

**IN THE UNITED STATES DISTRICT COURT
FOR THE DISTRICT OF DELAWARE**

VIDEOLABS, INC., and
VL COLLECTIVE IP LLC

Plaintiffs,

v.

NETFLIX, INC.

Defendant.

Civil Action No. 22-cv-229-CFC

JURY TRIAL DEMANDED

FIRST AMENDED COMPLAINT

Plaintiffs VideoLabs, Inc. (“VL”) and VL Collective IP LLC (“VL IP”) (collectively “VideoLabs” or “Plaintiffs”) file this First Amended Complaint against Defendant Netflix Inc. (“Netflix” or “Defendant”), and in support thereof alleges as follows:

NATURE OF THE ACTION

1. Digital video has become fundamental to how society interacts, communicates, educates, and entertains. In fact, video consumption now accounts for more than 82% of all Internet traffic.¹ The ability to reliably provide high-quality video drives the growth of digital platforms that are increasingly integral to the global economy.

2. The advent of high-quality video as a staple of digital consumption did not happen instantaneously. As with any complex technology, digital video presented implementation challenges. Many companies spent many years and resources to develop new and innovative technologies that guide how video is created, streamed, secured, managed, and consumed.

¹ See Ex. 1, *The Sustainable Future of Video Entertainment*, INTERDIGITAL (Aug. 2020), https://www.interdigital.com/white_papers/the-sustainable-future-of-video-entertainment?submit_success=true (last visited Jan. 20, 2022).

3. Various inventions and technological advances have transformed digital video. Some of these technologies, such as techniques to efficiently compress video file size, address central challenges to storing and transmitting video. Others enable video content to be efficiently and securely streamed to the many user devices that exist today. Yet others involve managing and organizing videos to provide viewers easier access to content and address how they interact with content. Successful video streaming thus requires myriad technologies that necessarily coordinate with one another.

4. Because various companies played roles in developing the foundational technology for today's digital video, no single company can provide the high-quality video experiences that consumers have come to expect without using technology owned by other companies.

5. The founders of VideoLabs recognized this problem and understood that collective action was needed to address it. If the companies that developed critical video technologies worked together, everyone could benefit: innovators could receive fair compensation for their contributions, companies deploying video technology could respect the innovators' patents and license them on affordable and predictable terms, and consumers could experience better and more affordable video technology.

6. In 2019, with support from widely-recognized industry leaders, VideoLabs launched a platform to achieve these goals. VideoLabs spent millions of dollars and thousands of hours analyzing the video space and identifying the patents that reflect the innovations with the highest impact. VideoLabs then compiled a portfolio of these core patents, obtaining them from leading companies, including Hewlett Packard Enterprise, Alcatel-Lucent S.A., Siemens AG, Swisscom AG, 3Com, Panasonic, LG, and Nokia.

7. VideoLabs then opened-up membership on its platform to all willing companies.

In exchange for low-cost membership or licensing fees, VideoLabs provides access to its patent portfolio and a commitment to seek out the most important patents in the video industry and clear them. Many prominent companies recognized the benefits of the VideoLabs platform and worked with VideoLabs to efficiently and responsibly license its video technology patents.

8. Unfortunately, Netflix has not. Netflix is one of the world's largest users of video technologies and operates the world's most popular streaming TV service with over 213 million subscribers.² It is enmeshed in practically every aspect of video, from creation to processing, delivery, and display.³

9. VideoLabs contacted Netflix multiple times to offer Netflix the benefit of VideoLabs' platform and to alert it to its use of VideoLabs' patented technology. As an added incentive for engaging in good faith discussion, VideoLabs offered to conduct discussions with Netflix under an NDA that would eliminate any legal risk from participating in the discussions, including a 3-month mutual legal standstill which would provide sufficient time for full and open dialogue. After many months of ignoring VideoLabs' entreaties, Netflix finally responded. But Netflix foreclosed the possibility of good faith discussions by insisting that VideoLabs agree not

² See, e.g., Ex. 2 at 3, <https://www.cnn.com/2021/11/10/disney-netflix-and-other-streaming-services-subs-arpu-q3-2021.html#:~:text=Netflix%20continues%20to%20outpace%20the,around%20paying%20customers%20and%20ARPU>; Ex. 3 at 6-7 <https://screenrant.com/ten-most-popular-streaming-services-ranked-subscriber-numbers/>; Ex. 4 at 3-4, <https://www.businessofapps.com/data/netflix-statistics/>.

³ See, e.g., Ex. 5 at 1, <https://www.statista.com/statistics/883491/netflix-original-content-titles/>; Ex. 6 at 1-3, <https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746>; Ex. 7 at 1-9, <https://www.comparitech.com/blog/vpn-privacy/netflix-statistics-facts-figures/>; Ex. 8 at 1-9, <https://netflixtechblog.com/ava-the-art-and-science-of-image-discovery-at-netflix-a442f163af6>; Ex. 9 at 1-6, <https://www.webdesignerdepot.com/2020/02/3-lessons-ux-designers-can-take-from-netflix/>; Ex. 10 at 1-4, <https://uxmag.com/articles/how-netflix-uses-psychology-to-perfect-their-customer-experience>; Ex. 11 at 1-12, <https://uxplanet.org/the-netflix-conundrum-overcoming-the-paradox-of-choice-a-ux-case-study-95b19acdc28c>.

to enforce its patent rights against Netflix for an indefinite period of time. When VideoLabs declined this demand, Netflix ceased responding to any of VideoLabs' communications.

10. Faced with this Hobson's choice, VideoLabs felt that it had no recourse but to file an action to stop Netflix's unauthorized use of VideoLabs' patents. Failure to take action would undermine the viability of VideoLabs' platform and permit further free-riding by Netflix of the significant innovations of VideoLabs' patents.

11. This case is ultimately about ensuring the integrity of the patent system and compensating patent owners for their protected innovations. Respect for intellectual property, as the law requires, is essential to incentivize innovation and promote technological progress. Accordingly, VideoLabs brings this action under the patent laws, 35 U.S.C. § 1 *et seq.*, in order to stop Netflix's willful infringement of U.S. Patent Nos. 8,139,878, 7,440,559, 7,233,790, and 8,605,794 (collectively, "patents-in-suit").

THE PARTIES

12. VL was founded in 2018 as part of an industry-sponsored and -funded effort to reduce the cost and risk of technological gridlock associated with diverse patent ownership. VL's leadership has decades of experience in intellectual property licensing, during which they have completed over 1,000 intellectual property transactions worldwide and drawn more than \$6 billion in revenue.

13. VL is a corporation organized under the laws of the State of Delaware, with its principal place of business in Palo Alto, California.

14. VL IP was founded in 2019 as a subsidiary of VideoLabs, Inc.

15. VL IP is a corporation organized under the laws of the State of Delaware, with its principal place of business in Palo Alto, California.

16. On information and belief, Netflix is a publicly traded corporation organized and

existing under the laws of the State of Delaware and is registered to do business in the State of Delaware. Netflix's headquarters are located at 100 Winchester Circle, Los Gatos, California 95032.

JURISDICTION AND VENUE

17. This is an action for patent infringement arising under the patent laws of the United States. This Court has jurisdiction over the subject matter of this action under 28 U.S.C. §§ 1331 and 1338(a), 15 U.S.C. § 1121, and 28 U.S.C. § 1367(a).

18. This Court has personal jurisdiction over Netflix because, on information and belief, Netflix conducts business in and has committed acts of patent infringement in this District, and has established minimum contacts with this forum state such that the exercise of jurisdiction over Netflix would not offend traditional notions of fair play and substantial justice. Netflix is incorporated in this District. On information and belief, Netflix offers products and/or services, including those accused herein of infringement, to customers and potential customers located in this District.

19. Venue is proper in this Court under 28 U.S.C. §§ 1391 and 1400(b). Netflix resides in this District. Netflix has chosen to incorporate in the state of Delaware, thereby receiving the benefits offered to Delaware corporations. Netflix must accordingly assume responsibilities to Delaware and its citizens.

20. Further, on information and belief, Netflix has offered and sold, and continues to offer and sell, its infringing products and services in this District. On information and belief, Netflix designs, uses, distributes, sells, and/or offers to sell the infringing products and services to consumers and businesses in this District.

21. On information and belief, Netflix is a corporation with global reach and annual revenue in the billions of dollars. Netflix accordingly cannot reasonably claim it would be

inconvenient to litigate in the forum in which it is incorporated.

22. Moreover, litigating in this District is convenient and would serve the interests of judicial economy because of a related pending lawsuit in this District.⁴

THE VIDEOLABS PATENTS-IN-SUIT

A. U.S. Patent No. 8,139,878

23. U.S. Patent No. 8,139,878 (the “’878 Patent”), titled “Picture Coding Method and Picture Decoding Method,” issued on March 20, 2012. VL owns all rights and title to the ’878 Patent, as necessary to bring this action. A true and correct copy of the ’878 Patent is attached as Exhibit 12.

24. The ’878 Patent was developed by engineers at Panasonic, one of the largest consumer electronics companies at the time of the invention and a major innovator in Internet technologies. In 2002, when patent applications were first filed for the ’878 Patent, Panasonic was a world leader in digital video technologies.⁵ Panasonic developed video coding technologies and designed consumer electronics — including TVs, DVD players, and memory cards — for storing, processing, and displaying video content.⁶ The inventions of the ’878 Patent are the result of years of research by teams of Panasonic engineers working at the cutting edge of video processing and encoding.

25. Native video files are massive. Modern digital video cameras used by premier television and movie studios capture images at incredibly fast rates (ranging from 30 frames per

⁴ See *Starz Entertainment, LLC v. VL Collective IP, LLC*, 1-21-cv-01448 (D. Del. filed Oct. 13, 2021).

⁵ See, e.g., Ex. 13 at 6, 10-17, 41, *Annual Report 2002*, National/Panasonic Matsushita Electric, available at <https://www.annualreportowl.com/Panasonic/2002/Annual%20Report> (last accessed Jan. 20, 2022).

⁶ See *id.*

second up to 300 frames per second) and extremely high resolutions (up to “5k,” or 5120 x 2880, for a total size of 14,745,600 pixels per frame). Storing just an hour of this raw content requires more than 300 GB of memory.⁷ Most modern TVs, laptops, tablets, and smartphones cannot possibly store and play such large files.

26. Even if they could, there would be little point from the perspective of on-demand content delivery: Internet speeds are far too slow to stream such massive video files. The fact is that transmitting high quality audiovisual content is simply not possible without powerful compression technologies. Streaming even just standard high-definition content (720p) requires network bandwidth of approximately 1.5 Gbps,⁸ which is about 35 times faster than the average Internet speed in the United States.⁹ “Encoding” and “decoding,” which respectively refer to the processes of compressing and decompressing content, are thus essential to applications such as video streaming, digital television, and videoconferencing.

27. Encoding video content allows the content to be made small for storage and transmission, while decoding permits the viewer to watch high-quality content on his or her device. In addition to making real-time streaming of content possible, every incremental increase in compression efficiency yields substantial benefits to companies that store, process, transmit, or access video. For example, if a video streaming company can cut the size of each of its movie files in half, then it reasons that it only needs half the numbers of servers to store its movies, half

⁷ See Ex. 14, *How Many GB Is a 2 Hour 4k Movie?*, <https://gamingsection.net/news/how-many-gb-is-a-2-hour-4k-movie/> (last visited Jan. 20, 2022).

⁸ See Ex. 15, Bryan Samis, *Back to Basics: GOPs Explained*, AWS MEDIA BLOG (May 28, 2020), <https://aws.amazon.com/blogs/media/part-1-back-to-basics-gops-explained/> (last visited Jan. 20, 2022).

⁹ See Ex. 16, *Average U.S. Internet Speed is 42.86 Mbps*, ETI (Feb. 2, 2021), <https://etisoftware.com/resources/blog/report-average-u-s-internet-speed-is-42-86-mbps/> (last visited Jan. 20, 2022).

the network bandwidth to transmit its movies, and half of all other related expenses, such as energy costs and staffing resources.

28. The '878 Patent describes breakthrough techniques for encoding and decoding audiovisual content so that it can be transmitted and stored with fewer resources. The patent vastly improves upon existing methods, and the core technology it describes has been used throughout the industry for years as the gold standard for coding video.

1. Background On Coding Technology

29. Video “coding” refers to both the encoding and decoding of video content. Video compression techniques minimize the size of the data that is sent between the encoder and the decoder by removing redundancies and imperceivable changes and then efficiently representing the remaining data for transmission.

30. Video is comprised of a series of frames. These frames are successively output to create the moving pictures that we recognize as video.



Figure 3.10 Frame 1



Figure 3.11 Frame 2

Ex. 17, Iain E. Richardson, *The H.264 Advanced Video Compression Standard* (2d. ed. 2010) (hereinafter “Richardson”), at 33.

31. In the early 2000s, certain techniques existed to reduce the amount of data needed to describe each frame without any loss in picture quality. For example, if there are a series of 50 white pixels in a row followed by 75 green pixels, then it is more efficient to store the fact that

there are 50 white pixels followed by 75 green pixels than to store the value of all 125 pixels. This algorithm, which reduces the redundancy stemming from repeating pixels within a frame, yielded substantial benefits.

32. Video engineers also realized that, very often, not much changes between successive frames. In the images shown above, for example, the changes between frames 1 and 2 are largely concentrated in the area near the book. As a result, it is not necessary to send the complete data for every frame of a video. Instead, frames can be sent periodically at strategic points, such as when there is a scene change that creates major differences between successive frames. Those strategic frames — called “key frames” — could be used to “predict” other frames nearby in time by analyzing each frame and storing the differences from one frame to the next.

33. Further research yielded additional advances in what became known as predictive coding. Video engineers realized that it was advantageous to divide each frame into blocks, as shown below.

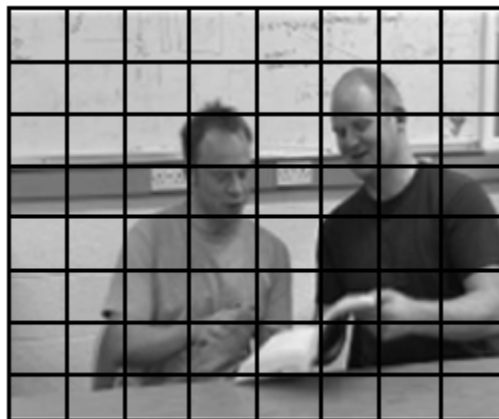


Figure 3.10 Frame 1

34. These blocks could be analyzed and used to predict the pixels in the same block in *surrounding* frames (“*inter-picture* prediction,” also called “temporal compression”). Additionally, these blocks could be analyzed to predict the pixels in surrounding blocks in the

same frame (“*intra*-picture prediction,” also called “spatial compression”). While predictive coding does not always recreate frames that are identical to the original frames, the differences are so minor as to be imperceptible. For example, in the middle of an intense action sequence, a frame might display a pixel as blue even though it should be green because doing so enables the image to be represented more efficiently. This minor alteration from the original content will go unnoticed by the viewer, who is distracted by all the activity.

35. Once redundancy in the video content has been minimized and imperceptible details have been streamlined, a process called “entropy encoding” further compresses the data by using as few bits to represent the data as possible, while still ensuring fidelity to the original visual content. This is achieved by allocating the fewest bits to commonly appearing bit sequences, and the most bits to infrequently occurring bit sequences. By way of analogy, when training your dog, the commands you use most frequently are likely the shortest, single-word commands, like “sit” and “no.” But commands that you need less frequently may be longer, such as “wait for it” and “roll over.” In this way, over the course of a week, you expend fewer (verbal) resources. Entropy encoding applies this same principle to the bits of data that comprise video content.

36. There are standardized ways to represent sequences of bits, and depending on the type of entropy coding, these sequences are stored in either “coding tables” or “probability tables.” Entropy coding involves selecting the optimal table for the data being transmitted and ensuring that the decoder knows the proper table to use when decoding the data.

37. It was in this context that the inventors of the ’878 Patent made their contributions.

2. The ’878 Patent

38. The ’878 Patent is directed to encoding audio and video content. With respect to video, the ’878 Patent describes a type of coding called “Context-based Adaptive Variable Length Coding,” or “CAVLC.” *See, e.g.,* Ex. 12, ’878 Patent at col. 1, ll. 49-52. Content encoded using

the techniques of the '878 Patent would then be stored or transmitted before ultimately being decoded for playback.

39. When encoded, the image data in a particular image block is represented by, among other things, its “coefficients.” *Id.* at col. 1, ll. 63-67; col. 7, ll. 38-43; col. 21, ll. 60-66; col. 25, ll. 29-36. Roughly speaking, larger coefficients for a block indicate a larger amount of changes in that block as compared with a reference block. *See id.* For many blocks, there are no such changes, and so all the coefficients have a value of zero. *See id.* at col. 21, ll. 60-66. The inventors of the '878 Patent recognized that these “zero-coefficient” blocks presented an opportunity for further compression. *See, e.g., id.* at col. 1, ll. 49-52.

40. They realized that the decoder did not need to know every single time a zero-coefficient block existed; rather, the decoder needs to know only when blocks have *non-zero* coefficients. They devised a technique wherein data about zero-coefficient blocks are effectively not encoded at all, and only non-zero coefficient block data is stored and transmitted. *See, e.g., id.* at col. 1, ll. 49-52, 56-62; col. 1, l. 65 – col. 2, l. 10. The inventors thereby achieved nearly perfect compression for these zero-coefficient blocks by communicating them practically without sending any information whatsoever. *See id.* at col. 2, ll. 11-14.

41. The inventors also made a substantial contribution to the efficiency of entropy coding. They recognized that the coefficients in neighboring blocks were a good predictor of the coefficients in the block being analyzed, and so could be used to select the optimal coding table for the block, yielding enhanced compression. *See, e.g., id.* at col. 9, ll. 34-37; col. 13, ll. 4-11. Prior techniques lacked this level of sophistication. They did not take advantage of the predictive power provided by analyzing the coefficients of the surrounding blocks. They would also use the same coding table for both inter- and intra-predictive coding, which was inefficient because there

could be significant differences between neighboring blocks in the current frame and blocks in subsequent frames. *See, e.g., id.* at col. 1, ll. 33-38. Due to these limitations in the use of coding tables, compression efficiency in previously known entropy coding techniques would vary significantly between different types of content, and generally decreased as the quality of content increased. *Id.* at col. 1, ll.39-44. These problems (and others) were overcome by the inventors of the '878 Patent.

42. The innovations of the '878 Patent provided a significant advance in compression that was recognized throughout the industry. In fact, the compression techniques of the '878 Patent are used in the ubiquitous video codec, H.264. H.264 was revolutionary in the video industry, as it provided a quantum leap of improvement over the video codecs that had previously been commonly used, such as Motion JPEG video and MPEG-2. In particular, H.264 “has an 80% lower bitrate than Motion JPEG video” and “the bitrate savings can be as much as 50% or more compared to MPEG-2.”¹⁰

B. U.S. Patent No. 7,440,559

43. U.S. Patent No. 7,440,559 (the “’559 Patent”), titled “System and Associated Terminal, Method and Computer Program Product for Controlling the Flow of Content,” issued on October 21, 2008. VL IP owns all rights and title to the ’559 Patent, as necessary to bring this action. A true and correct copy of the ’559 Patent is attached as Exhibit 18.

44. The original assignee of the ’559 Patent is Nokia Corporation, one of the largest consumer electronics and information technology companies in the world at the time of the invention and a major innovator of digital communications technologies. In 2003, the year in

¹⁰ See Ex. 19, *What is H264 Encoding?*, BlackBox, <https://www.blackbox.co.uk/gb-gb/page/38313/Resources/Technical-Resources/Black-Box-Explains/Multimedia/What-is-H264-video-encoding/>, at 2 (last visited Jan. 20, 2022).

which Nokia first filed for patent protection for the innovations of the '559 Patent, Nokia was a world leader in mobile device sales and technology. That year, Nokia launched its first media device, the Nokia 7700, and invested nearly one billion euros in research and development.¹¹

45. Customers are consuming more content via streaming services, commonly referred to in the industry as OTT (Over-The-Top) services, than ever before. At the same time, competition among video services is increasing. The number of OTT providers is constantly growing, and consumer confusion is mounting. Consumers expect the same level of innovation and development for OTT video as they do for other online services, and broadcasters and content providers are under constant pressure to distinguish their offerings through personalization and availability of innovative apps that entice and retain customers. The management, curation and optimization of audience viewing experiences across screens is becoming a core customer need, and at the same time an opportunity for service differentiation.

46. In the early 2000s, the deployment of high bit-rate mobile networks such as 3G enabled the delivery of new digital services, including video calling and streaming. *See, e.g.,* Ex.18, '559 Patent, col. 1, ll. 17-40. While audio could be delivered adequately using the bit rates available at the time, the limited transfer rates made it difficult to handle data-intensive tasks like delivering high quality full-motion video. *See, e.g., id.* For this and other reasons, alternative broadband delivery techniques were being investigated to support the delivery of data-intensive content. As digital broadband data broadcast networks evolved, there was increasing interest in combining use of mobile telecommunications with a broadband delivery technique to achieve efficient delivery of digital services to users on the move. But this led to new technical challenges

¹¹ *See* Ex. 20, Press Release, Nokia, Nokia Closes 2003 With Excellent Fourth Quarter, (Jan. 24, 2004), at 6, 9, available at <https://www.nokia.com/system/files/files/q4-2003-earnings-release-pdf.pdf>.

for content providers as they had to learn new techniques to efficiently deliver content to the myriad mobile devices that could consume broadband content over mobile networks.

47. At the time, mobile terminals would typically download content by “pulling” it from a server. *See, e.g., id.* at col. 2, ll. 25-39. This is because content providers tended to use content flow policies that had been used in non-mobile networks. *See id.* In those cases, the content provider typically maintained control over the content flow policy to the mobile terminal to enforce content access rights requirements. *See id.* The “pull” technique was thus rooted in the industry’s established habits, which ignored input from the devices consuming the content that might otherwise affect an operator’s content flow policy. Such outdated content flow policies were inefficient and undesirable as broadband content became accessible to mobile users everywhere and with myriad devices. When controlling content sent to a mobile device, they did not take into account, for example, the user preferences, terminal capabilities, previous content downloads, and/or use of previous content for that device. *See id.* at col. 2, ll. 40-53.

