IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE PATENT TRIAL AND APPEAL BOARD

In the *Inter Partes* Review of:	)
	)
U.S. Patent No.: 9,742,824	)
	)
	)
	)
For: STREAMING MEDIA DELIVERY SYSTEM	)
	)
	)
	)
	)
	)
	)

Mail Stop Patent Board
Patent Trial and Appeal Board
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF KEVIN JEFFAY

TABLE OF CONTENTS

Page

TABLE OF CONTENTS
(Continued)

Page

TABLE OF CONTENTS
(Continued)

Page

I, Kevin Jeffay, Ph.D., declare as follows:

I. INTRODUCTION

1. My name is Kevin Jeffay, and I have been asked by the parties requesting this review, Amazon.com, Inc., Amazon Web Services, Inc., and Amazon.com Services LLC (collectively "Petitioner") to analyze U.S. Patent No. 9,742,824 (the "'824 patent") (EX1001) and to provide my opinions regarding the patentability of claims 1-12 of the '824 patent.

2. I am being compensated for my time. This compensation is not contingent upon my performance, the conclusions I reach in my analysis, the outcome of this matter, or any issues involved in or related to this matter.

II. BACKGROUND AND QUALIFICATIONS

3. I am a tenured professor in the Department of Computer Science at the University of North Carolina at Chapel Hill where I currently hold the position of Gillian T. Cell Distinguished Professor of Computer Science. I also recently served as the Chairman of the Department (from 2014-2022). I have been a faculty member at UNC since 1989.

4. I received a Ph.D. in computer science from the University of Washington in 1989. Before that I received a M.Sc. degree in computer science from the University of Toronto in 1984, and a B.S. degree with Highest Distinction in mathematics from the University of Illinois at Urbana-Champaign in 1982.

5. I have been involved in the research and development of computing systems for nearly 40 years. I have been a faculty member at the University of North Carolina since 1989 where I perform research, and I teach in the areas of multimedia networking, computer networks, distributed systems, real-time systems, and operating systems, among others. A major theme of my research has been the development of technology to improve the performance of data transfers on the Internet. My research has examined problems ranging from network and operating system support for real-time multimedia applications such as audio and video streaming, voice-over-Internet protocol (VoIP) and Internet videoconferencing, to the design of congestion control mechanisms in network routers, to measurements and analysis of network traffic to passively assess the performance of servers on the Internet.

6. For example, starting in the late 1980s, the focus of my research was the development of network and operating system technology to enable the real-time transfer of streams of audio and video data across the Internet. This involved, among other things, the development of media encoders, media players, and network communication protocols for adaptive transmission of live audio and video data on the Internet. This work culminated in my research group developing some of the first videoconferencing systems for the Internet. Several of the papers authored by myself and members of my research group on this project won awards for their

technical contributions. For example, a 1993 paper authored by my research group on client-side playout buffer management won a best paper award at the Fourth International Workshop on Network and Operating System Support for Digital Audio and Video and was republished in 1995 in the journal ACM Multimedia Systems.

7. The videoconferencing and adaptive streaming research attracted the attention of industry groups such as IBM, Intel, Digital Equipment Corporation, Cabletron, and AT&T Bell Laboratories. For example, starting in 1991 IBM supported aspects of my research at UNC, and these efforts resulted in U.S. Patent No. 5,892,754 on adaptive media streaming being issued to IBM and UNC.

8. During this time, I was also collaborating with researchers at the Intel Architecture Labs in Hillsboro, OR, to modify an Intel product, the ProShare videoconferencing system, to use technology developed in my research lab at UNC for adaptive media transmission. By the late 1990s, the ProShare product included the ability to also transfer desktop information of a PC to a remote PC.

9. My research group also developed other "data conferencing" systems, also known as "shared window systems," that were capable of transferring the desktop information on a PC to a remote PC in real-time. One system that we built, called "XTV," was operational by 1991. The source code for XTV was made freely available and, by 1993, had been downloaded by over 600 users and institutions.

The system was functionally and visually equivalent to today's Cisco's WebEx and Zoom's screen sharing products and services. It provided a sophisticated means of transferring the desktop information of a PC to a remote site. In the XTV system, individual windows displayed on the desktop, or the entire desktop itself, could be selected for sharing with remote users. The XTV system also allowed remote users to remotely control and manipulate the desktop whose information was being distributed, as well as remotely control the applications that were generating the desktop information that was being shared.

10. In much of my research, I regularly build and use clusters of computers interconnected by network switches, bridges, and routers to form and evaluate experimental and production networks. For example, in the late 1990s and early 2000s, my research evolved to consider router-based mechanisms for controlling the performance of network traffic. In much of this research, my students and I built and instrumented network routers and performed large scale experiments with this equipment. The instrumentation included, for example, the development of network monitors that would receive copies of packets flowing on a network link and analyze and store packets or the results of analyses on these packets, all in real-time. Based on these experiments, in 2003, my group at UNC won the most prestigious research award for original research in computer networking. UNC applied for, and received, U.S. Patent No. 7,447,209 for aspects of this research.

11. This project, and others, took place in a networking lab my students and I constructed at UNC over a number of years. The lab consists of several hundred computers and networking devices. Managing this lab required establishing and configuring VLANs, firewalls, and other security appliances to isolate the lab from the campus network (and vice versa).

12. More recently, my research group has considered the design and operation of next generation aerial networks. UNC applied for, and received, U.S. Patent No. 9,832,705 for aspects of this research.

13. I have authored or co-authored over 100 articles in peer-reviewed journals, conference proceedings, texts, and monographs in the aforementioned areas of computer science and others. In addition, I have edited and co-edited numerous published proceedings of technical conferences and have edited a book of readings in multimedia computing and networking (with Hong-Jiang Zhang) published by Morgan Kaufman. I am a co-author (with Long Le and F. Donelson Smith) of a monograph related to computer network protocols, and a co-author (with Jay Aikat and F. Donelson Smith) of a second monograph related to experimental computer networking.

14. My research extends to leadership positions in several journals, conferences, and committees. I have previously served as Editor-in-Chief for the journal Multimedia Systems and Associate Editor for the journal Real-Time

Systems. In addition, I have been an active participant in the Association for Computing Machinery ("ACM") and the Institute of Electrical and Electronics Engineers ("IEEE"). Specifically, I have been a member of the steering committees for the ACM Special Interest Group on Multimedia, and the ACM Special Interest Group on Data Communications ("SIGCOMM") and its subgroup on Internet Measurement. I have also served as a member of the IEEE Technical Committee on Real-Time Systems. As a result of this involvement, I have served as a program chair or member of the technical program committee for over 100 professional, international, and technical conferences, workshops, and symposia. I was previously on the steering committee for the ACM technical committee on Internet measurement, where we worked on community standards for measurement-based research. I have also served on numerous proposal review panels for the National Science Foundation and other international funding agencies in the aforementioned areas of computer science.

15. In addition to the U.S. patents referenced above, I am a named inventor on a fourth patent. Each of these patents is generally related to computer networking and the delivery of services over networks including audio and video transmission.

16. I have developed and taught undergraduate and graduate level courses on operating systems, computer networking, computer security, and multimedia

computing. These courses have been developed for, and delivered to, students at UNC as well as for industry professionals.

17. I have served as an expert witness and technical consultant in litigation and *inter partes* review matters concerning videoconferencing, data conferencing and screen sharing, cellular and wireline telephony, voice over IP (VoIP) telephony, multimedia networking, distributed systems, operating systems, computer networks, datacenter networking, embedded systems and embedded software, and real-time systems, among others. I have testified in several trials, arbitrations, and claim construction hearings as an expert witness.

18. My Curriculum Vitae, attached as Exhibit 1003, includes a more detailed summary of my background, experience, and publications.

III. MATERIALS CONSIDERED

19. In preparing this declaration, I reviewed the '824 patent, including the claims of the patent in view of the specification, and I have reviewed the prosecution history of the '824 patent and numerous prior art and technical references from and before the time of the alleged invention. These references are discussed below.

20. Petitioner's counsel has asked me to consider whether certain references disclose or suggest, alone or in combination, the features recited in certain claims of the '824 patent. I have also been asked to consider the state of the art and

the prior art available before the time of the alleged invention of the '824 patent. My opinions are provided in this declaration.

21. My opinions in this declaration are based on my review of the documents above, my understanding as an expert in the relevant field, and my education, training, research, knowledge, and personal and professional experience.

22. To my knowledge, I have no financial interest in Petitioner. Counsel for Petitioner has informed me that WAG Acquisition, LLC purports to own the '824 patent. To the best of my knowledge, I have no financial interest in WAG Acquisition, LLC and, to my recollection, have had no contact with WAG Acquisition, LLC or the named inventor of the '824 patent, Harold Edward Price. To the best of my knowledge, I do not have any financial interest in the '824 patent.

23. To the extent any mutual funds or other investments that I own have a financial interest in the Petitioner, the Patent Owner, or the '824 patent, I am not aware of, and do not control, any financial interest that would affect or bias my judgment.

IV. LEGAL STANDARDS

24. Petitioner's counsel has informed me that, in an *inter partes* review proceeding, a patent claim may be deemed unpatentable if it is shown by a preponderance of the evidence that the claim was either anticipated by a prior art

patent or publication or rendered obvious by one or more prior art patents or publications.

25. Petitioner's counsel has informed me that a claim is unpatentable if the differences between the subject matter of the patent and the prior art are such that the subject matter as a whole would have been obvious to a person of ordinary skill in the art at the time of the invention, a "POSITA."

26. Petitioner's counsel has informed me that a determination of whether a claim would have been obvious should be based upon several factors, including the following:

- The level of ordinary skill in the art at the time the application was filed;
- The scope and content of the prior art; and
- What differences, if any, existed between the claimed invention and the prior art.

27. Petitioner's counsel has informed me that a single reference can render a patent claim obvious if any differences between that reference and the claims would have been obvious to a person of ordinary skill in the art. Alternatively, the teachings of two or more references may be combined in the same way as disclosed in the claims, if such a combination would have been obvious to one having ordinary skill in the art. In determining whether a combination based on either a single

reference or multiple references would have been obvious, I understand from Petitioner's counsel that it is appropriate to consider the following factors:

- Whether the teachings of the prior art references disclose known concepts combined in familiar ways, and when combined, would yield predictable results;
- Whether a POSITA could implement a predictable variation and would see the benefit of doing so;
- Whether the claimed elements represent one of a limited number of known design choices and would have a reasonable expectation of success by those skilled in the art;
- Whether a person of ordinary skill would have recognized a reason to combine known elements in the manner described in the claim;
- Whether there is some teaching or suggestion in the prior art to make the modification or combination of elements claimed in the patent; and
- Whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way.

28. Petitioner's counsel has informed me that a POSITA has ordinary creativity and is not an automaton.

29. Petitioner's counsel has informed me that all prior art references are to be looked at from the viewpoint of a POSITA.

30. Petitioner's counsel has informed me that, in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered, and that obviousness is analyzed from the perspective of a POSITA at the time of the invention.

V. OVERVIEW OF THE '824 PATENT

31. The '824 patent, titled "Streaming media delivery system," was filed on October 3, 2016, and issued on August 22, 2017.

32. The '824 patent issued from U.S. Patent Application No. 15/283,578 ("'578 application"), filed on October 3, 2016. The '578 application is a continuation of Application No. 13/815,040, filed on Jan 25, 2013, which is a continuation of Application No. 13/385,375, filed on February 16, 2012, which is a continuation of Application No. 12/800,177, filed on May 10, 2010, which is a continuation of Application No. 10/893,814, filed on July 19, 2004, which is a continuation-in-part of Application No. 09/819,337, filed on March 28, 2001, which claims priority to Application No. 60/231,997 ("'997 application"), filed on September 12, 2000 (the "Critical Date").

A. Summary of the Alleged Invention

33. The '824 patent provides in the "Field of Invention" section of the specification that it relates generally to "multimedia computer communication

systems" and more specifically describes "systems and methods for delivering streaming media, such as audio and video, on the Internet." EX1001, 1:52-55.

34. According to the '824 patent, systems purportedly use a "pre-buffering technique to store up enough audio or video data in the user's computer so that it can play the audio or video with a minimum of dropouts." *Id.* at 2:42-45. The user would "start[] the audio or video stream, typically by clicking on a 'start' button, and wait[] ten to twenty seconds or so before the material starts playing." *Id.* at 2:58-62. During that time, audio or video data would be delivered to the user's computer and fill the media player's buffer. *Id.* at 2:62-63.

