **118** in order to assure smooth playback of the movie data. (*E.g.*, Hill, 6:58-67.) Hill thus teaches requesting additional frames (*i.e.*, additional requests for subsequent media data elements) to store in buffer manager **118** (*i.e.*, for storage in the memory of the media player) until the number of future frames stored in the buffer manager **118** equals the request threshold, such as 10 future frames (*i.e.*, as required to maintain about a predetermined number of media data elements in the memory of

the media player).

142. Hill further teaches this process of prefetching future frames occurs “as the received media data elements are played.” Hill teaches the prefetching process occurs as buffer manager **118** provides frames to the control unit **114** (via cliplist manager **116**) for subsequent playback via the display **108**. At step 212, “buffer manager **118** provides a [requested] frame to the cliplist manager **116**...then refills its buffers with new frames as they are needed. This refill operation is critical to the present invention, and is described in detail below with respect to FIG. 3 and FIG. 4.” (Hill, 5:10-15; *see also id.*, Fig. 2, step 212 (“Buffer Manager provides a frame to cliplist manager *and* fills buffer with new frames as needed.”) (emphasis added).)

143. Hill further teaches that when the buffer manager **118** sends a frame to control unit **114** via cliplist manager **116**, “two paths” are followed. In one path, the buffer manager continues receiving requests from control unit **114** and continues providing frames for playback. (Hill, 5:44-45 (“One path shows control passing back to step 214...”), Fig. 2.) In the second path, the buffer manager requests new future frames to fill the buffer to the request threshold. (*Id.*, 5:44-46 (“...the other [path proceeds] to step 308.”), Fig. 3 (describing process by which buffer manager **118** prefetches future frames).) Hill teaches these two paths “are performed concurrently.” (*Id.*, 5:46-47.) In other words, the buffer manager **118** sends frames

to be played out *while* it requests future frames to fill the buffer up to the request threshold. Hill thus teaches prefetching future frames occurs “as the received media data elements are played.”

144. I understand there are several claim construction disputes relating to claim limitation 1[e]. For the reasons that follow, the parties’ claim construction arguments do not affect my opinion that Hill renders obvious claim limitation 1[e].

145. *First*, the parties dispute the construction of “as required to maintain about a predetermined number of media data elements.” Petitioner contends this term is indefinite, while Patent Owner argues the term should be given its plain and ordinary meaning. (EX1040, 4-6.) Petitioner explains that the term is indefinite because the “claims, specification, and extrinsic record fail to provide sufficient guidance as to the numerical range that ‘about’ encompasses.” (*Id.*, 4.) In my opinion, Petitioner’s argument is not relevant to my analysis of whether Hill discloses claim limitation 1[e]. Hill clearly teaches a “request threshold,” which “represents the optimum number of future frames stored in the buffer at any given time.” (Hill, 6:63-65.) There is no question storing an “optimum number of future frames” renders obvious maintaining “about a predetermined number of media data elements.” While the term may be indefinite from the perspective of an ordinarily skilled artisan who is trying to decide infringement, it would be exceedingly obvious

to a person of ordinary skill in the art that a buffer maintaining an optimum number of future frames renders obvious maintaining a predetermined number of media data elements.

146. *Second*, I understand the parties disagree over the construction of the term “a predetermined number of media data elements.” Petitioner contends the term means “a specified number of media data elements, set prior to the start of playback of the audio or video program.” (EX1040, 6.) Patent Owner counters that the term should be given its plain and ordinary meaning. (*Id.*) I understand the crux of the dispute involves by when in the streaming process the number of media data elements must be predetermined: Petitioner contends the number must be predetermined “prior to the start of playback,” while Patent Owner includes a later time period, prior to sending additional requests *after* playback has started. (EX1043, 9-10.) Again, for purposes of my analysis of Hill, the dispute is not relevant. In my opinion, Hill teaches the request threshold, which sets the number of future frames that must be stored in buffer manager **118**, is set prior to any data being played, thus satisfying both parties’ constructions. There is nothing in Hill that suggests otherwise, and in fact, the request threshold *must* be set prior to the system beginning.

147. Hill teaches that at the outset, “control unit **114** initiates start-up

procedures when the user first requests playback of movie data. These procedures basically involve *[filling] the buffer manager 118 with the appropriate frames*, passing the first frame to the control unit **114**, and issuing other preparatory commands.” (Hill, 4:59-64 (emphasis added).) The request threshold sets the number of future frames that should be stored in the buffer manager **118**; Hill teaches the buffer manager **118** will be filled with future frames until the buffer fill level is equal to the request threshold. (*E.g., id.*, 6:58-7:32.) Therefore, a person of ordinary skill in the art would understand the request threshold must be predetermined *prior to start-up*, otherwise the buffer manager would fill continuously. The request threshold is a cap on the number of future frames that should be stored in the buffer manager **118**, and this number would naturally be set prior to start up, based on “the characteristics of a particular system.” (*Id.*, 6:65-67.)

148. Therefore, a person of ordinary skill in the art would understand Hill satisfies the “a predetermined number of media data elements” according to both parties’ proposed constructions. The parties’ argument simply creates no ambiguity concerning my analysis of Hill in relation to claim limitation **1[e]**.

149. Hill thus discloses “as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about

a predetermined number of media data elements in the memory of the media player during playing.”

2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

150. Hill discloses the method of claim 1, “further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

151. The buffer manager **118** records the identification of the last frame received and stored by the media player. Hill teaches the buffer manager **118** sends requests for new frames to ensure the buffer fill level equals the request threshold. Hill teaches “buffer manager **118** sends requests for new frames to the cliplist manager **116**,” which “converts the global frame number to a clip number and local frame number” and requests the relevant frame from source **122**. (Hill, 6:7-16.) Implicit in this description is the understanding that buffer manager **118**’s request for a new frame specifies the desired frame by global frame number. A person of ordinary skill in the art would understand cliplist manager **116** could only convert the global frame number of the requested frame to a clip number and local frame number if it had been provided the global frame number by buffer manager **118**. And if buffer manager **118** provides the global frame number of the next frame it

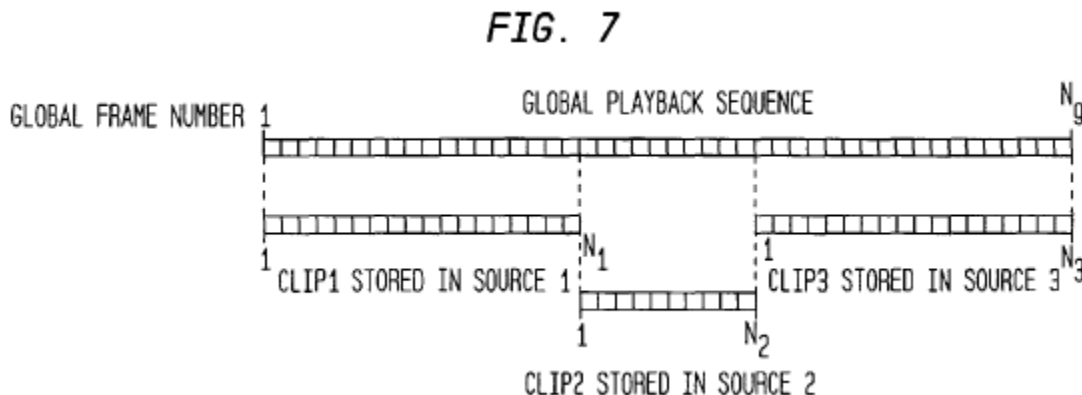
needs, buffer manager **118** must necessarily keep a record of the last frame received. If no record existed, buffer manager **118** would have no way of knowing what the “next “ frame should be.

152. Hill provides an explicit example of the buffer manager **118** keeping a record of the last frame it received. If, for example, “control unit **114** requests global frame 20,” the buffer manager **118** provides global frame 20 and then checks the buffer fill level to see if the level is below the request threshold. (Hill, 7:39-43.) The buffer manager does this “by checking to see if it has frames 21 through 31.” (*Id.*, 7:43-45.) Once it detects that a certain frame is missing – in this example, frame 31 – “the buffer manager **118** will then request global frame 31.” (*Id.*, 7:46-47.) The only way the buffer manager **118** could detect which frame is missing is if it has a record of the last frame it received – in this case, global frame 30. (*See also id.*, 12:27-34 (“...the buffer manager **118** will check to see if global frames 96 through 106 *are currently being stored*...”) (emphasis added).) This capability would be inherent or obvious to a person of ordinary skill in the art as a necessary function of buffer manager **118**.

153. Hill thus discloses the method of claim 1, “further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

3. Claim 3: “The method of claim 1 wherein the serial identifiers are sequential.”

154. Hill discloses the method of claim 1, “wherein the serial identifiers are sequential.” As I discussed above, the “serial identifier” of a frame is the global frame number or the clip number and local frame number, which the media player uses to synchronize the movie data and determine which frame to request from the sources **122**. The serial identifiers are sequential: “Frames may either be referenced according to their position within the *global playback sequence* (i.e., global frame number), or according to their position within the clip from which they originate (i.e., local frame number).” (Hill, 5:66-6:2, Fig. 7.) The frame numbers are based on the sequence of each frame within the media data; later-occurring frames have a higher frame number than earlier-occurring frames. It would be obvious to a person of ordinary skill in the art that the frame numbers are “sequential.”



(*Id.*, Fig. 7.)

155. Hill thus teaches the method of claim 1, “wherein the serial identifiers are sequential.”

4. Claim 4 (“The method of claim 1, wherein the receiving is via a reliable transmission protocol.”) and Claim 5 (“The method of claim 4, wherein the reliable transmission protocol is TCP.”)

156. Hill renders obvious claims 4 and 5. As I discussed above for claim 1, Hill teaches workstations **104** are connected to sources **122** via a network such as the Internet. (Hill, 3:48-56.) A person of ordinary skill in the art would find it obvious that the Transmission Control Protocol (TCP) could be used by the network. By the priority date of the '594 patent, TCP was widely known and used in the art and was, indeed, the most common transfer protocol used over the Internet. (*E.g.*, EX1048, Abstract (“Although the ubiquitous TCP/IP protocol is widely used to provide reliable data delivery...”); EX1049, Abstract (“The TCP/IP protocol suite, which is very widely used today...”); EX1050, 2315 (“TCP/IP is the standard networking protocol on the internet. It is the most widely used transport protocol for data services like file transfer, email and WWW browser.”); EX1051, 416-17 (“Recently, the TCP/IP protocols are widely used in various computer communications.”); (EX1017, 362 (“[T]he most widely used transport protocols today are TCP and its companion transport protocol, the User Datagram Protocol (UDP).”).) It would have therefore been obvious for Hill’s system to use TCP, which

is a reliable transmission protocol.

157. Hill therefore renders obvious claims 4 and 5.

5. Claim 6 (Independent)

158. Claim 6 is, for purposes of my analysis, materially the same as claim 1 except it is written as a system/device for performing the same method recited in claim 1. A side-by-side table comparing claim 1 with claim 6 is shown below (with differences noted in underlining):

Claim 1 (Method)	Claim 6 (System)
1. A <u>method for operating</u> a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the <u>method</u> comprising:	6. A <u>device comprising</u> a media player <u>operable</u> to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the <u>media player</u> comprising:
-	<u>[a] a processor, memory accessible to the processor, and a data connection;</u>
<u>[a(i)] sending</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,	<u>[b(i)] a machine-readable, executable routine containing instructions to cause the processor to send</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>an</u> audio or video program,
<u>[a(ii)]</u> each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,	<u>[b(ii)]</u> each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

[a(iii)] each request specifying one or more serial identifiers of the media data elements requested;	[b(iii)] each request specifying one or more serial identifiers of the media data elements requested;
[b(i)] <u>receiving</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and	[c(i)] <u>a machine-readable, executable routine containing instructions to cause the media player to receive</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and
[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;	[c(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;
[c] <u>storing</u> the received media data elements in <u>a</u> memory of the media player;	[d] <u>a machine-readable, executable routine containing instructions to cause the processor to store</u> the received media data elements in <u>the</u> memory of the media player;
[d] <u>playing</u> the received media data elements in series from the memory of the media player; and	[e] <u>a machine-readable, executable routine containing instructions to cause the received media data elements to be played</u> in series from the memory of the media player; and
[e] as the received media data elements are played, <u>the media player</u> automatically <u>sending</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the	[f] <u>a machine-readable, executable routine containing instructions to cause the processor to,</u> as the received media data elements are played, automatically <u>send</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

memory of the media player during playing.	element in the memory of the media player during playing.
--	---

159. As shown in the table, claim 6 presents no material differences from claim 1 in terms of the basic functions it recites. For purposes of claim 6, as explained in connection with the preamble of claim 1, the “device comprising a media player” is workstation **104**.

160. With respect to claim **6[a]**, Hill’s “device comprising a media player” includes “a processor, memory accessibly to the processor, and a data connection.” Hill teaches workstation **104** “may be a conventional personal computer, a high performance workstation, or dedicated hardware specially suited for audio/video playback.” (Hill, 3:29-32.) It was well known by the ’594 patent’s priority date that a conventional personal computer or workstation would include a processor, memory accessible to the processor, and a data connection. (*See also id.*, 4:8-11 (referencing “local digital memory, such as random access memory (RAM) or a local disk drive”).) The data connection would be, for example, connection to the Internet, which Hill clearly teaches. (*Id.*, 3:55-56.) This would have been obvious to a person of ordinary skill in the art.

161. For several limitations, claim 6 broadly recites “a machine-readable, executable routine containing instructions to cause” the processor or media player to perform a designated function. These limitations are not inventive and would

have been understood by a person of ordinary skill in the art to be inherent in the functionality of a computer. A computer is designed to include “machine-readable, executable routine[s] containing instructions.” It would have been obvious to a person of ordinary skill in the art that workstation **104** disclosed by Hill would include “machine-readable, executable routine[s] containing instructions to cause” workstation **104** to perform a designated function.

162. For example, with respect to claim limitations **6[b(i)]** through **6[b(iii)]**, it would be obvious to a person of ordinary skill in the art that the media player (*e.g.*, in workstation **104**) disclosed by Hill includes “a machine-readable, executable routine containing instructions to cause the processor to” send requests to sources **122** for media data elements. Hill teaches how the various components of the media player in workstation **104**, such as the control unit **114**, cliplist manager **116**, buffer manager **118**, and source manager **120** work together to request media data elements from source **122**. (*E.g.*, Hill, 3:57-4:34, 4:57-5:21, 6:35-7:56.) It would be obvious to a person of ordinary skill in the art that these different components would include hardware and/or software comprising “a machine-readable, executable routine containing instructions” to perform the described functions. I previously analyzed the substantive details of this capability of these components in my discussion of claim limitations **1[a(i)]** through **1[a(iii)]**, which I incorporate here.

163. With respect to claim limitations **6[c(i)]** and **6[c(ii)]**, it would be obvious to a person of ordinary skill in the art that the media player on workstation **104** includes “a machine-readable, executable routine containing instructions to” receive the requested media data elements at a rate as fast as the data connection allows, via a data connection that has a data rate more rapid than the playback rate of the media data elements. I previously analyzed the substantive details of this capability of workstation **104** in my discussion of claim limitations **1[b(i)]** and **1[b(ii)]**, which I incorporate here.