48. The ’559 Patent addresses these problems, among others, by giving a network entity control of the flow of content to the terminal based, in part, on status information from the terminal. *See* ’559 Patent, col. 2, ln. 57 – col. 3, ln. 9. Content flow is controlled, for example, by instructing the terminal to perform actions, such as downloading pieces of content from an origin server, or other content related actions based, in part, on the status information provided to the network entity from the terminal. *See id.* at col. 3, ll. 20-51. For example, the content provider can control the downloading and storage of content, as well as the deletion of content, at the terminal based upon status information regarding the terminal, and if so desired, further based upon status information regarding a source of content, such as the digital broadcast receiver, an origin server, or the like. *See id.* at col. 11, ll. 6-30. In that way, the flow of content to the terminal is more efficient since

the flow of new content to the terminal is affected by aspects of the terminal itself. *See id.* at col. 10, ll. 45-59.

C. U.S. Patent No. 7,233,790

49. U.S. Patent No. 7,233,790 (the “’790 Patent”), titled “Device Capability Based Discovery, Packaging and Provisioning of Content for Wireless Mobile Devices,” issued on June 19, 2007. VL owns all rights and title to the ’790 Patent, as necessary to bring this action. A true and correct copy of the ’790 Patent is attached as Exhibit 21.

50. The original assignee of the ’790 Patent is Openwave Systems, Inc. (“Openwave”), a leading developer of software applications for mobile devices. In the early 2000s, when the inventions of the ’790 Patent were in development, Openwave’s operating system and web browser software was being installed on billions of mobile phones.¹² This provided Openwave with a front seat to the many new products and services available to consumers on mobile devices.

51. The TV industry has been heavily affected by the rise of video on demand (“VOD”) and over-the-top (“OTT”) services, which allow users to conveniently stream over the Internet their favorite video content and watch it at any time, in any place, and in the format that best fits their needs. Today, digital video content is available from myriad streaming services and Pay TV operators and can be consumed on an ever-growing number of different connected consumer devices.

52. In the early 2000s, when digital video delivery over the Internet was in its nascent period, delivering media to large numbers of mobile users presented challenges due to the stringent

¹² See Ex. 22, *Openwave Announces Mobile Browser Integration for Qualcomm’s Brew Solution*, INTERNET ARCHIVE WAYBACK MACHINE, (Sept. 12, 2006), https://web.archive.org/web/20061127222501/http://www.openwave.com/us/news_room/press_releases/2006/20060912_opwv_brew_0912.htm, at 1 (last visited Jan. 11, 2022).

requirements of streaming media, mobility, wireless, and scaling to support large numbers of users. While advances in next-generation cellular networks and wireless networks were bringing higher bandwidths to mobile users, these higher bandwidths naturally created the demand for media-rich content, which in turn created requirements for a media delivery infrastructure that could handle the challenges of streaming media, user mobility, and scaling to large numbers of users accessing content with different types of devices. Traditional content delivery techniques that had previously served the market reasonably well at the time were no longer capable of meeting current needs.

53. Indeed, these techniques were rooted in the nature of the old technologies, in which content was prepared and packaged once, for distribution over a traditional broadcast medium and in a singular, conventional broadcast format. From a content supplier's perspective, an impediment to the efficient distribution of digital content was the fact that different connected devices often required different content packaging formats and provisioning protocols. In order for the content supplier to make a given item of digital content available to multiple connected devices supporting different provisioning models, a digital content supplier would normally have to deploy that item of content multiple times, packaging it differently for each of the provisioning models. Needing to package and provision digital content in a manner that is suitable for all of the connected devices in the marketplace is very burdensome. Moreover, it was a challenge for content suppliers to keep up with the constant changes in device capabilities for the many connected devices in the marketplace. As such, there was often a gap between the interoperability of a given digital product and how effective it could be used on a certain device.

54. The '790 Patent addresses this problem, among others, providing an efficient way for content providers to distribute digital content to different connected devices without the need to separately package and deploy the content differently for each device. *See* '790 Patent, col. 1,

ln. 60 – col. 2, ln. 17; col. 12, ll. 40-45. Since multiple different implementations of the content are kept in the content library, each associated with certain device capabilities, content can be packaged once, and distributed to the different user devices, leaving to the device the choice of which implementation version of the content to obtain from the service. *See* '790 Patent, col. 2, ll. 50-67; col. 9, ll. 37-50; col. 12, ll. 45-58; col. 11, ll. 42-62; col. 12, ll. 54-58. In the patented invention, the product information is separated from the content itself, the content is separated from how it is packaged for delivery, the packaged content is separated from the delivery mechanism, and the delivery mechanism is separated from the discovery mechanism. *See* '790 Patent, col. 12, ll. 46-54. By virtue of this separation, a content supplier can deploy content only once, targeting a wide range of devices, and trust it will successfully be delivered to those devices over a wide range of provisioning protocols. *See* '790 Patent, col. 12, ll. 54-58. Furthermore, access to different implementations of the content can be managed by the content supplier based on a device's capabilities ensuring the efficient distribution of compatible material. *See* '790 Patent, col. 2, ll. 7-17; col. 7, ll. 39-45; col. 8, ln. 66 – col. 9, ln. 12; col. 9, ll. 37-62.

D. U.S. Patent No. 8,605,794

55. U.S. Patent No. 8,605,794 (the "'794 Patent"), titled "Method for Synchronizing Content-Dependent Data Segments of Files," issued on December 10, 2013. VL IP owns all rights and title to the '794 Patent, as necessary to bring this action. A true and correct copy of the '794 Patent is attached as Exhibit 65.

56. The original assignee of the '794 Patent is Siemens Aktiengesellschaft ("Siemens"), one of the largest consumer electronics companies at the time of the invention and a major innovator in Internet technologies. In 2005 alone, the year in which Siemens first filed for patent protection for the inventions of the '794 Patent, Siemens invested €5.2 billion in research

and development.¹³

57. In the early 2000s, the inventors realized that the way that audiovisual content (e.g., television shows and movies) was transmitted to consumers was fundamentally changing. While content could be stored and accessed from media such as VHS tapes and DVDs, content was *transmitted* to consumers primarily through televisions — and had been for decades. Moreover, within each global region (e.g., the United States or Europe), all television content was encoded in a single formatting standard (e.g., the PAL standard in Europe and the NTSC standard in the United States) that could be played by all televisions. *See, e.g.*, Ex. 31, '794 Patent, col. 1, ll. 23-33.

58. But with the increasing importance of the Internet, the types of devices to which content could be transmitted was proliferating. *See, e.g., id.* at col. 1, ll. 34-43. Content was now being streamed to computers, laptops, PDAs, and other electronic devices. Unlike televisions, which were all designed to play content formatted in the same way, these new devices could play content encoded in any number of formats based on their capabilities. For example, a PDA, with its limited screen resolution and processing capabilities, could not process the higher quality content intended for high-resolution monitors connected to desktop computers. Additionally, a computer running a Windows operating system could play different content formats than an Apple computer.

59. The varying strength of Internet connections, particularly on wireless devices, also necessitated multiple content formats. For example, while a desktop computer might be capable of playing high resolution content, doing so is not desirable if the Internet connection for that computer is slow. Instead, it can be a better viewer experience for a lower quality version of the

¹³ Ex. 66, https://www.siemens.com/investor/pool/en/investor_relations/downloadcenter/e05_00_gb2005_1336469.pdf (last accessed Jan. 20, 2022).

content to be transmitted more quickly rather than having the user constantly waiting for higher quality content to download. Content delivery companies further realized that it would be beneficial to be able to change the quality of content *during a stream*. That is, when an Internet connection is weak, send lower quality content; when the connection is strong, send higher quality content. Thus, not only were different content formats necessitated by different device capabilities — even for the same device and during a single stream, it was advantageous to be able to vary the quality of the transmitted content.

60. Consumer expectations for the delivery of content also began to change. Whereas consumers could previously only watch whatever was “on TV,” consumers increasingly began to expect to watch whatever they wanted whenever they wanted, i.e., “on demand.” Consumers expected content to start playing at the click of a mouse, and to be able to jump to any point in the content and have playback resume immediately.

61. These changes in technology and consumer expectations led to new techniques for managing and processing audiovisual content. Content was no longer stored as a single file in a single location. Instead, for example, a movie’s audio and video data was broken up into numerous “segments” that might be stored on various Internet servers. These segments could be more easily transmitted over the Internet to consumer devices, and content could be played as soon as the first few segments were received instead of waiting until the entire file had been downloaded.

62. Prior to the innovations of the ’794 Patent, however, there was not a suitable method for aligning the various audio and video segments that comprised a piece of content. The need was all the greater when switching between content formats midstream (e.g., to account for changing Internet bandwidth) or skipping to different points within a piece of content.

63. Known techniques at the time would align the segments for playback using

timestamp information stored in each segment. Essentially, each segment includes metadata indicating when in the timeline of the content the segment should be played (e.g., audio content from 5 minutes and 30 seconds of the movie to 6 minutes and 30 seconds of the movie). Once a segment was downloaded, this information would have to be read out (which could require decoding the segment), and then additional processing would be needed to order this segment with the other segments. This technique was rooted in the nature of the old technologies, in which viewers received content in the order it was to be played, did not alternate in real time between different versions of the same content, and could not selectively play different parts of the content. Disadvantageously, this technique had a large overhead, and so could be slow and resource intensive. *See, e.g., id.* at col. 2, ll. 4-12, 36-54.

64. The '794 Patent improves upon these timestamp-based implementations. It describes a novel technique in which segments are ordered chronologically and aligned with corresponding segments (e.g., aligning a video segment with the correct audio segment) using predefined assignment rules. *See id.*, col. 2, ll. 36-42; col. 5, ll. 10-13. These assignment rules are not based on timestamps. *See id.*, col. 2, ll. 42-43. Instead, they flexibly permit the alignment of segments using rules appropriate for different contexts. This could include implementations in which, for example, each sequential video segment is aligned with every fourth audio segment. *See id.*, col. 2, ll. 55-60; col. 5, ln. 35 – col. 6, ln. 42. Alternatively, the assignment rules could be used to build pseudo-timelines that order and match audio and video segments based on the context of the content. *See id.*, col. 6, ll. 50-60. For example, key audio and video segments will align at the start of new scenes, changes in camera viewpoint, or the start of a song. The assignment rules of the '794 Patent require little overhead and are thus significantly faster than timestamp-based techniques while also providing more options in the management of segments. *See, e.g., id.* at col.

2, ll. 4-12, 36-54. This flexibility enables, for example, a user to jump to a key scene in a movie, and the corresponding segments to quickly be located, downloaded, and played. *See id.* at col. 3, ll. 20-28. This is because the context of content can be mapped to a particular segment, and then assignment rules can be used to quickly identify the corresponding and subsequent segments.

65. Today, online video streaming is ubiquitous, and the ability to alter the format of content mid-stream has been standardized and is known as “adaptive bitrate streaming.” There are two main protocols for this type of delivery: HTTP Live Streaming (“HLS”) and Dynamic Adaptive Streaming over HTTP (“DASH”). These protocols are used to stream the vast majority of online video. They are used by major streaming services (e.g., Amazon Prime Video, Disney+, Hulu).

66. The ’794 Patent is a core building block to these technologies, which has been recognized by the video technology industry. MPEG LA, which pioneered the concept of technology-specific patent pools and has created and maintained patent pools that efficiently license key technologies worldwide, launched a patent pool for DASH in November 2016.¹⁴ The ’794 Patent was submitted for inclusion into MPEG LA’s DASH patent pool, evaluated by MPEG LA’s patent experts, and declared as essential to using DASH to stream content. Indeed, the ’794 patent is one of just 10 U.S. patents that have been deemed essential to DASH, and its importance to the streaming technology and foundational nature is evidenced by the fact that it has the earliest invention date of all patents in the pool. Numerous companies have taken a license to the ’794 patent to obtain the right to use its technology to implement DASH.¹⁵

¹⁴ Ex. 67, *MPEG LA Releases MPEG-DASH Patent Portfolio License*, MPEG LA (Nov. 17, 2016), <https://www.mpegla.com/media/> (last visited Jan. 20, 2022).

¹⁵ Ex. 68, *DASH Licensees*, MPEG-LA, <https://www.mpegla.com/programs/dash/licensees/> (last visited Jan. 20, 2022).

ACCUSED NETFLIX PRODUCTS AND SERVICES

A. Netflix '878 Accused Products

67. The “Netflix '878 Accused Products” refers to all Netflix products, services, and functionalities that implement, in whole or in part, H.264 entropy encoding using CAVLC. This includes, for example, Netflix’s use of the H.264 baseline, main, and high profiles with CAVLC to encode its video content.

68. H.264 is the name for technology described in an industry standard that is widely used to encode and decode streaming video. H.264 reduces the file size of video files without any loss in quality of video, enabling companies to stream video in higher quality given the same network bandwidth.

69. H.264 focuses on the coding of the picture portions of the video content. To this end, H.264 defines a format, or syntax, for compressed video and a method for decoding this syntax to produce a displayable video sequence. An H.264 video encoder carries out prediction, transform, and encoding processes to produce a compressed H.264 bitstream. An H.264 video decoder carries out the complementary processes of decoding, inverse transform, and reconstruction to produce a decoded video sequence.

70. H.264 has been the dominant industry standard for compressing video for applications such as digital television, DVD video, video conferencing, and Internet video streaming. Standardizing video compression made it possible for products from different manufacturers to inter-operate. Recommendation H.264: Advanced Video Coding is a video decoding standard published by the international standards bodies ITU-T (International Telecommunication Union) and ISO/IEC (International Organisation for Standardisation / International Electrotechnical Commission) (hereinafter “H.264 Standard”) (available at <https://www.itu.int/rec/T-REC-H.264-202108-I/en>). It defines a format (syntax) for compressed

video and a method for decoding this syntax to produce a displayable video sequence. Products that support H.264 encoding and decoding are compliant with the H.264 standard.

71. According to Netflix, “given its wide support, our H.264/AVC Main profile family still represents a substantial portion of the members viewing hours and an even larger portion of the traffic.”¹⁶ As recently as August 10, 2020, Netflix announced that it was continuing to invest in its use of H.264, stating that “[k]eeping in mind our goal to maintain ubiquitous device support, we leveraged what we learned from innovations implemented during the development of newer encode families and have made a number of improvements to our H.264/AVC Main profile per-title encodes.”¹⁷ For its mobile devices, Netflix uses the high profile of H.264, which “enjoys broad decoder support.”¹⁸ Netflix’s use of H.264, and the resulting decrease in the bitrate of its content streams, “corresponds to a significant reduction in the overall Netflix traffic as well. These changes also lead to an improvement in Quality-of-Experience (QoE) metrics that affect the end user experience, such as play delays (i.e. how long it takes for the video to start playing), rebuffer rates, etc., as a result of the reduction in average bitrates. In addition, footprint savings will allow more content to be stored in edge caches, thus contributing to an improved experience for our members.”¹⁹

72. On information and belief, Netflix performs H.264 encoding using CAVLC to efficiently and seamlessly deliver video to its customers.

¹⁶ See Ex. 24, <https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9>, at 1-2.

¹⁷ See *id.* at 2; see also Ex. 25, <https://www.broadbandtvnews.com/2020/08/11/netflix-works-to-improve-h-264-experience/>, at 1-2.

¹⁸ See Ex. 26, <https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909>, at 1-2; Ex. 27, <https://www.vdocipher.com/blog/tech-update-netflix-updates-codecs-use-efficient-encoding/>, at 1-2.

¹⁹ See Ex. 24, <https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9>.

B. Netflix '559 Accused Products

73. “Netflix '559 Accused Products” refers to all Netflix products, services, features, and functionalities that control the flow of content to a client based on status information from a client and/or a content server. This includes, for example, all versions and implementations of the Netflix streaming service (including the service and the application).

74. Netflix controls the flow of streaming content to Netflix clients based, in part, on status information from the client and/or a content server.

75. On information and belief, Netflix uses status information from the client and/or a content server to “bookmark” content so that, when a viewer stops viewing content and then returns to view the same content later, the content begins at the time segment where the viewer left off. Without this feature, returning to view content would be significantly more inconvenient and probably cause viewers to view less content.²⁰

76. On information and belief, Netflix uses status information from the client and/or a content server to automatically download content based on the subscriber’s viewing history. Netflix’s automatic download feature operates so that, when a viewer “finish[es] watching a downloaded episode, Smart Downloads will delete it, and then automatically download the next episode.”²¹ The automatic download feature allows Netflix to “smartly download new episodes

²⁰ See, e.g., Ex. 28, <https://about.netflix.com/en/news/now-you-can-easily-tidy-up-your-continue-watching-row-on-all-devices>, at 1; See Ex. 29, <https://help.netflix.com/en/node/115312>; <https://www.addictivetips.com/web/get-continue-watching-on-top-in-netflix/>, at 1-3; Ex. 30, <https://discussions.apple.com/thread/4365964>, at 1-4 (various user complaints about a “bug” where Netflix videos were not updating to the most recent episode).

²¹ See Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, <https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1>, at 1-2 (last visited Feb. 17, 2022); Ex. 32, Amazon Prime Video to Get Netflix-like Smart Downloads Feature,

of TV shows so that users don't have to worry about doing the same manually.”²²

77. On information and belief, Netflix further uses status information from the client and/or a content server to prompt video features that significantly improve the consumer viewing experience. For example, Netflix provides viewers a “Skip Intro” option shortly after the opening credits sequence begins.²³ Similarly, when the beginning of an episode includes a summary of previous episodes in a TV series, Netflix provides viewers a “Skip Recap” option.²⁴ Additionally, immediately after the final scene of an episode, Netflix’s “autoplay” feature allows the viewer to automatically play the next episode in a TV series.²⁵ Without these features, viewing content would be more burdensome and the viewing experience would suffer.²⁶

C. Netflix ’790 Accused Products

78. “Netflix ’790 Accused Products” refers to all Netflix products, services, features,

MOBILESCOUT, at 1-2 (Nov. 19, 2019), <https://www.mobilescout.com/android/news/n114549/amazonprime-video-auto-downloads-feature-spotted.html> (last visited Feb 17, 2022).

²² See Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, <https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1>, at 1-2 (last visited Feb. 17, 2022).

²³ See Ex. 64, *How to Skip Intros on TV Shows*, NETFLIX, <https://help.netflix.com/en/node/63402>, at 1 (last visited Feb. 17, 2022).

²⁴ See Ex. 33, *Netflix Just Sneakily Added Another Button That Will Make You Binge So Much Better*, HELLOGIGGLES (Apr. 04, 2017) <https://hellogiggles.com/reviews-coverage/tv-shows/netflix-just-rolled-new-feature-will-seriously-help-show-facts-straight-binge/>; <https://help.netflix.com/en/node/2102>, at 1-6.

²⁵ See Ex. 35, *How to Autoplay the Next Episode*, NETFLIX, <https://help.netflix.com/en/node/121518>, at 1.

²⁶ See, e.g., Ex. 36, *Amazon Prime Video Advert Skip*, DIGITALSPY (July 12, 2021), <https://forums.digitalspy.com/discussion/2420457/amazon-prime-video-advert-skip> (last visited Jan. 20, 2022) at 6 (“The skip credits and Recap option is very useful as one click is a lot easier than having to forward to the place you want. It is obviously a popular feature as Netflix, Amazon and Disney Plus all have the feature”); Ex. 37, *Netflix’s “Skip Intro” Feature*, MEDIUM, <https://medium.com/an-attempt-at-writing/netflixs-skip-intro-feature-how-the-hell-do-they-do-that-7c5db9408f82> at 1-2 (“[Y]ou can skip intros of most shows. . . . [I]t’s a huge time saver and makes situations like binge watching especially very fluid.”).

and functionalities that maintain a product catalog of, and provide access to, content with a plurality of different implementations corresponding to device capabilities. This includes, for example, all versions and implementations of the Netflix streaming service (including the service and the application).

79. Netflix maintains a catalog of, and provides access to, streaming content, which has a plurality of different implementations corresponding to device capabilities.

80. On information and belief, Netflix receives and stores content, such as movies and TV shows, that will be provided to users.²⁷ By doing so, delivery of content to users is faster and less expensive.²⁸

81. On information and belief, Netflix maintains a plurality of different implementations of this content, where each implementation corresponds to device capabilities of the Netflix client.²⁹ Maintaining different implementations of content allows the Netflix clients to

²⁷ See, e.g., Ex. 38, *How Netflix Works with ISPs Around the Globe to Deliver a Great Viewing Experience*, NETFLIX (Mar. 17, 2016), <https://about.netflix.com/en/news/how-netflix-works-with-isps-around-the-globe-to-deliver-a-great-viewing-experience>, at 2-5 (last visited Feb. 21, 2022) (“Netflix Open Connect delivers 100% of our video traffic . . . almost all Netflix content is served from the local OCAs . . .”); see also Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021), <https://www.theverge.com/22787426/netflix-cdn-open-connect> (last visited Feb. 21, 2022), at 3 (“To avoid the traffic and fees, Netflix ships copies of its content to its own servers ahead of time.”).

²⁸ See, e.g., Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) at 3 (“To avoid the traffic and fees, Netflix ships copies of its content to its own servers ahead of time.”).

²⁹ See, e.g., Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021), at 6 (“Netflix effectively ships three copies of each of its titles to its servers, each with a different level of quality.”); see also Ex. 40, *Cloud Storage: How Does Netflix Store Their Content?*, GLOBAL TECH COUNCIL (Jan. 1, 2021), <https://www.globaltechcouncil.org/big-data/cloud-storage-how-does-netflix-store-their-content/> (last visited Feb. 21, 2022), at 2-4 (“To be available for viewing on a range of devices: from 44’ 4K Smart TV to 6-inch handset displays, they need to be encoded and compressed into different

smoothly and seamlessly stream content without “blips.”³⁰

82. On information and belief, Netflix maintains a product catalog which contains references to the content.³¹ By maintaining a product catalog that includes references to various content, Netflix is able to minimize and eliminate the risks of service disruption,³² and personalize its presentations for a particular user.³³

D. Netflix ’794 Accused Products

83. “Netflix ’794 Accused Products” refers to all Netflix products, services, features,

formats and qualities. . . . Many versions of the compressed encoded file are copied worldwide to hundreds of servers, forming a larger CDN for Netflix-like people around the world.”)