35. The '824 patent states that in such systems "audio or video data is delivered from the source at the rate it is to be played out." *Id.* at 2:63-3:1 ("[i]f, for example, the user is listening to an audio stream encoded to be played-out at 24,000 bits per second, the source sends the audio data at the rate of 24,000 bits per second"). After ten seconds of waiting, assuming the Internet connection has not been interrupted, "there [was] enough media data stored in the buffer to play for ten seconds." *Id.* at 3:1-4.

36. The '824 patent purportedly describes a streaming media system in which, in addition to a conventional buffer at the user computer, the server uses a first in, first out ("FIFO") server buffer to store streaming media data, and media data is sent from the server buffer to the user computer at a rate faster than the

playback rate, to allegedly protect against interruptions in playback. *Id.* at 6:13-29, 9:62-10:1, 13:66-14:5, 14:62-15:19.

37. The '824 patent states that "[w]ith the present invention . . . the server 12 transmits audio/video data as sequential data elements from its buffer 14 to the buffer 20 of the user, at a higher than playback rate." *Id.* at 9:38-41. The use of server-side buffering purportedly allows for a significant amount of media data—for example, one minute of data—to be quickly transferred from the server buffer to the user buffer, at a rate faster than the playback rate, so that the media data can be played out to the user continuously "despite data reception interruptions of less than a minute." *Id.* at 10:7-19; *see also id.* at 8:61-9:2. If the user buffer level has decreased due to interruptions in the flow of data to the user computer, the user computer requests additional media data elements to re-fill the user buffer while the media data continues to be played out. *Id.* at 6:25-29, 15:9-22; *see also id.* at 10:24-42.

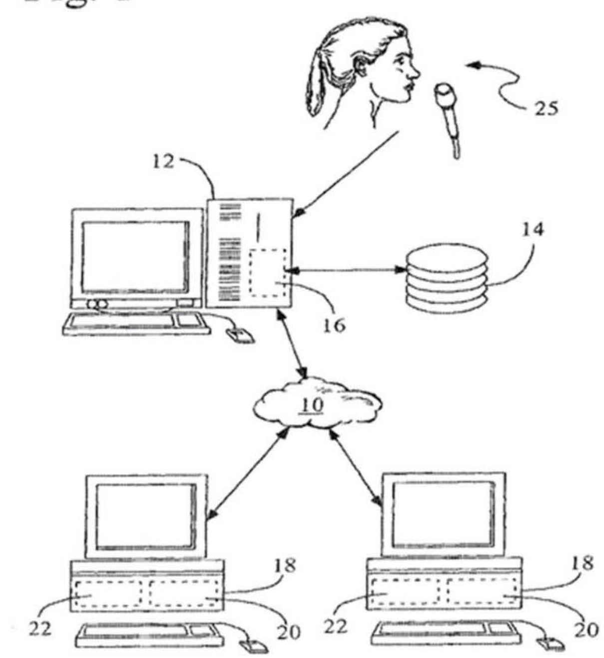
38. The '824 patent discloses two approaches to sending media data from the server buffer to the user computer: (1) a server-push embodiment in which the server selects the media data elements to send and sends them to the user computer on a pre-set schedule that is synchronized with the playback rate of the media data (*id.* at 9:38-63, 10:43-48), and (2) a client-pull embodiment, in which the user

computer sends requests to the server to send specific media data elements. *Id.* at 14:48-66.

39. The claims of the ’824 patent are directed to a client-pull embodiment, where the prerecorded media data elements stored in a server’s data structure are streamed based on requests from clients for such data by an identifier. *Id.* at claim 1, 3:65-4:12. This is known in the prior art as a client-pull system because the client requests the data from the server. This contrasts with a “server-push” system, where the server initiates the streaming to the client.

40. FIG. 1 of the ’824 patent (below) “is a schematic/block diagram illustrating the elements of a streaming media buffering system.” *Id.* at 4:23-25.

Fig. 1



41. In FIG. 1, "the system is provided with a server 12 connected to the Internet 10 for transmitting the streaming media data elements. Associated with the server 12 is a server buffer 14 for storing at least one of the data elements for transmission . . . [where] Buffer 14 is a conventional computer storage mechanism such as a hard disk." *Id.* at 6:32-41.

42. The '824 patent describes that using "standard data communications protocol techniques such as TCP, the user computer transmits a request to the server to send one or more data elements" which specifies "the serial numbers of the data elements." *Id.* at 14:56-59. In response to the request, the server sends the requested data elements to the user computer. *Id.* at 14:59-62. The media data is sent from the server buffer 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." *Id.* at 8:9-20, 9:62-67; *see also id.* at 14:66-15:1 (media data is sent to the user computer "as fast as the data connection between the user computer and the server will allow"). For example, if the media data is encoded for playback at 24,000 bits per second, and the user's Internet connection is at 56,000 bits per second, the server sends media data to the user computer at 56,000 bits per second. *Id.* at 9:64-10:6.

43. The '824 patent claims recite that the "data connection between the server system and each requesting user system has a data rate more rapid than the

playback rate" and "each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow." *Id.* at claim 1. Such client-pull systems existed well before the '824 patent.

44. The technology surrounding the delivery of streaming media over the Internet pre-dates the alleged priority date of the '824 patent. Each of the system components recited in the claims and described in the specification were well-known and conventional components.

45. The '824 patent contains twelve claims, of which claims 1, 5, and 9 are independent. Independent claim 1 recites:

> 1. A method for distributing over the Internet, from a server system to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media, the method comprising:
>
> reading, by at least one computer of the server system, the pre-recorded audio or video program from the computer-readable media;
>
> supplying, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate;
>
> serially identifying the media data elements, said serial identification indicating a time sequence of the media data elements;
>
> storing the media data elements in a data structure under the control of the server system;
>
> receiving requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial

identifiers of the requested one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and

responsive to the requests, sending, by the server system, the one or more media data elements having the one or more specified serial identifiers, to the requesting user systems corresponding to the requests; wherein

the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;

each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow;

the one or more media data element sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems;

all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests; and

all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein.

46. Independent claims 5 and 9 recite similar subject matter except in a server and computer program product, respectively. Dependent claims 2, 6, and 10 recite that the "serial identifiers are sequential"; claims 3, 7, and 11 recite that the

"sending is via a reliable transmission protocol"; and claims 4, 8, and 12 recite that the "reliable transmission protocol is TCP."

B. Prosecution History

47. I understand the '578 application was filed on October 3, 2016, and included claims 1-12, of which claims 1, 5 and 9 were independent. EX1004, 370-375. In an Office Action dated January 11, 2017, the pending claims were rejected based on (1) non-statutory double patenting over application no. 15/283,544 (now U.S. Patent No. 9,762,636) and (2) 35 U.S.C. § 103 in view of "Hooper et al. (Patent number US 5,414,455) . . . Omoigui (US 7,237,254) . . . Hodgkinson et al. (US 7,209,437) . . . Chen et al. (US 5,822,524)." *Id.* at 215-238.

48. Applicant amended independent claims 1, 5, and 9, in a Response dated March 29, 2017, to recite "<u>said serial identification indicating a time sequence of the media data elements</u>"[1] and argued that this claim amendment would be sufficient to overcome the prior art of record. *Id.* at 166-175. Applicant also filed a terminal disclaimer to overcome the non-statutory double patenting rejection. *Id.* at 176.

49. A Notice of Allowance subsequently issued on June 19, 2017, stating "the prior art . . . does not disclose . . . that the one or more media data elements sent are selected without depending on the server system maintaining a record of the last

---

[1] Unless otherwise noted, all underlining is added.

media data element sent to the requesting user systems." *Id.* at 37-43. The '578 application issued as the '824 patent on August 22, 2017. *Id.* at 8.

C. Priority Date

50. I do not offer any opinion as to whether the '824 patent can claim priority to provisional Application No. 60/231,997, filed on September 12, 2000.

VI. LEVEL OF ORDINARY SKILL IN THE ART

51. In my opinion, a person having ordinary skill in the art ("POSITA") for the '824 patent 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 streaming media systems for delivering audio and video. Additional education could have substituted for professional experience, and significant work experience could have substituted for formal education.

52. As of the filing date of the earliest application that the '824 patent claims priority to (i.e., September 12, 2000), including up to and including the filing date of the application (i.e., October 3, 2016) resulting in the '824 patent, I was a person having ordinary skill in the art.

VII. CLAIM CONSTRUCTION

53. Petitioner's counsel has informed me that claims subject to *inter partes* review are construed according to the plain and ordinary meaning of the claim as understood by a POSITA and the prosecution history of the patent being construed.

54. For purposes of this declaration, I have applied the plain and ordinary meaning of the claims when read in light of the '824 patent and the prosecution history of the '824 patent, as understood by a POSITA at the time of the invention.

VIII. SUMMARY OF OPINIONS

55. In my opinion, the challenged claims of the '824 patent, including claims 1-12, are invalid as obvious to a person of ordinary skill in the art as of the Critical Date. Further, in my opinion the Carmel and Feig references are prior art to the '824 patent under 35 U.S.C. § 102(e)[2] and the Willebeek reference is prior art under 35 U.S.C. § 102(b)[3].

56. This declaration reflects my opinions that I have formed to date, based on my review of the materials identified in Section III. I reserve the right to revise, supplement, or amend my opinions based on new information that becomes available to me, and by further continuing analysis of the materials identified in Section III.

---

[2] Carmel was filed as a patent application on March 24, 1999, and Feig was filed as a patent application on June 17, 1998, before the Critical Date (September 12, 2000) of the '824 patent. *See* 35 U.S.C. § 102(e).

[3] Willebeek was published in 1998, more than one year before the Critical Date of the '824 patent. *See* 35 U.S.C. § 102(b).

A. Ground 1: Claims 1-12 are rendered obvious over Carmel in view of Feig and Willebeek.

57. U.S. Patent No. 6,389,473 to Carmel et al. ("Carmel"), entitled "Network media streaming," was filed March 24, 1999, and granted on May 14, 2002. EX1005. Carmel is relevant prior art given the PTAB on December 26, 2017, found claims 10, 11, 13-21, and 23 of related U.S. Patent No. 8,122,141 ('141 patent) (EX1015) anticipated by Carmel and claims 12 and 22 obvious over Carmel. Final Written Decision, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 22 (Dec. 26, 2017) at 33. The '141 patent has overlapping claim language with the '824 patent. *See, e.g.*, EX1001, claim 1 ("media data elements" and "serial identifiers"); EX1015, claim 1 ("media data elements" and "serial identifiers"). On remand, the PTAB again found claims 10-18 unpatentable over Carmel. Final Written Decision on Remand, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 28 (July 16, 2020) at 25-26. Additionally, "Patent Owner did not appeal the Board's prior conclusion that claims 19-23 are unpatentable." *Id.* at 3.

58. Carmel is also relevant because it is analogous prior art to the claimed invention of the '824 patent. Carmel is from the same field of endeavor of multimedia streaming as the '824 patent. *See, e.g.*, EX1001, 1:51-55 ("The present invention relates to multimedia computer communication systems; and more particularly, to systems and methods for delivering streaming media, such as audio and video, on the Internet."); EX1005, 1:10-13 ("The present invention relates

generally to network data communications, and specifically to real-time multimedia broadcasting over a network."). Moreover, Carmel is reasonably pertinent to the problem faced by the inventor of the '824 patent. *See, e.g.*, EX1001, 2:34-45 ("users viewing or listening to streaming content over Internet connections often encounter interruptions . . . commonly referred to as '*dropouts*'"); EX1005, 12:10-12 ("when the slice durations are shorter, the effect of '*drop-out*' of a slice due to failure of the corresponding link is less marked").

59. Carmel was listed in an information disclosure statement, but it was not substantively considered during prosecution of the '824 patent. EX1004, 12. Critically, the Examiner never considered the PTAB's final written decision on remand above.

60. U.S. Patent No. 6,175,862 to Jeane Shu-Chun Chen and Ephraim Feig ("Feig"), entitled "Hot objects with sequenced links in web browsers," was filed on June 17, 1998, and granted on January 16, 2001. EX1031. Feig is analogous prior art to the claimed invention of the '824 patent. Feig is from the same field of endeavor of multimedia streaming as the '824 patent. *See, e.g.,* EX1031, 1:6-10 ("The present invention relates to Internet and Intranet Browsers, more specifically, to Browsers equipped with functionality to process a sequence of URL requests automatically. This capability allows the Browser to induce a non-streaming server to simulate a streaming server."). Moreover, Feig is reasonably pertinent to the

problem faced by the inventor of the ’824 patent. *See, e.g.*, EX1031, 2:10-13 (“Without explicit mechanisms to ensure isochronism, delivery rates of data to a browser are irregular, resulting in erratic playback quality at client machines.”). Feig was not before the Examiner during prosecution of the ’824 patent.