164. With respect to claim **6[d]**, it would be obvious to person of ordinary skill in the art that the processor in workstation **104** includes “a machine-readable, executable routine containing instructions to cause” the processor to store the received media data elements in memory in buffer manager **118**. I previously analyzed the substantive details of this capability of workstation **104** in my discussion of claim limitation **1[c]**, which I incorporate here.

165. With respect to claim **6[e]**, it would be obvious to person of ordinary skill in the art that workstation **104** includes “a machine-readable, executable routine containing instructions to cause” the media data elements to be played in series from the memory of the media player. In particular, Hill teaches control unit **114** causes frames to be displayed sequentially to a user. I previously analyzed the substantive

details of this capability of control unit **114** in workstation **104** in my discussion of claim limitation **1[d]**, which I incorporate here.

166. With respect to claim **6[f]**, it would be obvious to person of ordinary skill in the art that the processor in workstation **104** includes “a machine-readable, executable routine containing instructions to cause” the processor to automatically send additional requests for subsequent media data elements to maintain about a predetermined number of media data elements in memory of the media player. In particular, Hill teaches buffer manager **118** is responsible for sending requests for subsequent frames of movie data. I previously analyzed the substantive details of this capability of buffer manager **118** in my discussion of claim limitation **1[e]**, which I incorporate here.

6. Dependent claims 7-10

167. Claims 7 through 10 are substantially similar, for purposes of my analysis, to claims 2 through 5, as shown in the following chart (underlining showing differences):

Claims 2-5 (Method)	Claims 7-10 (System)
2. The <u>method</u> of claim <u>1</u> , further comprising <u>maintaining</u> in the memory of the media player a record identifying the last media data element received and stored by the media player.	7. The <u>device</u> of claim <u>6</u> , further comprising <u>a machine-readable, executable routine containing instructions to cause the processor to</u> maintain in the memory of the media player a record identifying the last

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

	media data element received and stored by the media player.
3. The <u>method</u> of claim <u>1</u> , wherein the serial identifiers are sequential.	8. The <u>device</u> of claim <u>6</u> , wherein the serial identifiers are sequential.
4. The <u>method</u> of claim <u>1</u> , wherein the receiving is via a reliable transmission protocol.	9. The <u>device</u> of claim <u>6</u> , wherein the receiving is via a reliable transmission protocol.
5. The <u>method</u> of claim <u>4</u> , wherein the reliable transmission protocol is TCP.	10. The <u>device</u> of claim <u>9</u> , wherein the reliable transmission protocol is TCP.

168. As shown above, the only differences are that claims 7 through 9 depend from system claim 6, claim 10 depends from 9, claims 2 through 4 depend from method claim 1, and claim 5 depends from claim 4. Claims 7 through 10 are thus not materially different from corresponding claims 2 through 5; the dependence of claims 7 through 10 on a system, rather than a method, does not affect the applicability of my analysis provided for claims 2 through 5.

169. Because Hill renders obvious claim 6 just as it renders obvious claim 1, claims 7 through 10 are thus obvious for the same reasons as claims 2 through 5.

7. Claim 11 (Independent)

170. Claim 11 is, for purposes of my analysis, materially the same as claim 1 except, like claim 6, it is written as a “computer program product” for performing the method recited in claim 1. A side-by-side table comparing claim 1 with claim 11 is shown below (with differences noted in underlining):

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Claim 1 (Method)	Claim 11 (System)
<p>1. A <u>method for operating a media player to receive and play</u> an audio or video program, from a remote media source via a data connection over the Internet, the <u>method</u> comprising:</p>	<p>11. A <u>computer program product for receiving and playing</u> an audio or video program <u>by a media player having a processor and a memory accessible to the processor</u>, from a remote media source via a data connection over the Internet, the <u>computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions comprising:</u></p>
<p>[a(i)] <u>sending</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,</p>	<p>[a(i)] <u>instructions executable to cause the processor to send</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,</p>
<p>[a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,</p>	<p>[a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,</p>
<p>[a(iii)] each request specifying one or more serial identifiers of the media data elements requested;</p>	<p>[a(iii)] each request specifying one or more serial identifiers of the media data elements requested;</p>
<p>[b(i)] <u>receiving</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and</p>	<p>[b(i)] <u>instructions executable to cause the processor to receive</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid</p>

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

	than the playback rate of the media data elements, and
[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;	[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;
[c] <u>storing</u> the received media data elements in <u>a</u> memory of the media player;	[c] <u>instructions executable to cause the processor to store</u> the received media data elements in <u>the</u> memory of the media player;
[d] <u>playing</u> the received media data elements in series from the memory of the media player; and	[d] <u>instructions executable to cause the received media data elements to be played</u> in series from the memory of the media player; and
[e] as the received media data elements are played, <u>the media player</u> automatically <u>sending</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.	[e] <u>instructions executable to cause the processor to,</u> as the received media data elements are played, automatically <u>send</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data element in the memory of the media player during playing.

171. As shown in the table, claim 11 presents no material differences from claim 1 in terms of the basic functions it recites. For purposes of claim 11, the “computer program product...comprising a non-transitory computer readable storage medium having program instructions embodied therewith” corresponds to the software and hardware of Hill’s workstation **104** that is programmed to carry out

the method outlined in claim 1. As I discussed above in reference to claim 6, a person of ordinary skill in the art would deem it obvious that Hill's system discloses a "computer program product...comprising a non-transitory computer readable storage medium having program instructions embodied therewith."

172. Accordingly, for the reasons I discussed with respect to independent claims 1 and 6, Hill renders obvious claim 11.

8. Dependent claims 12, 13, 16, 17

173. Claims 12, 13, 16, and 17 are not materially different from, for purposes of my analysis, claims 2 through 5, as shown in the following chart (underlining showing differences):

Claims 2-5 (Method)	Claims 12, 13, 16, 17 (System)
2. The <u>method</u> of claim <u>1</u> , further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.	12. The <u>computer program product</u> of claim <u>11</u> , further comprising <u>instructions executable to cause the processor to maintain</u> in the memory of the media player a record identifying the last media data element received and stored by the media player.
3. The <u>method</u> of claim <u>1</u> , wherein the serial identifiers are sequential.	13. The <u>computer program product</u> of claim <u>11</u> , wherein the serial identifiers are sequential.
4. The <u>method</u> of claim <u>1</u> , wherein the receiving is via a reliable transmission protocol.	16. The <u>computer program product</u> of claim <u>11</u> , wherein the receiving is via a reliable transmission protocol.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

5. The <u>method</u> of claim <u>4</u> , wherein the reliable transmission protocol is TCP.	17. The <u>computer program product</u> of claim <u>16</u> , wherein the reliable transmission protocol is TCP.
--	--

174. As shown above, the only differences are that claims 12, 13, 16, and 17 depend from system claim 11, while claims 2 through 5 depend from method claim 1. Claims 12, 13, 16, and 17 are thus materially the same as the corresponding claims 2 through 5; the dependence of claims 12, 13, 16, and 17 on a system, rather than a method, does not affect the applicability of my analysis provided for claims 2 through 5.

175. Because Hill renders obvious claim 11 just as it renders obvious claim 1, claims 12, 13, 16, and 17 are thus obvious for the same reasons as claims 2 through 5.

- 9. Claim 14 (“The computer program product of claim 11, wherein the computer program product is provided as a software application for the media player.”) and Claim 15 (“The computer program product of claim 11, wherein the computer program product is incorporated in the media player.”)**

176. As I discussed for Ground 1, claims 14 and 15 simply recite the different ways the computer program product recited in claim 11 may be implemented in a user computer. It would be obvious to a person of ordinary skill in the art that Hill’s system could be either provided as a software application that could be used by a personal computer’s media player or incorporated into a media

player on a personal computer. By the '594 patent's priority date, it was well-known that computer programs could be downloaded as specialized software applications. The internal components of Hill's client-side media player, from the user interface **124** and display **108**, to the source manager **120**, are neatly cabined and could be easily implemented either via a software application used by a media player or as components of a media player architecture. There is nothing inventive about how the '594's disclosed media player is implemented on a computer; each arrangement recited in claims 14 and 15 would have been readily understood and obvious to a person of ordinary skill in the art.

C. Ground 2: Claims 1-17 Are Obvious Over Hill In View Of Aharoni

177. Ground 2 incorporates Ground 1's analysis of Hill alone but adds Aharoni's teachings for limitation **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17. Accordingly, only claim limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are addressed in the discussion below. In light of the below analysis of these limitations and Ground 1's analysis of the remaining limitations and claims, claims 1-17 are obvious based on Hill in view of Aharoni.

178. Limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** recite each received media data element "is received at a rate as fast as the data connection between the media source and the media player allows." Claims 4, 9, and 16 further recite the receiving is via

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

“a reliable transmission protocol,” and Claim 5, 10, and 17 specify that reliably transmission protocol is TCP.

179. As I discussed in Ground 1, it would be obvious to a person of ordinary skill in the art that Hill’s system would use TCP to transmit frames from sources **122** to workstation **104**. By the priority date of the ’594 patent, TCP was widely known and used in the art and was, indeed, the most common transfer protocol used over the Internet. (*E.g.*, EX1048, Abstract (“Although the ubiquitous TCP/IP protocol is widely used to provide reliable data delivery...”); EX1049, Abstract (“The TCP/IP protocol suite, which is very widely used today...”); EX1050, 2315 (“TCP/IP is the standard networking protocol on the internet. It is the most widely used transport protocol for data services like file transfer, email and WWW browser.”); EX1051, 416-17 (“Recently, the TCP/IP protocols are widely used in various computer communications.”); (EX1017, 362 (“[T]he most widely used transport protocols today are TCP and its companion transport protocol, the User Datagram Protocol (UDP).”).) It would have been obvious to a person of ordinary skill in the art that a system streaming media data to clients connected via the Internet would use TCP. A person of ordinary skill would likewise know TCP is a reliable transport protocol. (*E.g.*, ’594 patent, 8:36-40 (“A data communications transport mechanism, such as the **TCP protocol**, may be used for the **reliable delivery** of data in an ordered

sequence...from the server to the media player software of the user computer.”)
(emphasis added), 14:50-56.)

180. A person of ordinary skill in the art would likewise understand sending data via Internet protocol, such as TCP, would send the data as fast as the connection allows. (EX1019, 271-72 (explaining file “is transferred to the client (browser station) as fast as TCP/IP can move it”); EX1042, 0002 (“The server sends data to the client as fast as the TCP connection will accept[.]”).).

181. To the extent using TCP, a reliable transfer protocol that sends data as fast as the connection allows, is not obvious based on Hill and the understanding of a person of ordinary skill in the art, using TCP is disclosed by Aharoni.

182. Aharoni teaches a system relating “generally to transport of video and audio information over networks and more particularly...to adapting the transport of video and audio information over IP networks having varying bandwidth capacities.” (Aharoni, 1:5-9.) Aharoni teaches “compressing a raw video source into a plurality of frames” and sending the frames “over the network channel.” (*Id.*, 2:54-65.) A video server **18** transmits the audio/video data over network **20** to clients **22**, who decompress the audio/video data for display. (*See generally id.*, 6:33-8:40.)

183. Aharoni teaches its system “has application to any type of network including those that utilize the Internet Protocol (IP) such as the Internet or other

TCP/IP networks.” (Aharoni, 2:12-15; *see also id.*, 6:37-40, 6:52-54, 7:31-34 (“The system, utilizing ‘best effort’ protocols such as those found on TCP/IP networks...”), 7:39-43.) Aharoni discloses some frames of video/audio data “are sent using a reliable mechanism. Such a reliable mechanism includes using a network protocol such as TCP[.]” (*Id.*, 9:64-66.) Aharoni thus teaches sending video data from a server to a client device using TCP, which is a reliable transmission protocol.

184. Hill in view of Aharoni thus renders obvious limitations 1[b(ii)]/6[c(ii)]/11[b(ii)] and claims 4, 5, 9, 10, 16, and 17.

185. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Hill with Aharoni, predictably resulting in the media streaming and playback system of Hill that uses Aharoni’s method of transporting frames using TCP. A person of ordinary skill in the art would have known that using Aharoni’s method of streaming video/audio data via TCP with Hill’s media streaming system would be successful.

186. Hill and Aharoni are analogous references in the field of streaming media from servers to client computers. Indeed, Hill and Aharoni teach remarkably similar systems. Both systems are advantageous in networks “having varying bandwidth capacities.” (Aharoni, 1:7-9; *compare id.*, 1:7-9 (“The present invention

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

relates generally to transport of video and audio information over networks...having ***varying bandwidth capacities.***”), with Hill, 1:7-10 (“The present invention relates generally to...playing movie data held in various storage devices with ***varying bandwidth capabilities.***”).) Hill and Aharoni both disclose breaking a stream of audio/video data into frames and transmitting frames over network to client devices for playback. (Hill, 3:37-56; Aharoni, 2:54-3:61, 4:35-5:12.) Hill and Aharoni are thus analogous references and disclose similar process (streaming media over internet networks) that are implemented in similar environments (when network connections have varying bandwidth capability).

187. A person of ordinary skill in the art would have been motivated to adopt Aharoni’s teachings of sending data via TCP for Hill’s streaming and playback system. As discussed at length, TCP was a ubiquitous transfer protocol as of the ’594 patent’s priority date. As a reliable transport protocol, a person of ordinary skill would have been motivated to use such a protocol in Hill’s system, where ensuring frames of movie data arrive to client workstations in a reliable manner would have been advantageous. (*E.g.*, Hill, 1:25-27 (“Conventionally, one must insure that the network provides the data at a constant rate in order to achieve an acceptable output.”).)

188. For the reasons discussed above and in Ground 1, Hill in view of

Aharoni render obvious claims 1-17.

D. Ground 3: Claims 1-17 Are Obvious Over Hill In View Of Ravi.

189. Ground 3 incorporates Ground 1's analysis of Hill alone but adds Ravi's teachings for limitations 1[b(ii)]/6[c(ii)]/11[b(ii)] and claims 4, 5, 9, 10, 16, and 17. Accordingly, only limitations 1[b(ii)]/6[c(ii)]/11[b(ii)] and claims 4, 5, 9, 10, 16, and 17 are addressed in the discussion below. In light of the below analysis of these limitations and Ground 1's analysis of the remaining limitations and claims, claims 1-17 are obvious based on Hill in view of Ravi.

190. Limitations 1[b(ii)]/6[c(ii)]/11[b(ii)] each recite "each received media data element is received at a rate as fast as the data connection between the media source and the media player allows." Claims 4, 9, and 16 further recite the receiving is via "a reliable transmission protocol," and Claims 5, 10, and 17 specify that a reliable transmission protocol is TCP. As discussed in Ground 1, Hill renders these limitations obvious. Hill teaches the media player can communicate with the media source via the Internet. (Hill, 3:53-56.) It would have been obvious to a person of ordinary skill in the art that Hill would use TCP to communicate between the media source, such as a network server, and the media player on a client device, particularly since TCP was widely used as of the '594 patent's priority date. A person of ordinary skill in the art further would understand that sending data via Internet protocol, such

as TCP, would send the data as fast as the connection allows.