³⁰ See, e.g., Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) (““We will adapt the content to the quality of the network and not vice versa . . . [t]hat’s the reason why you don’t see when your network has a blip – your streaming stays constant.””).

³¹ See, e.g., Ex. 38, *How Netflix Works with ISPs Around the Globe to Deliver a Great Viewing Experience*, NETFLIX (Mar. 17, 2016) (“Although the number and size of the files that make up our content library can be staggering, we are able to use sophisticated popularity models to make sure the right file is on the right server at the right time.”); see also Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) (“[W]e are placing content on all of these servers around the world”); Ex. 40, *Cloud Storage: How Does Netflix Store Their Content?*, GLOBAL TECH COUNCIL (Jan. 1, 2021) (“It is essential to sift through and organize the petabyte value of content that Netflix has, like a library.”); Ex. 41, *Why Your Netflix Thumbnails Don’t Look Like Mine*, VOX (Nov. 21, 2018), <https://www.vox.com/2018/11/21/18106394/why-your-netflix-thumbnail-coverart-changes> (last visited Feb. 21, 2022) (“Netflix has a catalog of thousands of videos to watch . . . Netflix doesn’t just use a film or show’s original art; it employs an algorithm with the daunting task of sourcing high-quality images from those videos.”).

³² See, e.g., Ex. 39, *A Look Under the Hood of the Most Successful Streaming Service on the Planet*, THE VERGE (Nov. 17, 2021) (““This pre-placement of our films and shows allows us, based on prime time viewing hours, to store 100 percent of our catalog locally. And that basically eliminates the whole risk associated with service disruption.””).

³³ See, e.g., Ex. 42, *Selecting the Best Artwork for Videos Through A/B Testing*, THE NETFLIX TECH BLOG (May 3, 2016), <https://netflixtechblog.com/selecting-the-best-artwork-for-videos-through-a-b-testing-f6155c4595f6> (last visited Feb. 21, 2022) (“Selecting the best artwork has improved the Netflix product experience in material ways. We were able to help our members find and enjoy titles faster.”); .); see also e.g., Ex. 63, “Learning a Personalized Homepage”, NETFLIX, <https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a> (last retrieved, Feb. 21, 2022).

and functionalities that implement, in whole or in part, Adaptive Bitrate (ABR) streaming to deliver content to consumers. This includes, for example, all versions and implementations of the Netflix streaming service (including the service and the application).

84. According to Netflix, its “goals are for every member to view high-quality video on every device, every time, and for our system to scale efficiently as our global member base grows. We focus on quality of experience, quantified by metrics like video load time, visual quality and bitrate of delivered video, and rate of video interruptions.”³⁴ To achieve these goals, Netflix uses ABR streaming: “When our members’ kids watch Carmen Sandiego in the back seats of their parents’ car or more generally when the network throughput varies, adaptive streaming technologies are applied to provide the best viewing experience under the network conditions.”³⁵ Netflix implements ABR streaming along with multimedia standards ISO Base Media File Format (ISOBMFF) and Common Media Application Format (CMAF).³⁶

85. Netflix has long used ABR streaming, and currently uses it for both video and audio. Indeed, Netflix “ha[s] spent years optimizing our adaptive streaming engine for video, so we know adaptive streaming has obvious benefits.”³⁷ When Netflix began implementing ABR streaming for its audio, it noted that “[a]n adaptive streaming algorithm’s goal is to provide the best overall playback experience — even under a constrained environment. A great playback experience should provide the best overall quality, considering both audio and video, and avoid

³⁴ Ex. 69, *Streaming*, <https://research.netflix.com/business-area/streaming>.

³⁵ Ex. 70, *Packaging award-winning shows with award-winning technology*, <https://netflixtechblog.com/packaging-award-winning-shows-with-award-winning-technology-c1010594ba39>.

³⁶ *See id.*

³⁷ Ex. 71, *Engineering a Studio Quality Experience With High-Quality Audio at Netflix*, <https://netflixtechblog.com/engineering-a-studio-quality-experience-with-high-quality-audio-at-netflix-eaa0b6145f32>.

buffer starvation which leads to a rebuffering event — or playback interruption.”³⁸ In short, by using ABR streaming, Netflix is able to provide an enhanced user experience that is truer to the original content and seamless even as the strength of the user’s Internet connection varies.

ALLEGATIONS OF PATENT INFRINGEMENT

COUNT I **INFRINGEMENT OF U.S. PATENT NO. 8,139,878**

86. VideoLabs incorporates by reference the foregoing paragraphs of this First Amended Complaint as if fully set forth herein.

87. VL is the assignee and lawful owner of all right, title, and interest in and to the ’878 Patent. The ’878 Patent is valid and enforceable.

88. On information and belief, Netflix has infringed and continues to infringe the ’878 Patent in violation of 35 U.S.C. § 271(a), either literally or through the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States products and/or methods that practice at least claim 1 of the ’878 Patent, including with respect to the Netflix ’878 Accused Products.

89. On information and belief, Netflix uses the Netflix ’878 Accused Products for its own business purposes. In addition, Netflix regularly conducts research, testing, and troubleshooting of the Netflix ’878 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix’s subsidiaries) use the Netflix ’878 Accused Products.

90. On information and belief, Netflix’s infringement through its use of H.264 CAVLC entropy encoding, described below, is exemplary of all of Netflix’s infringement with respect to all the Netflix ’878 Accused Products.

³⁸ *See id.*

91. The Netflix '878 Accused Products directly infringe at least claim 1 of the '878 Patent, for example, by performing variable length encoding of blocks of picture data using the block data, inter-, intra-, and context-aware prediction to generate a predictive block, calculating a residual block using orthogonal transformation and quantization, and using the number of non-zero coefficients in the predicted block to encode the picture data at the encoder.

92. Each of the Netflix '878 Accused Products meet every limitation of claim 1 of the '878 Patent, which recites:

1. A transmitting apparatus which transmits multiplexed data which is obtained by multiplexing coded audio data and coded picture data, said transmitting apparatus comprising:

an audio processing unit configured to code audio data to obtain coded audio data;

a picture coding unit configured to code picture data to obtain coded picture data; and

a multiplexing unit configured to multiplex the coded audio data and the coded picture data to obtain multiplexed data,

wherein said picture coding unit includes a block decoding unit configured to code a block image to obtain coded block data, the block image being obtained by dividing a picture signal into plural blocks, generating a residual block image from a block image of the respective blocks and a predictive block image obtained by intra-picture prediction or inter-picture prediction, and coding, on a block basis, coefficients obtained by performing orthogonal transformation and quantization on the residual block image,

wherein said block coding unit includes:

a coefficient number coding unit configured to code a total number of non-zero coefficients included in a current block to be coded, each of the non-zero coefficients being a coefficient having a value other than "0";

wherein said coefficient number coding unit includes:

a determining unit configured to determine a predictive value for the number of non-zero coefficients included in the current block based on the number of non-zero coefficients included in a coded block located on a periphery of the current block;

a selecting unit configured to select a variable length code table based on the determined predictive value; and

a variable length coding unit configured to perform variable length coding on the total number of the non-zero coefficients included in the current block, by using the selected variable length code table.

93. Each of the Netflix '878 Accused Products includes a transmitting apparatus that transmits multiplexed data, which is obtained by multiplexing coded audio data and coded picture data. In one representative example, H.264 carries audio and video multiplexed in a single stream. H.264 is directed to the picture portion of video, so devices containing H.264 encoders and decoders must multiplex/demultiplex audio and picture data in order to provide the H.264 picture data. For example, encoders can multiplex the audio and pictures into a single stream, so that decoders receive the complete video presentation, including sound. Alternatively, the coded audio and coded video can be time-multiplexed and delivered as independent content streams. Regardless, decoders receive the complete video presentation, including sound, and decode the stream to recreate the video. In the ISO Media File Format, which each Netflix '878 Accused Products, and the Netflix streaming service, is capable of processing, for example, a coded stream such as an H.264 video sequence or an audio stream is stored as a track, representing a sequence of coded data items or samples. Figure 8.32, below, illustrates an example of such multiplexed data, in which coded audio data (“audio track samples”) and coded picture data (“video track samples”) are multiplexed together.

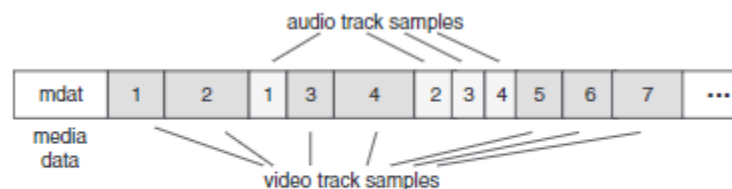
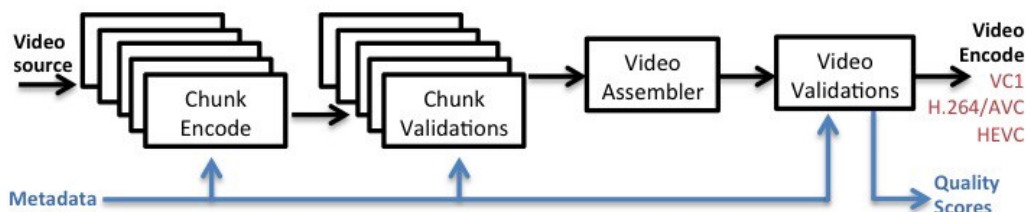


Figure 8.32 ISO Media File

Ex. 17, Richardson, at 247.

94. The Netflix '878 Accused Products have an audio processing unit configured to code audio data to obtain coded audio data. For example, the Netflix '878 Accused Products incorporate an audio codec that is configured to code audio data according to one of several input formats, including without limitation, Dolby Digital, Dolby Digital Plus, Dolby Atmos, and AAC.³⁹ The audio data is encoded as coded audio data (“audio track samples”).

95. The Netflix '878 Accused Products further include a picture coding unit configured to code picture data to obtain coded picture data. For example, the Netflix '878 Accused Products incorporate a H.264 video codec that is configured to code picture data to generate a H.264-compliant bitstream.⁴⁰ The picture data is encoded as coded picture data (“video track samples”). For example, Netflix employs a parallel video encoding process to generate multiple quality representations of a content program at different bitrates.⁴¹



A typical Netflix content program is encoded using multiple H.264 encoding profiles for encoding at the different bitrates:⁴²

³⁹ See Ex. 43, <https://netflixtechblog.com/optimizing-the-aural-experience-on-android-devices-with-xhe-aac-c27714292a33>; Ex. 44, <https://about.netflix.com/en/news/dolby-atmos-coming-to-netflix>;

⁴⁰ See Ex. 24, <https://netflixtechblog.com/improving-our-video-encodes-for-legacy-devices-2b6b56eec5c9>; Ex. 26, <https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909>.

⁴¹ See Ex. 6, <https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746>.

⁴² See e.g., filtered JSON manifest for the representative Netflix movie “IP Man” containing four H.264 High profile encodes.

```

Line 438:      "content_profile": "heaac-2-dash",
Line 480:      "content_profile": "heaac-2-dash",
Line 522:      "content_profile": "heaac-2hq-dash",
Line 590:      "content_profile": "playready-h264hpl30-dash",
Line 650:      "content_profile": "playready-h264hpl30-dash",
Line 710:      "content_profile": "playready-h264hpl30-dash",
Line 770:      "content_profile": "playready-h264hpl31-dash",

```

96. The Netflix '878 Accused Products include a multiplexing unit configured to multiplex the coded audio data and the coded picture data into multiplexed data. Encoders multiplex the audio track samples and video track samples so that decoders receive the complete video presentation, including sound. One non-limiting example of multiplexed data generated by the Netflix '878 Accused Products is shown in Figure 8.32 below, which shows the multiplexed stream of an ISO Media File including both coded audio track data and coded video track data.

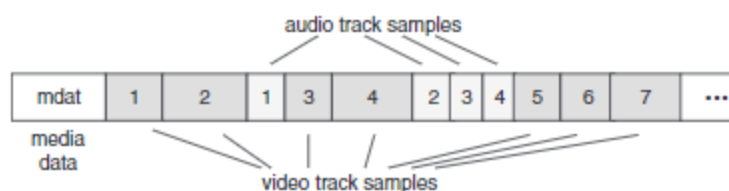


Figure 8.32 ISO Media File

Id. at 247. As noted, the coded audio data and the coded picture data can alternatively be time-multiplexed.

97. The picture coding unit in the Netflix '878 Accused Products includes a block coding unit configured to code a block image to obtain coded block data, the block image being obtained by dividing a picture signal into plural blocks.

98. In the Netflix '878 Accused Products picture data is coded by an H.264-compliant encoder by dividing a picture signal into macroblocks, as shown in Figure 6-7:

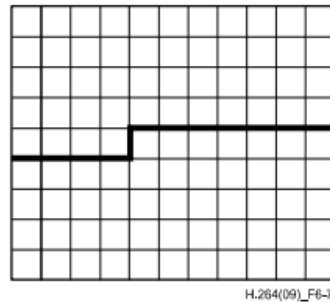


Figure 6-7 – A picture with 11 by 9 macroblocks that is partitioned into two slices

See ITU-T H.264, *Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of moving video, Advanced video coding for generic audiovisual services*, Section 6.3, p. 25 (03/2009), at Figure 6-7.

99. The macroblock consists of a 16 x 16 block of luma samples and two corresponding blocks of chroma samples. A macroblock can be further partitioned for inter-prediction forming segmentations for motion representation as small as a block of 4 x 4 luma samples.⁴³

100. In the Netflix '878 Accused Products a residual block image is generated from the block image of the respective blocks and a predictive block image is obtained by intra-picture prediction or inter-picture prediction by an H.264-compliant encoder. For example, Figure 6.6 below shows a picture signal to be coded, with the macroblock being coded highlighted. The macroblock is predicted using neighboring, previously-encoded samples, as shown in Figure 6.7; because this prediction looks only to the other macroblocks of the same picture, this is called intra-picture prediction. The predicted macroblock is shown in Figure 6.7. Figure 6.8 shows the prediction (Figure 6.7) subtracted from the original (Figure 6.6), which is called the residual.

⁴³ See generally ITU-T H.264, *Series H: Audio Visual and Multimedia Systems, Infrastructure of Audiovisual Services – Coding of Moving Video, Advanced Video Coding for Generic Audio Visual Services* [hereinafter “H.264 Standard”], Section 0.6.3, p. 5 (09/2019).



Figure 6.6 QCIF frame with highlighted macroblock



Figure 6.7 Predicted luma frame formed using H.264 intra prediction



Figure 6.8 Residual after subtracting intra prediction

Richardson at 141-143.

101. The Netflix '878 Accused Products can use intra- and inter- picture coding on the macroblocks of a picture signal. Inter-picture coding predicts the value of the macroblock using temporal statistical dependencies between different pictures. Both types of prediction can be used by the H.264-compliant encoder to calculate the residual.

102. In the Netflix '878 Accused Products the H.264-compliant encoder codes, on a block basis, coefficients obtained by performing orthogonal transformation and quantization on the residual block image. H.264 specifies an *entropy_coding_mode* flag that dictates the entropy encoding algorithm used to encode the picture data. When this flag is set to “0” the residual block data is coded using a CAVLC scheme.⁴⁴ In the case of Netflix’s AVC-Main and AVC-High profile H.264 encodes, the *entropy_coding_mode* flag is set to “0”:⁴⁵

⁴⁴ See H.264 Standard, Section 7.4.2.2, pp. 81-82 (09/2019).

⁴⁵ Picture Parameter Set analysis of two different Netflix content programs encoded using AVC High profile (first two images) and AVC Main profile (second two images).

MP4Box.js / ISOBMFF Box Structure Viewer (see other demos)

Star Fork Watch

Client-side tool: files are processed locally in your browser and not uploaded to any server

File: File Choose File -23 lp man first segme...throttled 300kbps.hex Load Loading Completed!

File Overview Box View Sample View Item View Segment View

Movie Info

File Size / Bitrate	0 bytes / NaN kbps
Duration / Timescale	0/24000 (0:00:00.000)
Brands (major/compatible)	iso6,iso6,dsm5,msix,dash
MIME	video/mp4; codecs="avc1.64001e"; profiles="iso6,dsm5,msix,dash"
Progressive	true
Fragmented	true
MPEG-4 IOD	false
Fragmented duration	992992 - 0:00:41.374
Creation / Modification Dates	06/13/2021 03:03 AM / 06/13/2021 03:03 AM

Video track(s) info

ID	References	Alternate Group	Presentation Duration	Presentation Edits	Duration	Timescale	Timelines Shift	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Width	Height	Lay
1	0	0	0 - 0:00:00.000	Presentation Track Time Speed Duration 2002 - 1 0:00:00.000 0:00:00.083	0 - 0:00:00.000	24000	23	688.84	avc1.64001e	und	-	768	432	0	

Fields Va

Box Tree

ftyp

free

moov

mvhd

trak

tkhd

edts

mdia

mdhd

hdr

minf

vmhd

dinf

stbl

stsd

avc1

avcc

stts

stsc

stsz

stco

mver

sidc

FIELDS

VALUES

sps.0.cpb_removal_delay_length...	24
sps.0.dpb_output_delay_length...	1
sps.0.time_offset_length	2
sps.0.low_delay_hrd_flag	false
sps.0.pic_struct_present_flag	false
sps.0.bitstream_restriction_flag	false
numOfPictureParameterSets	1
pps.0.pictureParameterSetLength	5
pps.0.pic_parameter_set_id	2
pps.0.seq_parameter_set_id	1
pps.0.entropy_coding_mode_flag	false
pps.0.pic_order_present_flag	false
pps.0.num_slice_groups_minus1	0
pps.0.num_ref_idx_l0_active_mi...	0
pps.0.num_ref_idx_l1_active_mi...	0
pps.0.weighted_pred_flag	false
pps.0.weighted_bipred_idc	2

MP4Box.js / ISOBMFF Box Structure Viewer (see other demos)

Star Fork Watch

Client-side tool: files are processed locally in your browser and not uploaded to any server

File: File Choose File -373 Archive 81 - 1st segment.hex Load Loading Completed!

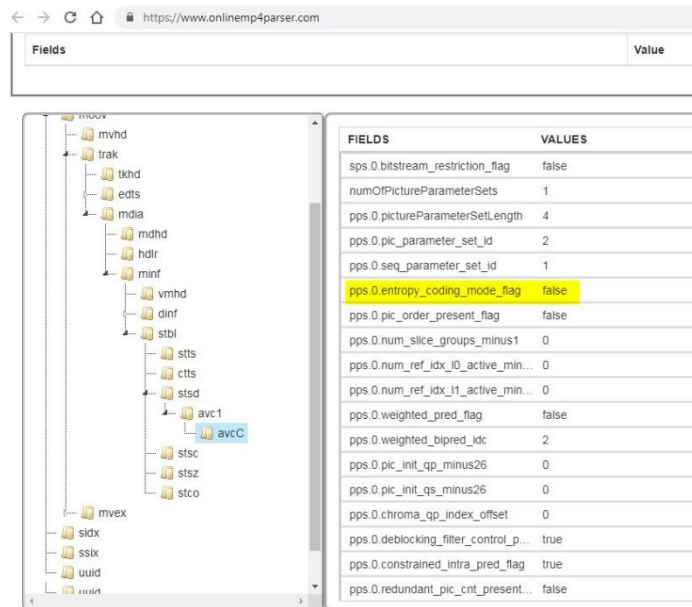
File Overview Box View Sample View Item View Segment View

Movie Info

File Size / Bitrate	0 bytes / NaN kbps
Duration / Timescale	0/10000000 (0:00:00.000)
Brands (major/compatible)	iso2,iso2,dash,iso6
MIME	video/mp4; codecs="avc1.4d401e"; profiles="iso2,dash,iso6"
Progressive	true
Fragmented	true
MPEG-4 IOD	false
Fragmented duration	898397500 - 0:01:29.839
Creation / Modification Dates	11/23/2021 05:37 PM / 11/23/2021 05:37 PM

Video track(s) info

ID	References	Alternate Group	Presentation Duration	Presentation Edits	Duration	Timescale	Timelines Shift	Number of Samples	Bitrate (kbps)	Codec	Language	Kind	Width	Height	Lay
2	0	0	0 - 0:00:00.000	Presentation Track Time Speed Duration 898397500 - 417083 - 1 0:01:29.839 0:00:00.041	0 - 0:00:00.000	10000000	0	NaN	avc1.4d401e	und	-	720	480	0	



103. In such case, the resulting prediction residual is split into 4x4 blocks, and transformation and quantization are applied. An integer transform is applied to the residual, outputting a set of coefficient weighting values. This process is shown in the figures below:

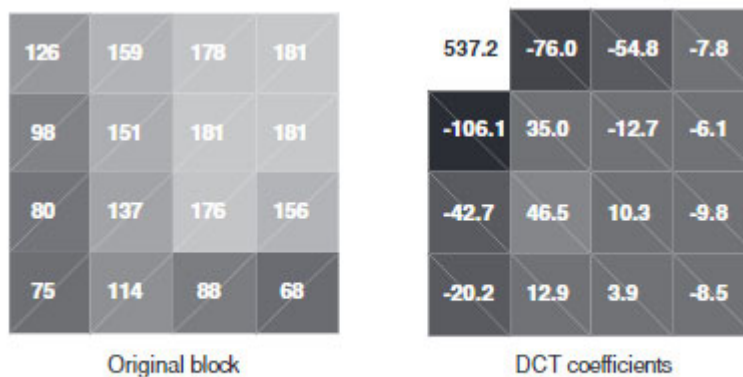


Figure 3.31 Close-up of 4 × 4 block; DCT coefficients

Ex. 17, Richardson, at 47.