61. “Bamba—Audio and video streaming over the Internet,” authored by M. H. Willebeek-LeMair et al. (“Willebeek”), was published March 1998 in the IBM Journal of Research and Development. EX1006, 1. Willebeek is relevant prior art at least because it was listed in an information disclosure statement during prosecution of the ’824 patent. EX1004, 257. The Examiner did not substantively consider Willebeek during prosecution.

62. Willebeek is also relevant because it is analogous prior art to the claimed invention of the ’824 patent. Willebeek is from the same field of endeavor of multimedia streaming as the ’824 patent. *See, e.g.*, EX1001, 1:51-55; EX1006, 1 (“we present a media-streaming system, called Bamba, that delivers audio and video”). Moreover, Willebeek is reasonably pertinent to the problem faced by the inventor of the ’824 patent. *See, e.g.*, EX1001, 2:41-56 (“Even with this *pre-buffering* process, interruptions in playback still occur.”); EX1006, 5-6 (“[t]he amount of *pre-buffering* is” based on “the initial download rate”).

63. Willebeek is also relevant prior art because the PTAB found claim 21 of the '141 patent obvious over Willebeek. Final Written Decision, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 22 (Dec. 26, 2017) at 33.

1. Overview of Carmel

64. Carmel is directed towards a method for streaming live or prerecorded media from a server to multiple client computers over the Internet. EX1005, 2:1-21, 6:24-27, 6:57-60, 2:29-31.

65. Figure 2 of Carmel (below) is a schematic illustration of a computer system (32) for broadcasting of a multimedia sequence of a network (28). *Id.* at 6:24-27.

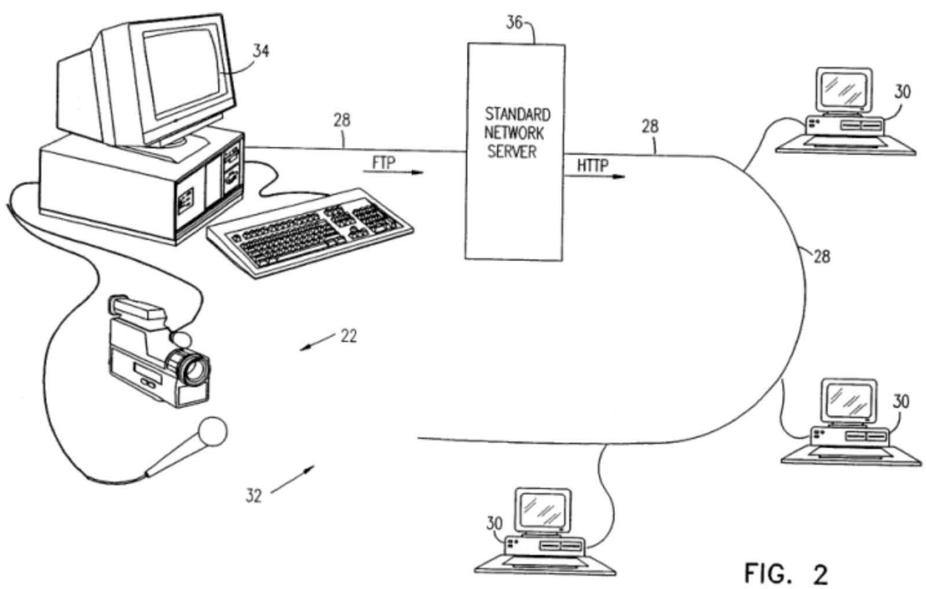
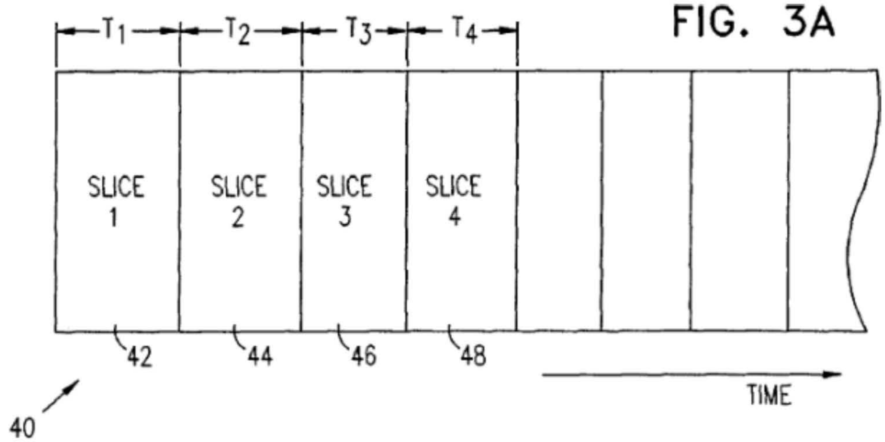


FIG. 2

66. System 32 comprises transmitting computer 34 that receives audio visual input from devices 22, a plurality of clients 30, and network server 36. *Id.* at 6:28-35.

67. The transmitting computer 34 generates a multimedia data sequence (data stream 40), which comprises a series of sequential data slices 42, 44, 46, 48, etc., with each slice containing a segment of video and/or audio data that corresponds to a respective, successive time interval T1, T2, T3, etc. *Id.* at 7:18-25, Fig. 3A (below).

FIG. 3A



68. The server sends media data elements to the client using Internet protocols, such as HTTP. *Id.* at 5:25-28, 6:28-31, 6:36-38. Clients 30 connected with server 36 read an index file containing such numbered slices and request or pull the sequential slices by identifier at a fast rate over the network. *Id.* at 10:25-48, Fig. 6A, 7:39-8:5, 2:51-59, 11:9-22. Furthermore, Carmel teaches sending media data elements at a rate more rapid than the playback rate. *Id.* at 2:51-59 ("the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer").

2. Overview of Feig

69. Feig is directed toward a method for causing an Internet browser to "induce a non-streaming server to simulate a streaming server." EX1031, 1:9-10. Feig's method is "well suited for viewing video whose compressed data is transmitted from a server, without the need for storing all the data in advance at the local site, and without the need for the server to have streaming capabilities." *Id.* at 6:14-19.

70. Feig operates by partitioning the video to be streamed into sequenced segments. *Id.* at 6:17-19. Feig describes a data type it calls a "Uniform Resource Locator Sequence (URLS). A URLS consists of a Header, containing a header file, and an ordered list URLS(j), where j is an index ranging from 1 to n." *Id.* at 3:19-22. Feig shows a URLS in FIG. 1.

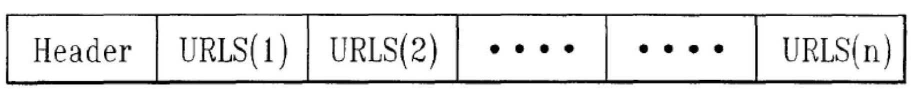


FIG. 1

Id. at FIG. 1.

71. The header file describes the data contained in the subsequent URLS, such as video data. A URLS(j) within the URLS contains a URL(j), which "is the URL for the jth entry of the URLS." *Id.* at 3:32-37. A URLS(j) also includes a time

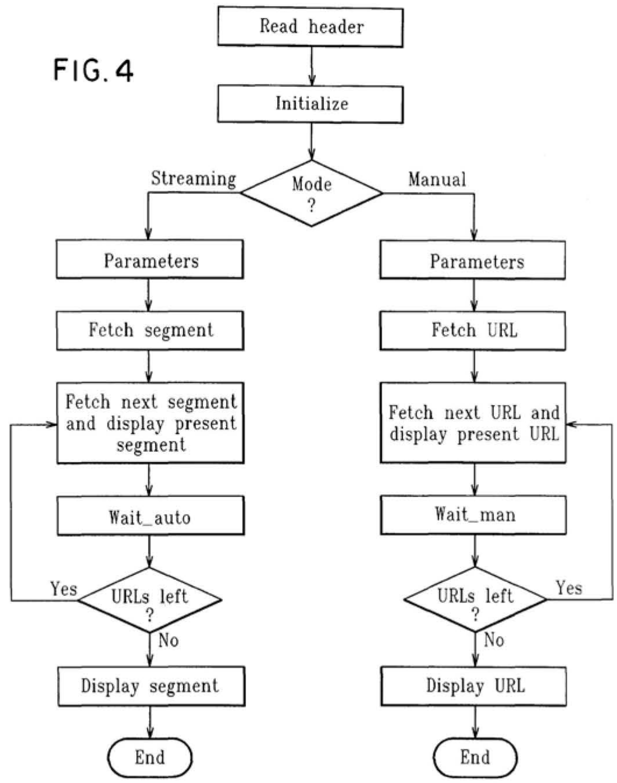
duration parameter T(j) and a value describing the size of the data B(j). *Id.* Feig shows a URLS(j) in FIG. 2.

FIG. 2

Id. at FIG. 2.

72. In operation, a user using a browser executing on a computer launches a hyperlink by clicking a link on a web page. This action causes the browser to recognize the URLS data type and execute a series of instructions to stream a video. *Id.* at 4:20-26, 4:27-5:54. Feig's FIG. 4 illustrates a set of steps and modules used to perform the streaming.

FIG. 4



Id. at FIG. 4. The "Parameters" module "partitions the URLs comprising the URLS into segments." *Id.* at 4:66-67. In other words, the Parameters module forms segments of video, with each segment including data referenced by one or more URLs. Feig illustrates three sample segments as follows:

segment 1—(URL(A_0), . . . , URL(A_1)),
segment 2—(URL(A_1+1), . . . , URL(A_2)),
segment 3—(URL(A_2+1), . . . , URL(A_3)), etc., such that

$$T(1)+ \ldots +T(A_1-1)<T \text{ and } T(1)+ \ldots +T(A_1)>T,$$

$$T(A_1+1)+ \ldots +T(A_2-1)<T \text{ and } T(A_1+1)+ \ldots +T(A_2)>T, \text{ etc.}$$

Id. at 5:1-8. "T" is a time parameter describing the length of a segment. *Id.* at 4:59-65.

73. Feig's method allocates two client buffers for storing the segments, BUFF_A and BUFF_B. *Id.* at 5:14-15. The "'Fetch segment' module fetches the first segment and stores it in BUFF_A. This involves making requests for URL(l), URL(2), and so on, until URL(Al)." *Id.* at 5:16-18. A POSITA would have understood and found obvious that these requests are HTTP GET requests for a file containing the requested data stored at the locations specified by the URLs.

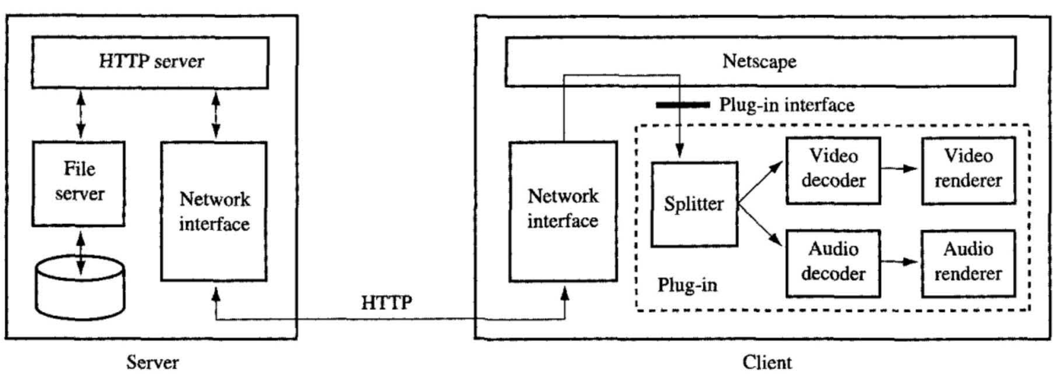
74. "As soon as all the data for the first segment arrives, the 'Fetch next segment-and display present-segment' module begins fetching the second segment and storing it in BUFF_B and simultaneously displaying the decoded contents from the first segment stored in BUFF_A." *Id.* at 5:16-23. Feig's method continues fetching and displaying the segments until "there are no more segments left to fetch." *Id.* at 5:37-40. The end result is that a "video will be displayed from beginning to end in a continuous manner as is typical when using a streaming server and a stream capable player." *Id.* at 5:47-50. Feig's method may operate in a streaming mode in which segments are fetched automatically and a manual mode in which it fetches a segment after a signal from the user. *Id.* at 5:24-36; 5:55-63.