191. To the extent not rendered obvious based on Hill alone, limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are obvious based on Hill in view of Ravi.

192. Ravi discloses a method for “the efficient and reliable delivery of multimedia streams over a diverse computer network with dynamically variable bandwidth capacity.” (Ravi, 1:55-60; *see also id.*, 3:1-10 (“The present invention provides efficient transmission of multimedia streams from a server to a client computer over a diverse computer network...such as the internet. Examples of multimedia streams provided to the client computer include a compressed video stream [or] a compressed audio stream[.]”).) Ravi teaches a method wherein “the client computer includes a playout buffer, and the transmission rate [from the server to the client] is dynamically matched to the available bandwidth capacity of the network connection between the server and the client computer.” (*Id.*, 3:11-14.) Ravi discloses an “efficient and reliable streaming of data packets” from a server to a client computer, where the server “dynamically selects transmission rates in order to better match the varying bandwidth capacity of the network connection.” (Ravi, 6:32-33, 6:41-44; *see generally id.*, 6:32-47.) Ravi thus teaches transmitting packets of media data from a server to a client computer and selecting a transmission rate to

“match the...bandwidth capacity of the network connection. (*Id.*, 6:43-44.)

193. A person of ordinary skill in the art would understand that Ravi’s system sends data “as fast as the data connection...allows.” If the transmission rate “matches” the capacity of the data connection’s bandwidth, the transmission rate is “as fast as” the data connection allows. If the bandwidth of the data connection increases, Ravi’s servers would increase the transmission rate to “match” the allowable transmission rate. Likewise, if the bandwidth of the data connection decreases, Ravi’s servers would decrease the transmission rate until the rate matches the allowable data rate.

194. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Hill with Ravi, predictably resulting in the media streaming and playback system of Hill that uses Ravi’s efficient and reliable method of transmitting frames by matching the transmission rate to the available bandwidth. A person of ordinary skill in the art that combining Ravi’s transmission methods with Hill’s media streaming and playback system would be successful.

195. Hill and Ravi are analogous references in the field of streaming media from servers to media players on personal computers. Hill explains its system relates to “playing movie data held in *various storage devices* with varying bandwidth

capabilities, where the movie player adapts to these varying conditions.” (Hill, 1:8-11 (emphasis added).) Ravi likewise states its methods “relate[] to multimedia communications” and more particularly to “delivery of multimedia streams over a *diverse computer network*[.]” (Ravi, 1:55-60 (emphasis added).) Ravi’s system is substantially similar to Hill’s: it discloses a stream server that is “coupled to client computer 240 via a computer network [such as] the internet.” (*Id.*, 5:43-51.) Ravi also teaches media data is transmitted to clients as “streamed [data] packets,” which is similar to Hill’s disclosure of transmitting movie data as frames. Ravi and Hill are thus analogous references and disclose similar processes in the field of streaming media data, particularly multimedia data, over the Internet.

196. A person of ordinary skill in the art would have been motivated to adopt Ravi’s method of reliably streaming media data across the internet for Hill media streaming system. Indeed, Hill recognizes “problems arise” in streaming multimedia data “when the data source must be access via a network.” (Hill, 1:19-22.) “Conventionally, one must insure that the network provides the data at a constant rate in order to achieve an acceptable output.” (*Id.*, 1:25-27.) Hill thus recognizes the problem with limited bandwidth. Hill attempts to solve this problem by limiting the number of requests the media player makes to sources **122** so as to not overload the sources, particularly when the sources are struggling with low bandwidth. (*E.g.*,

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Id., 2:35-41 (“The present invention monitors the fill level of the buffer and if it has difficulty maintaining an acceptable level of data, the system begins to selectively make fewer requests.”), 7:11-24, 8:20-9:35.) Ravi provides an additional and more effective solution to low bandwidth, which is modulating the rate to match the bandwidth capacity. It would be natural for a person of ordinary skill in the art to combine Hill’s system with Ravi’s dynamic method of matching the transmission rate to the available bandwidth. Hill explicitly suggests Ravi’s method could be readily adopted, characterizing its system as “playing movie data held in various storage devices with *varying bandwidth capabilities*, where the movie player *adapts to these varying conditions*.” (Hill, 1:7-11 (emphasis added).) In addition, Hill does not offer details concerning the methods the source managers **120** use to communicate with sources **122**; a person of ordinary skill in the art would have looked to Ravi’s detailed disclosure of multimedia transmission techniques.

197. A person of ordinary skill in the art would have had a reasonable expectation of success in modifying Hill to employ Ravi’s dynamic packet transmission techniques. As I discussed, Hill and Ravi disclose substantially similar systems that carry out analogous streaming processes. Implementing Ravi’s teachings into Hill’s system would involve nothing more than routine adaptations to components in Hill’s structure. In particular, workstation **104** in Hill’s system would

be modified to have the capability to compute “performance variables” and “average client packet computational rate[,]” as disclosed in Ravi, to determine whether “the data packets [are] arriving at [workstation **104**] at a faster, equal or slower rate than [workstation **104**] is able to decompress and render the data packets.” (Ravi, 11:15-26.) This would be well within the skills of a person of ordinary skill in the art. Indeed, Hill and Ravi disclose identical workstations **104**. (*Compare* Hill, 3:29-32 (workstation **104** “may be a conventional personal computer [or] a high performance workstation”) *with* Ravi, 5:46-49, 4:36-41 (client computers “can be a personal computer (such as an Apple computer, e.g., an Apple Macintosh, an IBM personal computer, or one of the compatibles thereof), a workstation computer (such as Sun Microsystems or Hewlett-Packard workstation), or some other type of computer.”).) It would be obvious that features present in Ravi’s client computers could be easily added, or are already present in, Hill’s workstation **104**.

E. Ground 4: Claims 1-17 Are Obvious Over Carmel in view of Hill.

1. Claim 1 (Independent)

198. Claim 1 is rendered obvious by Carmel in view of Hill.

- (a) **“A method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet” (claim 1, preamble)**

199. I understand that Petitioner and Patent Owner stipulated in the district

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

court litigation that the preamble is limiting. (EX1033, 0003.) The preamble is disclosed by Carmel.

200. Carmel discloses a method for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet. (*E.g.*, Carmel, Abstract (“A method for real-time broadcasting from a transmitting computer to one or more client computers over a network...”), 1:11-13 (“The present invention relates to network data communications, and specifically to real-time multimedia broadcasting over a network.”), 2:1-4 (“In preferred embodiments of the present invention, a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients.”) Carmel teaches a system that comprises “a transmitting computer **34**, which generates the [multimedia] sequence, a plurality of clients **30**, and a network server **36**, all of which communicate over network **28**, preferably using the well-known Internet Protocol (IP).” (Carmel, 6:28-31.)

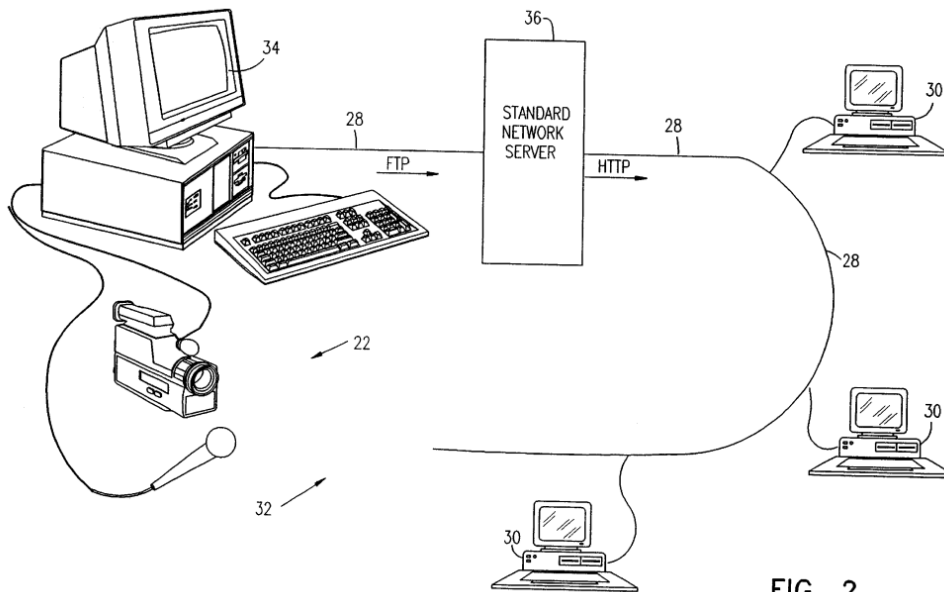


FIG. 2

(*Id.*, Fig. 2.)

201. Carmel discloses a “media player to receive and play an audio or video program.” Clients 30 “preferably comprise conventional personal computers or workstations.” (Carmel, 6:38-40.) The clients 30 “connect to server 36 and receive the multimedia sequence, substantially in real time.” (*Id.*, 7:4-5.) A person of ordinary skill in the art would understand the multimedia sequence to be “an audio or video program.” Furthermore, Carmel teaches “clients 30 will typically need only add a Java applet or plug-in to their existing Web browsers...in order to receive and play back the broadcast.” (Carmel, 7:14-17; *see also id.*, 10:24-34 (“The operation of client is controlled by a Java applet, which may be downloaded from server 36, and includes facilities for carrying out the steps shown in FIG. 6A...”), Fig. 6A

(showing steps carried out by Java applet, including “Output Data”).) A person of ordinary skill in the art would understand the Java applet or plug-in disclosed in Carmel is a “media player.” (E.g., EX1052, 0006 (“We have implemented an MPEG-1 video player in as a Java applet...”), EX1053, 33 (“For the time being, Java is used to add functionality to network applications. Applets included in WWW pages introduce new levels of multimedia functionality through the use of text, graphics, images and audio.”); EX1054, Abstract (“The Cmew media player has been implemented as a Java applet, which enables its use in the current WWW environment.”); EX1055, Abstract (“We also describe a component based SMIL player written in Java. Different components can [be] put together to create a multimedia player. The player can run on major web browsers as a Java applet.”).

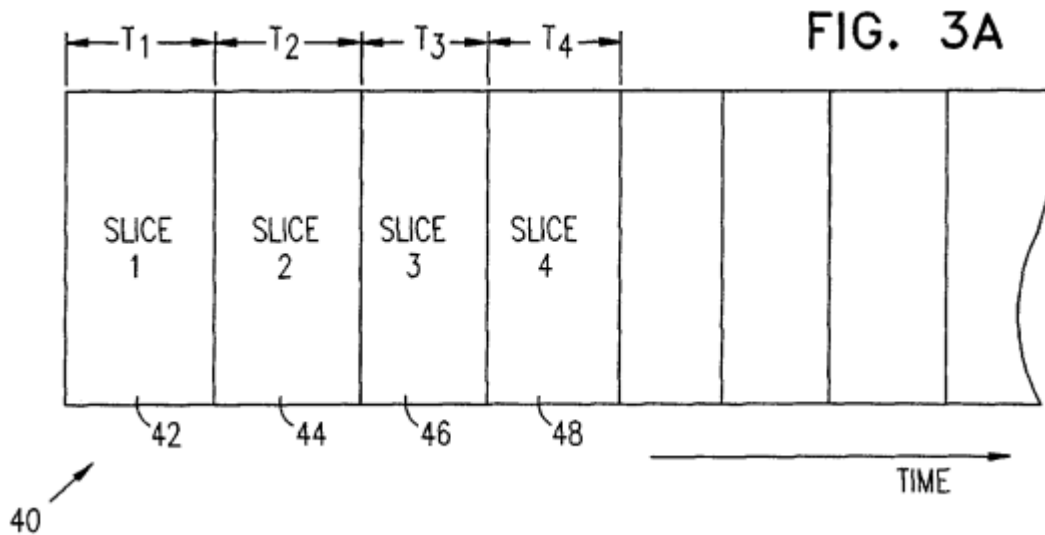
202. Carmel also discloses the media player receives the audio or video program “from a remote media source.” Computer **34** receives multimedia data and “uploads the [multimedia] sequence” to server **36**. (Carmel, 6:50-51; *see generally id.*, 7:50-58, 9:53-10:5 (describing computer **34** uploading multimedia sequence to server **36**).) The client **30**, which includes the media player, “connect[s] to server **36** and receive[s] the multimedia sequence.” (Carmel, 7:4-5; *see generally id.*, 10:24-54 (describing clients **30** downloading multimedia data from server **36**).) Computer **34** and server **36** are collectively the “media source.”

203. Clients **30** receive the multimedia data from server **36** “via a data connection over the Internet.” Carmel teaches “[c]lients **30** preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP[.]” (Carmel, 7:5-9.) Carmel further teaches server **36** is connected to clients **30** via network **28**, which “preferably comprises the Internet.” (*Id.*, 6:36-37, Fig. 2.)

(b) “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program,” (claim 1[a(i)])

204. Carmel discloses “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program.”

205. Carmel teaches the media source has “one or more serially identified media data elements representing the audio or video program.” Carmel teaches that after the audiovisual data is input to the computer **34** from the input devices **22** (*i.e.*, supplied), “[t]he data are compressed at step **80**[.]” (Carmel, 9:66-67.) The data are compressed into “a stream of broadcast data...corresponding to a multimedia data sequence.” (*Id.*, 7:18-21.) This data stream **40** is “then ‘sliced’ at step **82** into files **42, 44, 46, 48**, etc., as shown in FIG. 3A.” (*Id.*, 9:53-10:1, Fig. 3A.)



206. Each “slice” of the data stream **40** corresponds to a “media data element.” The ’594 patent explains media data elements are segments of the media data provided to the server system. (*E.g.*, ’594 patent, 6:30-32 (“In one embodiment, the invention provides a system for distributing via the Internet **streaming media composed of a plurality of time-sequenced data elements.**”) (emphasis added), 7:16-35, 8:4-6 (“This buffer may, for example, hold enough data elements for about one minute of play.”), 9:36-38 (“...the server **12** transmits **audio/video data as sequential data elements...**”) (emphasis added), 9:44-45 (“As each data element is played, it is deleted or displaced from the user’s buffer[.]”), 10:9-12 (“In one embodiment, the number of data elements in the server buffer **14** actually never changes, it always maintains one minute of audio/video data.”).) Additionally, Carmel teaches “[e]ach slice contains a segment of video and/or audio data,

corresponding to a respective, successive time interval T_1 , T_2 , T_3 , etc.” (Carmel, 7:23-25.) The compilation of slices of the data stream **40** thus “represent[] the program” of audio/visual data that was provided to computer **34**.

207. Carmel further discloses the data slices (*i.e.*, media data elements) are serially identified. Carmel discloses that “[e]ach slice is preferably assigned a respective slice index.” (Carmel, 2:6-7, *see also id.*, Abstract (“The slices are encoded in a corresponding sequence of files, **each file having a respective index...**”) (emphasis added), 3:33-34.) The slice indices are serially stored in the running slice index: “[c]omputer **34** stores each slice as a corresponding file, having a running slice index 1, 2, 3...N.” (*Id.*, 7:27-28.) Each data slice is thus identified, via the corresponding slice index, in the running slice index.