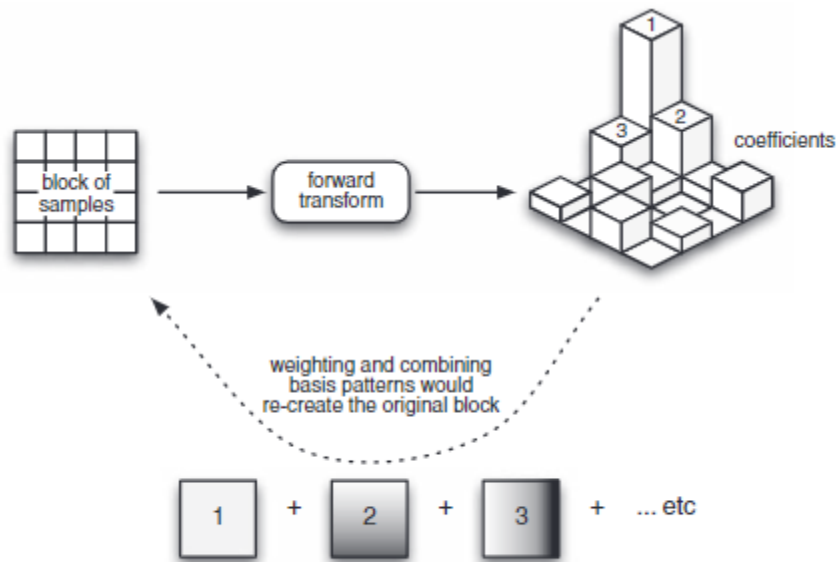


Figure 4.11 Forward transform

Ex. 17, Richardson, at 88.

104. The coefficients are then quantized, meaning that insignificant coefficient values are rounded down (for example, to zero), while a small number of significant, non-zero coefficients are retained. The quantization step is shown in Figure 4.12:

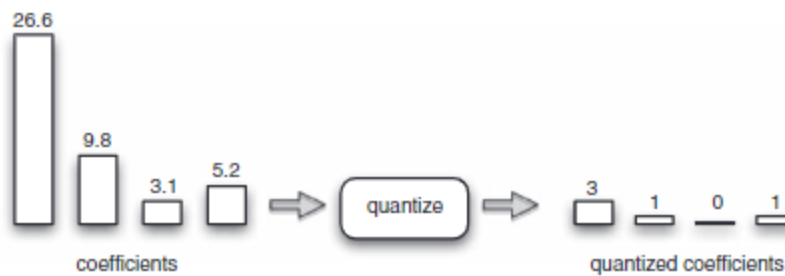


Figure 4.12 Quantization example

Ex. 17, Richardson, at 88.

105. The encoding process is shown in Figure 4.4:

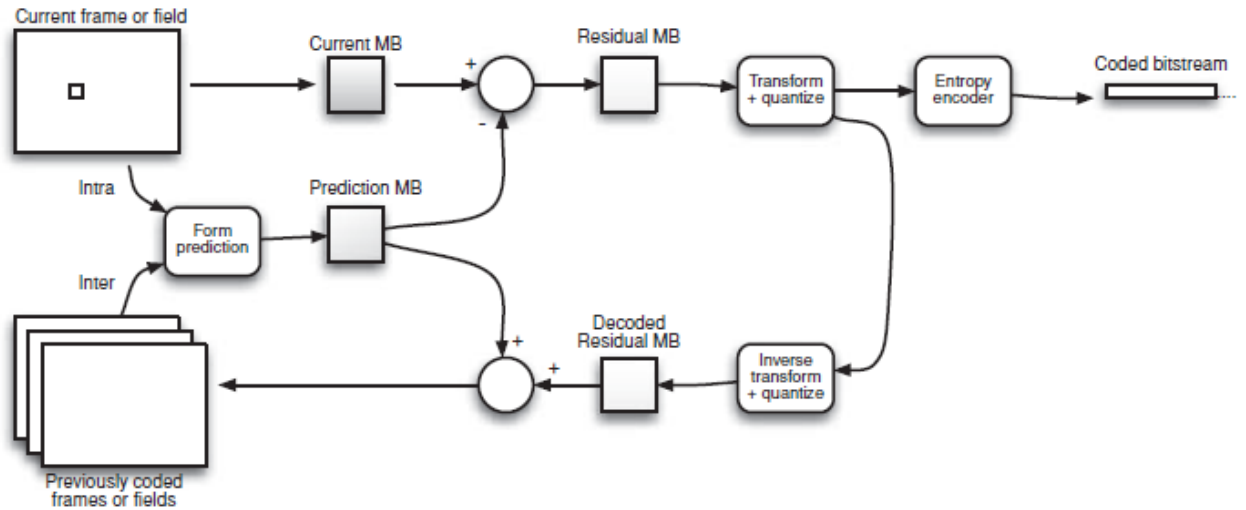


Figure 4.4 Typical H.264 encoder

Ex. 17, Richardson, at 84.

106. In the Netflix '878 Accused Products, the block coding unit includes a coefficient number coding unit configured to code a total number of non-zero coefficients included in a current block to be coded, where each of the non-zero coefficients have a value other than "0". In H.264 CAVLC encoding generally, which the Netflix '878 Accused Products support and perform, the total number of non-zero coefficients included in a current block to be coded is derived in order to generate a H.264-compliant bitstream.

107. To this end, the coefficient number coding unit in the Netflix '878 Accused Products includes a determining unit configured to determine a predictive value for the number of non-zero coefficients included in the current block based on the number of non-zero coefficients included in a block located on a periphery of the current block. Since CAVLC is a context-adaptive variable length coding technique, the number of non-zero coefficients in neighboring blocks is correlated as part of the entropy coding process. An H.264-compliant encoder uses previously-processed macroblocks to help encode the currently-processed macroblock. The number of non-zero coefficients in previously-processed blocks on the periphery—including the blocks to the left

and above the current macroblock—are used to predict the number of non-zero coefficients in the current block. This use of the blocks of the periphery is shown in Figure 6-14:

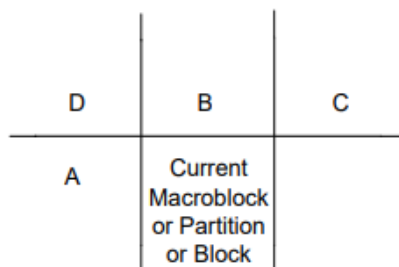


Figure 6-14 – Determination of the neighbouring macroblock, blocks, and partitions (informative)

H.264 Standard, at 33.

108. The H.264-compliant encoder in the Netflix '878 Accused Products obtains the number of non-zero coefficients in the left and above blocks to set the variable nC , the prediction of the current macroblock's number of non-zero coefficients based on the neighboring macroblocks' number of non-zero coefficients.⁴⁶

109. The coefficient number coding unit in the Netflix '878 Accused Products also includes a selecting unit configured to select a variable length code table based on the determined predictive value. The H.264-compliant encoder in the Netflix '878 Accused Products uses the predictive value nC to select one of six variable length coding tables specified in Table 9-5 of the H.264 standard in order to generate a H.264-compliant bitstream.⁴⁷

110. This selection of a variable length code table based on a determined predictive value is exemplified in Figure 7.19:

⁴⁶ See H.264 Standard, Section 9.2.1, pp. 214-216 (09/2019).

⁴⁷ See H.264 Standard, Section 9.2.1 and Table 9-5, pp. 214-218 (09/2019).

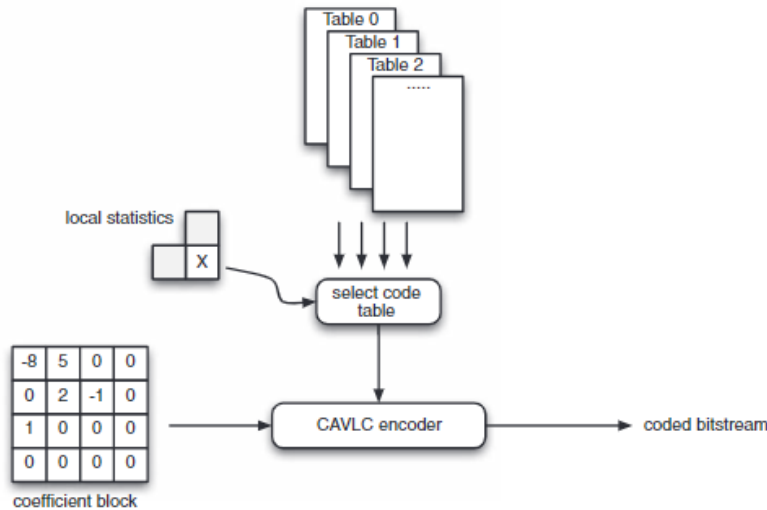


Figure 7.19 CAVLC encoder overview

Richardson at 211.

111. The coefficient number coding unit in the Netflix '878 Accused Products further includes a variable length coding unit configured to perform variable length coding on the number of the non-zero coefficients included in the current block, by using the selected variable length code table. Figure 7.19, *supra*, shows a CAVLC coding unit. The unit uses the selected variable length code table to perform variable length coding on the syntax element *coeff_token* representing the number of non-zero coefficients of the current macroblock to generate a H.264-compliant bitstream.⁴⁸

112. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.⁴⁹ Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.⁵⁰ Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory

⁴⁸ See generally H.264 Standard, Section 9.2.1 and Table 9-5, pp. 214-218 (09/2019).

⁴⁹ See Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

⁵⁰ See Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

call on March 19, 2021.⁵¹ Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.⁵² After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a “standstill” for an indefinite period of time.⁵³ Given that nearly two years had already passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs’ communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the ’878 Patent, identifying the Netflix products that infringe the ’878 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the ’878 Patent.⁵⁴

113. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the ’878 Patent, has begun investigating the ’878 Patent and its infringement of the Netflix ’878 Accused Products. Netflix has been given further notice of the ’878 Patent and its infringement of the ’878 Patent through the filing of the Complaint (ECF No. 1) on February 23, 2022, and the filing of this First Amended Complaint. On information and belief, Netflix is either knowingly infringing the ’878 Patent or is willfully blind to its infringement — including by ignoring VideoLabs’ communications and/or using a disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs’ patents — and continues to act in wanton disregard of VideoLabs’ patent rights.

114. Despite becoming aware of or willfully blinding itself to its infringement of the ’878 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing

⁵¹ See Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

⁵² See *id.* at 8-10.

⁵³ See *id.* at 2-7.

⁵⁴ See *id.* at 1-2.

activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '878 Accused Products. On information and belief, Netflix has made no attempts to design around the '878 Patent or otherwise stop its infringing behavior.

115. Netflix's infringement of the '878 Patent therefore has been and remains willful.

116. Netflix also indirectly infringes the '878 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '878 Accused Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the '878 Patent since at least February 22, 2022.

117. On information and belief, Netflix has actively induced the infringement of the '878 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '878 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '878 Patent in violation of 35 U.S.C. § 271(a).

118. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '878 Patent by, for example, requiring that third parties provide content to Netflix that has been encoded using H.264, working with third parties to ensure that this requirement is met, and otherwise actively encouraging, supporting, and assisting such infringing acts.⁵⁵

119. On information and belief, Netflix contributorily infringes the '878 Patent under 35 U.S.C. § 271(c) by importing, selling, and/or offering to sell within the United States the Netflix

⁵⁵ See, e.g., Ex. 51, <https://partnerhelp.netflixstudios.com/hc/en-us/articles/360057627253-VFX-Media-Review-Delivery-Specifications>; Ex. 52, https://drive.google.com/file/d/1_bwU4s4NaF7Y-07_NHwQqsL5L6zPVMPN/view; Ex. 53, https://fta-media.konsole-labs.com/pdf_files/Netflix_Full_Technical_Spec_%26_Operators_Manual_v8.1.pdf.

'878 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the hardware and/or software for encoding content with H.264 using CAVLC is material, has no insubstantial non-infringing uses, and is known by Netflix to be especially made or adapted for use in a manner that infringes the '878 Patent.

COUNT II
INFRINGEMENT OF U.S. PATENT NO. 7,440,559

120. VideoLabs incorporates by reference the foregoing paragraphs of this First Amended Complaint as if fully set forth herein.

121. VL IP is the assignee and lawful owner of all right, title, and interest in and to the '559 Patent. The '559 Patent is valid and enforceable.

122. On information and belief, Netflix has infringed and continues to infringe the '559 Patent in violation of 35 U.S.C. § 271, either literally or through the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States products and/or methods that practice at least claims 1 and 13 of the '559 Patent, including with respect to the Netflix '559 Accused Products.

123. On information and belief, Netflix uses the Netflix '559 Accused Products for its own business purposes. In addition, Netflix regularly conducts research, testing, and troubleshooting of the Netflix '559 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix's subsidiaries) use the Netflix '559 Accused Products.

124. On information and belief, Netflix's infringement through certain instrumentalities of the Netflix streaming service, described below, are exemplary of all of Netflix's infringement with respect to all the Netflix '559 Accused Products.

125. The Netflix '559 Accused Products directly infringe at least claim 1 of the '559 Patent, for example, by providing one or more servers configured to receive, from a remote terminal accessing the Netflix streaming and content delivery services, content status information, including content bookmark information, and configured to receive server status information regarding available content from a source of content, and in response to the content status information, instructing the terminal to perform actions, such as download control instructions, based on the terminal status information and the server status information, to control the flow of content to the terminal.

126. The Netflix '559 Accused Products meet every limitation of at least claim 1 of the '559 Patent, which recites:

1. An apparatus comprising:
a processor configured to receive, from a terminal located remote from the apparatus, a content status including terminal status information, and configured to receive server status information regarding a source of content, wherein the server status information comprises a listing of at least one piece of content available from the source, wherein the processor is configured to send, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information, and
wherein the at least one piece of content available from the source, and the content for which the processor is configured to control the flow, comprise multimedia content.

127. The architecture of the Netflix streaming and content delivery services generally constitutes (among other things) a back-end control plane, which controls and manages the myriad of microservices employed by Netflix to manage and deliver streaming content to Netflix subscribers, a terminal application, accessible directly from a subscriber's browser, or via the Netflix app installed on one or more of a subscriber's connected devices, and a content distribution

network in which implementations of Netflix's content library are stored, distributed and managed, and subsequently delivered, to Netflix subscribers on demand.

128. The Netflix back-end control plane incorporates a processor configured to receive, from a terminal located remote from the apparatus, a content status including terminal status information. During a streaming session with a remote terminal accessing the Netflix streaming and content delivery services using, for example, a browser by accessing www.netflix.com, or via the Netflix application installed on the subscriber's connected device, the remote terminal routinely provides content status messages to Netflix servers in the back-end control plane. One such content status message provided in this regard is a content status message ("cl2") provided by the remote terminal to the Netflix domain www.netflix.com, via an API message (highlighted in blue below). This content status message is provided to the Netflix servers in the back-end control plane via a HTTPS POST method and includes, in its message payload, terminal status information about the content the Netflix subscriber is watching or has recently finished watching (red highlighting below).

105353800-1056988937o=1&v=82&e=1645683111&t=Z1jyRw...8r%10%04_qU%18%1E%00R%04%1Cg%05%07...	GET	200	ipv4-c073-sjc005-ix.1.ocanflxvideo...
cl2	POST	200	www.netflix.com
105698894-1059750867o=1&v=82&e=1645683111&t=Z1jyRw...%10%04_qU%18%1E%00R%04%1Cg%05%07...	GET	200	ipv4-c075-sjc005-ix.1.ocanflxvideo...
28585535-288473847o=1&v=82&e=1645683111&t=LuTUs82...%10%04_qU%18%1E%00R%04%1Cg%05%07...	GET	200	ipv4-c012-sjc005-ix.1.ocanflxvideo...
ws	GET	101	push.prod.netflix.com
cl2	POST	200	www.netflix.com
cl2	POST	200	www.netflix.com
blobhttps://www.netflix.com/8d903673-e913-41b8-8065-078ba8bc5e87	GET	200	www.netflix.com
router?reqAttempt=1&reqName=release/license&client...rsion=98.0.4758.102&osname=windows&osversion=1...	POST	200	www.netflix.com
1?reqAttempt=1&reqName=logblob&clienttype=akira&ui...rsion=98.0.4758.102&osname=windows&osversion...	POST	200	www.netflix.com
1?reqAttempt=1&reqName=events/stop&clienttype=akir...rsion=98.0.4758.102&osname=windows&osversion...	POST	200	www.netflix.com
adtech_iframe_target_05.html?data=%7B%22membership...22%3Afalse%2C%22referrer%22%3A%22homeScre...	GET	200	ae.nflximg.net
pathEvaluator?avif=false&webp=true&drmSystem=widev...iginal_path=%2Fshakti%2Fv3f540b6%2FpathEvaluator	POST	200	www.netflix.com
0-40957o=1&v=82&e=1645683083&t=kiENbVb8Uunb5_6f12...54xmzJaUfh-EZcV7ozLIGNhu6KE4dKuneE6XP...	GET	200	ipv4-c013-sjc005-ix.1.ocanflxvideo...
0-40957o=1&v=81&e=1645683083&t=GUUYm68clmNa6oh9oRXC...yduqZ5Gv1VmY05KG6Dvm1rYtoUqaXnueCZ...	GET	200	ipv4-c083-sjc005-ix.1.ocanflxvideo...

```

X Headers Payload Preview Response Initiator Timing Cookies
▶ 1175322639132537: {level: 1, muting: false, type: ["Volume"], id: 1175322639132537}
▶ 1175323076938168: {playerId: 1175322362394568, mediaOffset: 1456162, type: ["MediaOffset"], id: 1175323076938168}
▶ 1175355360731979: {playerId: 1175355245368350, sharedUid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854",...}
▶ 1175355413153934: {level: 1, muting: false, type: ["Volume"], id: 1175355413153934}
▶ 1175355562312364: {mediaOffset: 2455, playerId: 1175354479222568, type: ["MediaOffset"], id: 1175355562312364}
▶ 1175355718699006: {playerId: 1175355245368350, mediaOffset: 1504696, type: ["MediaOffset"], id: 1175355718699006}
▶ 1175375260109464: {...}
▶ 1175375294819797: {time: 1645641572753, trackingInfo: {trackId: 14170286, videoId: 80103688, row: 7, rank: 0,...},...}
▶ 1175375508069244: {type: ["MediaPlayer"], id: 1175375508069244}
▼ 1175375751126441: {playerId: 1175375508069244, sharedUid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854",...}
  id: 1175375751126441
  playerId: 1175375508069244
  sharedUid: "watch-ebb33031-f64b-4a56-b7c4-8d074d065854"
  ▼ trackingInfo: {videoId: 80103688, trackId: 14170286}
    trackId: 14170286
    videoId: 80103688
  ▼ type: ["Xid"]
    0: "Xid"
  xid: "164564157278930780"
▶ 1175375807792956: {level: 1, muting: false, type: ["Volume"], id: 1175375807792956}
▶ 1175375979528660: {level: 1, muting: false, type: ["Volume"], id: 1175375979528660}
▶ 1175376170250698: {mediaOffset: 2698, playerId: 1175374746692948, type: ["MediaOffset"], id: 1175376170250698}
▶ 1175376358086557: {inputTime: 1645641604993, type: ["UserInputTime"], id: 1175376358086557}
▼ 1175376426934557: {playerId: 1175375508069244, mediaOffset: 1536313, type: ["MediaOffset"], id: 1175376426934557}
  id: 1175376426934557
  mediaOffset: 1536313
  playerId: 1175375508069244
  ▼ type: ["MediaOffset"]
    0: "MediaOffset"
  ▶ reverseDeltas: [{"1",...}]
  type: "CompactConsolidatedLoggingEnvelope"
  version: 2

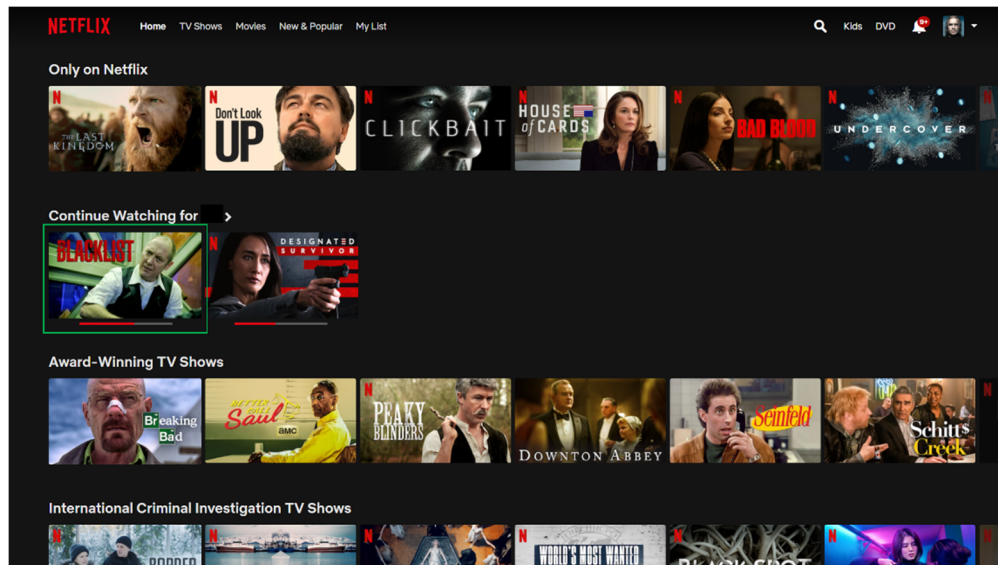
```