3. Overview of Willebeek

75. Willebeek describes "a media streaming system, called Bamba, that delivers audio and video over low-bandwidth modem connections with the use of standard compression technologies." EX1006, 1. The Bamba system was "fully

implemented and deployed both internally at IBM and externally" by 1998. *Id.* In addition, Willebeek recognizes that it was conventional by 1998 to stream audio and video from "both stored and live sources on the Web." *Id.*

76. As shown in FIG. 1 of Willebeek below, the Bamba system includes a "client equipped with the appropriate Bamba software is able to communicate with the server and receive the Bamba audio/video file." *Id.* at 2. The "server is a standard HTTP Web server, which contains the stored Bamba audio and video files." *Id.* at 3. Moreover, "the underlying transport protocol used by HTTP is TCP/IP, which provides reliable end-to-end network connections." *Id.* at 3-4. If a Bamba clip is selected, the server transfers the clip "to the client (browser station) as fast as TCP/IP can move it, and the client begins decoding and displaying the Bamba file as soon as the first few bytes arrive." *Id.* at 4.



Id. at Fig. 1.

4. Motivation to combine Carmel, Feig, and Willebeek

77. In my opinion, a POSITA would have been motivated to combine Carmel, Feig, and Willebeek. All three references relate to streaming multimedia data over the Internet. *See, e.g.*, EX1005, 1:10-13 ("The present invention relates generally to network data communications, and specifically to real-time multimedia broadcasting over a network."); EX1031, 1:6-10 ("The present invention relates to Internet and Intranet Browsers, more specifically, to Browsers equipped with functionality to process a sequence of URL requests automatically. This capability allows the Browser to induce a non-streaming server to simulate a streaming server."); EX1006, 1 ("we present a media-streaming system, called Bamba, that delivers audio and video"). Additionally, all three references provide purported improvements to remedy drawbacks in prior art streaming solutions. *See, e.g.*, EX1005, 1:43-47 ("real-time broadcasting is normally possible only for hosts having a suitable, dedicated encoder and broadcast server and cannot be offered by Internet service providers (ISPs) to their general clientele"); EX1031, 2:10-13 ("Without explicit mechanisms to ensure isochronism, delivery rates of data to a browser are irregular, resulting in erratic playback quality at client machines."); EX1006, 1 ("A primary objective in developing Bamba is to stream audio and video across the Web through very-low-bit-rate connections."). A POSITA in possession of Carmel would have been aware of the difficulties of streaming multimedia data over the types of

links offered by ISPs to their general clientele and would have looked to solutions like those in Feig and Willebeek that address these difficulties, and incorporated aspects of those solutions into Carmel's approach. I provide additional motivations to combine the references below.

5. Independent claims 1, 5, and 9 are obvious over Carmel in view of Feig and Willebeek.

78. Independent claim 1 is a method claim that recites steps for distributing over the Internet, from a server system to one or more user systems, a pre-recorded multimedia program stored in digitally encoded form on computer-readable media. EX1001, 16:36-17:15. This is known in the prior art as a client-pull system because the client requests the data from the server. This contrasts with a "server-push" system, where the server initiates the streaming to the client. *Id.* For example, U.S. Patent No. 6,728,763 (EX1021) describes both a push and pull system for a streaming media server and describes a pull system as "starting pull process" when "web client 170 continues to request the media content." EX1021, 11:44-47, 12:13-14, Fig. 3B.

79. In my opinion, independent claim 5 is similar in scope to independent claim 1, though independent claim 5 recites a server system with at least one computer and a machine-readable, executable routine containing instructions to be executed by the at least one computer to perform steps similar in scope to the method of claim 1. EX1001, 17:22-18:5.

80. Furthermore, in my opinion, independent claim 9 is similar in scope to independent claim 1, though independent claim 9 recites a computer program product comprising a non-transitory computer readable storage medium having program instructions to be executed by at least one computer to perform steps similar in scope to the method of claim 1. *Id.* at 18:12-65.

81. To the extent the limitations recited in claims 5 and 9 differ in scope compared to the limitations recited in claim 1, for the same reasons for claim 1, and as shown below, it is my opinion that Carmel in view of Feig and Willebeek discloses claims 5 and 9.

(a) Preamble Limitations

[1.a] "A method for distributing over the Internet, from a server system to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media, the method comprising:"

[5.a] "A server system for distributing a pre-recorded audio or video program over the Internet to one or more user systems, the server system comprising:"

[9.a] "A computer program product for distributing over the Internet from a server system comprising at least one computer to one or more user systems, a pre-recorded audio or video program stored in digitally encoded form on computer-readable media, the computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith, the program instructions comprising:"

82. To the extent the preambles are limiting, in my opinion Carmel teaches the preambles.

83. As illustrated in Figure 2 below, Carmel discloses "a transmitting computer [34] [that] generates a data stream and broadcasts the data stream via a

network server [36] to a plurality of clients [30]." EX1005, 2:1-4, 3:24-27, 7:4-5, Fig. 2. Moreover, the broadcasting of the data stream from the "transmitting computer to [the] client computers" is "real-time," where the "data stream [has] a given data rate." *Id*. at Abstract.

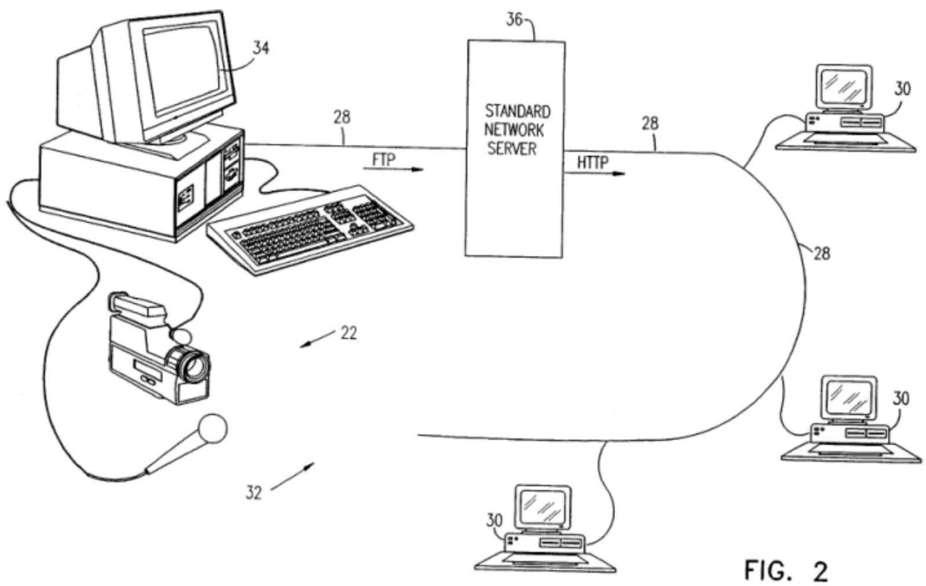


FIG. 2

84. Carmel's server system ("standard network server 36") includes "any suitable type of computer or computer system, for example, a Sun Microsystems UltraSPARC station or a Windows NT server, as are commonly used by Internet Service Providers (ISPs)." *Id*. at 6:40-43. A POSITA would have known that these well-known server systems comprise one or more computers.

85. Internet is used as a network in Carmel for distribution of the data stream from the transmitting computer to the server and/or the server to the client computers. For example, "[n]etwork 28 preferably comprises the Internet, although it may equally comprise a LAN, WAN, intranet or other computer network as is

known in the art." *Id.* at 6:36-38; *see also id.* at 6:28-31, 1:16-22. Further, "preferably using an Internet protocol" as the communications protocol, "clients download the data stream from the server." *Id*. at 2:11-12.

86. Carmel's "data stream comprises multimedia data" which is broadly defined to include an audio or video program. For example, Carmel's data stream "include[s] still images, video, graphics, animation or any combination thereof." *Id.* at 2:30-37. The multimedia data is generated by "[o]ne or more input devices 22 (for example, a video camera and/or microphone)," and "transmitted to a plurality of clients 30 via a network 28." *Id.* at 1:24-28.

87. Carmel's data stream that includes the audio or video program is prerecorded and stored on a disk or a tape, which is a form of computer-readable media. For example, "a prerecorded sequence . . . [to] be broadcast[ed]" (*Id.* at 6:59-60) is "stored on disk or tape." *Id.* at 9:64-66. Further, the stored program is in a digitally encoded form. For example, a data stream provided to the transmitting computer is "divid[ed] . . . into a sequence of slices" which are in turn "encoded in a corresponding sequence of files, each file having a respective index." *Id.* at Abstract.

88. Carmel teaches a computer program product comprising a non-transitory computer readable storage medium having program instructions embodied therewith. For example, Carmel provides "[a]n appendix [with] computer-readable files which exemplify aspects of the operation of system 32 (FIG. 2) and of the file

structures and methods described hereinabove." *Id.* at 13:55-59. These computer-readable files are "stored on disk in a common folder or directory." *Id.* at 14:8-9.

(b) Limitations reciting reading the pre-recorded program

[1.b] "reading, by at least one computer of the server system, the pre-recorded audio or video program from the computer-readable media;"

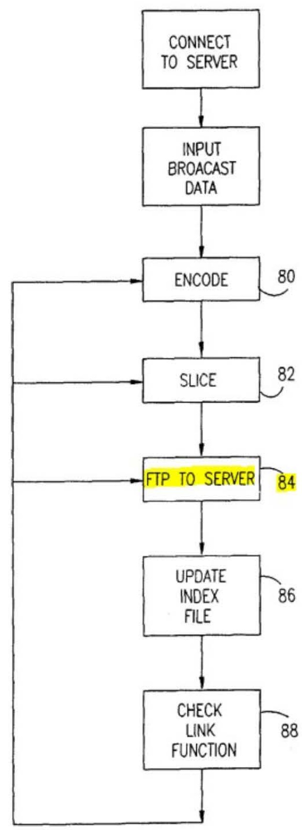
[5.b] "at least one computer having a connection to the Internet; a machine-readable, executable routine containing instructions to cause one of the at least one computers to read the pre-recorded audio or video program from computer-readable media,"

[9.b] "instructions executable to cause one of the at least one computers to read the pre-recorded audio or video program from the computer-readable media,"

89. In my opinion, Carmel teaches these limitations.

90. Carmel teaches the part of claim [5.b] that recites "at least one computer having a connection to the Internet," as previously discussed in ¶¶ 84-85. In addition, Carmel teaches that the broadcast data "stored on disk or tape" is first "input[ted] to the [transmitting] computer [34]," after which the "[c]omputer 34 conveys file 40 [i.e., broadcast data] to server 36 . . . using FTP, at step 84." *Id.* at 9:64-10:3. This is shown in FIG. 5 below (step 84 highlighted).

FIG. 5



91. This is also disclosed by various other parts of Carmel. For example, "[t]he transmitting computer uploads the sequence of slices to the server substantially in real time, preferably using an Internet protocol." *Id.* at 2:7-10. Additionally, "[c]omputer 34 continues to upload files 42, 44, 46, etc., until data stream 40 is finished or terminated by a user of computer 34. All of the files in the data stream may be saved on server 36 for any desired period of time." *Id.* at 7:50-55; *see also id.* at 10:64-11:8, 8:21-29, 14:33-35.

92. Further, in my opinion, a POSITA would have understood that the server system would necessarily "read" the data files 40 that are uploaded to the server via an FTP or other protocols from computer 34. This is because when client 30 later requests specific slices of the recorded broadcast stream, the server must read the requested slices from the storage medium in order to deliver them to the client 30. *See id.* at 10:36-42.

(c) Limitations reciting supplying media data elements

[1.c] "supplying, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate;"

[5.c] "to supply, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate,"

[9.c] "to supply, at the server system, media data elements representing the program, each media data element comprising a digitally encoded portion of the program and having a playback rate,"

93. In my opinion, Carmel teaches these limitations.

94. Carmel teaches that the prerecorded audio or video program comprises a series of data slices (i.e., media data elements). For example, "[d]ata stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc." *Id.* at 7:22-25.

95. These data slices are encoded or compressed using various known standards for compression. For example, "[t]he data stream is divided into a

sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed." *Id.* at 2:1-15.