208. Furthermore, the slices are identified “serially” because each new slice (and corresponding slice index) is identified by the next available spot in the running slice index, *i.e.*, the slice index for the 1st created slice is listed 1st in the running slice index, the slice index for the 2nd created slice is listed 2nd in the running slice index, and the slice index for the Nth created slice is listed Nth in the running slice index.⁶

⁶ Carmel’s definition of the overall index storing each slice index as the “running slice index” suggests the slice indices identify the slices serially. The word

209. Carmel teaches “sending requests from the media player to the media source via the data connection.” As discussed previously, clients **30** include a Java applet media player or plug-in. Clients **30** are also connected to server **36** via a data connection, such as the Internet. (*E.g.*, Carmel, 6:30-31 (clients communicate with server “preferably using the well-known Internet Protocol (IP)”), 7:5-9.) Carmel further teaches the Java applet or plug-in (*i.e.*, media player) sends requests to the media source for the multimedia data. When clients **30** connect to the server **36**, the client “first reads the index file in order to identify at what point in stream **40** to begin and to start receiving the data stream substantially in real time[.]” (*Id.*, 8:1-5; *see also id.*, 10:24-54.) The index file “comprises a slice ID **52**, indicating the index of the file in data stream **40** that was most recently uploaded by computer **34**.” (*Id.*, 7:62-64.) By reading the index file, the client **30** decides it wants to begin downloading the most recently uploaded data slice and requests that data slice be transmitted from server **36**. Alternatively, Carmel teaches a user “may choose to begin downloading data stream **40** from an earlier point in time than that indicated by ID **52**.” (Carmel, 8:7-9.) In either scenario, the client **30** selects the data slice to

“running” suggests to a person of ordinary skill in the art that the running slice index is updated as each new data slice and slice index is created, one after the other.

download and requests that data slice be sent from server **36** (i.e., the media source).⁷ (*Id.*, 10:42-48 (“Responsive to a user input, client **30** selects an appropriate starting slice and begins to download and decode (decompress) files **42**, **44**, **46**, etc.”); see also *id.*, Fig. 6A (“Select Slice”).)

210. Carmel thus teaches “sending requests from the media player to the media source via the data connection for one or more serially identified media data elements representing the audio or video program.”

(c) **“each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,” (claim 1[a(ii)])**

211. Carmel discloses “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate.”

212. As I discussed for claim limitation **1[a(ii)]**, Carmel teaches computer

⁷ As discussed for Ground 1, I understand the parties dispute the construction of “media source.” For similar reasons as I explained in Ground 1, the dispute does not affect my analysis of Carmel. Server **36** meets either construction of “media source.” Server **36** is the storage device “from which the streaming material originates,” as only after the data is passed to server **36** from computer **34** is the data ready to be transmitted.

34 produces a “stream of broadcast data 40” comprising “a series of data slices 42, 44, 46, 48, etc.” (Carmel, 7:19-23.) “Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T_1 , T_2 , T_3 , etc.” (*Id.*, 7:23-25.) Carmel further teaches the data stream 40 is encoded prior to the stream being sliced into data slices 42, 44, 46, 48, etc. (*E.g., id.*, 11:23:38 (“In encoding data stream 40, computer 34 preferably compresses the data using any suitable compression method known in the art.”), Figs. 5, 7 (detailing encoding step 80).) Carmel thus teaches each data slice (*i.e.*, media data element) is an encoded segment of video and/or audio data (*i.e.*, digitally encoded portion of the audio or video program).

213. Carmel also discloses that each media data element has a playback rate. A person of ordinary skill in the art would understand that “playback rate” of a media file is the rate at which the media is played at normal speed, from beginning to end, through the computer. (*E.g.*, ’594 patent, 5:60-65 (“In conventional systems for streaming media over the Internet, media data (whether real-time or file based) is simply transmitted from the server to the user at **the rate at which it will be played out** (the ‘playback rate’)...”)) (emphasis added).) I understand Patent Owner submitted the expert declaration of Keith J. Teruya in support of its claim construction arguments in the district court litigation. (EX1041.) I read Mr.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Teruya’s declaration and noted his opinion that “[a]s used in the specification, a POSITA would understand ‘playback rate’ to refer to the bitrate of the media data at a normal (1.0) rendition.” (EX1041, ¶ 26.) Mr. Teruya’s understanding of the claims is consistent with my analysis, with the caveat that I would clarify that a “playback rate” need not be defined in terms of bitrate. A person of ordinary skill in the art would understand the rate at which media data is played could be measured in a number of ways, using a number of different metrics well known in the art. As Mr. Teruya explained, “playback rate” is “described by a metric, which is not a data rate itself, but rather a characteristic of how the data is to be played back.” (*Id.*, ¶ 24.) The “playback rate” *may* be described in terms of bitrate, but it need not *always* be described in terms of bitrate.⁸

⁸ I understand in a separate litigation between Patent Owner and Amazon.com, the parties dispute the construction of “playback rate.” *WAG Acquisition, L.L.C. v. Amazon.com, Inc. et al.*, No. 6:21-cv-00815 (W.D. Tex. 2021). Amazon.com argues the term means “a rate at which the data is encoded for playback to a user,” while Patent Owner argues the term should be given its plain and ordinary meaning. (EX1058, 6-8.) These respective arguments do not change my opinion that Carmel teaches limitation **1[a(ii)]**.

214. Accordingly, a person of ordinary skill in the art would understand that “playback rate” of a media file is the rate at which the media is played at normal speed, from beginning to end, through the computer. A person of ordinary skill in the art would therefore understand that a slice of audiovisual data, such as a segment of a video or a segment of an audio recording, would naturally have a playback rate, *i.e.* the rate at which that segment of video or audio plays at normal speed from beginning to end. This is confirmed by Carmel, which explains each slice corresponds to “a respective, successive time interval labeled T_1 , T_2 , T_3 , etc.” (Carmel, 7:23-25.) If the time interval T_1 for slice #1 is 0.1 seconds, the playback rate of slice #1 is likewise 0.1 seconds. Playback rate is an inherent property of multimedia data.

215. Carmel thus discloses “each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate.”

(d) “each request specifying one or more serial identifiers of the media data elements requested;” (claim 1[a(iii)])

216. Carmel discloses “each request specifying one or more serial identifiers of the media data elements requested.”

217. As I discussed for claim limitation 1[a(i)], Carmel teaches each data slice (*i.e.*, media data element) is identified by a corresponding slice index (*i.e.*, serial identifier). (Carmel, 2:6-7; *see also id.*, Abstract (“The slices are encoded in a

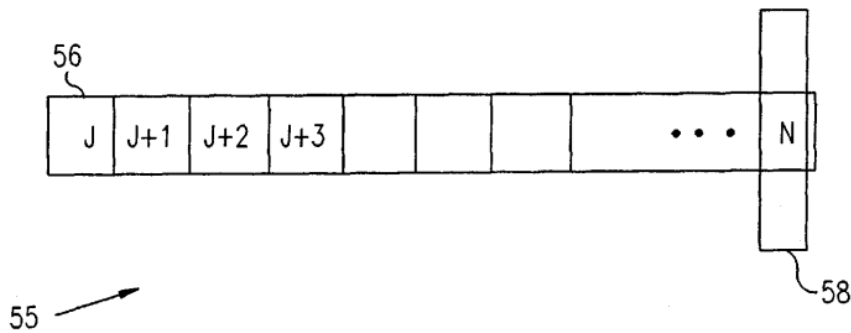
corresponding sequence of files, **each file having a respective index...**) (emphasis added), 3:33-34.) The slice indices are serially stored in the running slice index: “[c]omputer **34** stores each slice as a corresponding file, having a running slice index 1, 2, 3...N.” (*Id.*, 7:27-28.)

218. Each client request for a data slice specifies the slice index of the data slice the client wants to play. As discussed, when client **30** connects to server **36** and “begins to download the data stream, it first reads the index file in order to identify at what point in stream **40** to begin and to start receiving the data stream[.]” (Carmel, 8:1-5.) The index file “comprises a slice ID **52**, indicating the index of the file in data stream **40** that was most recently uploaded by computer **34**.” (*Id.*, 7:62-64.) Put another way: when client **30** requests the most recent data slice in the stream, it points to the slice index (*i.e.*, serial identifier) to identify the requested data slice.

219. Alternatively, Carmel teaches the client **30** may request a data slice that occurs earlier in the stream. Carmel discloses a user may select an earlier slice, such as “J, J+1, J+2,...N,” where each symbol represents “the indices of the slices of stream **40** that are stored on server **36**, wherein N is the index of the most recent slice, and J is the index of the earliest stored slice.” (Carmel, 8:18-31.) To request an earlier data slice, client **30** specifies the slice index (*i.e.*, serial identifier) of the

earlier data slice.

FIG. 3C

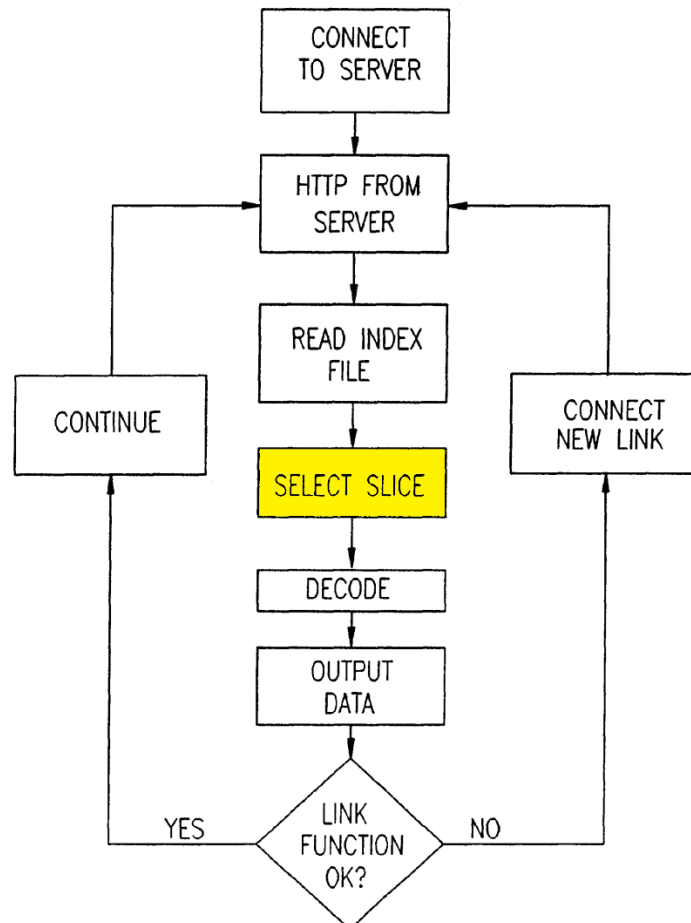


(*Id.*, Fig. 3C.)

220. In both options discussed above, the request from a client **30** specifies where in the data stream the client **30** wants to begin playing the data. In one case, the client requests to download the stream at the last slice uploaded by computer **34**. In another case, the client requests to download the stream at an earlier point in time. But either way, the request specifies a serial identifier of the desired slice. The request for a particular slice of data stream **40** thus specifies the serial indicator (*i.e.*, the slice index) of the requested slice.

221. Carmel further confirms that for each request a slice is selected. For example, Figure 6A shows “a flow chart illustrating the operation of clients **30** in downloading and playing back data stream **40**[.]” (Carmel, 10:24-25.) Figure 6A shows that the client **30** “select[s] [the] slice” the user wants to download, downloads

and decodes the slice, and then repeats the process for each subsequent slice.



(*Id.*, Fig. 6A (highlighting added).)

222. Carmel thus discloses “each request specifying one or more serial identifiers of the media data elements requested.”

- (e) **“receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements,” (claim 1[b(i)])**

223. Carmel discloses “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than

the playback rate of the media data elements.”

224. After the client **30** requests a data slice, client **30** “begins to download the data stream[.]” (Carmel, 8:1-3; *see also id.*, 7:4-5 (“Clients **30** connect to server **36** and receive the multimedia sequence, substantially in real time.”), 8:32-33, 10:24-54 (“FIG. 6A is a flow chart illustrating the operations of clients **30** in downloading and playing back data stream **40**...”).) As discussed at length, the client **30** downloads the data stream via a data connection, such as the Internet. (*E.g.*, *Id.* 2:11-15 (“The clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP)...”), 6:30-31, 7:5-9.) Carmel thus discloses “receiving each of the requested media data elements via the data connection.”

225. Carmel further teaches “wherein the data connection has a data rate more rapid than the playback rate of the media data elements.” A person of ordinary skill in the art would understand a “data rate” is the rate at which data is carried across a data connection.⁹ (EX1039, 0006.) As I discussed above in the context of

⁹ I understand in the separate litigation between Patent Owner and Amazon.com, the parties dispute the construction of “data rate.” *WAG Acquisition, L.L.C. v. Amazon.com, Inc. et al.*, No. 6:21-cv-00815, Dkt. No. 38 at 10 (W.D. Tex. 2021).

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

“playback rate,” “data rate” may be expressed by different unit measurements. Once such measurement of data rate is bitrate, such as bits per second. The ’594 patent often expresses data rate as a bitrate; for example, it states that “[t]he bandwidth requirement of audio or video media refers to the **data rate in bits per second** that must be transmitted and received in order to listen to or view the material uninterrupted.” (’594 patent, 4:40-43 (emphasis added).)

226. As previously discussed, the “playback rate” of a media file is the rate at which the media is played at normal speed, from beginning to end, through the computer. Therefore, the data connection has a data rate “more rapid than the playback rate” of the one or more “media data elements” sent “via [that] connection” when the rate of sending a given multimedia file across a data connection (such as the Internet) is more rapid than the rate at which the given multimedia file is

Amazon.com argues the term means “the actual rate at which the data connection delivers data to the requesting user system at any given time,” while Patent Owner argues the term should be given its plain and ordinary meaning. *Id.* These respective arguments do not change my opinion that Carmel teaches limitation 1**[b(i)]** because Carmel teaches that the rate at which the connection delivers data at any given time may be faster than the playback rate.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

subsequently played at normal speed. (*E.g.*, '594 patent, 9:32-35.) As a practical example, consider a multimedia file “file.mp4” that has a playback rate of 24,000 bits per second. A data connection has a data rate more rapid than the playback rate of “file.mp4” if “file.mp4” can be transmitted across the data connection faster than 24,000 bits per second, such as 36,000 bits per second.

227. Carmel discloses “wherein the data connection has a data rate more rapid than the playback rate of the media data elements.” Carmel discloses multiple ways its system satisfies this claim limitation.