129. The content status message received by the Netflix servers in the back-end control plane from the remote terminal specifies, among other information, the view progress within the content the Netflix subscriber has recently watched, but not yet finished watching (in the above example, the content program is represented by videoID 80103688, which correlates to the Netflix show Blacklist: Season 3, Episode 15 – Drexel). For example, the terminal status information relating to the view progress in the above example is reported by the Netflix client player and includes the parameter *mediaOffset*, which represents the offset position within the video program the Netflix client player is/was streaming. This information is passed through to the Netflix back-end control plane and Netflix maintains this terminal status information in its back-end control plane and uses this information at least to manage the content bookmarking feature in its streaming and content delivery services. For example, Netflix updates the bookmarking information in near real-time based on this received information to reflect the bookmark change, if any, in the client GUI, as shown below:

```

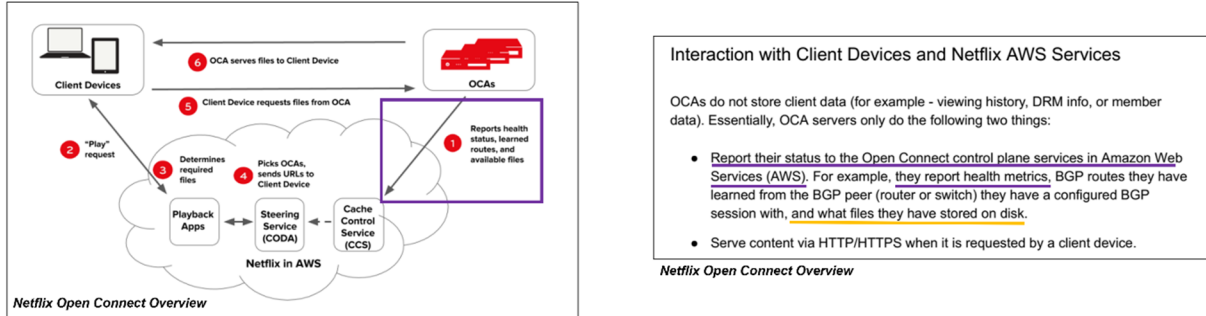
X Headers Payload Preview Response Initiator Timing Cookies
▼ {paths: [{"videos": 70281312, "current": "-"}],...}
▼ jsonGraph: {videos: {70281312: {current: {type: "ref", value: [{"videos": "80103688"}]}},...}
▼ videos: {70281312: {current: {type: "ref", value: [{"videos": "80103688"}]}},...}
▼ 70281312: {current: {type: "ref", value: [{"videos": "80103688"}]}},...}
▼ current: {type: "ref", value: [{"videos": "80103688"}]}
  type: "ref"
  value: [{"videos": "80103688"}]
    0: "videos"
    1: "80103688"
▼ 80103688: {bookmarkPosition: {type: "atom", value: 1536, $expires: -900000, $timestamp: 1645641615303},...}
  bookmarkPosition: {type: "atom", value: 1536, $expires: -900000, $timestamp: 1645641615303}
  $expires: -900000
  $timestamp: 1645641615303
  type: "atom"
  value: 1536
  interactiveBookmark: {type: "atom", value: null}
  runtime: {type: "atom", value: 2588}
  summary: {type: "atom",...}
  title: {type: "atom", value: "Drexel (No. 113)"}
  paths: [{"videos": 70281312, "current": "-"}]

```



130. The Netflix back-end control plane also incorporates a processor configured to receive server status information regarding a source of content, wherein the server status information comprises a listing of at least one piece of content available from the source. For example, the Netflix back-end control plane manages status information received from the origin servers hosting Netflix content within its content delivery network. This status information includes health metrics available from the source (*e.g.*, Open Connect Appliances (or “OCAs”)) (*see e.g.*, purple references below) and a listing of content streams of different encoded formats

available from the source (*see e.g.*, yellow reference below).⁵⁶



131. The Netflix back-end control plane processor is further configured to send, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information. For example, when the Netflix subscriber requests to stream the content program from the Netflix streaming and content delivery services, the Netflix back-end control plane sends to the subscriber's terminal a response to the content status. The response is sent, for example, as payload information in a JSON container to the subscriber's terminal:

```
{
  "id": "164564349052688260",
  "serverTime": 1645643491387,
  "result": {
    "from": "playapi",
    "id": "164564349052688260",
    "result": {
      "audioShortcuts": [
        {
          "id": "d21d77a-b945-8b67-8e13-eb6647fd7bd",
          "audioTrackId": "A:2:1:2:en:1:0:,-,-"
        }
      ],
      "audioTracks": [
        {
          "trackType": "PRIMARY",
          "rawTrackType": "primary",
          "channels": "2.0",
          "surroundFormatLabel": "2.0,-,-"
        }
      ],
      "auxiliaryManifests": [
        {
          "bookmark": -1
        }
      ],
      "conResponseData": {
        "pbid": "6.KYgw8I4uQRR89QHQCF35QepEC6-xleuqCw6Z5nVA",
        "clientId": "90.42.50.172",
        "qpid": null,
        "drmContextId": "default",
        "drmType": "widevine",
        "drmVersion": 25,
        "duration": 58000,
        "eligibleAssetsMap": {
          "expiration": 1645686691374,
          "hasClearProfile": false,
          "hasClearStreams": true,
          "hasDrmProfile": true,
          "hasDrmStreams": false,
          "isBranching": false,
          "isSupplemental": true
        },
        "links": [
          {
            "license": {
              "id": {
                "events": {
                  "key": "1-33651-high",
                  "rank": 1,
                  "weight": 160,
                  "level": 1
                }
              },
              "manifestExpirationDuration": 28799979,
              "maxRecommendedAudioRank": 0,
              "maxRecommendedTextRank": 1,
              "movieId": 81043687,
              "packageId": "1527827",
              "partiallyHydrated": false,
              "playbackContextId": "E3-RQFRAADLIP7eGdU77IurwL6Ipsc-y8SK-cel3CPzrKHsKey_2788558hLIEvRISBAH358Cqjhw4yV5Q7CvQm_uKD8otcuru9Ffw0p3CgwhutUaE8KLLHq2u4tfnu3eV5qnv2u0",
              "recommendedMedia": {
                "videoTrackId": "V:2:1:2:default:1:none:1:1",
                "audioTrackId": "A:2:1:2:en:1:0:,-,-"
              },
              "servers": [
                {
                  "id": 44067,
                  "key": "1-33651-high",
                  "longname": false,
                  "name": "c006.sjc005.ix.nflxvideo.net",
                  "rank": 1,
                  "type": "standard"
                }
              ],
              "steeringAdditionalInfo": {
                "additionalGroupNames": [
                  {
                    "steeringId": "6.KYgw8I4uQRR89QHQCF35QepEC6-xleuqCw6Z5nVA",
                    "type": "standard"
                  }
                ]
              },
              "timeTextTracks": [
                {
                  "type": "timedtext",
                  "trackType": "PRIMARY",
                  "rawTrackType": "subtitles",
                  "id": "none",
                  "type": "standard"
                }
              ],
              "trickPlays": [
                {
                  "id": "518003068",
                  "interval": 10,
                  "pixelsAspectX": 1,
                  "pixelsAspectY": 1,
                  "width": 240,
                  "height": 122,
                  "type": "standard"
                }
              ],
              "urlExpirationDuration": 43199987,
              "videoTracks": [
                {
                  "trackType": "PRIMARY",
                  "rawTrackType": "primary",
                  "channels": "2.0",
                  "surroundFormatLabel": "2.0,-,-"
                }
              ]
            }
          }
        ]
      }
    }
  }
}
```

132. The response includes information to instruct the terminal to perform one or more

⁵⁶ See e.g., Ex. 49, “Open Connect Overview”, NETFLIX, <https://openconnect.netflix.com/Open-Connect-Overview.pdf>.

actions to thereby control the flow of content to the terminal based upon the terminal status information and the server status information. For example, the response includes a federated listing of the different origin servers from where the client can obtain the program stream (red highlighting below), as well as the segment index references and base URLs for respective audio and video tracks (green highlighting below) for the content program to enable the Netflix client to obtain the individual segments of the content program stream from the ideal source(s) depending on bandwidth constraints, etc.

```
X Headers Payload Preview Response Initiator Timing Cookies
hasClearProfile: false
hasClearStreams: true
hasDrmProfile: true
hasDrmStreams: false
isBranching: false
isSupplemental: true
▶ links: {license: {,}, ldl: {,}, events: {,}}
▶ locations: [{key: "1-33651-high", rank: 1, weight: 160, level: 1},...]
manifestExpirationDuration: 28799979
maxRecommendedAudioRank: 0
maxRecommendedTextRank: 1
movieId: 81063687
packageId: "1527827"
partiallyHydrated: false
playbackContextId: "E3-Q0FRAAELEPZeGAUG77IurkL6Ipsc--y85K-cel3CPzXHz5XKey_2708558nLZehriGNAV358eCqzyhwp4yv5Q7CvQMu_KDsabtcuru9f9ko6p7CvhdUaEbKLLHq2utq4tfnuJevSqrv2xqj"
▶ recommendedMedia: {videoTrackId: "Vi2i1j2;defaultj-1;nonej-1j", audioTrackId: "Ai2i1j2;enj1j0;"}
▼ servers: [{id: 44067, key: "1-33651-high", lowgrade: false, name: "c006.sjc005.ix.nflxvideo.net", rank: 1,-},...]
▶ 0: {id: 44067, key: "1-33651-high", lowgrade: false, name: "c006.sjc005.ix.nflxvideo.net", rank: 1,-}
▶ 1: {id: 44103, key: "1-33651-high", lowgrade: false, name: "c022.sjc005.ix.nflxvideo.net", rank: 2,-}
▶ 2: {id: 44109, key: "1-33651-high", lowgrade: false, name: "c028.sjc005.ix.nflxvideo.net", rank: 3,-}
▶ 3: {id: 63499, key: "1-33651-high", lowgrade: false, name: "c088.sjc005.ix.nflxvideo.net", rank: 5,-}
▶ 4: {id: 63532, key: "1-33651-high", lowgrade: false, name: "c094.sjc005.ix.nflxvideo.net", rank: 6,-}
▶ 5: {id: 53074, key: "1-33651-high", lowgrade: false, name: "c080.sjc005.ix.nflxvideo.net", rank: 7,-}
▶ 6: {id: 53069, key: "1-33651-high", lowgrade: false, name: "c075.sjc005.ix.nflxvideo.net", rank: 8,-}
▶ 7: {id: 48029, key: "1-33651-high", lowgrade: false, name: "c031.sjc005.ix.nflxvideo.net", rank: 10,-}
▶ 8: {id: 44093, key: "1-33651-high", lowgrade: false, name: "c012.sjc005.ix.nflxvideo.net", rank: 11,-}
▶ 9: {id: 63494, key: "1-33651-high", lowgrade: false, name: "c083.sjc005.ix.nflxvideo.net", rank: 12,-}
▶ 10: {id: 63527, key: "1-33651-high", lowgrade: false, name: "c089.sjc005.ix.nflxvideo.net", rank: 13,-}
▶ 11: {id: 68519, key: "2-33651-high", lowgrade: false, name: "c229.sjc002.ix.nflxvideo.net", rank: 14,-}
▶ 12: {id: 68446, key: "2-33651-high", lowgrade: false, name: "c235.sjc002.ix.nflxvideo.net", rank: 15,-}
▶ 13: {id: 50068, key: "2-33651-high", lowgrade: false, name: "c176.sjc002.ix.nflxvideo.net", rank: 16,-}
▶ 14: {id: 68450, key: "2-33651-high", lowgrade: false, name: "c239.sjc002.ix.nflxvideo.net", rank: 18,-}
▶ steeringAdditionalInfo: {additionalGroupNames: [], steeringId: "6.XYgw8I4MDQNR98QmXQCF35qepE6-XleUgCw6Z5NVA,..."}
▶ timedTextTracks: [{type: "timedText", trackType: "PRIMARY", rawTrackType: "subtitles", id: "none",...}]
▶ trickplays: [{id: "518003068", interval: 10, pixelsAspectX: 1, pixelsAspectY: 1, width: 240, height: 122,-},...]
type: "standard"
urlExpirationDuration: 43199987
▼ video_tracks: [{trackType: "PRIMARY",...}]
▼ 0: {trackType: "PRIMARY",...}
dimensionsCount: 2
```

```
X Headers Payload Preview Response Initiator Timing Cookies
▼ audio_tracks: [{trackType: "PRIMARY", rawTrackType: "primary", channels: "2.0", surroundFormatLabel: "2.0",...}]
▼ 0: {trackType: "PRIMARY", rawTrackType: "primary", channels: "2.0", surroundFormatLabel: "2.0",...}
channels: "2.0"
channelFormat: "2.0"
codeName: null
defaultText: null
disallowedSubtitleTracks: []
hydrated: true
id: "664e7c1232307c5072696d6172797c747275657c6e6f6e657c756e6d6e6f776e7c756e6d6e6f776e7c4175644696f7c756e6d6e6f776e7c7c6e6f6e657c38"
isNative: true
isNoneTrack: false
language: "en"
languageDescription: "English [Original]"
new_track_id: "Ai2i1j2;enj1j0;"
offTrackDisallowed: false
profile: "heaac-2-q-dash"
profileType: "AbstractDynamicEnum(name='AUDIO')"
rank: 0
rawTrackType: "primary"
stereo: false
▼ streams: [{trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 489110,-}]
▼ 0: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 489110,-}
audioKey: null
bitrate: 64
channels: "2.0"
channelFormat: "2.0"
content_profile: "heaac-2-dash"
downloadable_id: "1022456349"
isDrm: false
language: "en"
▶ moov: {offset: 241, size: 662}
new_stream_id: "1022456349"
▶ sidx: {offset: 909, size: 300}
size: 489110
▶ sidx: {offset: 1201, size: 364}
surroundFormatLabel: "2.0"
tags: []
trackType: "PRIMARY"
type: 0
▼ urls: [{cdn_id: 63499,-}, {cdn_id: 63494,-}, {cdn_id: 50068,-}]
▼ 0: {cdn_id: 63499,-}
cdn_id: 63499
url: "https://ip4-c088-sjc005-ix.1.cdn.nflxvideo.net/10-18w818e=164568669185-pv1t14uWj3ZvYao535uHLPSPF8okQv6sp5Ief5jPvXpKfE8b0vMLdgaVw61n7d3jU12515r7Tg8hVw6d"
...}
```

```

profileType: "AbstractDynamicEnum(name='VIDEO')
stereo: false
streams: [...]
w: {trackType: "PRIMARY", content_profile: "avi-main-l30-dash-cbcs-prk", bitrate: 76, peakBitrate: 99,...}
  bitrate: 76
  content_profile: "avi-main-l30-dash-cbcs-prk"
  crop_h: 270
  crop_w: 480
  crop_x: 0
  crop_y: 0
  dimensionsCount: 2
  dimensionsLabel: "2D"
  downloadable_id: "476157"
  drmHeaderId: ""
  framerate_scale: 1001
  framerate_value: 24000
  isDrm: false
  mov: {offset: 111, size: 690}
  new_stream_id: "000"
  peakBitrate: 99
  plv_h: 1
  plv_w: 1
  res_h: 270
  res_w: 480
  segmentVmf: []
  sidx: {offset: 601, size: 236}
  size: 561159
  ssi: {offset: 1037, size: 220}
  startByteOffset: 1257
  tags: ["EQUIV", "Ladder", "avi-hd-bitrate-capped", "HCCLEARAVI", "no-letterbox"]
  trackType: "PRIMARY"
  type: 1
  uris: [{cdn_id: 63532,...}, {cdn_id: 53069,...}, {cdn_id: 60446,...}]
    w: {cdn_id: 63532,...}
      cdn_id: 63532
      url: "https://ip04-c094-sj005-lx.1.cca.netflixvideo.net/fo18v81se16456060915t55facvtn0WLA_iUd0u51e0apHuxXZ18yXK53j2ERvVCKZM2G_rmo7Lac3mIfdRajHqgqivAf201V90N"
      s: {cdn_id: 53069,...}
        cdn_id: 53069
        vmf: 69
    z: {trackType: "PRIMARY", content_profile: "avi-main-l30-dash-cbcs-prk", bitrate: 107, peakBitrate: 130,...}
    z: {trackType: "PRIMARY", content_profile: "avi-main-l30-dash-cbcs-prk", bitrate: 140, peakBitrate: 179,...}
    z: {trackType: "PRIMARY", content_profile: "avi-main-l30-dash-cbcs-prk", bitrate: 185, peakBitrate: 235,...}
    z: {trackType: "PRIMARY", content_profile: "avi-main-l31-dash-cbcs-prk", bitrate: 321, peakBitrate: 406,...}

```

133. The response also includes the bookmark position within the content program at which the Netflix client player should initiate content streaming (blue highlighting below):

```

{,...}
  jsonGraph: {videos: {80103688: {current: {type: "ref", value: ["videos", "80103688"]},...},...}
    prePlayExperiences: {,...}
    videos: {80103688: {current: {type: "ref", value: ["videos", "80103688"]},...}
      80103688: {current: {type: "ref", value: ["videos", "80103688"]},...}
        availability: {type: "atom",...}
        bookmarkPosition: {type: "atom", value: 1538, $expires: -900000, $timestamp: 1645644934759}
          $expires: -900000
          $timestamp: 1645644934759
          type: "atom"
          value: 1536
        creditsOffset: {type: "atom", value: 2545}
        current: {type: "ref", value: ["videos", "80103688"]}
        preplay: {14170286: {type: "ref", value: ["prePlayExperiences", "80103688", "14170286"]}}
        summary: {type: "atom",...}
    paths: [{"videos", 80103688, "current", ["availability", "bookmarkPosition", "creditsOffset", "summary"]},...}

```

134. The Netflix application and/or the Netflix browser, under the programming control by Netflix, carries out the instructions by issuing a series of consecutive HTTP GET requests starting at the designated media offset point to the designated content source at specific time intervals thereby controlling the flow of multimedia content to the terminal.

94504207-94628996to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
94628997-94774992to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
metadata?movieid=80103688&imageFormat=webp&withSize=true&materialize=true&_e=1645644936296	GET	200	www.netflix.com
94774993-95005764to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
25473813-25736115to=1&v=82&e=1645668122&t=HgvMTppL...fkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c021-sjc005-ix.1.ocan.netflixvideo.net
95005765-95059304to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
95059305-95406992to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
95406993-95760405to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
25736116-25997839to=1&v=82&e=1645668122&t=HgvMTppL...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1C...	GET	200	ip4-c021-sjc005-ix.1.ocan.netflixvideo.net
95760406-95904384to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
95904385-96264560to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
96264561-96420413to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
96420414-96480423to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
96480424-96641959to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
96641960-96751916to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
25997840-26260397to=1&v=82&e=1645668122&t=HgvMTppL...fkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c021-sjc005-ix.1.ocan.netflixvideo.net
96751917-96908238to=1&v=82&e=1645668122&t=ik71bO4s...fkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
96908239-97020111to=1&v=82&e=1645668122&t=ik71bO4s...fkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97020112-97151800to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVk...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97151801-97403322to=1&v=82&e=1645668122&t=ik71bO4s...fkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj%78...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97403323-97490087to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj%7...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
26260398-26522069to=1&v=82&e=1645668122&t=HgvMTppL...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CV...	GET	200	ip4-c021-sjc005-ix.1.ocan.netflixvideo.net
97490088-97651445to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97651446-97656559to=1&v=82&e=1645668122&t=ik71bO4s...attfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj%7...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97656560-97876535to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net
97876536-98207681to=1&v=82&e=1645668122&t=ik71bO4s...Gzattfkdv(%0Aq%01pV%17Wn%16%0E%10F%04%1CVj...	GET	200	ip4-c081-sjc005-ix.1.ocan.netflixvideo.net

135. At least one piece of content available from the source, and the content for which the processor is configured to control the flow, comprise multimedia content. As noted, all the content available for streaming from the Netflix streaming and content delivery services is multimedia content.

136. On information and belief, the “Skip Intro,” “Skip Recap,” and “Autoplay”/“Next Episode” features operate substantially the same as described above, and thus infringe for substantially the same reasons.

137. The Netflix ’559 Accused Products also directly infringe at least claim 13 of the ’559 Patent, for example, by receiving, at a network entity, from a remote terminal accessing the Netflix streaming and content delivery services, content status information, including content download information, and in response to the content status information, sending instructions to the terminal to perform actions, such as delete and download control instructions, based on the terminal status information, to control the flow of multimedia content to the terminal.

138. The Netflix ’559 Accused Products meet every limitation of at least claim 13 of the ’559 Patent, which recites:

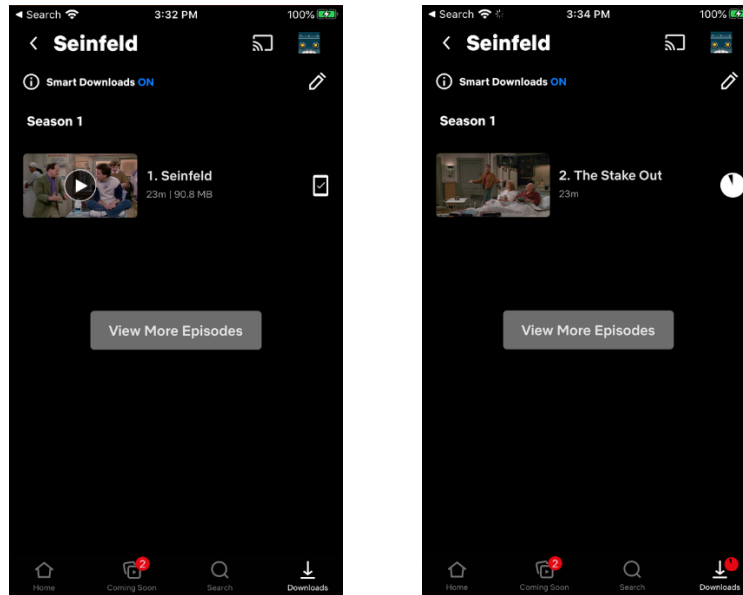
13. A method for controlling a flow of content, the method comprising:

receiving, at a network entity from a terminal located remote therefrom, a content status including terminal status information comprising a listing of at least one piece of content stored in a memory of the terminal; and sending, from the network entity to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information , and

wherein the at least one piece of content stored in the memory of the terminal, and the content for which the flow is controlled, comprise multimedia content.

139. The architecture of the Netflix streaming and content delivery services generally constitutes (among other things) a back-end control plane, which controls and manages the myriad of microservices employed by Netflix to manage and deliver streaming content to Netflix subscribers, a terminal application, accessible directly from a subscriber's browser, or via the Netflix app installed on one or more of a subscriber's connected devices, and a content distribution network in which implementations of Netflix's content library are stored, distributed and managed, and subsequently delivered, to Netflix subscribers on demand.