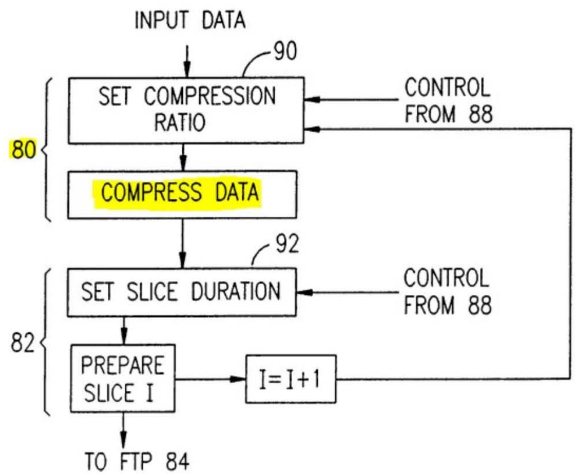
96. It was well known in the art that compression is a form of digital encoding. Numerous other patents and publications describe the compression of video as encoding the video. For example: U.S. Patent No. 6,668,088, titled "Digital signal compression encoding with improved quantisation" "relates to the compression of digital video, audio, or other signals." EX1016, 1:7-8. U.S. Patent No. 5,533,138, titled "Image compression encoding and decoding method and apparatus therefor" describes how "[r]ecently, video information compressing technology has been used in processing a video signal as digital data in digital video apparatus such as digital VCR, HDTV, digital video camera and video phone." EX1017, 1:17-20.

97. U.S. Patent No. 5,469,212, titled "Apparatus for compression-encoding and decoding video signals" describes "[a]n apparatus for compression-encoding and decoding digital video signals, capable of additionally transmitting a differential signal indicative of a difference between an original video signal and an encoded video signal obtained by encoding the original video signal by an existing video compression system, so as to efficiently cope with a varied bandwidth of a recording medium or a channel. The apparatus comprises an encoding device for encoding an original digital video in a sampling manner . . . ." EX1018, Abstract. U.S. Patent

No. 6,314,137, titled "Video data compression system, video recording/playback system, and video data compression encoding method" states that "[a]n MPEG encoding circuit performs inter-frame predictive-encoding for each specified number of frames." EX1019, Abstract. A POSITA would have known that MPEG refers to a compression and encoding algorithm for video.

98. Carmel teaches various known standards for compression. For example, "[i]n encoding data stream 40, computer 34 preferably compresses the data using any suitable compression method known in the art . . . if data stream 40 comprises audio data, GSM 6.10 standard encoding may be used, . . . [a]lternatively or additionally . . . H.263 standard compression . . . [or] MPEG data compression, may similarly be used." EX1005, 11:26-38; *see also id.* at 3:63-4:2; 1:29-33. FIG. 7 below shows, at step 80, encoding or compressing of the data slices.

FIG. 7



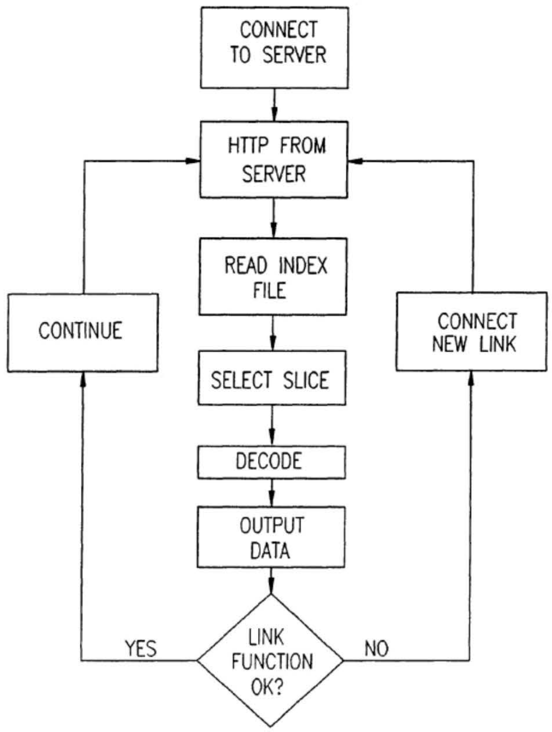
99. Carmel teaches that the transmitting computer uploads or supplies media data elements to the server. For example, "[t]he files are uploaded to server 36, such that while any given slice (other than first slice 42) is being created, one or more preceding slices are in the process of being uploaded." EX1005, 7:22-34. Additionally, "a transmitting computer generates a data stream and broadcasts the data stream via a network server to a plurality of clients." *Id.* at 2:1-4.

100. The encoded or compressed data slices or media data elements are then stored at the server. For example, "[c]omputer 34 continues to upload files 42, 44, 46, etc., until data stream 40 is finished or terminated by a user of computer 34. All of the files in the data stream may be saved on server 36 for any desired period of time . . . ." *Id.* at 7:50-55; *see also id.* at 10:64-11:8, 8:21-29, 14:33-35.

101. Furthermore, Carmel teaches that each of the media data elements has a playback rate. Carmel's client will "play back the broadcast" received from the server. *Id.* at 7:4-17. Thus, it would have been obvious to a POSITA that Carmel's encoded slices have a playback rate. In addition, Carmel's client "reconstructs and outputs the multimedia data for the appreciation of a user" using time stamps "to synchronize the data." The result is that "the multimedia sequence is played back just as it was input at computer 34." *Id.* at 10:48-54. A POSITA would have understood this to be further evidence that slices of the data stream have a playback rate. Otherwise, the client would not be able to play the multimedia data as it was previously inputted.

102. Carmel teaches that the server supplies the media data elements to the clients. This is evident in Figure 6A below, which is a flow chart including a step of "HTTP from Server." *Id.* at Fig. 6. The server supplies the slices to the clients by making the slices available to be streamed via a HTTP link. *See id.* In my opinion, a POSITA would have understood that "communications received by the clients" include media data elements supplied by Carmel's server. *See id.* at 10:27-35. The server would supply the media data elements to the clients in order for the clients to play back "the multimedia data for the appreciation of a user." *See id.* at 10:48-50.

FIG. 6A



(d) Limitations reciting serially identifying the media data elements

[1.d] "serially identifying the media data elements, said serial identification indicating a time sequence of the media data elements;"

[5.d] "to serially identify the media data elements, said serial identification indicating a time sequence of the media data elements, and"

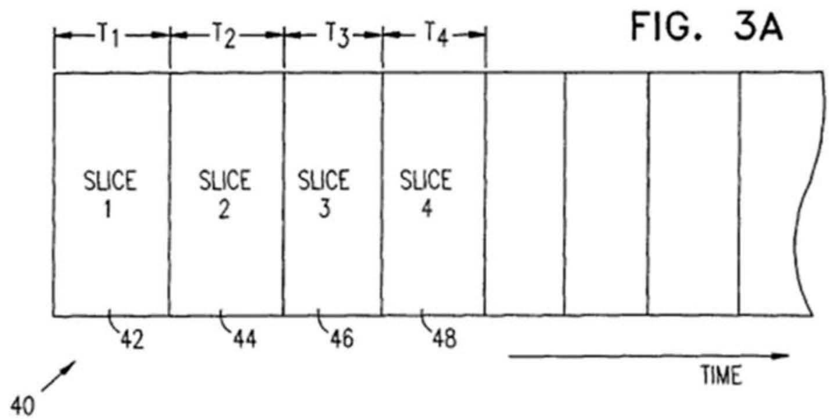
[9.d] "to serially identify the media data elements, said serial identification indicating a time sequence of the media data elements, and"

103. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

104. Carmel's data stream is divided into a sequence of segments or slices, indicating a time sequence. For example, "[t]he data stream is divided into a

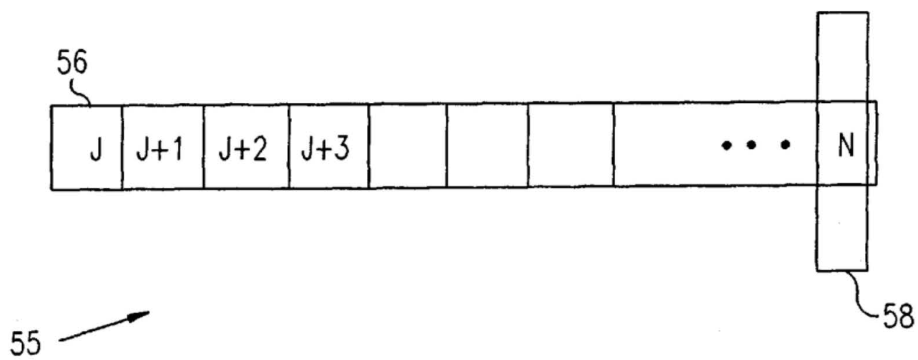
sequence of segments or slices of the data, preferably time slices." *Id.* at 2:1-6. Additionally, the "[d]ata stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc." *Id.* at 7:22-25.

105. Furthermore, Carmel teaches that "[e]ach slice [of the data stream] is preferably assigned a respective slice index," thereby providing a serial identification indicating a time sequence of the slices. *Id.* at 2:6-7, 7:27-32. Carmel's FIG. 3A below is a block diagram showing data slices 42, 44, 46, etc. corresponding to successive time intervals T1, T2, T3, etc.



106. Carmel's FIG. 3C below shows a time sequence of slices, specifically "a user interface graphic 'slider' 55, available to users of computers 30 . . . [where] [t]he symbols J, J+1, J+2, . . . N in the figure are 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." *Id.* at 8:18-31.

FIG. 3C



107. Carmel teaches that the most recent slice of the sequence of slices can be identified using an indicator, shown by element 58 above. For example, "[w]hen one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown in FIG. 3C. . . . If the user wishes to begin the download at an earlier point, he may move indicator 58 to the left along bar 56 to that point . . . ." *Id.* at 8:32-41.

108. To the extent not disclosed in Carmel, Feig discloses these limitations. Feig discloses the browser making a series of requests to a server for "URL(1), URL(2), and so on." EX1031, 5:16-18. A POSITA would have found it obvious that Feig's server stores the media data elements at respective locations specified by the URLs, such as in files having names incorporating serial identifiers, because Feig's illustrative URLs use serial identifiers and the media data elements being requested by the URLs are sequential. *See id.* at 2:51-57 ("The URLS consists of . . . a sequence of URLS. The URLS may be a sequence of pre-existing files, or a new

sequence of URL's may be created by partitioning a data resource such as video into contiguous time segments . . . ."). A POSITA would have found it natural and convenient to use serial identifiers for sequential data elements.

109. A POSITA would thus have found it obvious that Feig's server serially identifies the media data elements, and that the serial identification indicates a time sequence of the media data elements. A POSITA would have been motivated to combine the teachings of Feig related to using serial identifiers for the media data elements into the disclosure of Carmel because it would simplify the operation of the combined system by using serial identifiers for sequential media data elements and allow the client to precisely control and select which segments it receives from the server, which supports Carmel's goal of allowing a user to "decide and indicate at which slice of data stream 40 to begin downloading." EX1005, 10:42-45. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way.

(e) Limitations reciting storing the media data elements

[1.e] "storing the media data elements in a data structure under the control of the server system;"

[5.e] "to store the media data elements in a data structure under the control of the server system;"

[9.e] "to store the media data elements in a data structure under the control of the server system;"

110. In my opinion, Carmel teaches these limitations.

111. Carmel teaches that the media data elements are stored in the server system and are thus under the server's control. For example, "[a]ll of the files in the data stream may be saved on server 36 for any desired period of time, as long as the server has sufficient free memory that is accessible to computer 34." *Id.* at 7:52-55.

112. Furthermore, the media data elements stored on the server are in a sequence in a data structure. For example, "[t]he symbols J, J+1, J+2, . . . N in the figure are 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. J may indicate the first slice in the sequence, if all of the files are stored on server 36, or it may be the earliest file not yet erased." *Id.* at 8:23-29; *see also id.* at 6:40-43, 9:62-10:5.

113. To the extent Carmel's slices were not already stored in a data structure, a POSITA would have been motivated to modify Carmel to maintain the slices in a linked list, a set of numbered files, or other form of data structure to ensure the server could organize the slices and more readily respond to requests from clients for numbered slices. A POSITA would also have had a reasonable expectation of success given a data structure is well-known. A POSITA would also have been familiar with utilizing a mixture of data structures, such as creating a linked list of files, or an array of files, which would simplify accessing the next file segment or jumping to a particular file segment.