228. *First*, under normal operation, Carmel teaches that the system “monitor[s] the uploading and downloading of data to and from the server...in order to determine the amount of time required to convey each slice and to verify that the slices are conveyed at a sufficient rate.” (Carmel, 2:51-56.) Carmel explains “[w]hen the data stream comprises multimedia data, the **data rate** should be generally **equal to or faster** than the **rate at which the data are generated at the transmitting computer.**” (*Id.*, 2:56-59 (emphasis added).)

229. As previously discussed, the “data rate” is the rate at which data is carried across a data connection, measured in bits per second.

230. The “rate at which the data are generated at the transmitting computer” is the rate at which the multimedia data, such as video data or audio data or both, is

created on computer **34** from input devices **22**. For example, a video file is generated by recording a live event. It takes 1 second to generate 1 second of video data. (*E.g.*, Carmel, 6:57-60 (“The sequence is preferable generated and compressed in real time, and could comprise, for example, an interview program or an entertainment or sports event[.]”).) Thus, a person of ordinary skill in the art would understand it would take 1 second to generate a 1-second audio or video clip; and it would further take 1 second to play back this same clip at normal speed.

231. Carmel explains how the data rate is faster than the playback rate. Transmitting computer **34** “measures a slice transmission time T_{SL} , corresponding to the time required to transmit the entire file to server **36**.”¹⁰ (Carmel, 12:61-64.) T_{SL} thus corresponds to the data rate, the rate at which data is transmitted over a connection: the lower T_{SL} is, the faster the data rate, and vice versa. Carmel further

¹⁰ While this discussion in Carmel describes the process of uploading data slices from the computer **34** to server **36**, Carmel later states that “[a]lthough details of these steps have been described primarily with reference to the uploading process [from computer **34** to server **36**], it will be understood that similar methods are applicable, mutatis mutandis, to the method of downloading the files from server **36** to clients **30**, as shown in FIG. 6A.” (Carmel, 13:29-35.)

states that “[f]or optimal, reliable functioning of the upload process from computer 34 to server 36, T_{SL} should desirably be close to or less than a predetermined minimum time T_{MIN} [,]” where T_{MIN} is “approximately equal to the slice duration T_1 , T_2 , etc.” (*Id.*, 13:7-11.) In other words, under optimal conditions, Carmel teaches the time required to transmit a given data slice should be *less than* the time required to play through the duration of the data slice. And a person of ordinary skill in the art would understand the time required to play through the duration of a data slice corresponds to the playback rate. Put another way: the speed at which the duration of a data slice is played is the playback rate; and Carmel teaches that T_{SL} (the time required to transmit the file) should optimally be less than the playback duration. Therefore, the transmission rate (or data rate) is more rapid than the playback rate.

232. A practical example may be helpful. Consider a data slice A. Data slice A represents a 5-second video clip; therefore, T_{MIN} of data slice A is 5 seconds. Carmel teaches the time to transmit data slice A, or T_{SL} , “should desirably be as close to or less than” T_{MIN} . So T_{SL} should be perhaps 4 seconds. The time to transmit data slice A is 4 seconds, and the time to play back data slice A is 5 seconds; since the file size of data slice A is constant, this necessarily means the rate at which data slice A is transmitted (data rate) is “more rapid than” the rate at which data slice A is played back (playback rate).

233. Carmel thus teaches that under normal operation the data connection between the server and the client device has a data rate that is more rapid than the playback rate of at least one of the data slices of media stream **40**.¹¹

234. ***Second***, Carmel discloses that the computer **34** and clients **30** simultaneously monitor the transmission of data to the clients **30** “in order to ensure that the transmission or reception is ‘keeping up’ with the input of the data to the computer.” (Carmel, 7:36-39.) If the system detects a lag, “steps are taken to increase the data transmission or reception rate[.]” (*Id.*, 7:39-41.) One step to increase the “transmission rate” of data to the clients **30** is varying the compression level of the transmitted data “so as to adjust the data streaming rate to the available

¹¹ I understand the Board previously found Carmel’s normal operation anticipated a similar limitation from a related patent. In IPR2016-01238, a proceeding involving the related ’141 patent, the Board concluded Carmel’s normal operation results in sending media data elements at a rate more rapid than the playback rate. *WebPower, Inc. v. WAG Acquisition, LLC*, IPR2016-01238, Paper 28 at 18-21 (P.T.A.B. July 16, 2020). The Board thus held Carmel anticipated causing “the server to send media data elements to the user...at a rate more rapid than the rate at which said streaming media is played back by a user.” (*Id.*, 16-23.)

bandwidth over one or more channels...between server **36** and client **30**.” (*Id*, 7:45-49.) For example, Carmel teaches that “if the rate is low, such that...the slices need to be played as fast or faster than they are being received, the client will preferably select a lower quality level if one is available.” (*Id*, 11:9-15.) Selecting a lower quality level of a given data slice will increase the data rate with which that slice is transmitted from server **36** to client **30**.

235. It would have been obvious to a person of ordinary skill that the increased data rate of the lower-quality data slice would be “more rapid than” the playback rate. According to Carmel, a client **30** can deploy the lag recovery technique if “the slices need to be played as fast or faster than they are being received,” indicating a low data rate. (Carmel, 11:9-15.) If data slices need to be played “**as fast**” as they are being received, this means the transmission rate of each data slice is **equal to** the playback rate: a first data slice is received by the client **30** and, as the first data slice begins playing, a second data slice is sent from server **36**. Once the first data slice finishes playing, the second data slice arrives and is immediately played. In this scenario, the rate of data slices arriving at the client **30** (data rate) is equal to the rate at which the data slices are exhausted at the client **30** (playback rate). In the same way, if data slices need to be played “**faster**” than they are being received, this means the transmission rate of each data slice is **slower than**

the playback rate: the transmission of data slices cannot keep up with the client device playing the data slices.

236. Thus, Carmel teaches the client **30** may increase the data rate by selecting a lower quality level in order to ensure the transmission rate is **equal to or faster than** the playback rate. Indeed, a person of skill in the art would understand that the necessary implication of selecting the lower quality-level data slices is so that the transmission is **faster rate** than the playback rate. When the client **30** uses the lag recovery technique, the transmission rate is “more rapid than” the playback rate: data slices are sent to and arrive at client **30** faster than the data slices are being played out.¹²

¹² I understand the Board previously found Carmel’s lag recovery anticipated a similar limitation from a related patent. In IPR2016-01238, a proceeding involving the related ’141 patent, the Board concluded Carmel’s lag recovery technique, which I just described, results in “the transmission of the media data elements [] faster than the media [is] being played[.]” *WebPower, Inc. v. WAG Acquisition, LLC*, IPR2016-01238, Paper 28 at 21-23 (P.T.A.B. July 16, 2020). The Board thus held Carmel anticipated causing “the server to send media data elements to the user...at a rate

237. This result makes sense. After all, Carmel teaches this lag recovery technique is appropriate if the data transfer rate between server **36** and client **30** is low. (Carmel, 11:9-23.) To recover from the lag, data slices must be transmitted faster than the data slices are being played out, otherwise the streaming system will never recover. It would therefore have been obvious to a person of ordinary skill in the art that a client **30** changing the quality level to increase the transmission rate and recover from a lag would result in the data rate being more rapid than the playback rate.

238. *Third*, to the extent the “data rate” refers to the speed of the overall connection between server **36** and client **30**, rather than the speed with which an individual slice is transmitted from the server **36** to client **30**, Carmel offers an additional way its system supports a data rate more rapid than the playback rate of one or more slices. Carmel discloses a client **30** can open additional links (*i.e.*, additional data connections) to speed up the transfer of data from the server **36**. If a client **30** determines “any of the links [between it and server **36**] is non-operative or is operating unacceptably slowly, that link is closed, and a new link is opened in its

more rapid than the rate at which said streaming media is played back by a user.”
(*Id.*, 16-23.)

place[.]” (Carmel, 10:55-59.) Additionally, or alternatively, client **30** “opens additional links with server **36** in order to **increase the overall data rate.**” (*Id.*, 10:59-63 (emphasis added).) By permitting multiple links between the server **36** and a single client, Carmel teaches the data rate can be increased, and a person of ordinary skill in the art would understand the increased rate could be more rapid than the playback rate. For example, if two data slices are simultaneously transmitted across two separate links at the playback rate, the overall data rate would be more rapid than the playback rate.

239. Accordingly, Carmel discloses “receiving each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements.”

(f) **“each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;” (claim 1[b(ii)])**

240. Carmel discloses “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.”

241. Claim limitation **1[b(i)]** recites that the data connection between the server system and the client device has a data rate more rapid than the playback rate of one or more of the media data elements sent from the server system to a client

device. Claim limitation **1[b(ii)]** further recites that when a media data element is received by a client device after being sent from the media source, the transmission rate of the sending is as fast as the data connection allows. In other words, this claim limitation recites that media data elements are transmitted to a client device as fast as the data connection allows.

242. Carmel renders obvious claim limitation **1[b(ii)]**. It would have been obvious to a person of ordinary skill in the art that Carmel's server **36** would send a slice of data stream **40** as fast as the data connection between server **36** and a client **30** would allow. Carmel discloses that slices of data stream **40** are sent to clients **30** via well-known Internet protocols, such as HTTP, UDP, RTP, or TCP. (Carmel, 6:50-54 ("Alternatively, [computer **34** may use] other Internet protocols[,] . . . such as the TCP/IP, UDP, or RT(x) protocols, which are known in the art."), 7:5-9 ("Clients **30** preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP, **as noted hereinabove with reference to uploading by computer 34.**") (emphasis added).) It would be obvious to a person of ordinary skill in the art that a server sending data via Internet protocol, such as HTTP or TCP, would send the data as fast as the connection allows. (EX1019, 271-72 (explaining file "is transferred to the client (browser station) as fast as TCP/IP can move it"); EX1042, 0002 ("The

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

server sends data to the client as fast as the TCP connection will accept[.]”).) Indeed, the ’594 patent itself says as much, stating that “media data is sent to the user computer at a rate faster than the playback rate, which may be the highest rate that the data connection between the server and the user computer will support[.]” (’594 patent, 8:14-20.)

243. I understand in the district court litigation, Petitioner and Patent Owner disagree on the construction of a similar limitation from a related patent. I understand Petitioner argues this limitation is indefinite, while Patent Owner argues this limitation should be given its plain and ordinary meaning. (EX1040, 8-11.) I reviewed Petitioner’s indefiniteness argument and conclude the argument does not affect my analysis of this limitation in view of Carmel.

244. Petitioner argues in the district court litigation that a person of ordinary skill in the art would understand this limitation to encompass three possible transmission rates: “(1) the maximum possible capacity of the data connection, (2) the available bandwidth on a data connection, and/or (3) the permitted bandwidth for uses on a data connection.” (EX1040, 8.) I understand Petitioner argues these three plausible interpretations render the limitation indefinite because the claim “does not provide an objective basis for those of skill in the art to make an informed and confident choice among the contending definitions.” (*Id.*) However, having

three plausible interpretations of limitation **1[b(ii)]** does not affect my analysis because Carmel teaches and/or renders obvious each alternative transmission rate. It would have been understood by a person of ordinary skill in the art that Carmel's use of HTTP and TCP teaches or renders obvious sending data at either (1) the maximum possible speed supported by the protocol, or (2) the maximum available speed of the protocol, taking into consideration external drains on the connection's bandwidth. (Carmel, 6:50-54, 7:5-9.) In addition, a person of skill in the art would recognize Carmel teaches (3) an artificially enforced maximum permitted data rate. For example, Carmel teaches tracking the data rate between server **36** and clients **30**. The system has the option of setting "upper and lower data rate thresholds[.]" (Carmel, 11:19-22.) An upper data rate threshold is simply the permitted data rate for a given data connection. Carmel thus renders obvious sending data slices at the maximum permitted data rate, *i.e.*, at the upper data rate threshold. Therefore, Petitioner's indefiniteness argument does not affect my opinion that Carmel renders teaches or renders obvious sending frames as fast as the data connection allows.

245. Accordingly, Petitioner's indefiniteness argument does not disturb my opinion that Carmel renders obvious claim **1[b(ii)]** under all three plausible interpretations of "transmission rate."

246. Patent Owner's understanding of this term is consistent with my

obviousness analysis for claim limitation **1[b(ii)]** – that data sent via well-known internet protocols, such as TCP, would naturally send the data as fast as the connection allows. During claim construction, I understand Patent Owner argued that the specification of the '636 patent tied delivery rate for data “to the full speed of the connection with the client established by...‘[a] data communications transport mechanism, such as the TCP protocol[.]’” (EX1043, 15 (citing '636 patent, 8:38-48).) Patent Owner echoed my opinion that “[i]t is second nature to a POSITA what a TCP connection does with data delivered to it.” (*Id.*)

247. Likewise, I understand Patent Owner submitted an expert declaration of Keith J. Teruya, who opined that the “[a]s fast as the data connection...allows” limitation “requires nothing more than simply handing the data to the transport mechanism.” (EX1041, ¶ 30.) This further supports my opinion that Carmel’s disclosure of data slices being sent via a well-known internet protocol, such as TCP, renders obvious the limitation requiring the transmission rate being “as fast as the data connection...allows.”

248. Carmel thus teaches “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.”

(g) “storing the received media data elements in a memory of the media player;” (claim 1[c])

249. Carmel discloses “storing the received media data elements in a memory of the media player.”

250. It would be obvious to a person of ordinary skill in the art that the media player in client **30** would have a memory and that such memory would store the received media data elements as “clients **30** preferably comprise conventional personal computers or workstations” and “[s]erver **36** may comprise any suitable type of computer or computer system[.]” (Carmel, 6:38-43.) Memory is a common and often necessary component of any computer program; a media player, such as a Java applet or plug-in, could not function if it could not store the media data it receives and intends to play. In line with this understanding, Carmel teaches the media player in client **30** “download[s] and decode[s]” the data slices, and “reconstructs and outputs the multimedia data for the appreciation of a user.” (Carmel, 10:45-50.) It would be obvious to a person of ordinary skill in the art that the data slices would need to be stored in memory so client **30** could decode and reconstruct the data.

251. Carmel thus discloses “storing the received media data elements in a memory of the media player.”

(h) “playing the received media data elements in series from the memory of the media player; and” (claim 1[d])

252. Carmel discloses “playing the received media data elements in series from the memory of the media player.”

253. Carmel teaches the data stream “reconstructs and outputs the multimedia data for the appreciation of a user.” (Carmel, 10:47-50; *see also id.*, 10:24-28, Figs. 6A and 6B (illustrating “operation of clients **30** in downloading and playing back data stream **40**,” including “Output Data”).) As discussed above, a person of skill in the art would understand that the data slices would be reconstructed and output from a memory of the client **30**. Carmel thus discloses “playing the received media data elements...from the memory of the media player.”

254. Carmel further discloses playing the received data elements “in series.” Carmel teaches using “[t]ime stamps” to “synchronize the data, so that the multimedia sequence is played back just as it was input at computer **34**[.]” (Carmel, 10:50-54.) A person of ordinary skill in the art would understand playing out the multimedia data “just as it was input” to the computer, by the live source, is the same as playing the data elements “in series.” Playing out multimedia data “in series” means playing it out in the order in which the data was created or recorded by the live source. A person of ordinary skill in the art would have understood this is

disclosed by Carmel.