140. One important feature that Netflix offers its subscribers is content downloads. For example, Netflix subscribers can download Netflix programming content, such as episodes of Seinfeld, to their connected mobile device(s) and watch the programming off-line. Netflix remotely manages the remote client by controlling the deletion and addition of Netflix content programming on the client device.



141. Netflix refers to this important feature as “smart downloads”.⁵⁷ With smart downloads, when a subscriber is finished watching a downloaded episode, Smart Downloads will delete it, and then automatically download the next episode.

142. To accomplish this, Netflix servers receive, from a remote terminal, a content status including terminal status information. During a streaming session with a remote terminal accessing the Netflix streaming and content delivery services using, for example, the Netflix application installed on the subscriber’s connected device, the remote terminal routinely provides content status messages to Netflix servers in the back-end control plane. One such content status message provided in this regard is a content status message (“/iosui/user/13.38”) provided by the remote terminal to the Netflix domain ios.prod.ftl.netflix.com, via an API message [red highlighting below]. This content status message is provided to the Netflix servers in the back-end control plane via a HTTPS POST method and includes, in its message payload, terminal status information about downloaded content the Netflix subscriber is watching or has recently finished

⁵⁷ See Ex. 31, <https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1>.

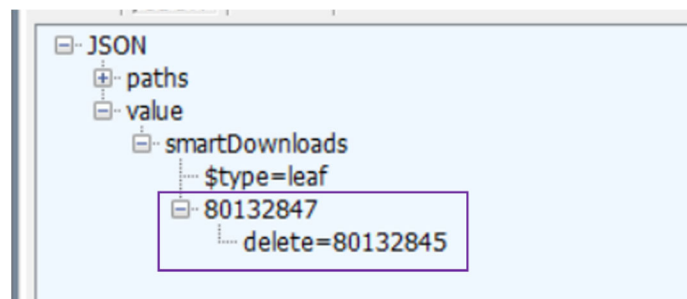
watching (for example, the first episode of season 1 of Seinfeld, which correlates to the program reference “80132845”) [green highlighting below].

687	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
688	CONN...	200	HTTP	Tunnel to	ios.prod.cloud.netflix.com:443
689	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
690	POST	200	HTTPS	ios.prod.ftl.netflix.com	/ms/playapi/ios/event
691	POST	200	HTTPS	ios.prod.cloud.netflix.com	/ms/playapi/ios/logblob
692	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/warmer/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&clientAppCanHandleFlo
693	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
694	CONN...	200	HTTP	Tunnel to	ios.prod.ftl.netflix.com:443
695	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
696	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&config=%7B%22kidsTop1
697	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
698	GET	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38?appInternalVersion=13.38.0&appVersion=13.38.0&config=%7B%22kidsTop1
699	POST	200	HTTPS	ios.prod.ftl.netflix.com	/iosui/user/13.38
700	POST	200	HTTPS	ios.prod.ftl.netflix.com	/nq/iosplatform/pbo_manifest/~1.0.0/router
701	CONN...	200	HTTP	Tunnel to	occ-0-2311-999-1.nflxso.net:443
702	GET	200	HTTPS	occ-0-2311-999-1.nflxso.net	/dmn/api/v6/K194eJsgW8DE2aQbaIdmCXGUP-Y/AAAAABQko7mlelSuuah9pyudBatqkQowXQv7r
703	CONN...	200	HTTP	Tunnel to	pv4-c321-nyc001-ix.1.ocn.nflxvideo.net:443
704	CONN...	200	HTTP	Tunnel to	pv4-c321-nyc001-ix.1.ocn.nflxvideo.net:443
705	CONN...	200	HTTP	Tunnel to	pv4-c002-avp001-penteledata-sp.1.ocn.nflxvideo.net:443
706	CONN...	200	HTTP	Tunnel to	pv4-c001-avp001-penteledata-sp.1.ocn.nflxvideo.net:443
707	CONN...	200	HTTP	Tunnel to	pv4-c321-nyc001-ix.1.ocn.nflxvideo.net:443
708	CONN...	200	HTTP	Tunnel to	pv4-c003-abe001-penteledata-sp.1.ocn.nflxvideo.net:443
709	CONN...	200	HTTP	Tunnel to	pv4-c003-avp001-penteledata-sp.1.ocn.nflxvideo.net:443
710	CONN...	200	HTTP	Tunnel to	pv4-c003-avp001-penteledata-sp.1.ocn.nflxvideo.net:443

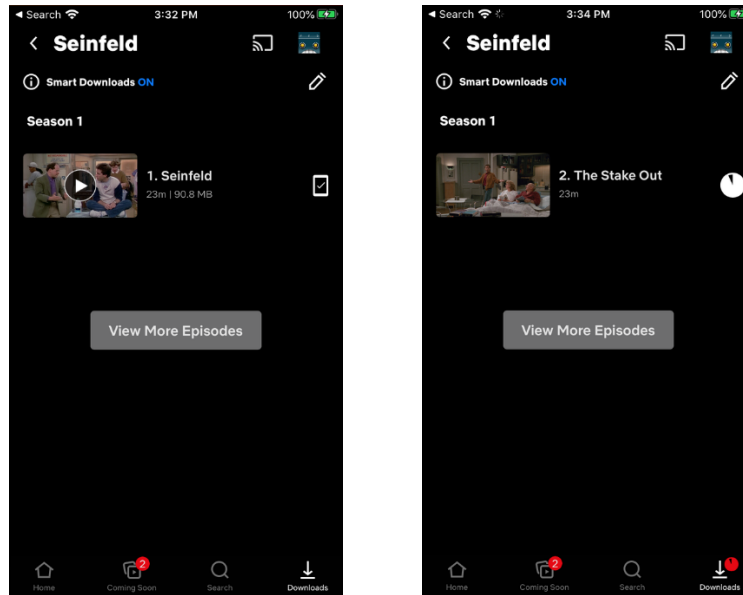
Body	
Name	Value
appInternalVersion	13.38.0
appVersion	13.38.0
callPath	["smartDownloads"]
config	{"kidsTopTenListEnabled": "true", "seasonRenewalPostPlayEnable
device_type	NFAPPL-02-
esn	NFAPPL-02-IPHONE8=1-540D4F0040F36ED4DB7F00768D66C03
idiom	phone
iosVersion	13.6.1
isTablet	false
isTop10KidsEnabled	true
languages	en-DE
locale	en-DE
maxDeviceWidth	375
method	call
model	saget
modelType	IPHONE8-1
odpAware	true
param	{"currentDownloads": {"80132845": {"bookmark": 1396}}}
pathFormat	graph
pixelDensity	2.0
progressive	false
responseFormat	json

143. The content status message received by the Netflix servers in the back-end control plane from the remote terminal specifies, among other information, the view progress within the content the Netflix subscriber has recently watched locally on their connected device. For example, the terminal status information relating to the view progress in the above example is specified by an API schema defined by Netflix and includes the parameter *bookmark*. Netflix maintains this terminal status information in its back-end control plane and uses this information at least to manage the smart download feature in its streaming and content delivery services.

144. The Netflix back-end control plane sends, to the terminal, a response to the content status that instructs the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information. For example, when the content status indicates that the Netflix subscriber has completed watching the locally stored content program, the Netflix back-end control plane sends to the subscriber's terminal a response to the content status [purple highlighting below]. The response is sent, for example, as payload information in a JSON container to the subscriber's terminal that includes information to instruct the terminal to perform one or more actions to thereby control the flow of content to the terminal based upon the terminal status information.



145. The Netflix application, under the programming control by Netflix, carries out the instructions by flagging the episode ("80132845") for deletion from the local memory of the client and initiate downloading of the next episode ("80132847") in the series to the client. The Netflix client automatically executes the instructions accordingly to update the Smart Downloads on the client device.



146. At least one piece of content stored in the memory of the terminal, and the content for which the flow is controlled, comprise multimedia content. As noted, all the content available for streaming from the Netflix streaming and content delivery services is multimedia content.

147. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.⁵⁸ Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.⁵⁹ Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.⁶⁰ Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.⁶¹ After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.⁶² Given that nearly two years had already

⁵⁸ See Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

⁵⁹ See Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

⁶⁰ See Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

⁶¹ See *id.* at 8-10.

⁶² See *id.* at 2-7.

passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the '559 Patent, identifying the Netflix products that infringe the '559 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the '559 Patent.⁶³

148. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the '559 Patent, has begun investigating the '559 Patent and its infringement of the Netflix '559 Accused Products. Netflix has been given further notice of the '559 Patent and its infringement of the '559 Patent through the filing of the Complaint (ECF No. 1) on February 23, 2022, and the filing of this First Amended Complaint. On information and belief, Netflix is either knowingly infringing the '559 Patent or is willfully blind to its infringement — including by ignoring VideoLabs' communications and/or using a disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

149. Despite becoming aware of or willfully blinding itself to its infringement of the '559 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '559 Accused Products. On information and belief, Netflix has made no attempts to design around the '559 Patent or otherwise stop its infringing behavior.

150. Netflix's infringement of the '559 Patent therefore has been and remains willful.

151. Netflix also indirectly infringes the '559 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '559 Accused

⁶³ See *id.* at 1-2.

Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the '559 Patent since at least February 22, 2022.

152. On information and belief, Netflix has actively induced the infringement of the '559 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '559 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '559 Patent in violation of 35 U.S.C. § 271(a).

153. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '559 Patent by, for example, marketing the Netflix '559 Accused Products and infringing functionalities to consumers; working with consumers to implement, install and/or operate the Netflix '559 Accused Products and infringing functionalities; fully supporting and managing consumers' continuing use of the Netflix '559 Accused Products and infringing functionalities; and providing technical assistance to consumers during their continued use of the '559 Accused Products and infringing functionalities.⁶⁴

154. For example, Netflix induces third parties to infringe the '559 Patent by encouraging them to install and operate the Netflix streaming service, which alone and/or in combination with the third parties' devices constitutes infringement of the '559 Patent. Netflix advertises and promotes its Netflix streaming service on its website and in various app stores such

⁶⁴ See e.g., Ex. 54, *Netflix Help Center*, <https://help.netflix.com/en/> (last accessed Feb. 15, 2022); Ex. 35, *How to Autoplay the Next Episode*, NETFLIX, <https://help.netflix.com/en/node/121518>; Ex. 31, *Downloading on Netflix Just Got Smarter*, NETFLIX, <https://about.netflix.com/en/news/downloading-on-netflix-just-got-smarter-1> (last visited Feb. 17, 2022); Ex. 32, *How to Skip Intros on TV Shows*, NETFLIX, <https://help.netflix.com/en/node/63402> (last visited Feb. 17, 2022); Ex. 55, *How to Use "Download Next Episode"*, NETFLIX, <https://help.netflix.com/en/node/101262> (last visited Feb. 21, 2022).

as Apple's app store and Android's app store in connection with the Netflix mobile application that can be installed on consumers' respective connected iOS and Android devices (as well as others), and encourages consumers to configure and operate their mobile and computer devices in an infringing manner.⁶⁵ In response, consumers acquire, configure, and operate the Netflix streaming service in an infringing manner.

155. On information and belief, Netflix contributorily infringes the '559 Patent under 35 U.S.C. § 271(c) by importing, selling, and/or offering to sell within the United States the Netflix '559 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the Netflix streaming application, and the code for sending, receiving, and/or processing status information, are material, has no insubstantial non-infringing uses, and are known by Netflix to be especially made or adapted for use in a manner that infringes the '559 Patent.

COUNT III
INFRINGEMENT OF U.S. PATENT NO. 7,233,790

156. VideoLabs incorporates by reference the foregoing paragraphs of this First Amended Complaint as if fully set forth herein.

157. VL is the assignee and lawful owner of all right, title, and interest in and to the '790 Patent. The '790 Patent is valid and enforceable.

158. On information and belief, Netflix has infringed and continues to infringe the '790 Patent in violation of 35 U.S.C. § 271(a), either literally or through the doctrine of equivalents, by using methods and/or taking steps that practice at least claim 2 of the '790 Patent, including with respect to the Netflix '790 Accused Products.

⁶⁵ See e.g., Ex. 54, *Netflix Help Center*, <https://help.netflix.com/en/> (last accessed Feb. 15, 2022).

159. On information and belief, Netflix uses the '790 Accused Products for its own business purposes. In addition, Netflix regularly conducts testing and troubleshooting of the Netflix '790 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix's subsidiaries) use the '790 Accused Products.

160. On information and belief, Netflix's infringement through certain instrumentalities of the Netflix streaming service, described below, are exemplary of all of Netflix's infringement with respect to all the Netflix '790 Accused Products.

161. The Netflix '790 Accused Products directly infringe at least claim 2 of the '790 Patent, for example, by providing subscribers access to content on their respective connected devices through the receipt and storage of different implementations of Netflix content programming, each corresponding to a different set of device capabilities, in its content management system, and the selection of a portion of the Netflix content programming catalog which includes references to each content program implementation based on capabilities of the subscriber's device.

162. The Netflix '790 Accused Products meet every limitation of at least claim 2 of the '790 Patent, which recites:

2. A method of providing access to digital content for use on wireless communication devices, the method comprising:

receiving and storing in a server system a plurality of items of digital content to be made available for use in wireless communication devices used by a plurality of wireless services subscribers, including receiving and storing a plurality of different implementations of at least one of the items of digital content, where each implementation of any given item of digital content corresponds to a different set of device capabilities;

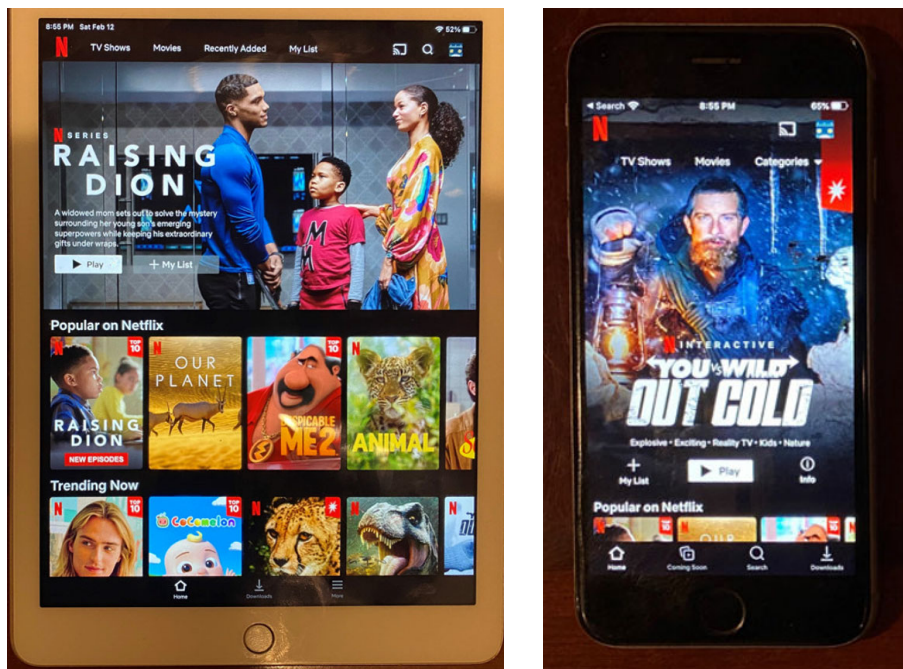
operating the server system to maintain a product catalog containing a description of the items of digital content, wherein the product catalog includes, in association with each item of digital content, a reference to each implementation of said item of digital content;

receiving a request from a wireless device used by one of the subscribers;

in response to the request, selecting a portion of the product catalog to be presented to the subscriber, based on device capabilities of the wireless device used by the subscriber; and

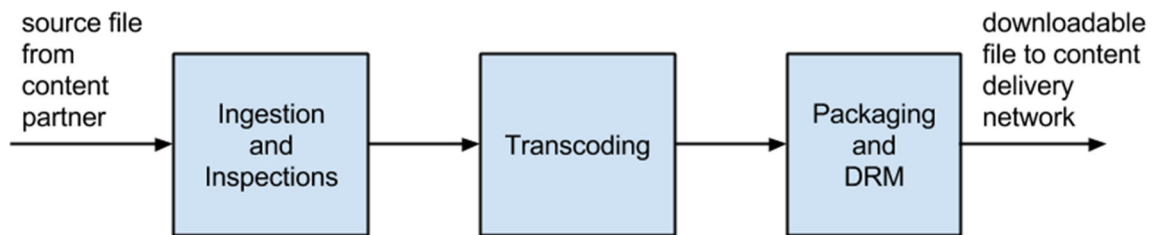
presenting the selected portion of the product catalog to the subscriber via a wireless network, such that the selected portion, as presented to the subscriber, provides only a single description of each item of digital content in said portion, regardless of the number of implementations of each said item.

163. The Netflix streaming service, when accessed by a Netflix subscriber, provides access to content for use on subscribers' wireless communication devices. Netflix subscribers access the Netflix streaming service, for example, via the Netflix app on their connected device or by accessing the website from their browser at www.netflix.com:



164. Netflix receives and stores a plurality of items of content to be made available for use in wireless communication devices used by a plurality of wireless services subscribers. Specifically, Netflix controls the ingestion and management of a comprehensive catalog of

programming content that is made available to mobile device users via the Netflix app or by accessing the Netflix OTT streaming service at www.netflix.com. Original format content programming is received by Netflix in its content processing pipeline.⁶⁶



165. For every Interoperability Master Format Package (IMP) delivered to Netflix for a specific title, Netflix performs transfer/delivery validations on the contents of the IMP before updating the content catalog in its asset management system as the content is ingested into the Netflix content management system.⁶⁷ Netflix receives IMF package contents via its Backlot portal.⁶⁸ Netflix also receives production related materials from Netflix production content creators via its Content Hub asset storage, management, collaboration, and delivery platform.⁶⁹

166. As part of its content ingestion workflow Netflix transcodes the raw mezzanine content into multiple different implementation formats, each corresponding to a different set of device capabilities. This video encoding pipeline runs on AWS EC2 spot instances to seamlessly scale up additional encoders when more titles need to be processed into the system. Specifically, Netflix employs a parallel encoding process to ingest high quality video sources and generate video

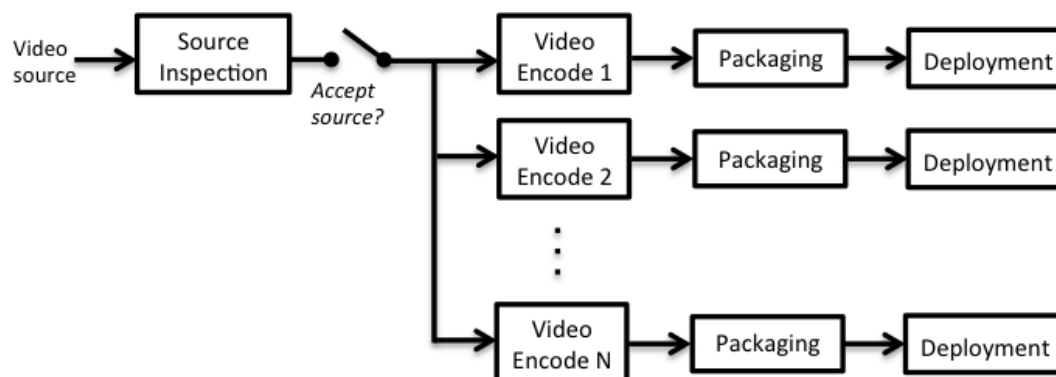
⁶⁶ See Ex. 56, “The Netflix IMF Workflow”, NETFLIX, <https://netflixtechblog.com/the-netflix-imf-workflow-f45dd72ed700>, (last retrieved, Feb. 21, 2022).

⁶⁷ *Id.*

⁶⁸ See e.g., Ex. 57, <https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004926427-Backlot-Overview-for-Content-Partners> (last retrieved, Feb. 21, 2022).

⁶⁹ See e.g., Ex. 58, <https://partnerhelp.netflixstudios.com/hc/en-us/articles/360000509487-Content-Hub-Overview-Introduction> (last retrieved, Feb. 21, 2022).

encodes of various codec profiles, at multiple quality representations per profile.⁷⁰ Exemplary codec profiles supported by Netflix transcodes include VP9, AV1, H.264/AVC Main, H.264/AVC Mobile-Hi, and HEVC.⁷¹



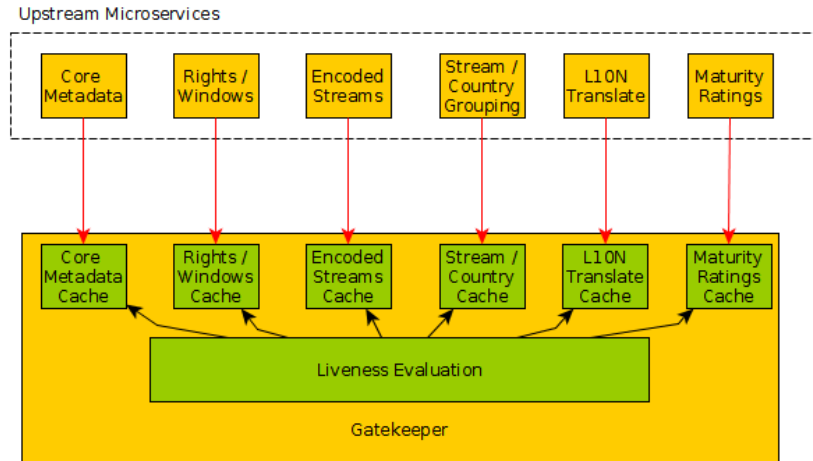
167. Netflix operates its content management services to maintain a product catalog containing a description of the items of digital content. Netflix maintains an enormous, ever-changing content library. Recently, its US content library alone exceeded 6,000 titles.⁷² Netflix leverages its Gatekeeper service to evaluate the “liveness” of the titles in its catalog – a title is not available to a subscriber until Gatekeeper approves it.⁷³

⁷⁰ See e.g., Ex. 6, “High Quality Video Encoding at Scale”, NETFLIX, <https://netflixtechblog.com/high-quality-video-encoding-at-scale-d159db052746> (last retrieved, Feb. 21, 2022)

⁷¹ *Id.*; see also Ex. 59, “Netflix Now Streaming AV1 on Android”, NETFLIX, <https://netflixtechblog.com/netflix-now-streaming-av1-on-android-d5264a515202> (last retrieved, Feb. 21, 2022); Ex. 26, “More Efficient Mobile Encodes for Netflix Downloads”, NETFLIX, <https://netflixtechblog.com/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909> (last retrieved Feb. 21, 2022).