(f) **Limitations reciting receiving requests at the server system**

[1.f] "receiving requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of the requested one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and"

[5.f] "a machine-readable, executable routine containing instructions to cause one of the at least one computers to receive requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of the one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and"

[9.f] "instructions executable to cause one of the at least one computers to receive requests at the server system via one or more data connections over the Internet, for one or more of the media data elements stored in the data structure, each received request specifying one or more serial identifiers of the one or more media data elements, each received request originating from a requesting user system of the one or more user systems; and"

114. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

115. Carmel's server system receives over the Internet requests from users' systems specifying one more serial identifiers (slice IDs) for one or more media data elements (slices), e.g., "42, 44, 46, etc." that are stored in the server's data structure. For example, "[e]ach client 30 connects to server 36, optionally using multiple HTTP links . . . [where] client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation." EX1005, 10:36-42; *see also id.* at 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

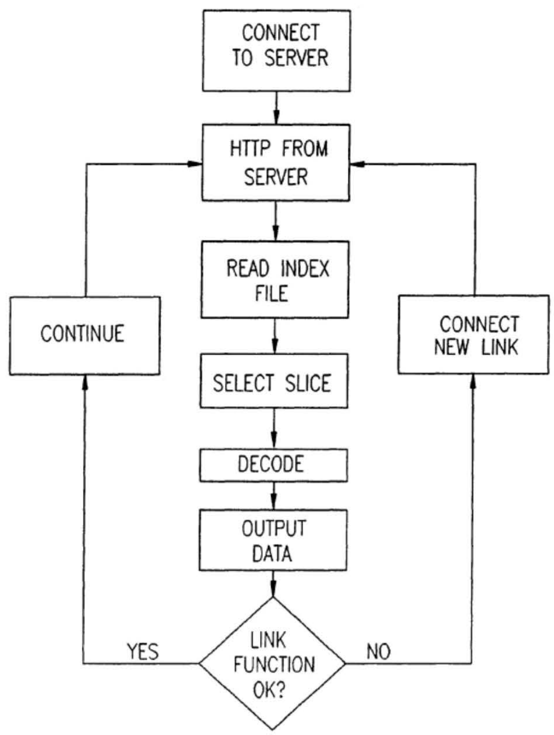
(HTTP) . . . ."), 7:4-17 ("other Internet protocols may also be used"); *supra* Section VIII.A.3(a).

116. The requests received by Carmel's server system are based on the user system reading the index file with the numbered slices and requesting such numbered slices. For example:

> The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc.

EX1005, 10:42-48. A POSITA would have understood Carmel's disclosure of "[r]esponsive to a user input, client 30 selects an appropriate starting slice and begins to download" to mean that the client makes HTTP requests for the slices. *See id.* Indeed, this is evident in Figure 6A below, which "is a flow chart illustrating the operation of clients 30 in downloading and playing back data stream." *Id.* at 10:24-25.

FIG. 6A



117. Carmel teaches that the user systems' requests are for those slices stored in the server's data structure. For example:

> Slider 55, which is preferably displayed on the screens of computers 30, includes a bar 56 and a movable indicator 58. The symbols J, J+1, J+2, . . . N in the figure are 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. J may indicate the first slice in the sequence, if all of the files are stored on server 36, or it may be the earliest file not yet erased. . . .
>
> When one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown

> in FIG. 3C. This is the point at which the download will begin, unless the user of the computer chooses otherwise. If the user wishes to begin the download at an earlier point, he may move indicator 58 to the left along bar 56 to that point . . . .

Id. at 8:9-41; *see also id.* at 15:3-7, 8:56-62, 10:64-11:8, 15:13-17, Figs. 3C-3D.

118. To the extent not disclosed in Carmel, Feig discloses these limitations. In Feig, a user initiates streaming by clicking on a link of a web page displayed by a browser of a client computer. EX1031, 4:20-26. Subsequently, a "Fetch segment" module executing on the client computer makes a series of requests to a server for "URL(1), URL(2), and so on, until URL(A1)" is fetched. *Id.* at 5:16-18. These requests are for media data elements. *See id.* at 3:23-26 (the data type of a URLS includes video); 4:33-35 (the data in the URLs comprise segments from a video stream); 4:49-52 ("the URLs comprising contiguous segments from a video sequence").

119. A POSITA would have recognized and found obvious from Feig's disclosure that these requests originate from a requesting user system and are received by a server system via one or more data connections. *See id.* at 6:14-18 ("The present invention is well suited for viewing video whose compressed data is transmitted from a server, without the need for storing all the data in advance at the local site, and without the need for the server to have streaming capabilities."); 6:25-26 ("The bandwidth from the server to the browser is 3 Mbps."); 1:9-10 ("This

capability allows the Browser to induce a non-streaming server to simulate a streaming server."). A POSITA would also have found it obvious that the requests specify one or more serial identifiers of the one or more media data elements for the reasons discussed above in [1.d], [5.d], and [9.d].

120. A POSITA would have been motivated to combine the teachings of Feig related to reciting receiving requests at the server system with Carmel because it would allow the client to precisely control and select which segments it receives from the server, which supports Carmel's goal of allowing a user to "decide and indicate at which slice of data stream 40 to begin downloading." EX1005, 10:42-45. The combination would also expand the types of servers that could be used within Carmel's system by inducing "a non-streaming server to simulate a streaming server." EX1031, 1:9-10. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way. The combination is also a simple substitution of one known element for another to obtain predictable results. Both Carmel and Feig use HTTP requests from the client to the server to request media data elements, and a POSITA would find it simple and predictable to modify Carmel to use Feig's disclosure.

(g) Limitations reciting sending media data elements to the requesting user systems

[1.g] "responsive to the requests, sending, by the server system, the one or more media data elements having the one or more specified serial identifiers, to the requesting user systems corresponding to the requests; wherein"

[5.g] "a machine-readable, executable routine containing instructions to cause one of the at least one computers to send, responsive to the requests, the one or more media data elements having the one or more specified serial identifiers, to the requesting user systems corresponding to the requests; wherein"

[9.g] "instructions executable to cause one of the at least one computers to send, responsive to the requests, the one or more media data elements having the one or more specified serial identifiers to the requesting user systems corresponding to the requests; wherein"

121. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

122. Carmel teaches that responsive to the requests from user systems (discussed above in [1.f], [5.f], and [9.f]), the server system sends the media data elements with the specified serial identifiers to the corresponding user systems. For example, "[r]esponsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc." *Id.* at 10:44-47. A POSITA would have understood that this downloading occurs by the server sending the slices to the clients in response to the requests. "In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user." *Id.* at 10:47-49; *see id.* at Fig. 6A, 6:3-6 (describing Fig. 6A as "a method of downloading broadcast data from a server to a client"); *id.* at FIG. 3C, 8:1-7, 8:32-34 ("When one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown in FIG. 3C."); *see also id.* at 2:11-15, 15:13-17.

123. To the extent not disclosed in Carmel, Feig discloses these limitations. Feig teaches that responsive to the requests from user systems (discussed above in [1.f], [5.f], and [9.f]), the server system sends the media data elements with the specified serial identifiers to the corresponding user systems. *See, e.g.,* EX1031, 6:14-18 ("The present invention is well suited for viewing video whose compressed data is transmitted from a server, without the need for storing all the data in advance at the local site, and without the need for the server to have streaming capabilities."); 6:25-26 ("The bandwidth from the server to the browser is 3 Mbps."); 1:9-10 ("This capability allows the Browser to induce a non-streaming server to simulate a streaming server.").

124. A POSITA would have been motivated to combine the teachings of Feig related to sending media data elements to the requesting user systems with Carmel because it would allow the client to precisely control and select which segments it receives from the server, which supports Carmel's goal of allowing a user to "decide and indicate at which slice of data stream 40 to begin downloading." EX1005, 10:42-45. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way.

(h) Limitations reciting that the data connection has a data rate more rapid than the playback rate

[1.h] "the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;

[5.h] "the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;"

[9.h] "the data connection between the server system and each requesting user systems has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection;"

125. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

126. Petitioner's counsel has informed me that in a related case, a similar data "rate" limitation in claim 10 of the '141 patent was previously construed by the Federal Circuit. *See* EX1015, 14:25-28 ("server to send media data elements to the user system responsive to said requests, at a rate more rapid than the rate at which said streaming media is played back by a user"). Specifically, I understand that the Federal Circuit construed this data rate as "the rate at which each requested data element is transmitted from the server to the user computer." *WAG Acquisition v. WebPower, Inc.*, 781 Fed. App'x 1007, 1012 (Fed. Cir. 2019). I see that the PTAB adopted the Federal Circuit's construction of this data rate term. Final Written Decision on Remand, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 28 (July 16, 2020) at 10. I also understand that the PTAB explained that the Federal Circuit "distinguished its construction as excluding the 'overall rate,'" that

"might be achieved with multiple links over which data elements are sent to the recited user system."[4] *Id.*

127. I understand that the PTAB, in view the Federal Circuit's construction of this data rate term, found that Carmel anticipates claim 10 of the related '141 patent. Final Written Decision on Remand, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 28 (July 16, 2020) at 25-26. In particular, I see that the PTAB relied on Carmel's teaching that "[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." EX1005, 2:51-59. I understand that the PTAB determined this teaching from Carmel discloses the '141 patent claim 10 limitation, "instructions to cause the server to send media data elements to the user system responsive to said requests, <u>at a rate more rapid than the rate at which said streaming media is played back by a user</u>." Final Written Decision on Remand, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 28 (July 16, 2020) at 6, 18-21. I agree with this determination by the PTAB.

128. I understand that the PTAB, in reaching its determination, cited admissions by the Patent Owner regarding Carmel. For example, according to the

---

[4] In view of this distinction by the Federal Circuit, Carmel can achieve the data rate "using a single HTTP link" between a client and the server. EX1005, 10:67-11:2.

Patent Owner's expert, Carmel at 2:51-59 (quoted above) means that "Carmel adjusts the slices so that they are transmitted at about the playback rate." Final Written Decision on Remand, *WebPower v. WAG Acquisition, LLC*, IPR2016-01238, Paper No. 28 (July 16, 2020) at 18. Moreover, the Patent Owner's expert conceded that "[i]f it is transmitted slightly faster than playback rate and then slightly lower, slightly higher, slightly lower, which is what 'about playback rate' means." *Id.* at 19. The Patent Owner did "not specifically dispute that Carmel teaches transmission faster than the playback rate." *Id.* Thus, Carmel's teaching that "the data rate should be generally equal to or faster than the rate at which the data are generated at the transmitting computer" (EX1005, 2:51-59) discloses the limitation, "the data connection between the server system and each requesting user system has a data rate more rapid than the playback rate of the one or more media data elements sent via that connection."

129. To the extent this limitation is construed as requiring "a data rate more rapid than the playback rate" for *all* of the media data elements sent, a POSITA would have found it obvious that Carmel teaches this limitation. As I explain above, Carmel discloses that the data rate is more rapid than the playback rate for at least some of the media data elements sent. Based on this disclosure, it is my opinion that a POSITA would have found it obvious that Carmel can send *all* of the media data elements (not only at least some of the media data elements) at a data rate more rapid

than the playback rate. A POSITA would have been motivated to modify Carmel to do so "in order to ensure that the transmission or reception is 'keeping up' with the input of the data" to the client. EX1005, 7:36-40. Furthermore, "[i]n the event that a lag is detected," Carmel will "increase the data transmission or reception rate." *Id.* at 7:40-42. It is desirable for Carmel to send media data elements at the playback rate or <u>faster to ensure that the clients' buffers</u> do not become depleted and cause interruptions during playback.

130. To the extent not disclosed in Carmel, Feig discloses that the data connection has a data rate more rapid than the playback rate. Feig provides an example in which "[a] 60 minute MPEG encoded video stream is partitioned into one minute segments" of 90 megabits each and "[t]he bandwidth from the server to the browser is 3 Mbps." EX1031, 6:21-27. Here, "[t]he time to deliver a one minute segment of video over the channel is 30 seconds." *Id.* at 6:29-31. Thus, Feig contemplates that a portion of video that requires one minute to be played is delivered over a data connection in 30 seconds. A POSITA would have recognized and found obvious from this example that Feig discloses a data connection with a data rate more rapid than the playback rate.

131. A POSITA would have found it obvious in view of Feig to modify Carmel to ensure that the data transmission rate is always faster than the playback rate of the requested slice. This modification ensures that there will be no

interruptions in playback due to data connection issues because it will "ensure that the transmission or reception is 'keeping up' with the input of the data" to the client. EX1005, 7:36-40. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way.

132. Furthermore, in my opinion, it was well-known in the art that the data rate or bandwidth capacity can be more rapid than the encoding or playback rate. I describe this below using two examples.

133. U.S. Patent No. 6,848,004 (EX1020) ("Chang") describes that "there are four bandwidth-sensitive video contents encoded at 20 kbps, 50 kbps, 80 kbps, and 140 kbps, respectively. These video clips may be contained in one-single HotMedia file or multiple HotMedia files." EX1020, 10:15-19. Chang further describes that it will estimate the bandwidth between the server (e.g., delivery station) and user systems. For example, "[t]he predicted available bandwidth Bi* can be used to make a decision which content should reach the client." *Id.* at 10:19-21. In addition, Chang provides "prediction of bandwidth for adaptive content delivery of rich media according to available user bandwidth. The rich media file . . . is transmitted by the delivery station to the client station . . . ." *Id.* at Abstract.