255. Carmel thus discloses “playing the received media data elements in series from the memory of the media player.”

- (i) **“as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e])**

256. Carmel in view of Hill renders obvious this limitation. For this limitation, I incorporate my discussion of Hill in the context of claim limitation 1[e] in Ground 1. As I discussed in Ground 1, Hill teaches a technique called “prefetching,” the goal of which is “to always have available the next frame required by the control unit 114 when that frame is requested.” (Hill, 4:7-15.) Hill’s buffer manager 118 uses a “request threshold,” which “represents the optimum number of future frames stored in the buffer at any given time,” to maintain the buffer fill level at a predetermined level, *e.g.*, 10 frames. (*Id.*, 6:63-65.) A person of ordinary skill in the art would understand the “request threshold” corresponds to “about a predetermined number of media data elements in the memory of the media player during playing.” Hill thus teaches requesting additional frames (*i.e.*, additional requests for subsequent media data elements) to store in buffer manager 118 (*i.e.*,

for storage in the memory of the media player) until the number of future frames stored in the buffer manager **118** equals the request threshold, such as 10 future frames (*i.e.*, as required to maintain about a predetermined number of media data elements in the memory of the media player).

257. Carmel in view of Hill therefore renders obvious limitation **1[e]**.

258. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Carmel with Hill, predictably resulting in the media streaming system of Carmel that uses Hill's buffering technique, including maintaining predetermined number of data slices in the memory of client **30**. A person of ordinary skill in the art would have known that combining Hill's buffering technique with Carmel's media streaming system would be successful.

259. Carmel and Hill are analogous references in the field of streaming media data from servers to client computers and playing back the media data with minimal delay. Carmel and Hill both disclose systems where media data is sent from a network server to a client's personal computer or workstation for subsequent playback. (*Compare*, Carmel, 6:24-7:17, *with* Hill, 3:57-4:34.) Carmel and Hill both disclose breaking media data into smaller segments and transmitting the segments sequentially. (*Compare* Carmel, 2:1-28, *with* Hill, 3:37-47.) Moreover, a

POSITA would understand from both reference that there are multiple ways to break media data into smaller segments, and that segments of different time periods can be selected. (Carmel, 13:4-6 and 42-46; Hill, 3:37-56.) Carmel and Hill are thus analogous references and disclose similar processes in the field of streaming media data over the Internet.

260. A person of ordinary skill in the art would have been motivated to adopt Hill's buffering technique, including Hill's "request threshold," with Carmel's media streaming system. While Carmel discloses multimedia data can be played back via a Java applet or plug-in on a client **30**, Carmel does not provide a detailed analysis of the client-side software and hardware. Hill, in contrast, provides significant detail for what happens with frames send from a source **122** to the workstation **104** prior to those frames being played back by the media player. (*E.g.*, Hill, 6:35-7:55.) A person of ordinary skill in the art would have been motivated to look to Hill's disclosure of a client-side prefetching technique to pair with Carmel's streaming system. Hill's prefetching technique combined with Carmel's system would help prevent, or at least minimize, delays that result from bandwidth connectivity problems. Hill's prefetching technique would benefit Carmel's system: Carmel teaches opening multiple links if any single link "is operating unacceptably slowly" or if "there is a significant lag" in data receipt. (Carmel, 10:55-64.)

Including a buffer manager **118** at client **30** would avoid such lags. Hill expressly teaches that one advantage of its buffering technique is that the “request threshold” is set based on “the characteristics of a particular system[.]” (Hill, 6:63-67.) A person of skill would be motivated to use Hill’s adaptive buffering technique to better respond to the characteristics of a particular system.

261. Furthermore, buffer manager **118** would be a convenient and advantageous storage place for data slices transmitted across multiple links; buffer manager **118** could hold the data slices provided by multiple links, order them according to the slice indices, and provide the data slices to the display in an ordered fashion. Hill is well-suited for this task, as Hill teaches its system can include multiple sources and “one source manager **120** for each source[.]” (Hill, 4:33-34.) This is analogous to Carmel’s teaching of opening multiple links between server **36** and client **30**; Hill naturally has one source manager **120** for each link between server **36** and client **30**. Hill thus fits neatly within Carmel’s disclosure.

2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

262. Carmel discloses the method of claim 1, “further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

263. It would be obvious to a person of ordinary skill in the art that Carmel's client **30** would maintain in the memory of the media player a record identifying the last data slice received and stored. This would be an implied capability in Carmel's system, and indeed would be necessary for Carmel to function as intended. For example, Carmel teaches a user "may choose to begin downloading data stream **40** from an earlier point in time than that indicated by ID **52**[" where ID **52** is the latest data slice uploaded to server **36**. (Carmel, 8:7-9.) In this case, client **30** would need to maintain a record of the last data slice it received and stored. Suppose the latest data slice uploaded to server **36** is slice **100**, and the user wishes to begin downloading and playing data stream **40** from slice 25. After slice 25 is downloaded and the client **30** is ready to download the next slice, client **30** would *need* to know it already received slice 25 and should next request slice 26. This "back in time" capability would depend on client **30** having a record of the latest data slice received.

264. As yet another example, Carmel teaches client **30** can "open additional links with server **36**." (Carmel, 10:62-63.) When client **30** is operating multiple links, client **30** would need to know the last data slice received to avoid the multiple links duplicating efforts or overwriting data slices already stored, thus frustrating the purpose of opening multiple links, which is to speed up the rate at which data slices are received at client **30**.

265. It would thus be obvious to a person of ordinary skill in the art that Carmel discloses claim 2. To the extent not disclosed by Carmel, Carmel in view of Hill renders obvious claim 2. For this limitation, I incorporate my discussion of claim 2 in Ground 1. As I discussed there, Hill teaches “maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.” As discussed in Ground 1, Hill teaches buffer manager **118** recording the last frame received and requesting the next frame, by frame number, from cliplist manager **116**. (*E.g.*, Hill, 6:7-16, 7:39-56, 12:7-44.)

266. **Manner, Rationale, and Motivation to Combine:** The manner, rationale, and motivation to combine Carmel with Hill is discussed above in Ground 3, limitation **1[e]**. I incorporate that discussion by reference.

3. Claim 3: “The method of claim 1 wherein the serial identifiers are sequential.”

267. Carmel further discloses the method of claim 1, “wherein the serial identifiers are sequential.”

268. As discussed for claim limitations **1[a(i)]**, **1[a(ii)]**, and **1[a(iii)]**, the “serial identifier” of each data slice is the slice index. The slice indices are “sequential;” as Carmel explains, “[c]omputer **34** stores each slice as a corresponding file, having a running slice index 1, 2, 3...N.” (Carmel, 7:27-31.) The serial identifiers (slice indices) of the slices are thus stored and exist

sequentially. “[E]ach file also includes one or more time stamps, indicating a real time at which the data in the file were recorded or an elapsed time relative to the beginning of stream **40**.” (*Id.*, 7:28-31.) It would be obvious to a person of ordinary skill in the art that the serial identifier of the first data slice would reflect an earlier time in the data stream, and the second data slice would reflect a later time in the data stream, and so on. This sequential ordering of data slices is “used to synchronize the data, so that the multimedia sequence is played back just as it was input at computer **34**[.]” (*Id.*, 10:50-54.)

269. Carmel thus discloses the method of claim 1, “wherein the serial identifiers are sequential.”

4. Claim 4: “The method of claim 1, wherein the receiving is via a reliable transmission protocol.”

270. Carmel further discloses the method of claim 1, “wherein the receiving is via a reliable transmission protocol.”

271. As previously discussed, Carmel teaches the server **36** sends data slices to clients **30** “using the well-known Internet Protocol (IP).” (Carmel, 6:29-31.) Carmel specifies that “[c]lients **30** preferably download the sequence using the Hypertext Transfer Protocol (HTTP), although other Internet protocols may also be used, such as UDP or RTP,” or “TCP/IP, UDP or RT(x) protocols[.]” (Carmel, 6:50-54, 7:4-12.) Carmel thus teaches that the server **36** can send data slices to clients **30**

via one of several Internet protocols, including TCP.

272. A person of ordinary skill in the art would understand that at least TCP is a “reliable transmission protocol.” (See EX1045, 1. Abstract (“As a reliable, end-to-end transport protocol, the ARPA Transmission Control Protocol (TCP) uses positive acknowledgements and retransmissions to guarantee delivery.”); EX1046, Abstract (“TCP is a reliable transport protocol...”).) Indeed, the ’594 patent treats TCP as a reliable transmission protocol. (’594 patent, 8:36-40 (“A data communications transport mechanism, such as the **TCP protocol**, may be used for the **reliable delivery** of data in an ordered sequence...from the server to the media player software of the user computer.”) (emphasis added), 14:50-56.)

273. Because Carmel teaches data slices are sent to clients **30** via the TCP protocol, Carmel discloses “wherein the receiving is via a reliable transmission protocol.”

5. Claim 5: “The method of claim 4, wherein the reliable transmission protocol is TCP.”

274. Carmel discloses the method of claim 4, “wherein the reliable transmission protocol is TCP.” Carmel discloses this limitation for the reasons stated for claim 4. I incorporate my analysis of claim 4 here. Carmel teaches server **36** sends data slices to clients **30** using the TCP protocol, which a person of ordinary skill in the art would understand is a “reliable transmission protocol.”

6. Claim 6 (Independent)

275. As discussed for Ground 1, claim 6 is, for purposes of my analysis, materially the same as claim 1 except it is written as a system/device for performing the method recited in claim 1. I copy again the side-by-side table comparing claim 1 with claim 6 (with differences noted in underlining):

Claim 1 (Method)	Claim 6 (System)
1. A <u>method</u> for operating a media player to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the <u>method</u> comprising:	6. A <u>device comprising</u> a media player <u>operable</u> to receive and play an audio or video program, from a remote media source via a data connection over the Internet, the <u>media player</u> comprising:
-	[a] a processor, memory accessible to the processor, and a data connection;
[a(i)] <u>sending</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,	[b(i)] <u>a machine-readable, executable routine containing instructions to cause the processor to send</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>an</u> audio or video program,
[a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,	[b(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,
[a(iii)] each request specifying one or more serial identifiers of the media data elements requested;	[b(iii)] each request specifying one or more serial identifiers of the media data elements requested;
[b(i)] <u>receiving</u> each of the requested media data elements via the data connection, wherein the data connection	[c(i)] <u>a machine-readable, executable routine containing instructions to cause the media player to receive</u> each of the requested media data elements via the

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

has a data rate more rapid than the playback rate of the media data elements, and	data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and
[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;	[c(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;
[c] <u>storing</u> the received media data elements in <u>a</u> memory of the media player;	[d] a machine-readable, executable routine containing instructions to cause the processor to store the received media data elements in the memory of the media player;
[d] <u>playing</u> the received media data elements in series from the memory of the media player; and	[e] <u>a machine-readable, executable routine containing instructions to cause the received media data elements to be played in series from the memory of the media player; and</u>
[e] as the received media data elements are played, <u>the media player automatically sending</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.	[f] <u>a machine-readable, executable routine containing instructions to cause the processor to, as the received media data elements are played, automatically send</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data element in the memory of the media player during playing.

276. As shown in the table above, claim 6 presents no material differences from claim 1 in terms of the basic functions it recites. For purposes of claim 6, as explained in connection with the preamble of claim 1, the “device comprising a media player” is client **30**.

277. With respect to claim **6[a]**, Carmel’s “device comprising a media player” includes “a processor, memory accessible to the processor, and a data connection.” Carmel discloses “clients **30** preferably comprise [a] conventional personal computer[] or workstation[.]” (Carmel, 6:38-40.) It was well known by the ’594 patent’s priority data that a conventional personal computer or workstation would include a processor, memory accessible to the processor, and a data connection. The data connection would be, for example, a connection to the Internet, which Carmel clearly teaches. (*Id.*, Fig. 2, 6:50-54, 7:4-9.) This would have been obvious to a person of ordinary skill in the art.

278. As I discussed for Ground 1, for several limitations, claim 6 broadly recites “a machine-readable, executable routine containing instructions to cause” the processor or media player to perform a designated function. These limitations are not inventive and would have been understood by a person of ordinary skill in the art to be inherent in the functionality of a computer. A computer is designed to include “machine-readable, executable routine[s] containing instructions.” It would have been obvious to a person of ordinary skill in the art that client **30** disclosed by Carmel would include “machine-readable, executable routine[s] containing instructions to cause” client **30** to perform a designated function.

279. For example, with respect to claim limitations **6[b(i)]** through **6[b(iii)]**,

it would be obvious to a person of ordinary skill in the art that the media player (*e.g.*, Java applet and/or plug-in) on client **30** includes “a machine-readable, executable routine containing instructions to cause the processor to” send requests to server **36** for media data elements. I previously analyzed the substantive details of this capability of client **30** in my discussion of claim limitations **1[a(i)]** through **1[a(iii)]**, which I incorporate here.

280. With respect to claim limitations **6[c(i)]** and **6[c(ii)]**, it would be obvious to a person of ordinary skill in the art that the media player on client **30** includes “a machine-readable, executable routine containing instructions to” receive the requested media data elements at a rate as fast as the data connection allows, via a data connection that has a data rate more rapid than the playback rate of the media data elements. I previously analyzed the substantive details of this capability of client **30** in my discussion of claim limitations **1[b(i)]** and **1[b(ii)]**, which I incorporate here.

281. With respect to claim **6[d]**, it would be obvious to person of ordinary skill in the art that the processor in client **30** includes “a machine-readable, executable routine containing instructions to cause” the processor to store the received media data elements in memory. I previously analyzed the substantive details of this capability of client **30** in my discussion of claim limitation **1[c]**, which

I incorporate here.

282. With respect to claim 6[e], it would be obvious to person of ordinary skill in the art that client 30 includes “a machine-readable, executable routine containing instructions to cause” the media data elements to be played in series from the memory of the media player. I previously analyzed the substantive details of this capability of client 30 in my discussion of claim limitation 1[d], which I incorporate here.

283. With respect to claim 6[f], it would be obvious to person of ordinary skill in the art that the processor in client 30 includes “a machine-readable, executable routine containing instructions to cause” the processor to automatically send additional requests for subsequent media data elements to maintain about a predetermined number of media data elements in memory of the media player. I previously analyzed the substantive details of this capability of client 30 in my discussion of claim limitation 1[e], which I incorporate here.

7. Dependent claims 7-10

284. As I discussed for Ground 1, claims 7 through 10 are materially the same as, for purposes of my analysis, claims 2 through 5, as shown in the following chart (underlining showing differences):

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

Claims 2-5 (Method)	Claims 7-10 (System)
2. The <u>method</u> of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.	7. The <u>device</u> of claim 6, further comprising a <u>machine-readable, executable routine containing instructions to cause the processor to maintain in the memory of the media player a record identifying the last media data element received and stored by the media player.</u>
3. The <u>method</u> of claim 1, wherein the serial identifiers are sequential.	8. The <u>device</u> of claim 6, wherein the serial identifiers are sequential.
4. The <u>method</u> of claim 1, wherein the receiving is via a reliable transmission protocol.	9. The <u>device</u> of claim 6, wherein the receiving is via a reliable transmission protocol.
5. The <u>method</u> of claim 4, wherein the reliable transmission protocol is TCP.	10. The <u>device</u> of claim 9, wherein the reliable transmission protocol is TCP.