⁷² See e.g., Ex. 60, <https://www.whats-on-netflix.com/news/netflix-library-by-the-numbers-2021/> (last retrieved, Feb. 21, 2022).

⁷³ See Ex. 61, “Re-Architecting the Video Gatekeeper”, NETFLIX, <https://netflixtechblog.com/re-architecting-the-video-gatekeeper-f7b0ac2f6b00> (last retrieved, Feb. 21, 2022).



168. In association with each title in the content catalog, Netflix maintains a reference association to each individual program encode so that only certain encodes are chosen for a particular user device. For example, Netflix employs a video metadata service that includes links to content program images, trailers, encoded video files, subtitles, individual episodes and seasons (if applicable), title, genre, synopsis, cast, maturity ratings, etc.⁷⁴

169. Netflix receives a request from a wireless device used by one of the subscribers, such as when the Netflix subscriber launches the Netflix app on their connected mobile device to access the Netflix service. When Netflix subscribers access the streaming service from their respective connected device, an HTTP GET request is sent from the subscriber's device to Netflix:

⁷⁴ See e.g., Ex. 62, <https://netflixtechblog.com/object-cache-for-scaling-video-metadata-management-c3c17830983e>.

```

Request Headers
GET /iosui/warmer/14.23?
appInternalVersion=14.23.0&clientAppCanHandleFloatingCharacterRow=true&clientAppMaxRows=
h=6&config=781:2:useCDSCGGalleryEnabled%22%3A%22true%22%2C%22fastLaughsRowV2Enabled%22%3A%22false
%22true%22%2C%22codeSearchForceGameGridEnabled%22%3A%22false%22%2C%22sharksEnabled%22%3A%22true
%22false%22%2C%22billboardEnabled%22%3A%22true%22%2C%22bypassContextualAssetsEnabled%22%3A%22false
Client
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US;q=1
User-Agent: Argo/14.23.0 (Pad; iOS 14.8.1; Scale/2.00)
Cookies
Netflixdid=v%3D2%26ct%3D3DBQAOAEEBEG6fUp_VTIDjIVZBellyvoCB4jQ0UgEBYXhFDe0EU6FmM9__w3HuzC
nFvdd=6QFmAAEBEPH__2kaWTDYGFqmirjV-JgJpRJC64CHPZeAdixFzwWrlDj-64FU68h3F8Mw10GZ7e2
SecureNetflixdid=v%3D2%26mac%3DAQEAEQABABtsUmfGh9OexGhVge6ogC_3eZpAKDno4.%26dt%3D:
Miscellaneous
X-Netflix-Argo.abTests: 895:2|1023:1|1062:0|2143:1|2144:1|2973:2|3285:2|3287:2|5987:3|6059:2|6220:5|6
X-Netflix-Argo.NFNSM: 7
X-Netflix-Argo.translated: true
X-Netflix.client.appVersion: 14.23.0
X-Netflix.client.ftl.esn: NFAPPL-01-IPAD6=11-F08BFFA5F8DA43653EA15C5B0134AAD073786A9E7E0286E3F9
X-Netflix.client.idiom: pad
X-Netflix.client.iOSVersion: 14.8.1
X-Netflix.client.type: argo
X-Netflix.Request.Attempt: 1
X-Netflix.Request.Client.Context: ("appState": "foreground")
X-Netflix.request.client.supportsGames: true
X-Netflix.request.client.user.guid: R6QMUHSEFDEKE4TG7LZTFPL4Y
X-Netflix.Request.Routing: ("path": "/hq/mobile/hqios/~14.23.0/warmer", "control_tag": "iosui_argo_warmer")
X-Netflix.request.toplevel.uid: 35EA9828-E88A-4E37-98B0-CB067A2FA4F7
X-Netflix.tracing.d.userActionId: 7276558D-ABC5-4517-A511-99AC965925D6
Transport
Connection: keep-alive
Host: ios-prod.ftl.netflix.com

```

170. Associated with the request are various capabilities of the subscriber's device, including, for example, the maximum number of rows that can be displayed on the device ("clientAppMaxRows"); the maximum number of content items that can be shown in each row ("clientAppMaxTitlesPerRow"); the type of device the subscriber is using ("idiom"); the version of operating system the device is using ("iOSVersion"); and a flag indicating whether the device is a tablet or mobile device ("isTablet"). Representative images from an iPad (left) and an iPhone (right) accessing the Netflix service are shown below:

QueryString	Name	Value
	appInternalVersion	14.23.0
	appVersion	14.23.0
	clientAppCanHandleFloating	true
	clientAppMaxRows	40
	clientAppMaxTitlesPerRow	75
	clientAppViewPortWidth	6
	config	["UseCDSCGGalleryEnabled": "true", "FastLaughsRowV2Enabled": "false", "seasonR
	device_type	NFAPPL-01-
	esn	NFAPPL-01-IPAD6=11-F08BFFA5F8DA43653EA15C5B0134AAD073786A9E7E0286E3F9
	fragment.colomolimitX	15
	fragment.colomolimitY	14
	hydratool.level	fragment
	idiom	pad
	iOSVersion	14.8.1
	isTablet	true
	languages	en-DE
	locale	en-DE
	maxDeviceWidth	1024
	method	get
	model	hall
	modelType	IPAD6-11
	odpAware	true
	pathFormat	graph
	pixelDensity	2.0
	progressive	false
	responseFormat	json
	type	kolomo-eth
	warmerHasCategories	1

QueryString	Name	Value
	appInternalVersion	13.38.0
	appVersion	13.38.0
	clientAppCanHandleFloating	true
	clientAppMaxRows	20
	clientAppMaxTitlesPerRow	75
	clientAppViewPortWidth	3
	config	["UseCDSCGGalleryEnabled": "true", "seasonRenewalPostPlayEnabled": "true", "syno
	device_type	NFAPPL-02-
	esn	NFAPPL-02-IPHONE8=1-540D4F0D40F36ED4087F00760D66C0B30276183DA1
	fragment.colomolimitX	11
	fragment.colomolimitY	5
	fragment.colomoOffsetY	13
	hydratool.level	fragment
	idiom	phone
	iOSVersion	13.6.1
	isTablet	false
	isTop10IndexEnabled	true
	languages	en-DE
	locale	en-DE
	kolomoId	3e952571-dd2e-4168-87c8-db770d5ee877_ROOT
	maxDeviceWidth	375
	method	get
	model	saget
	modelType	IPHONE8-1
	odpAware	true
	pathFormat	graph
	pixelDensity	2.0
	progressive	false
	responseFormat	json
	type	kolomo-eth
	warmerHasCategories	1

[illegible]

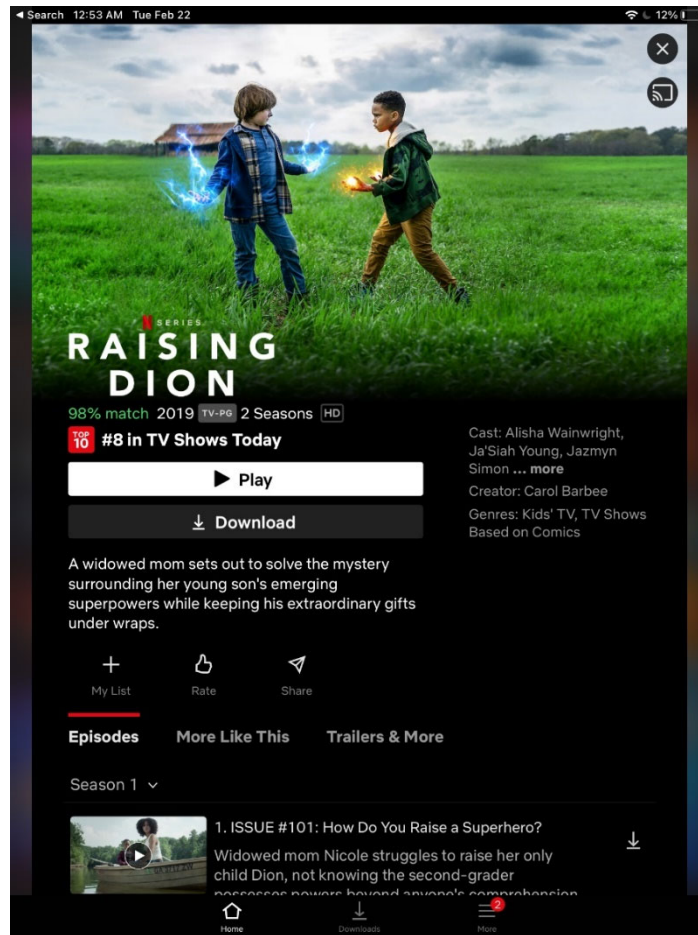
67

a streaming session as a function of its content personalization microservice.⁷⁵ In particular, according to Netflix, each device has different hardware capabilities that can limit the number of videos or rows displayed at any one time and how big the whole page can be. As such, the page generation process must be aware of the constraints of the device for which it is creating the page, including the number of rows, the minimum and maximum length of a row, the size of the visible portion of the page, and whether or not certain rows are required or are not applicable to a certain device.⁷⁶

173. The portion of the product catalog as presented provides only a single description of each item of digital content, regardless of the number of implementations of each item. For example, although each digital program in the selected portion of the product catalog has multiple associated different bitrate and format encodes, as discussed previously, in presenting the selected portion of the product catalog to the Netflix subscriber, Netflix provides only a single description of each content program to the subscriber.

⁷⁵ See Ex. 63, “Learning a Personalized Homepage”, NETFLIX, <https://netflixtechblog.com/learning-a-personalized-homepage-aa8ec670359a> (last retrieved, Feb. 21, 2022).

⁷⁶ *Id.*



In contrast, the manifest listing for the program Raising Dion, for example, indicates at least eight different video encoding implementations (first image below) and three different audio encoding implementations (second image below) for the content program:

```
streams: [...]  
  > 0: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 232, peakBitrate: 259,...}  
  > 1: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 373, peakBitrate: 411,...}  
  > 2: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 559, peakBitrate: 632,...}  
  > 3: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 747, peakBitrate: 831,...}  
  > 4: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 1046, peakBitrate: 1171,...}  
  > 5: {trackType: "PRIMARY", content_profile: "playready-h264mpl30-dash", bitrate: 1729, peakBitrate: 1957,...}  
  > 6: {trackType: "PRIMARY", content_profile: "playready-h264mpl31-dash", bitrate: 2332, peakBitrate: 2678,...}  
  > 7: {trackType: "PRIMARY", content_profile: "playready-h264mpl31-dash", bitrate: 2970, peakBitrate: 3387,...}  
  
  < streams: [{trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 615564,...},...]  
    > 0: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 64, size: 615564,...}  
    > 1: {trackType: "PRIMARY", content_profile: "heaac-2-dash", bitrate: 96, size: 908429,...}  
    > 2: {trackType: "PRIMARY", content_profile: "heaac-2hq-dash", bitrate: 128, size: 1201292,...}
```

174. On information and belief, to the extent applicable, VideoLabs has complied with 35 U.S.C. § 287(a) with respect to the '790 Patent.

175. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.⁷⁷ Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.⁷⁸ Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.⁷⁹ Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.⁸⁰ After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.⁸¹ Given that nearly two years had already passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications. On February 22, 2022, VideoLabs sent an email to Netflix apprising it of the '790 Patent, identifying the Netflix products that infringe the '790 Patent, and requesting that the parties discuss the terms for Netflix to take a license to the '790 Patent.⁸²

176. Netflix of course knows how its products operate, and on information and belief, upon receiving notice of the '790 Patent, has begun investigating the '790 Patent and its infringement of the Netflix '790 Accused Products. Netflix has been given further notice of the '790 Patent and its infringement of the '790 Patent through the filing of the Complaint (ECF No. 1) on February 23, 2022, and the filing of this First Amended Complaint. On information and belief, Netflix is either knowingly infringing the '790 Patent or is willfully blind to its infringement —

⁷⁷ See Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

⁷⁸ See Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

⁷⁹ See Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

⁸⁰ See *id.* at 8-10.

⁸¹ See *id.* at 2-7.

⁸² See *id.* at 1-2.

including by ignoring VideoLabs' communications and/or using a disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

177. Despite becoming aware of or willfully blinding itself to its infringement of the '790 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '790 Accused Products. On information and belief, Netflix has made no attempts to design around the '790 Patent or otherwise stop its infringing behavior.

178. Netflix's infringement of the '790 Patent therefore has been and remains willful.

179. Netflix also indirectly infringes the '790 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '790 Accused Products in this District and throughout the United States. As described above, on information and belief, Netflix has known about the '790 Patent since at least February 22, 2022.

180. On information and belief, Netflix has actively induced the infringement of the '790 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '790 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '790 Patent in violation of 35 U.S.C. § 271(a).

181. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '790 Patent by, for example, marketing the Netflix '790 Accused Products and infringing functionalities to consumers; working with consumers to implement, install and/or operate the Netflix '790 Accused Products and infringing functionalities; fully supporting and managing consumers' continuing use of the Netflix '790 Accused Products and infringing

functionalities; and providing technical assistance to consumers during their continued use of the '790 Accused Products and infringing functionalities.⁸³

182. For example, Netflix induces third parties to infringe the '790 Patent by encouraging them to install and then operate the Netflix streaming service, which constitutes infringement of the '790 Patent. Netflix advertises and promotes its Netflix streaming service on its website and in various app stores such as Apple's app store and Android's app store in connection with the Netflix mobile application that can be installed on consumers' respective connected iOS and Android devices (as well as others), and encourages consumers to configure and then operate their mobile and computer devices in an infringing manner.⁸⁴ In response, consumers acquire, configure, and then operate the Netflix streaming service in an infringing manner.

183. Netflix further induces third parties to infringe the '790 Patent by encouraging and instructing Netflix content partners to provide their original content programming to be ingested into the Netflix service through the Netflix Backlot portal and/or the Netflix Content Hub.⁸⁵ In response, Netflix content partners follow Netflix guidelines and instructions to submit their raw mezzanine content to Netflix for ingestion into the Netflix service.

184. On information and belief, Netflix contributorily infringes the '790 Patent under 35 U.S.C. § 271(c) by importing, selling, and/or offering to sell within the United States the Netflix

⁸³ See e.g., Ex. 54, *Netflix Help Center*, <https://help.netflix.com/en/> (last accessed Feb. 15, 2022).

⁸⁴ See e.g., Ex. 54, *Netflix Help Center*, <https://help.netflix.com/en/> (last accessed Feb. 15, 2022).

⁸⁵ See e.g., Ex. 57, <https://partnerhelp.netflixstudios.com/hc/en-us/articles/115004926427-Backlot-Overview-for-Content-Partners> (last retrieved, Feb. 21, 2022).

⁸⁵ See e.g., Ex. 58, <https://partnerhelp.netflixstudios.com/hc/en-us/articles/360000509487-Content-Hub-Overview-Introduction> (last retrieved, Feb. 21, 2022).

'790 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the content management and delivery techniques used by Netflix to provide subscribers access to Netflix content on their respective connected devices is material, has no insubstantial non-infringing uses, and is known by Netflix to be especially made or adapted for use that practices at least claim 2 of the '790 Patent with respect to the '790 Accused Products.

COUNT IV
INFRINGEMENT OF U.S. PATENT NO. 8,605,794

185. VideoLabs incorporates by reference the foregoing paragraphs of this First Amended Complaint as if fully set forth herein.

186. VL IP is the assignee and lawful owner of all right, title, and interest in and to the '794 Patent. The '794 Patent is valid and enforceable.

187. On information and belief, Netflix has infringed and continues to infringe the '794 Patent in violation of 35 U.S.C. § 271(a), either literally or through the doctrine of equivalents, by making, using, selling, offering for sale, and/or importing into the United States products and/or methods that practice at least claim 1 of the '794 Patent, including with respect to the Netflix '794 Accused Products.

188. On information and belief, Netflix uses the Netflix '794 Accused Products for its own business purposes. In addition, Netflix regularly conducts testing and troubleshooting of the Netflix '794 Accused Products. Further, VideoLabs is informed and believes companies related to Netflix (e.g., Netflix's subsidiaries) use the Netflix '794 Accused Products.

189. On information and belief, Netflix infringes through its use of Adaptive Bitrate (ABR) streaming, Common Media Application Format (CMAF), and ISO Base Media File Format

(ISOBMFF), as described below.”⁸⁶ This use is exemplary of all of Netflix’s infringement with respect to all the Netflix ’794 Accused Products.

190. The Netflix ’794 Accused Products directly infringe at least claim 1 of the ’794 Patent by, for example, synchronizing content-related first and second data segments of data files by sequentially outputting, by a device for synchronizing content-related data in such a way that each of the content-related data segments is output together on the basis of an assignment rule assigning each of the content-related second data segments to one of the content-related first data segments.

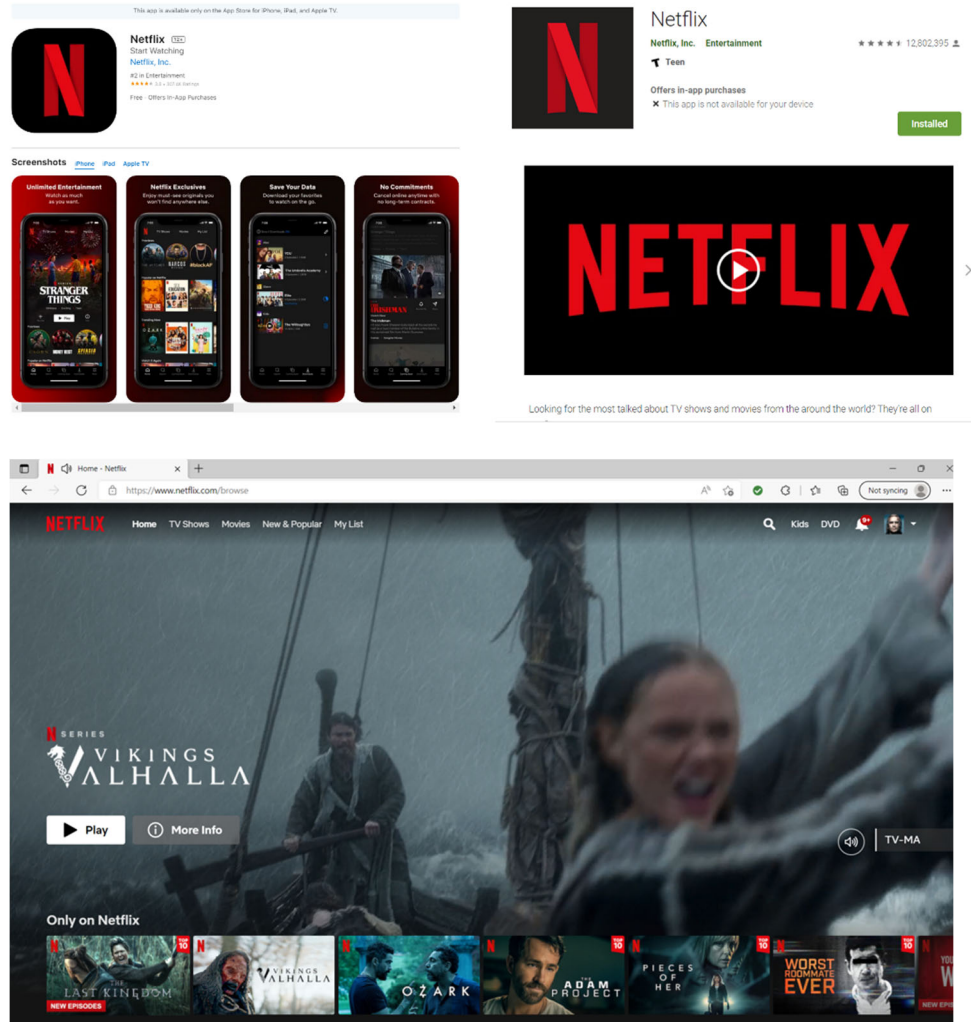
191. The Netflix ’794 Accused Products meet every limitation of claim 1 of the ’794 Patent, which recites:

1. A method for synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file, the method comprising:

sequentially outputting, by a device for synchronizing content-related data, the content-related first data segments and the content-related second data segments according to their chronological sequence in such a way that each of the content-related second data segments is output together with an associated one of the content-related first data segments on the basis of an assignment rule for assigning each one of the content-related second data segments to one of the content-related first data segments.

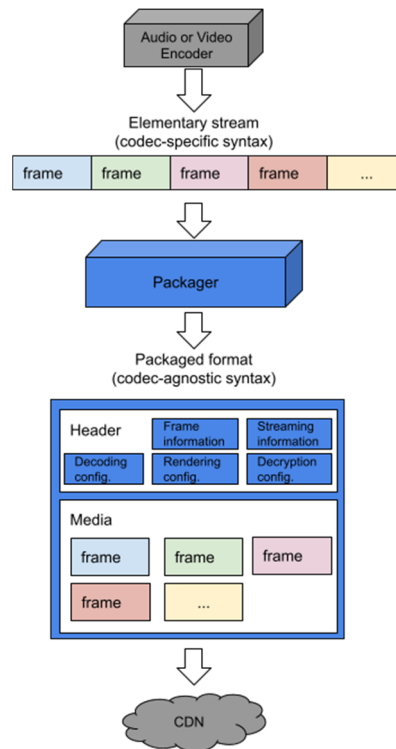
192. Subscribers to the Netflix streaming service can access the service from different connected devices, including PC/laptop devices and connected mobile devices, among others. To access the Netflix streaming service using a PC or laptop, for example, a Netflix subscriber would access the service by directing their web browser to www.netflix.com. Access on mobile devices, on the other hand, is provided through the Netflix app.