134. Chang also teaches that this data connection rate will be more rapid than the encoding rate. For example, "[t]he following rules will be used in the decision maker. If B3* less than 28 kbps, then the client receives 20 kbps video content. Or

else, if B3* less than 56 kbps, then the client receives 50 kbps video content. Else if B3* less than 128 kbps, then the client receives 80 kbps video content. Else, the client receives 140 kbps video content." *Id.* at 10:45-51. Therefore, in Chang, if the bandwidth of the data connection is 120 kbps, then the 80 kbps encoded file will be selected and transmitted to the client, i.e., the data connection is more rapid than the playback rate.

135. As another example, WO1997044942 (EX1014) ("Kliger") describes that "[t]he communication channel 11 is a typical telephone line or other transmission/communication cable handling a 28,800 baud data rate or the like." EX1014, 7:2-4. Kliger further describes that it provides video objects with different playback rate files (compression or bandwidth). *Id.* at 13:9-15 ("'object specific compression' is provided by the server, such that on an object by object basis, the object data stream 300 may be compressed one object at a time depending upon client criteria"), 9:32-10:4 ("[A]n object class may consist of various versions of a particular graphic object. . . . Quality may also have other meaning such as bandwidth . . . ."), 12:33-13:1 ("The object classes may also be defined as bandwith [sic] selectable objects."), 9:24-9:27 ("objects may include text, graphics, audio, video . . . [and] objects may be complete data files or only portions of such files").

136. Kliger teaches that the data connection rate between the server and clients is more rapid than the playback rate of the various versions because it

explicitly describes that the server transmits "ahead of consumption" or "sufficiently fast to prevent . . . lagging behind." *Id.* at 15:10-13 ("the server 17 transmission of data is ahead of the consumption and thus no pausing during the transmission of the object data stream 300 is warranted"), 18 at claim 5 ("rate at which the requesting processor receives requested objects transmitted from the server processor is sufficiently fast to prevent the data stream from lagging behind the requesting processor use of requested objects"), 13:33-14:2 ("client 13a continually monitors the real data throughput versus data consumption. The client 13a wants the delivery-to-consumption ratio to be greater than one so that the throughput (supply) is keeping up with the consumption (demand).").

137. This teaching is reinforced by Kliger describing that the server transmits a specific bandwidth or playback rate version based on the data connection or bandwidth capacity and thus the data connection rate would be more rapid than the playback rate. For example, "for a given requested object 04, a specific version 04a, 04b . . . 04x may be selected for transfer from the server 17 to the client 13a as a function of available bandwidth. This function is termed 'bandwidth scalability' of the object stream." *Id.* at 13:1-8. Thus, Kliger teaches this limitation in multiple ways given the transmission is faster than consumption and separately given the server transmits a certain bandwidth file based on data connection rate (or bandwidth capacity).

(i) Limitations reciting that "each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow[s]"

[1.i] "each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow;"

[5.i] "each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allow;"

[9.i] "each sending is at a transmission rate as fast as the data connection between the server system and each requesting user system allows;"

138. In my opinion, Carmel and/or Carmel in combination with one or both of Feig and Willebeek teaches these limitations.

139. Carmel teaches using HTTP between the server system and the requesting clients. For example, "client computers download the encode sequence using an Internet download protocol, most preferably HTTP . . . ." EX1005, 5:25-28; *see also id.* at 7:4-9, 6:28-31. A POSITA would have understood that HTTP was used in conjunction with TCP. *See* EX1006, 3-4 (explaining that HTTP uses TCP/IP as "the underlying transport protocol"). A POSITA would thus have understood that Carmel's disclosure of HTTP would disclose this claim limitation because Carmel would "simply hand[] the data to" TCP given the "technical underpinnings of TCP . . . is well-established." EX1026, 9; EX1025, 15. Furthermore, TCP was a well-known protocol for reliable transmission of data described by 1981 and published as a "Request for Comments" ("RFC") by the Internet Engineering Task Force ("IETF"). *See* EX1029 (IETF RFC793, the standards document for TCP). TCP

incorporates a method of congestion control, which allows "the capacity of the internet [to] be reached" thus maximizing the bandwidth between the TCP endpoints. EX1030, 54-60.

140. To the extent that Carmel alone does not teach these limitations, Carmel in combination with Feig and Willebeek teaches them. Feig discloses an example in which "[t]he bandwidth from the server to the browser is 3 Mbps." EX1031, 6:25-26. This bandwidth means that a one minute segment of video encoded with 90 megabits is delivered "over the channel [in] 30 seconds." *Id.* at 6:29-31. A POSITA would have recognized that Feig is here describing a sending at a transmission rate as fast as the data connection between the server system and the requesting user system allows.

141. Willebeek teaches sending a video or audio clip from a server to a client "as fast as TCP/IP can move it." EX1006, 4. A POSITA would have recognized that when Willebeek says "as fast as TCP/IP can move it," Willebeek is disclosing that the media data elements are being sent from the server to the client "at a rate as fast as the data connection" can move it, i.e., as fast as is allowed by the data connection.

142. Moreover, Willebeek teaches that a "base requirement of the Bamba streaming system is to function within the WWW standard HTTP-based client-server architecture." *Id.* at 3. Indeed, "Bamba was designed to stream clips from standard

HTTP Web servers" and uses "communication mechanisms provided by the HTTP protocol." *Id.*

143. Carmel similarly teaches sending video or audio data using HTTP because "clients download the data stream from the server, preferably using an Internet protocol, as well, most preferably the Hypertext Transfer Protocol (HTTP)." EX1005, 2:11-15; *see also id.* at 3:63-66, 7:5-7, 10:38-42 ("Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded . . . ."). In addition, "HTTP is supported by substantially all modern Web browsers." *Id.* at 7:12-17. Feig likewise teaches or suggests sending video or audio data using HTTP because a preferred embodiment of Feig's invention "extends a standard HTML browser." EX1031, 3:17-19.

144. A POSITA would have found it obvious to incorporate Feig's disclosure of sending at a transmission rate as fast as the data connection between the server system and the requesting user system allows and/or Willebeek's disclosure of using HTTP to transfer data to the client "as fast as TCP/IP can move it" within Carmel's method. As I describe above, Carmel, Feig, and Willebeek all relate to streaming multimedia data and the references also disclose use of HTTP, which is based on TCP/IP, for such streaming. In my opinion, a POSITA would have known that HTTP runs on TCP. Indeed, Willebeek explains that HTTP uses TCP/IP as "the underlying transport protocol." EX1006, 3-4.

145. A POSITA would have been motivated to combine the teachings of Feig and/or Willebeek related to the use of HTTP and/or TCP with Carmel to quickly move data slices from the server to the clients to support playback with a minimum of dropouts. *See, e.g.*, *id.* at 2 ("the file is played once uninterrupted playback can be ensured"); EX1005, 8:1-7 (describing that the client computer 30 preferably downloads "the data stream . . . with only a minimal lag"). Such a combination is no more than using a known technique disclosed in Feig and/or Willebeek to improve a similar device or method in Carmel in the same way. Thus, Carmel could seamlessly implement the teachings of Feig and/or Willebeek.

(j) Limitations reciting that the elements are sent without depending on the server system to maintain a record of the last element sent

[1.j] "the one or more media data element sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems;"

[5.j] "the one or more media data elements send are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems;"

[9.j] "the one or more media data elements sent are selected without depending on the server system maintaining a record of the last media data element sent to the requesting user systems;"

146. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

147. In Carmel, each client requests numbered slices from the server and even maintains a graphical slider to track each slice, including the last media data element, without regard to the server. For example:

> Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation, but as in the case of transmitting computer 34, a greater number of links may similarly be opened. The client first reads index file 50 (FIG. 3B), and graphic 56 (FIG. 3C) is displayed by the client, so that a user can decide and indicate at which slice of data stream 40 to begin downloading. Responsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user.

Id. at 10:37-50; *see also id.* at FIG. 3C, 8:32-41 ("When one of computers 30 reads index file 50 and begins to download stream 40, indicator 58 preferably marks the most recent slice, as shown in FIG. 3C.").

148. Furthermore, the client computer 30, based on a user input, identifies a point in the data stream to begin receiving the data stream. For example, "[w]hen one of computers 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." *Id.* at 8:1-4. Additionally, "a user of one of computers 30 may choose to begin downloading data stream 40 from an earlier point in time than that indicated by ID 52." *Id.* at 8:7-9.

149. Because this point in the data stream is identified based on a user input received at the client computer, Carmel teaches a client-side control for requesting sequential media data elements (*id.* at 8:18-31), which does not depend on the server system maintaining a record of the last media data element sent to the requesting user systems. Since the slider allows for random access into the file stream, and playback occurs by requesting sequential segments, it is unnecessary for the server to track the segment state. Furthermore, the server tracking the segment state could conflict and/or cause race conditions with new input selections made by the user.

150. To the extent not disclosed in Carmel, Feig discloses these limitations. In Feig, a user initiates streaming by clicking on a link of a web page displayed by a browser of a client computer. EX1031, 4:20-26. Subsequently, a "Fetch segment" module executing on the client computer makes a series of requests to a server for "URL(1), URL(2), and so on, until URL(A1)" is fetched. *Id.* at 5:16-18. Each request is for video data forming a media data element. *See id.* at 3:23-26 (the data type of a URLS includes video); 4:33-35 (the data in the URLs comprise segments from a video stream); 4:49-52 ("the URLs comprising contiguous segments from a video sequence"). These media data elements are sequential. *See id.* at 5:44-47 ("if the streaming criteria are met, then the entire content of all the URLs comprising the URLS will be displayed in proper timing order as prescribed by the content creator");

6:17-19 ("The video data which will be streamed is partitioned into sequenced segments.").

151. Feig thus teaches a client-side control for requesting sequential media data elements, which does not depend on the server system maintaining a record of the last media data element sent to the requesting user systems. It is unnecessary for the server system to track the state of media data elements since the elements are requested individually by the client browser and HTTP is a stateless protocol.

152. A POSITA would have found it obvious to incorporate Feig's disclosure of the client making requests for media data elements within Carmel's method. As I describe above, both Carmel and Feig relate to the streaming multimedia data and both references disclose use of client-side requests for media data elements referenced by URLs. *See* EX1005, 10:38-40 ("Typically, client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation . . . ."); EX3031, 5:16-18 (fetching a segment "involves making requests for URL(l), URL(2), and so on, until URL(Al)").

153. A POSITA would have been motivated to combine the teachings of Feig related to sending media data elements from the server to the client without depending on the server system maintaining a record of the last media data element sent with Carmel because it would allow the client to precisely control and select which segments it receives from the server, which supports Carmel's goal of

allowing a user to "decide and indicate at which slice of data stream 40 to begin downloading." EX1005, 10:42-45. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way.

(k) Limitations reciting that all of the elements are sent in response to the requests

[1.k] "all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests; and"

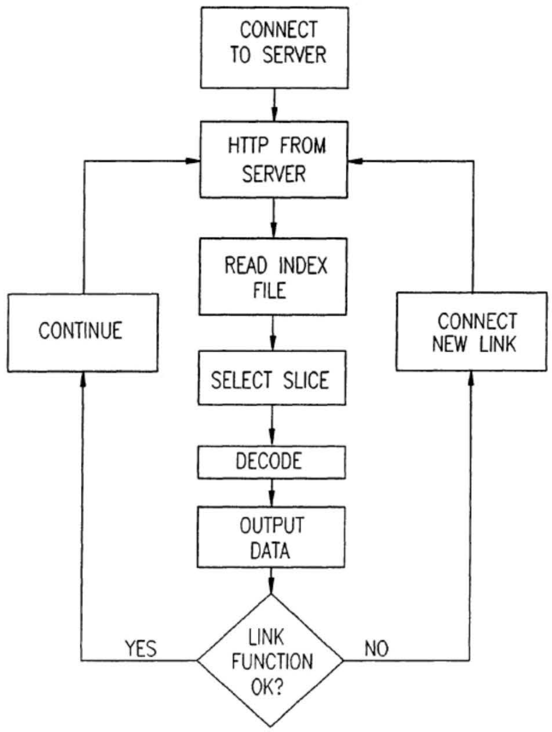
[5.k] "all of the media data elements that are sent by the server to the one or more user systems are sent in response to the requests; and"

[9.k] "all of the media data elements that are sent by the server to the one or more user systems are sent in response to the requests; and"

154. In my opinion, Carmel and/or Carmel in combination with Feig teaches these limitations.

155. As discussed previously with respect to limitations [1.g], [5.g], and [9.g], Carmel teaches sending the media data elements by the server system to the user systems in response to the requests by the user systems because the server, "[r]esponsive to a user input" for slices (e.g., "42, 44, 46, etc."), provides all such requested slices to the client and no other slices. EX1005, 10:36-50. This is shown in Figure 6A below ("HTTP from Server," "Read Index File," and "Select Slice"), which is "a method of <u>downloading broadcast data from a server to a client</u>." *Id.* at 6:3-6.