285. As shown above, the only differences are that claims 7 through 9 depend from system claim 6, claim 10 depends from claim 9, claims 2 through 4 depend from method claim 1, and claim 5 depends from claim 4. Claims 7 through 10 are thus materially the same as corresponding claims 2 through 5; the dependence of claims 7 through 10 on a system, rather than a method, does not affect the applicability of my analysis provided for claims 2 through 5.

286. Because Carmel renders obvious claim 6 just as it renders obvious claim 1, claims 7 through 10 are thus obvious for the same reasons as claims 2 through 5.

8. Claim 11 (Independent)

287. As I discussed for Ground 1, claim 11 is, for purposes of my analysis, materially the same as claim 1 except, like claim 6, it is written as a “computer program product” for performing the method recited in claim 1. I copy again a side-by-side table comparing claim 1 with claim 11 is shown below (with differences noted in underlining):

Claim 1 (Method)	Claim 11 (System)
1. A <u>method for operating a media player to receive and play</u> an audio or video program, from a remote media source via a data connection over the Internet, the <u>method</u> comprising:	11. A computer program product for receiving and playing an audio or video program by a media player having a processor and a memory accessible to the processor, from a remote media source via a data connection over the Internet, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions comprising:
[a(i)] <u>sending</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,	[a(i)] <u>instructions executable to cause the processor to send</u> requests from the media player to the media source via the data connection for one or more serially identified media data elements representing <u>the</u> audio or video program,
[a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,	[a(ii)] each media data element comprising a digitally encoded portion of the audio or video program and having a playback rate,
[a(iii)] each request specifying one or	[a(iii)] each request specifying one or

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

more serial identifiers of the media data elements requested;	more serial identifiers of the media data elements requested;
[b(i)] <u>receiving</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and	[b(i)] <u>instructions executable to cause the processor to receive</u> each of the requested media data elements via the data connection, wherein the data connection has a data rate more rapid than the playback rate of the media data elements, and
[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;	[b(ii)] each received media data element is received at a rate as fast as the data connection between the media source and the media player allows;
[c] <u>storing</u> the received media data elements in <u>a</u> memory of the media player;	[c] <u>instructions executable to cause the processor to store</u> the received media data elements in <u>the</u> memory of the media player;
[d] <u>playing</u> the received media data elements in series from the memory of the media player; and	[d] <u>instructions executable to cause the received media data elements to be played</u> in series from the memory of the media player; and
[e] as the received media data elements are played, <u>the media player</u> automatically <u>sending</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.	[e] <u>instructions executable to cause the processor to</u> , as the received media data elements are played, automatically <u>send</u> additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.

288. As shown in the table, claim 11 presents no material differences from claim 1 in terms of the basic functions it recites. For purposes of claim 11, the “computer program product...comprising a non-transitory computer readable

storage medium having program instructions embodied therewith” corresponds to the software and hardware of Carmel’s clients **30** that are programmed to carry out the method outlined in claim 1. As I discussed above in reference to claim 6, a person of ordinary skill in the art would deem it obvious that Carmel’s system discloses a “computer program product...comprising a non-transitory computer readable storage medium having program instructions embodied therewith.”

289. Accordingly, for the reasons I discussed with respect to independent claims 1 and 6, Carmel renders obvious claim 11.

9. Dependent Claims 12, 13, 16, 17

290. As I discussed for Ground 1, claims 12, 13, 16, and 17 are materially the same as, for purposes of my analysis, claims 2 through 5, as shown in the following chart (underlining showing differences):

Claims 2-5 (Method)	Claims 12, 13, 16, 17 (System)
2. The <u>method</u> of claim <u>1</u> , further comprising <u>maintaining</u> in the memory of the media player a record identifying the last media data element received and stored by the media player.	12. The <u>computer program product</u> of claim <u>11</u> , further comprising <u>instructions executable to cause the processor to maintain</u> in the memory of the media player a record identifying the last media data element received and stored by the media player.
3. The <u>method</u> of claim <u>1</u> , wherein the serial identifiers are sequential.	13. The <u>computer program product</u> of claim <u>11</u> , wherein the serial identifiers are sequential.
4. The <u>method</u> of claim <u>1</u> , wherein the receiving is via a reliable transmission	16. The <u>computer program product</u> of claim <u>11</u> , wherein the receiving is via a reliable transmission protocol.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

protocol.	
5. The <u>method</u> of claim <u>4</u> , wherein the reliable transmission protocol is TCP.	17. The <u>computer program product</u> of claim <u>16</u> , wherein the reliable transmission protocol is TCP.

291. As shown above, the only differences are that claims 12, 13, and 16 depend from system claim 11, claim 17 depends from 16, claims 2 through 4 depend from method claim 1, and claim 5 depends from claim 4. Claims 12, 13, 16, and 17 are thus materially the same as the corresponding claims 2 through 5; the dependence of claims 12, 13, 16, and 17 on a system, rather than a method, does not affect the applicability of my analysis provided for claims 2 through 5.

292. Because Carmel renders obvious claim 11 just as it renders obvious claim 1, claims 12, 13, 16, and 17 are thus obvious for the same reasons as claims 2 through 5.

- 10. Claim 14 (“The computer program product of claim 11, wherein the computer program product is provided as a software application for the media player.”) and Claim 15 (“The computer program product of claim 11, wherein the computer program product is incorporated in the media player.”)**

293. Claims 14 and 15 simply recite the different ways the computer program product recited in claim 11 may be implemented on a user computer. Claim 14 recites that the computer program product is provided as a software application; that is, the computer program product is an optional application that the media player

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

can use in particular situations, such as if the user expressly selects the application. Claim 15, in contrast, recites that the computer program product is embedded in the media player and is therefore used whenever the user's media player is used to stream media data, such as videos or audio.

294. Carmel renders obvious claims 14 and 15. It would be obvious to a person of ordinary skill in the art that Carmel's system could be either provided as a software application that could be used by a personal computer's media player or incorporated into a media player on a personal computer. By the '594 patent's priority date, it was well-known that computer programs could be downloaded as specialized software applications. In fact, Carmel itself provides a software application implementing the disclosed system. (*E.g.*, Carmel, 10:33-34 ("A sample applet of [the sort described above] is incorporated herein in the software appendix, as further described hereinbelow."), 13:53-14:16 (describing contents of "Software Appendix" and explaining "[t]he above files should be entered into a personal computer, preferably a desktop PC (Pentium 166 MHz, with 32 MB RAM), with a Windows 95 or Windows NT Operating system installed").) Carmel thus explicitly teaches its system provided as a "software application" that may be used by a media player installed on a Windows 95 or Windows NT operating system. And a person of ordinary skill in the art could easily implement the same system "incorporated in

the media player” of a Windows 95 or Windows NT operating system. Such was well within the capability of a person of ordinary skill in the art.

295. Carmel thus renders obvious claims 14 and 15.

F. Ground 5: Claims 1-17 Are Obvious Over Carmel In View Of Galensky.

296. Ground 5 incorporates Ground 4’s analysis of Carmel alone but replaces Ground 4’s reliance on Hill with Galensky. Accordingly, limitations 1[e]/6[f]/11[e] will be addressed below. In addition, to the extent Ground 4 relied on combining Carmel with Hill to render obvious dependent claims 2, 7, and 12, Carmel in view of Galensky likewise renders obvious claim 2, 7, and 12. In light of the below analysis and Ground 4’s analysis of the remaining limitations and claims in view of Carmel alone, claims 1-17 are obvious over Carmel in view of Galensky.

1. **“as the received media data elements are played, the media player automatically sending additional requests for subsequent media data elements for storage in the memory of the media player as required to maintain about a predetermined number of media data elements in the memory of the media player during playing.” (claim 1[e])**

297. Carmel in view of Galensky renders obvious claim limitation 1[e]/6[f]/11[e].

298. Galensky teaches a system and method “for receiving, playing and storing streamed multimedia files over a wireless telecommunications network from

a multimedia server.” (Galensky, 3:13-16.) Galensky teaches multimedia server **10** is connected, via telecommunications network **40**, to multiple wireless multimedia devices **80**. (*Id.*, 3:19-63.) Multimedia device **80** is “capable of decoding streamed, compressed data for playback to the user in real time[.]” (*Id.*, 3:52-54.) Galensky teaches “multimedia server **10** [] streams [] selected multimedia files (e.g., digital MP3 audio files, video data files, etc.) [] to the wireless device **80** over link **20** [and] network **40**[.]” (*Id.*, 5:37-45.) The multimedia file “is preferably transmitted...to the network **40** in TCP/IP packets[.]” (*Id.*, 5:41-44.)

299. Device **80** receives the multimedia file and “creates a temporary buffer” to ensure “there is a continuous transmission of data so that the multimedia file (eg., digital MP3 audio file) sounds of high quality.” (Galensky, 5:46-53.) Galensky teaches the buffer helps conserve bandwidth. (*Id.*, 6:2-10.) Galensky teaches the buffer has a “minimum threshold level” of data, such as 2-3 seconds – if the amount of data in the buffer falls below the minimum threshold level, the system increases the rate at which data is sent to the device **80** to maintain the minimum threshold until all the data has been transmitted/received. (*Id.*, 6:10-27, 2:29-47.)

300. A person of ordinary skill in the art would understand Galensky’s user-side buffer “maintain[s] about a predetermined number of media data elements in the memory of the media player during playing.” A person of ordinary skill would

know the “minimum threshold level” of data stored in the buffer means a certain number of media data elements, whose total duration equals the minimum threshold, would be stored within the buffer. Maintaining enough media data for the minimum threshold of playback is “about a predetermined number.”

301. Galensky thus teaches claim limitations 1[e]/[6f]/11[e].

302. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Carmel with Galensky, predictably resulting in the media streaming system of Carmel that uses Galensky’s client-side buffer technique to build up a cache of future data during playback. A person of ordinary skill in the art would have known that combining Galensky’s buffer technique with Carmel’s media streaming system would be successful.

303. Carmel and Galensky are analogous references in the field of streaming media from servers to client devices. Both Carmel and Galensky disclose media data being stored in a server connected to client devices over the Internet. (*Compare* Carmel, 6:24-7:17, *with* Galensky 3:20-4:41.) Carmel and Galensky both acknowledge the problems of bandwidth capacity and strive to solve these problems. (*Compare* Carmel, 7:35-49, 10:55-11:22, *with* Galensky, 6:1-27.) Galensky and Carmel are thus analogous references and disclose similar processes in the field of streaming media data, particularly audio and video data, over the internet.

304. A person of ordinary skill in the art would have been motivated to adopt Galensky's buffering technique for Carmel's media streaming system. As discussed in Ground 3 with respect to Hill, a person of skill would recognize buffering data on the client's device would facilitate smoother playback and minimize dropouts and lags; storing a predetermined number of data slices, such as enough slices total 10 seconds of playback, would benefit Carmel's system. Indeed, client-side buffering was known in the art and would have required little effort to incorporate into Carmel's clients **30**. Although Galensky relates to telecommunications devices, such as mobile communications devices, a person of ordinary skill in the art would not understand Galensky's disclosures to be so limited. The art of streaming data from a server to a client device includes streaming to both mobile devices and computers or workstations; a method for one client device would work equally well for another. And Galensky describes streaming to personal computers in the "Background of the Invention" section, suggesting that the invention was created with that background in mind and could be translated to personal computers. (*E.g.*, Galensky, 1:13-40.)

305. A person of ordinary skill in the art would have been motivated to combine Galensky with Carmel. Accordingly, Carmel in view of Galensky renders obvious limitation **1[e]**.

2. Claim 2: “The method of claim 1, further comprising maintaining in the memory of the media player a record identifying the last media data element received and stored by the media player.”

306. For Ground 4, I relied on Carmel alone as well as Carmel in view of Hill for claim 2. Ground 4 adds Galensky and removes Hill. Nonetheless, it is my opinion that Carmel alone and in view of the knowledge of a person of ordinary skill in the art renders obvious claims 2, 7, and 12.

307. As I discussed above in the context of Ground 4, it would have been obvious to a person of ordinary skill in the art that Carmel teaches maintaining a record identifying the last data slice received and stored in client **30**. Particularly given Carmel’s teachings of downloading a data stream from an earlier time and opening multiple links between server **36** and a single client **30**, this capability would have been inherent in Carmel’s system.

308. My opinion is supported by the state of the art and knowledge of a person of ordinary skill in the art at the time of the ’594 patent’s priority date. It was well known that devices receiving streamed data should keep track of the data elements they already received. Hill is but one example. Maintaining a record of received data was a known way “to ensure the files do not overwrite each other.” (EX1056, 3:39-48 (“When a file is received, the Receive File Manager saves it in a pending directory until the entire file is received.... Within the pending directory, if

a filename conflict occurs, the conflict is resolved to ensure the files do not overwrite each other.”.) Another reference, titled “System For Just-In-Time Retrieval Of Multimedia Files Over Computer Networks By Transmitting Data Packets At Transmission Rate Determined By Frame Size,” disclosed that “[t]o detect lost packets” of data, the client “uses a register to maintain a variable Last Pkt. Seq. No. (51), which is the packet sequence number of the last received packet.” (EX1027, 7:24-28.) Furthermore, in client-pull streaming architectures – which Carmel is – “the complexity and the task of maintaining continuous playback (or in general, meeting the QoS requirements of the applications)” is delegated to the client device, so that the server can be “relatively simple and maintain[] very little state information.” (EX1028, 0003 (§ 4.1).) It thus would have been obvious that the client **30** in Carmel would maintain a record of each data slice is received.

G. Ground 6: Claims 1-17 Are Obvious Over Carmel In View Of Hill And Ravi.

309. Ground 6 incorporates Ground 4’s analysis of Carmel and Hill but adds Ravi’s teachings for limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17. Accordingly, only limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are addressed in the discussion below. In light of the below analysis of these limitations and Ground 4’s analysis of the remaining limitations and claims, claims 1-17 are obvious based on Carmel in view of Hill and Ravi.

310. Limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** each recite “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.” Claims 4, 9, and 16 further recite the receiving is via “a reliable transmission protocol,” and Claims 5, 10, and 17 specify that reliable transmission protocol is TCP. As discussed in Ground 4, Carmel renders obvious these limitations. Carmel teaches data slices are transmitted to clients using any well-known Internet protocol, such as TCP. (Carmel, 6:50-54, 7:5-9.) It would have been obvious to a person of ordinary skill in the art that a server sending data via Internet protocol, such as TCP, would send the data as fast as the connection allows.

311. To the extent not rendered obvious based on Carmel alone, limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are obvious based on Carmel in view of Ravi.

312. Ravi discloses a method for “the efficient and reliable delivery of multimedia streams over a diverse computer network with dynamically variable bandwidth capacity.” (Ravi, 1:55-60; *see also id.*, 3:1-10 (“The present invention provides efficient transmission of multimedia streams from a server to a client computer over a diverse computer network...such as the internet. Examples of multimedia streams provided to the client computer include a compressed video stream [or] a compressed audio stream[.]”).) Ravi teaches a method wherein “the

client computer includes a playout buffer, and the transmission rate [from the server to the client] is dynamically matched to the available bandwidth capacity of the network connection between the server and the client computer.” (Ravi, 3:11-14.) Ravi discloses an “efficient and reliable streaming of data packets” from a server to a client computer, where the server “dynamically selects transmission rates in order to better match the varying bandwidth capacity of the network connection.” (Ravi, 6:32-33, 6:41-44; *see generally id.*, 6:32-47.) Ravi thus teaches transmitting packets of media data from a server to a client computer and selecting a transmission rate to “match the...bandwidth capacity of the network connection.” (*Id.*, 6:43-44.) A person of ordinary skill in the art would understand that Ravi’s system sends data “as fast as the data connection...allows.” If the transmission rate “matches” the capacity of the data connection’s bandwidth, the transmission rate is “as fast as” the data connection allows. If the bandwidth of the data connection increases, Ravi’s servers would increase the transmission rate to “match” the allowable transmission rate. Likewise, if the bandwidth of the data connection decreases, Ravi’s servers would decrease the transmission rate until the rate matches the allowable data rate.

313. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Carmel with Hill and Ravi, predictably resulting in the media streaming system of Carmel that uses Ravi’s

efficient and reliable method of transmitting data slices by matching the transmission rate to the available bandwidth and Hill's buffer manager. It is readily apparent to a person of ordinary skill in the art that combining Ravi's transmission methods and Hill's buffer manager with Carmel's media streaming system would be successful.

314. Carmel, Hill, and Ravi are analogous references in the field of streaming media from servers to client computers. Carmel and Hill both disclose systems where media data is sent from a network server to a client's personal computer or workstation for subsequent playback. (*Compare*, Carmel, 6:24-7:17, *with* Hill, 3:57-4:34.) Carmel and Hill both disclose breaking media data into smaller segments and transmitting the segments sequentially. (*Compare* Carmel, 2:1-28, *with* Hill, 3:37-47.) Ravi states its methods "relate[] to multimedia communications" and more particularly to "delivery of multimedia streams over a diverse computer network[.]" (Ravi, 1:55-60.) Ravi's system is substantially similar to Carmel's and Hill's: it includes a "[s]ource[] for generat[ed] video streams[.]" including "a video camera." (*Id.*, 6:10-11.) It also includes a stream server which is "coupled to client computer 240 via a computer network [such as] the internet." (*Id.*, 5:43-51.) Ravi also teaches media data is transmitted to clients as "streamed [data] packets," which is similar to Carmel's disclosure of transmitting data slices to clients and Hill's disclosure of breaking movie data into units called frames. (*Id.*,

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

6:27-61.) Ravi, Hill, and Carmel are thus analogous references and disclose similar processes in the field of streaming media data, particularly audio and video data, over the internet.

315. A person of ordinary skill in the art would have been motivated to adopt Ravi's method of reliably streaming media data across the internet for Carmel's media streaming system combined with Hill's buffer manager. A person of ordinary skill in the art would understand the problems associated with streaming data over the internet, which can result in dropouts or data loss due to fluctuations in bandwidth capacity. Carmel even recognizes this problem. (*E.g.*, Carmel, 1:23-47 (describing issues in the prior art, including "bandwidth limitations of a network").) It would therefore be natural for a person of ordinary skill in the art to combine Carmel's system with Ravi's dynamic method of matching the transmission rate to the available bandwidth. Ravi teaches computing performance variables and client computing rates to determine to the appropriate transmission rate at which to send data packets from the server to the client. (*E.g.*, Ravi, 11:8-41.) A person of ordinary skill in the art would have been motivated to use this method in Carmel's system to assure optimal streaming performance and therefore optimal media playback at client computers. Carmel itself suggests using feedback to alter the rate at which data slices are transmitted. (Carmel, 4:63-65 ("Preferably, the transmitting computer

compares the upload rate to the data rate and adjusts the upload rate responsive to the comparison.”).) A person of ordinary skill in the art would recognize Ravi teaches a method of using feedback to alter the rate at which data slices are transmitted. (*E.g.*, Ravi, 6:40-44 (“Accordingly, server 220, in response to **feedback** from client computer 240, dynamically selects transmission rates in order to better match the varying bandwidth capacity of the network connection.”) (emphasis added).)

316. A person of ordinary skill in the art would have had a reasonable expectation of success in modifying Carmel to add Hill’s buffering technique and buffer manager and employ Ravi’s dynamic packet transmission techniques. As I discussed, Carmel, Hill, and Ravi disclose substantially similar systems that carry out analogous streaming processes. Implementing Ravi’s teachings into Carmel’s system would involve nothing more than routine adaptations to components in Carmel’s structure. In particular, clients **30** in Carmel’s system would be modified to have the capability to compute “performance variables” and “average client packet computational rate[,]” as disclosed in Ravi, to determine whether “the data packets [are] arriving at client computer [**30**] at a faster, equal or slower rate than client computer [**30**] is able to decompress and render the data packets.” (Ravi, 11:15-26.) This would be well within the skills of a person of ordinary skill in the

art. Indeed, Carmel and Ravi disclose identical client computers **30**. (*Compare* Carmel, 6:38-40 (clients **30** “preferably comprise conventional personal computers or workstations”) *with* Ravi, 5:46-49, 4:37-41 (client computers “can be a personal computer (such as an Apple computer, e.g., an Apple Macintosh, an IBM personal computer, or one of the compatibles thereof), a workstation computer (such as Sun Microsystems or Hewlett-Packard workstation), or some other type of computer.”).) It would be obvious that features present in Ravi’s client computers could be easily added, or are already present in, Carmel’s clients **30**.

317. Carmel in view of Hill and Ravi therefore renders obvious limitations **1[b(ii)]**, **6[c(ii)]**, and **11[b(ii)]**. For the reasons provided in Ground 1, Carmel in view of Hill and Ravi renders obvious claims 1-17.

H. Ground 7: Claims 1-17 Are Obvious Over Carmel In View Of Galensky And Ravi.

318. Ground 7 incorporates Ground 5’s analysis of Carmel and Galensky but adds Ravi’s teachings for limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 as found in Grounds 5 and 6. Accordingly, only limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are addressed in the discussion below. In light of the below analysis of these limitations and the analysis of the remaining limitations and claims in Grounds 5 and 6, claims 1-17 are obvious based on Carmel in view of Galensky and Ravi.

319. Limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** each recite “each received media data element is received at a rate as fast as the data connection between the media source and the media player allows.” Claims 4, 9, and 16 further recite the receiving is via “a reliable transmission protocol,” and Claims 5, 10, and 17 specify that reliable transmission protocol is TCP. As discussed in Ground 4, Carmel renders obvious this limitation. Carmel teaches data slices are transmitted to clients using any well-known Internet protocol, such as TCP. (Carmel, 6:50-54, 7:5-9.) It would have been obvious to a person of ordinary skill in the art that a server sending data via Internet protocol, such as TCP, would send the data as fast as the connection allows.

320. To the extent not rendered obvious based on Carmel alone, limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]** and claims 4, 5, 9, 10, 16, and 17 are obvious based on Carmel in view of Ravi.

321. Ravi discloses a method for “the efficient and reliable delivery of multimedia streams over a diverse computer network with dynamically variable bandwidth capacity.” (Ravi, 1:55-60; *see also id.*, 3:1-10 (“The present invention provides efficient transmission of multimedia streams from a server to a client computer over a diverse computer network...such as the internet. Examples of multimedia streams provided to the client computer include a compressed video

stream [or] a compressed audio stream[.]”).) Ravi teaches a method wherein “the client computer includes a playout buffer, and the transmission rate [from the server to the client] is dynamically matched to the available bandwidth capacity of the network connection between the server and the client computer.” (*Id.*, 3:11-14.) Ravi discloses an “efficient and reliable streaming of data packets” from a server to a client computer, where the server “dynamically selects transmission rates in order to better match the varying bandwidth capacity of the network connection.” (*Id.*, 6:32-33, 6:41-44; *see generally id.*, 6:32-47.) Ravi thus teaches transmitting packets of media data from a server to a client computer and selecting a transmission rate to “match the...bandwidth capacity of the network connection.” (*Id.*, 6:43-44.) A person of ordinary skill in the art would understand that Ravi’s system sends data “as fast as the data connection...allows.” If the transmission rate “matches” the capacity of the data connection’s bandwidth, the transmission rate is “as fast as” the data connection allows. If the bandwidth of the data connection increases, Ravi’s servers would increase the transmission rate to “match” the allowable transmission rate. Likewise, if the bandwidth of the data connection decreases, Ravi’s servers would decrease the transmission rate until the rate matches the allowable data rate.

322. **Manner, Rationale, and Motivation to Combine:** It would have been obvious for a person of ordinary skill in the art to combine Carmel with Galensky

and Ravi, predictably resulting in the media streaming system of Carmel that uses Ravi's efficient and reliable method of transmitting data slices by matching the transmission rate to the available bandwidth and Galensky's client-side buffer technique to build up a cache of future data during playback. It is readily apparent to a person of ordinary skill in the art that combining Ravi's transmission methods and Galensky's buffer technique with Carmel's media streaming system would be successful.

323. Carmel, Galensky, and Ravi are analogous references in the field of streaming media from servers to client computers. Both Carmel and Galensky disclose media data being stored in a server connected to client devices over the Internet. (*Compare* Carmel, 6:24-7:17, *with* Galensky 3:20-4:41.) Carmel and Galensky both acknowledge the problems of bandwidth capacity and strive to solve these problems. (*Compare* Carmel, 7:35-49, 10:55-11:22, *with* Galensky, 6:1-27.) Ravi is a solution to this problem and states its methods "relate[] to multimedia communications" and more particularly to "delivery of multimedia streams over a diverse computer network[.]" (Ravi, 1:55-60.) Ravi's system is substantially similar to Carmel's: it includes a "source[] for generat[ed] video streams," including "a video camera." (*Id.*, 6:10-11.) It also includes a stream server which is "coupled to client computer 240 via a computer network [such as] the internet." (*Id.*, 5:43-51.)

Ravi also teaches media data is transmitted to clients as “streamed [data] packets,” which is similar to Carmel’s disclosure of transmitting data slices to clients and Galensky’s preference for transmitting files in packets. (*Id.*, 6:27-61.) Carmel, Galensky, and Ravi are thus analogous references and disclose similar processes in the field of streaming media data, particularly audio and video data, over the internet.

324. As discussed in Ground 5, a person of ordinary skill in the art would have been motivated to adopt Galensky’s buffering technique for Carmel’s media streaming system. A person of skill would recognize buffering data on the client’s device would facilitate smoother playback and minimize dropouts and lags; storing a predetermined number of data slices, such as enough slices to total 10 seconds of playback, would benefit Carmel’s system. Although Galensky relates to telecommunications devices, such as mobile communications devices, a person of ordinary skill in the art would not understand Galensky’s disclosures to be so limited. The art of streaming data from a server to a client device includes streaming to both mobile devices and computers or workstations; a method for one client device would work equally well for another. And Galensky describes streaming to personal computers in the “Background of the Invention” section, suggesting that the invention was created with that background in mind and could be translated to personal computers. (*E.g.*, Galensky, 1:13-40.)

325. A person of ordinary skill in the art would have been motivated to adopt Ravi's method of reliably streaming media data across the internet for Carmel's media streaming system combined with Galensky's buffering technique. A person of ordinary skill in the art would understand the problems associated with streaming data over the Internet, which can result in dropouts or data loss due to fluctuations in bandwidth capacity. It would therefore be natural for a person of ordinary skill in the art to combine Carmel's system with Ravi's dynamic method of matching the transmission rate to the available bandwidth. Ravi teaches computing performance variables and client computing rates to determine the appropriate transmission rate at which to send data packets from the server to the client. (*E.g.*, Ravi, 11:8-41.) A person of ordinary skill in the art would have been motivated to use this method in Carmel's system to assure optimal streaming performance and therefore optimal media playback at client computers. Carmel itself suggests using feedback to alter the rate at which data slices are transmitted. (Carmel, 4:63-65 ("Preferably, the transmitting computer compares the upload rate to the data rate and adjusts the upload rate responsive to the comparison.")) A person of ordinary skill in the art would recognize Ravi teaches a method of using feedback to alter the rate at which data slices are transmitted. (*E.g.*, Ravi, 6:40-44 ("Accordingly, server 220, in response to **feedback** from client computer 240, dynamically selects transmission

rates in order to better match the varying bandwidth capacity of the network connection.”) (emphasis added).)

326. A person of ordinary skill in the art would have had a reasonable expectation of success in modifying Carmel to add Galensky’s buffering technique and to employ Ravi’s dynamic packet transmission techniques. As I discussed, Carmel, Galensky, and Ravi disclose substantially similar systems that carry out analogous streaming processes. Implementing Ravi’s teachings into Carmel’s system would involve nothing more than routine adaptations to components in Carmel’s structure. In particular, clients **30** in Carmel’s system would be modified to have the capability to compute “performance variables” and “average client packet computational rate,” as disclosed in Ravi, to determine whether “the data packets [are] arriving at client computer [30] at a faster, equal or slower rate than client computer [30] is able to decompress and render the data packets.” (Ravi, 11:15-26.) This would be well within the skills of a person of ordinary skill in the art. Indeed, Carmel and Ravi disclose identical client computers **30**. (*Compare* Carmel, 6:38-40 (clients **30** “preferably comprise conventional personal computers or workstations”) *with* Ravi, 5:46-49, 4:37-41 (client computers “can be a personal computer (such as an Apple computer, e.g., an Apple Macintosh, an IBM personal computer, or one of the compatibles thereof), a workstation computer (such as Sun

Microsystems or Hewlett-Packard workstation), or some other type of computer.”.)

It would be obvious that features present in Ravi’s client computers could be easily added, or are already present in, Carmel’s clients **30**.

327. Carmel in view of Galensky and Ravi therefore renders obvious limitations **1[b(ii)]/6[c(ii)]/11[b(ii)]**. For the reasons provided above and in Ground 5, Carmel in view of Galensky and Ravi renders obvious claims 1-17.

VII. NO SECONDARY CONSIDERATIONS OF NONOBVIOUSNESS

328. I understand from counsel that Patent Owner in the underlying district court litigation has not yet identified any evidence with respect to secondary considerations of non-obviousness. To the extent Patent Owner subsequently identifies any such evidence, I reserve the right to consider and revise any of my opinions should it be necessary to do so.

Declaration of Nathaniel Polish in Support of
Petition for *Inter Partes* Review of
U.S. Patent No. 9,729,594 B2

VIII. CONCLUSION

329. I hereby declare that all statements made herein of my own knowledge are true and that all statements are made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

Dated: August 11, 2022

Respectfully submitted,



Nathaniel Polish, Ph.D.