FIG. 6A



156. Because Carmel's data stream is divided into various media data elements, a POSITA would have recognized that each slice file may be requested individually using an HTTP GET request, and the server would send each requested slice to the user system in response. *See id.* at 2:4-5 ("[t]he data stream is divided into a sequence of segments or slices of the data"). For example, "client 30 opens one or two HTTP links, over which files 42, 44, 46, etc., are downloaded in successive alternation." *Id.* at 10:38-40.

157. To the extent Carmel does not disclose all of the media data elements that are sent by the server system to the one or more user systems are sent in response

to the requests, Feig does. As I mentioned above, Feig allocates two client buffers for storing respective segments. The size of the buffers is set to the amount "required to store the segment with the largest amount of data." EX1031, 5:10-13; *see id.* at 3:33-34 ("a value B(j) representing the size of the corresponding data"). Feig's method fills the buffers by making requests for the media data elements located at the various URLs in the segment. *Id.* at 5:16-43. A POSITA would have recognized and found obvious from this disclosure that all of the media data elements that are sent by the server system to the one or more user systems are sent in response to the requests because there is a risk a buffer would overflow if the server sent media data elements other than those requested by the client.

158. A POSITA would have been motivated to combine the teachings of Feig related to all of the media data elements that are sent by the server system to the one or more user systems being sent in response to the requests with Carmel because it would allow the client to precisely control and select which segments it receives from the server, which supports Carmel's goal of allowing a user to "decide and indicate at which slice of data stream 40 to begin downloading." EX1005, 10:42-45. Such a combination is no more than using a known technique in Feig to improve a similar device or method in Carmel in the same way.

(l) Limitations reciting that all of the elements are sent from the data structure as the elements were first stored therein

[1.l] "all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein."

[5.l] "all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein."

[9.l] "all of the media data elements that are sent by the server system to the requesting user systems are sent from the data structure under the control of the server system as the media data elements were first stored therein."

159. In my opinion, Carmel teaches these limitations.

160. As discussed above, Carmel's server as described in [1.e] stores media data elements in a data structure under the control of the server system; [1.f-1.g] receives requests for media data elements in the data structure and sends such requested media data elements; and in [1.k] sends all such media data elements to the user system in response to the requests.

161. Carmel further teaches that the same slices (same format), e.g., "42, 44, 46, etc.," are stored in the server's data structure to be sent to the requesting user systems. *See supra* claim [1.e]. For example, Carmel teaches that "the memory available on server 36 is limited, and files 42, 44, 46, etc., will be stored on the server and erased therefrom in a 'first-in-first-out' sequence." EX1005, 7:55-58; *see also id.* at 8:23-29 ("[t]he symbols J, J+1, J+2, . . . N in the figure are 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. J may indicate the first slice in the sequence, if all of the files are stored on server 36, or it may be the earliest file not yet erased.").

162. Carmel also teaches sending the same stored slices, e.g., "42, 44, 46, etc.," in the data structure to each requesting user system. *See supra* claims [1.f-1.g, 1.k]; *see also* EX1005, 10:46-50 ("[r]esponsive to a user input, client 30 selects an appropriate starting slice and begins to download and decode (decompress) files 42, 44, 46, etc. In the case of a multimedia stream, client 30 reconstructs and outputs the multimedia data for the appreciation of a user."), Fig. 6A, 6:3-6 (describing Fig. 6A as "a method of downloading broadcast data from a server to a client").

163. Nevertheless, a POSITA would have understood that Carmel sends all the slices to the requesting clients in the same format and order as first stored in the server's data structure because Carmel's server stores and sends them in a "first-in-first-out" sequence. For such a "first-in-first-out" sequence, the order in which the media data elements are sent from the server to the requesting user systems remains the same as the order in which they were received from the recording client. A POSITA would have understood that the sequence of recorded media stream segments would be the same sequence used for continuous playback, otherwise the played stream would not maintain the same temporal continuity and would be subject to jumps that may disorient the user. For example:

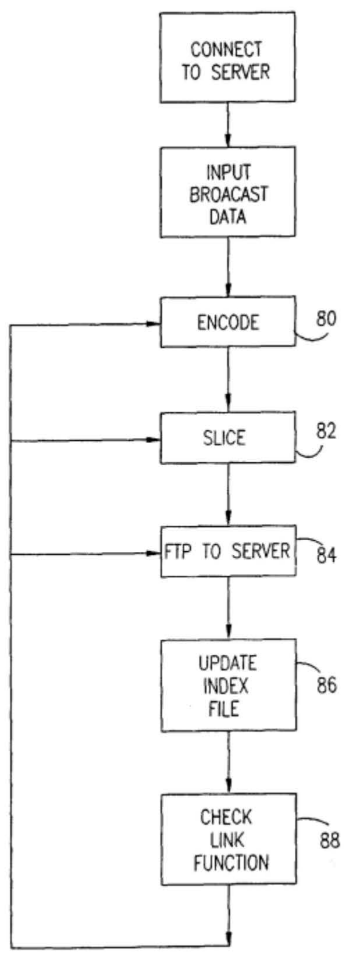
> The slices are encoded in a corresponding sequence of files, each file having a respective index, and the sequence is uploaded to a server at an upload rate generally equal to the data rate of the stream, such that the one or more client computers can download the sequence over the network from the server at a download rate generally equal to the data rate.

Id. at Abstract. Therefore, the clients receive the data stream in its original format.

164. Additionally, Carmel teaches this limitation to the extent it is construed as "all of the media data elements that are sent by the server system to the requesting user systems are sent from the same data structure under the control of the server system and in the same format as the media data elements were 'first stored therein.'" This is because, in Carmel, after the transmitting computer (34) prepares the media data elements (multimedia sequence) and uploads them to the server (36) to be first stored, the format and order of the media data elements does not change when they are downloaded by the requesting user systems (clients 30). For example, "[a]fter preparing the multimedia sequence, computer 34 uploads the sequence over network 28 . . . the data in the sequence are compressed . . . . Computer 34 is preferably equipped with suitable software for preparing and compressing the multimedia sequence." *Id.* at 6:50-62. Additionally, "[c]lients 30 connect to server 36 and receive the multimedia sequence, substantially in real time." *Id.* at 7:4-5. This is shown in Carmel at Figure 5, including the steps of "Input Broadcast Data," "Encode," "Slice," and "FTP to Server." For example, Carmel teaches that the

sequence of storing uploaded media segments from the user ("42, 44, 46, etc.") (*id.* at 7:50-58) is later used as the same sequence for playback when the media is requested by the user so that "files 42, 44, 46, etc., are downloaded in successive alternation." *Id.* at 10:36-48.

FIG. 5

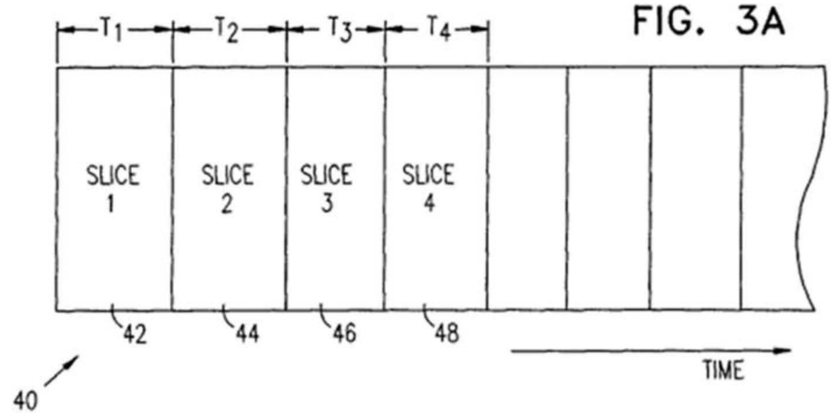


6. Claims 2, 6, and 10 are obvious over Carmel in view of Feig and Willebeek.

165. As I describe above, Carmel in view of Willebeek teaches claims 1, 5, and 9. Claims 2, 6, and 10 depend on claims 1, 5, and 9 respectively, and further recite the limitation "wherein the serial identifiers are sequential." In my opinion, Carmel teaches claims 2, 6, and 10.

166. Carmel teaches that the data stream broadcasted via a network server to a plurality of clients "is divided into a sequence of segments or slices of the data, preferably time slices, wherein the data are preferably compressed. Each slice is preferably assigned a respective slice index." EX1005, 2:1-21; *see also id.* at 8:21-29. Furthermore, Carmel provides an example of a time sequence of media segments or slices. For instance, "[d]ata stream 40 comprises a series of data slices 42, 44, 46, 48, etc. Each slice contains a segment of video and/or audio data, corresponding to a respective, successive time interval labeled T1, T2, T3, etc. . . . Computer 34 stores each slice as a corresponding file, having a running slice index 1, 2, 3 . . . N." *Id.* at 7:18-28; *see also id.* at 3:24-34; *supra* limitations [1.d], [5.d], and [9.d].

167. For example, FIG. 3A of Carmel below shows sequential and numbered slices 1-4.



7. Claims 3, 7, and 11 are obvious over Carmel in view of Feig and Willebeek.

168. As I describe above, Carmel in view of Feig and Willebeek teaches claims 1, 5, and 9. Claims 3, 7, and 11 depend on claims 1, 5, and 9 respectively, and further recite the limitation "wherein the sending is via a reliable transmission protocol." Carmel teaches claims 3, 7, and 11.

169. Carmel teaches that the server system sends media data elements to the requesting user systems via a reliable transmission protocol well known in the art such as HTTP. For example, "[c]lients 30 connect to server 36 and receive the multimedia sequence, substantially in real time. 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." EX1005, 7:4-9; *see also id.* at 6:28-31, 5:25-28. A POSITA would have understood that HTTP was used in conjunction with TCP (a reliable transport protocol) and hence would result in the sending to the

client being by a reliable transmission protocol. *See* EX1006, 3-4 (explaining that HTTP uses TCP/IP as "the underlying transport protocol . . . which provides reliable end-to-end network connections").

170. While it is my opinion that Carmel, alone, teaches "wherein the sending is via a reliable transmission protocol," Willebeek also explicitly teaches this limitation because in the Bamba system, "the underlying transport protocol used by HTTP is TCP/IP, which provides reliable end-to-end network connections." *Id.*

8. Claims 4, 8, and 12 are obvious over Carmel in view of Feig and Willebeek.

171. As I describe above, Carmel in view of Feig and Willebeek teaches claims 3, 5, and 11. Claims 4, 8, and 12 depend on claims 3, 5, and 11, respectively, and further recite the limitation "wherein the reliable transmission protocol is TCP." Carmel teaches claims 4, 8, and 12.

172. Carmel teaches various communication protocols for communication between client 30 and server 36 such as HTTP. For example, "[s]ystem 32 comprises a transmitting computer 34, which generates the 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)." EX1005, 6:28-31; *see also id.* at 7:4-9 ("Clients 30 preferably download the sequence using the Hypertext Transfer Protocol (HTTP) . . . ."), 5:25-28.

173. A POSITA would have understood that HTTP was used in conjunction with TCP (a reliable transport protocol) and hence would result in sending data to the client via TCP. *See* EX1006, 3-4 (explaining that HTTP uses TCP/IP as "the underlying transport protocol . . . which provides reliable end-to-end network connections").

174. While it is my opinion that Carmel, alone, teaches "wherein the reliable transmission protocol is TCP," Willebeek also explicitly teaches this limitation because in the Bamba system, "the underlying transport protocol used by HTTP is TCP/IP, which provides reliable end-to-end network connections." *Id.*; *see also id.* at 4 ("Bamba uses TCP/IP as the underlying communication protocol").

IX. CONCLUSION

175. In my opinion, and for the reasons provided above, claims 1-12 are obvious over Carmel, Feig, and Willebeek.

I hereby declare that all statements made herein of my own knowledge are true and that all statements 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 Section 1001 of Title 18 of the United States Code.

Dated: August 22, 2022

Kevin Jeffay