⁸⁶ See Ex. 72, *How Netflix brings safer and faster streaming experiences to the living room on crowded networks using TLS 1.3*, <https://netflixtechblog.com/how-netflix-brings-safer-and-faster-streaming-experience-to-the-living-room-on-crowded-networks-78b8de7f758c>.



193. The Netflix '794 Accused Products provide a method of synchronizing content-related first data segments of a first data file and content-related second data segments of a second data file. For example, Netflix has adopted adaptive bitrate streaming techniques, including ISOBMFF and CMAF to simplify delivery of its HTTP-based streaming media.⁸⁷

⁸⁷ See, e.g., Ex. 70, *Packaging award-winning shows with award-winning technology*, <https://netflixtechblog.com/packaging-award-winning-shows-with-award-winning-technology-c1010594ba39>.



To this end, Netflix relies on the ISO-BMFF for packaging its content streams.⁸⁸

194. ISO-BMFF contains structural and media data information principally for timed presentations of media data, such as audio, video, etc., but also provides support for untimed data, such as relational metadata. The basic structures of ISO-BMFF include: a logical structure (*i.e.*, a movie that in turn contains a set of time-parallel tracks), a time structure (*i.e.*, the tracks contain sequences of samples in time, and those sequences are mapped into the timeline of the overall movie by optional edit lists), and a physical structure (a series of defined “boxes” which have a size and a type).⁸⁹

195. ISO-BMFF utilizes a sequence of bytes in an initialization segment to contain all

⁸⁸ See *id.*; see also Ex. 73, W3C, ISO-BMFF Byte Stream Format specification, <https://www.w3.org/2013/12/byte-stream-format-registry/isobmff-byte-stream-format.html> (July 17, 2014).

⁸⁹ See ISO-BMFF Specification (Dec. 15, 2015) (herein “ISO-BMFF Specification”).

the information required to decode a sequence of media segments.⁹⁰ In Netflix's case, an ISOBMFF initialization segment is defined as a single *ftyp* (File Type) box followed by a single *moov* (Movie Header) box. Other ISOBMFF "boxes" used by Netflix in this regard are the *sidx* (Segment Index) box and, in some cases, *ssix* (Subsegment Index) box.⁹¹

196. Netflix encodes its video content using a technique called shot-based encoding.⁹² In essence, Netflix applies a shot-change detection algorithm to the content which returns a list of shots and their timestamps. These resulting shots are used as the basic encoding blocks for its video, instead of fixed-duration segments for the entire program. By optimizing in this way, Netflix obtains better overall coding efficiency. Since shot-based encoding results in variable duration encoded segments, Netflix will typically specify the chunked segment listing in the ISOBMFF *sidx* (segment index) box. The *sidx* box provides a compact index of one media stream. It contains a sequence of references to contiguous subsegments and their respective durations.⁹³ Netflix encodes its audio separately, often using ISOBMFF with substantially fixed duration segments. The complete video and audio segment lists for a program can be obtained by parsing the respective audio and video *sidx* boxes at the Netflix client as shown in the excerpts below (video above; audio below):⁹⁴

⁹⁰ See Ex. 74, W3C Working Draft, Media Source Extensions, <https://www.w3.org/TR/media-source/#init-segment> (February 17, 2022).

⁹¹ See Ex. 75, DASH Industry Forum, Guidelines for Implementation: DASH-IF Interoperability Points, § 5.1 (August 7, 2019).

⁹² See Ex. 76, *Dynamic optimizer — a perceptual video encoding optimization framework*, <https://netflixtechblog.com/dynamic-optimizer-a-perceptual-video-encoding-optimization-framework-e19f1e3a277f>.

⁹³ See ISOBMFF Specification, § 8.16.3.

⁹⁴ Excerpts dumped from the ISOBMFF *sidx* boxes contained in the initialization segment information of the respective video and audio segments for the Netflix program IP Man.

Video segments

Audio segments

78

seconds.

198. Netflix establishes a dynamic bit ladder recipe to transcode each content program available on its streaming service into multiple distinct bitrate/resolution pairs.⁹⁵ The different encoded versions of a program are maintained on Netflix's content delivery network. Specific versions are dynamically chosen by Netflix on-demand to serve a particular Netflix content program stream to a requesting subscriber.⁹⁶ To this end, the Netflix '794 Accused Products make use of a primary media data structure to initially convey to the Netflix client the relevant information necessary for the client to obtain the audio and video content streams from the specified Netflix CDN server(s) to initiate and maintain playback of the Netflix content program. The data structure is provided as a JSON object in response to a manifest request issued from the Netflix client.

199. The data structure file dynamically generated by Netflix specifies multiple variant bitstreams for audio and video, which indicates at least one first data file that contains media encoded at a particular bit rate, in a particular format, and at a particular resolution for media containing video (blue outlining below); and another which indicates at least one second data file that specifies media encoded at a particular bit rate, in a particular format, for media containing audio (red outlining below). Metadata in the JSON object may contain other information such as the peak bitrate and the content profile scheme used (yellow outlining below).

⁹⁵ See Ex. 77, NETFLIX TECH BLOG, Per-Title Encode Optimization, <https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2>.

⁹⁶ *Id.*

Exemplary video bitstream manifest information

```

670 "video_tracks": [
671     {
672         "trackType": "PRIMARY",
673         "track_id": "c6dfe6c57c55e4bbe4f57e7c75de6bde6f776e7c666dc673657cfed6ef657c75de6bde6f776e7c62447c5669e4656f7c75de6bde6f776e7c7c7c6465666f756c7*",
674         "mov_track_id": "V12112;default-1;none-1;",
675         "dimensionCount": 0,
676         "dimensionalLabel": "2D",
677         "stereo": false,
678         "profileType": "AbstractDynamicNum(name='VIDEO')",
679         "type": 1,
680         "isC": false,
681         "hasClearProfile": false,
682         "hasDrnProfile": true,
683         "hasClearStreams": true,
684         "hasDrnStreams": false,
685         "streams": [
686             {
687                 "trackType": "PPHDMV",
688                 "content_profile": "playready-h264hp130-dash",
689                 "createTime": 0,
690                 "peakBitrate": 888,
691                 "dimensionCount": 0,
692                 "dimensionalLabel": "2D",
693                 "drmHeaderID": "",
694                 "pix_w": 1,
695                 "pix_h": 1,
696                 "res_w": 768,
697                 "res_h": 432,
698                 "frameRate_value": 24000,
699                 "frameRate_scale": 1001,
700                 "name": "H264",
701                 "startByteOffset": 1421,
702                 "moov": 60,
703                 "segmentRead": [],
704                 "crop_x": 0,
705                 "crop_y": 0,
706                 "crop_w": 768,
707                 "crop_h": 432,
708                 "downloadable_id": "1074379338",
709                 "tags": [
710                     {
711                         "mov_stream_id": "todo",
712                         "type": 1,
713                         "url": [
714                             {
715                                 "odm_id": "7947",
716                                 "url": "https://www-e330-shl01-iw.1.amazonaws.com/net/7?i=784ee1c44290491cfa08-HCAB28f6e8f232020WvQoFdaTvOinIHEnllTofauadITf0d8TastIwd8zNDKc1xYVZouqagP1OUW2Ze2a3T7oKe8QoeK6CVSDF",
717                             },
718                             {
719                                 "odm_id": "88209",
720                                 "url": "https://www-e330-shl01-iw.1.amazonaws.com/net/7?i=784ee1c44290491cfa0838T6t17TT8nLT7F390eCl03cde6d8MwUoLcoCFahPW_Q07479820av0B0Xk9a1VjaoP40FFaceITTeafECCIP0-met4doTohot1evajsvf",
721                             },
722                             {
723                                 "odm_id": "68820",
724                                 "url": "https://www-e330-shl01-iw.1.amazonaws.com/net/7?i=206e1c44290491cfa0838T6t17TT8nLT7F390eCl03cde6d8MwUoLcoCFahPW_Q07479820av0B0Xk9a1VjaoP40FFaceITTeafECCIP0-met4doTohot1evajsvf",
725                             }
726                         ],
727                         "tags": [
728                             {
729                                 "tag": "id3",
730                                 "value": "MP3",
731                             }
732                         ]
733                     }
734                 ]
735             }
736         ]
737     },
738     {
739         "type": 1,
740         "isC": false
741     }
742 ],
743 "isC": false
744 }

```

```

573:         "video_tracks": [
574:             {
575:                 "trackType": "PRIMARY",
576:                 "track_uri": "def4ed407554eb4ef574e7c756ed4bde776a70c6d6616c73657c0ed64bde776a70c6d6646567c756ed4bde776a70c7d64666d1756d74",
577:                 "new_track_id": "V12122;default-1;none-1;",
578:                 "dimensionCount": 2,
579:                 "dimensionLabel": "2D",
580:                 "stereo": false,
581:                 "profileType": "AbstractDynamicEnum(name='VIDEO')",
582:                 "type": 1,
583:                 "isrc": false,
584:                 "hasClearProfile": false,
585:                 "hasDrmProfile": true,
586:                 "hasClearStreams": true,
587:                 "hasDrmStreams": false,
588:                 "streams": [
589:                     {
590:                         "trackType": "PRIMARY",
591:                         "content_profile": "playready-h264p130-dash",
592:                         "bitrate": 1115,
593:                         "peakBitrate": 1571,
594:                         "dimensionCount": 2,
595:                         "dimensionLabel": "2D",
596:                         "drmHeaderId": "",
597:                         "pfx_w": 1,
598:                         "pfx_h": 1,
599:                         "cep_w": 768,
600:                         "cep_h": 432,
601:                         "frameRate_Value": 24000,
602:                         "frameRate_scale": 1001,
603:                         "alpha": 0.0000,
604:                         "startByteOffset": 1421,
605:                         "mval": 75,
606:                         "segmentUrfid": [],
607:                         "crop_x": 0,
608:                         "crop_y": 0,
609:                         "crop_w": 768,
610:                         "crop_h": 432,
611:                         "downloadable_id": "1074380171",
612:                         "tag": {
613:                             "new_stream_id": "todo",
614:                             "type": 1,
615:                             "urlSet": {
616:                                 {
617:                                     "cdn_id": "1471",
618:                                     "url": "https://www-c01-sh1001-iw.i.cdn.cloudflare.net/3o-iw=784e1c4428504941eA5843aa5tU7M600m7Y2ubWYebU00q1CvN0h5t-811ePw025eTer_kbn0c11X0C0;ZXuh0027T1a37AWC0V7u5a10dehU7cU70N0_Y0d",
619:                                 },
620:                                 {
621:                                     "cdn_id": "83209",
622:                                     "url": "https://www-c08-sh1001-iw.i.cdn.cloudflare.net/3o-iw=784e1c4428504941eA5843aa5tU7M600m7Y2ubWYebU00q1CvN0h5t-811ePw025eTer_kbn0c11X0C0;ZXuh0027T1a37AWC0V7u5a10dehU7cU70N0_Y0d",
623:                                 },
624:                                 {
625:                                     "cdn_id": "68920",
626:                                     "url": "https://www-c187-gw202-iw.i.cdn.cloudflare.net/3o-iw=204e1c4428504941eA5843aa5tU7M600m7Y2ubWYebU00q1CvN0h5t-811ePw025eTer_kbn0c11X0C0;ZXuh0027T1a37AWC0V7u5a10dehU7cU70N0_Y0d",
627:                                 }
628:                             },
629:                         },
630:                     },
631:                 ],
632:                 "isDrm": false
633:             }
634:         ]
635:     }
636: }

```


Exemplary audio bitstream manifest information

[illegible][illegible]

[illegible][illegible]

84

an assignment rule that assigns each one of the content-related second data segments (*i.e.*, the audio segments) to one of the content-related first data segments (*i.e.*, the video segments).

203. Playback of content in the Netflix client application on the subscriber's connected device or in the subscriber's browser essentially consists of a three-step process: (1) loading and parsing the JSON data structure and related ISOBMFF *sidx* boxes to create a list of segments for the various content types, associating an approximate position on the overall media presentation timeline for each; (2) using the resulting segment lists and approximate timing information to obtain video and audio segments from the specified Netflix CDN(s) and load the appropriate content segments into a playback buffer sufficient to begin content presentation by the presentation engine; and (3) managing the playback buffer as presentation continues, including the loading of subsequent segments for consumption, and the removal of segments which have been presented and are no longer needed.

204. To this end, the Netflix client on the subscriber's connected device, or the Netflix application programmed into the web browser when accessing the Netflix streaming service, parses the respective video and audio ISOBMFF *sidx* box information of the video and audio initialization segments and obtains the complete, chronological listing of the individual video and audio segments contained therein, as noted above. The *sidx* box information specifies not only segment references that provide a chronological ordering to the listed content-related data segments, but other metadata, including at least offset and approximate duration information (shown highlighted below), that enable the Netflix client to order and assign the different content-related data segments relative to each other and with respect to an approximate media presentation timeline.

segment lists, in conjunction with other *sidx* box metadata. Collectively, at least this metadata serves as an assignment rule that assigns each one of the content-related second data segments (*i.e.*, the audio segments) to one of the content-related first data segments (*i.e.*, the video segments) on the media presentation timeline. Given the Netflix '794 Accused Products' reliance on the use of shot-based video encoding, the relative number and duration of video and audio segments differ significantly. Realizing Netflix's goal of a "best streaming experience" relies on at least each of the audio segments being output together with an associated one of the video segments to maintain content synchronization during playback so that the correct audio is played with the correct sequence of video and at the correct time.

206. VideoLabs first reached out to Netflix on October 22, 2019, to provide information about the many benefits and value of VideoLabs' platform.⁹⁷ Netflix never responded. VideoLabs again contacted Netflix on at least February 20, 2020, and again received no response.⁹⁸ Finally, in response to an email on March 1, 2021, Netflix responded and agreed to have an introductory call on March 19, 2021.⁹⁹ Following the initial call, VideoLabs provided a draft non-disclosure agreement so that the parties could continue their discussions with certain mutual protections.¹⁰⁰ After some back-and-forth, it became clear that Netflix would not agree to an NDA unless it contained a "standstill" for an indefinite period of time.¹⁰¹ Given that nearly two years had already passed since VideoLabs first reached out to Netflix, this was an untenable position. After Netflix

⁹⁷ See Ex. 47, VideoLabs email to Netflix dated October 22, 2019.

⁹⁸ See Ex. 48, VideoLabs email to Netflix dated February 20, 2020.

⁹⁹ See Ex. 50, VideoLabs-Netflix email chain dated March 1, 2021 through February 22, 2022, at 10-12.

¹⁰⁰ See *id.* at 8-10.

¹⁰¹ See *id.* at 2-7.

refused to agree to a more reasonable standstill length, it stopped responding to VideoLabs' communications.

207. On information and belief, Netflix has long known about the '794 Patent through its involvement with DASH. In 2010, Netflix was "actively participating in the MPEG committee for Dynamic Adaptive Streaming over HTTP (DASH) to define an industry standard for adaptive streaming."¹⁰² The DASH standard, which Netflix was involved with creating, was published in 2012.¹⁰³ Then, in 2016, MPEG LA created a pool of patents that are necessarily practiced when the DASH standard is used.¹⁰⁴ The '794 Patent is one of just 10 U.S. patents in the pool. On information and belief, from its longstanding interest and involvement in adaptive bitrate streaming and DASH, Netflix either knew about the '794 Patent and its relevance to its streaming platform or intentionally decided not to investigate the '794 Patent.

208. Netflix of course knows how its products operate, and on information and belief, upon becoming aware of the '794 Patent, began investigating the '794 Patent and its infringement of the Netflix '794 Accused Products. Netflix has been given further notice of the '794 Patent and its infringement of the '794 Patent through the filing of this First Amended Complaint. On information and belief, Netflix is either knowingly infringing the '794 Patent or is willfully blind to its infringement — including by ignoring VideoLabs' communications and/or using a disagreement over a standstill in an NDA as a pretext to avoid having discussions with VideoLabs

¹⁰² Ex. 78, *HTML5 and Video Streaming* (Dec. 22, 2010), <https://netflixtechblog.com/html5-and-video-streaming-a3563b19eb02>.

¹⁰³ Ex. 79, *Dynamic Adaptive Streaming over HTTP*, Wikipedia, https://en.wikipedia.org/wiki/Dynamic_Adaptive_Streaming_over_HTTP.

¹⁰⁴ Ex. 67, *MPEG LA Releases MPEG-DASH Patent Portfolio License*, MPEG LA (Nov. 17, 2016), <https://www.mpegla.com/media/>.

and learning more about VideoLabs' patents — and continues to act in wanton disregard of VideoLabs' patent rights.

209. Despite becoming aware of or willfully blinding itself to its infringement of the '794 Patent, Netflix has nonetheless continued to engage in and has escalated its infringing activities by continuing to develop, advertise, make available, and use the infringing functionalities of the Netflix '794 Accused Products. On information and belief, Netflix has made no attempts to design around the '794 Patent or otherwise stop its infringing behavior.

210. Netflix's infringement of the '794 Patent therefore has been and remains willful.

211. Netflix also indirectly infringes the '794 Patent by inducing others to infringe and contributing to the infringement of others, including third-party users of the Netflix '794 Accused Products in this District and throughout the United States. As described above, on information and belief, Netflix has long known about the '794 Patent.

212. On information and belief, Netflix has actively induced the infringement of the '794 Patent under 35 U.S.C. § 271(b) by actively inducing the infringement of the Netflix '794 Accused Products by third parties in the United States. Netflix knew or was willfully blind to the fact that its conduct would induce these third parties to act in a manner that infringes the '794 Patent in violation of 35 U.S.C. § 271(a).

213. Netflix actively encouraged and continues to actively encourage third parties to directly infringe the '794 Patent by, for example, marketing the Netflix '794 Accused Products and infringing functionalities to consumers; working with consumers to implement, install and/or operate the Netflix '794 Accused Products and infringing functionalities; fully supporting and managing consumers' continuing use of the Netflix '794 Accused Products and infringing

functionalities; and providing technical assistance to consumers during their continued use of the '794 Accused Products and infringing functionalities.¹⁰⁵

214. For example, Netflix induces third parties to infringe the '794 Patent by encouraging them to install and operate the Netflix streaming service, which alone and/or in combination with the third parties' devices constitutes infringement of the '794 Patent. Netflix advertises and promotes its Netflix streaming service on its website and in various app stores such as Apple's app store and Android's app store in connection with the Netflix mobile application that can be installed on consumers' respective connected iOS and Android devices (as well as others), and encourages consumers to configure and operate their mobile and computer devices in an infringing manner.¹⁰⁶ In response, consumers acquire, configure, and operate the Netflix streaming service in an infringing manner.

215. On information and belief, Netflix contributorily infringes the '794 Patent under 35 U.S.C. § 217(c) by importing, selling, and/or offering to sell within the United States the Netflix '794 Accused Products (or components thereof) that constitute a material part of the claimed invention and are not staple articles of commerce suitable for substantial non-infringing use. For example, the hardware and/or software for sequentially outputting content-related data segments is material, has no insubstantial non-infringing uses, and is known by Netflix to be especially made or adapted for use in a manner that infringes the '794 Patent.

Prayer For Relief

WHEREFORE, VideoLabs prays for judgment as follows:

¹⁰⁵ See e.g., Exs. 80-85, <https://www.netflix.com/app>; <https://help.netflix.com/en/node/101653>; <https://help.netflix.com/en/>; <https://www.youtube.com/watch?v=IYx8Keoxl50>; <https://www.youtube.com/watch?v=Ha8ofuzpriE>; <https://www.nytimes.com/2019/12/16/business/media/netflix-commercials.html>.

¹⁰⁶ See e.g., Ex. 81, <https://help.netflix.com/en/node/101653>.

a) That Netflix directly and/or indirectly infringes the '878, '559, '790, and '794 Patents;

b) That such infringement is willful;

c) That Netflix and its respective officers, directors, agents, partners, servants, employees, attorneys, licensees, successors, and assigns, and those in active concert or participation with any of them, be permanently enjoined from engaging in infringing activities with respect to the '878, '559, '790, and '794 Patents;

d) In the alternative, in the event injunctive relief is not granted as requested by VideoLabs, an award of a mandatory future royalty payable on each future product sold by Netflix that is found to infringe one or more claims of the '878, '559, '790, and '794 Patents, and on all future products which are not colorably different from products found to infringe;

e) That Netflix be required to pay VideoLabs damages in an amount adequate to compensate VideoLabs for Netflix's infringement, but in no event less than a reasonable royalty under 35 U.S.C. § 284, including supplemental damages for any continuing post-verdict infringement up until entry of judgment and beyond, with accounting, as needed;

f) That VideoLabs be awarded all statutory and actual damages to which it is entitled, including the profits reaped by Netflix through its illegal conduct, and prejudgment and post-judgment interest;

g) That VideoLabs be awarded enhanced damages, up to and including trebling of the damages awarded to VideoLabs;

h) That VideoLabs be awarded recovery of the costs of this suit, including reasonable attorneys' fees; and

i) That VideoLabs be awarded such further relief as this Court deems just and proper.

Demand For Jury Trial

216. Pursuant to Rule 38 of the Federal Rules of Civil Procedure and D. Del. LR 38.1, VideoLabs demands a jury trial on its claims for patent infringement and any and all issues triable of right before a jury.

Dated: May 6, 2022

Respectfully submitted,

FARNAN LLP

Christine E. Lehman (*pro hac vice*)
clehman@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
1909 K Street NW, Suite 800
Washington, DC 20006
Telephone: (202) 894-7325
Fax: (650) 623-1449

/s/ Michael J. Farnan
Brian E. Farnan (Bar No. 4089)
Michael J. Farnan (Bar No. 5165)
919 N. Market St., 12th Floor
Wilmington, DE 19801
Telephone: (302) 777-0300
Fax: (302) 777-0301
bfarnan@farnanlaw.com
mfarnan@farnanlaw.com

Wesley L. White (*pro hac vice*)
wwhite@reichmanjorgensen.com
Jaime F. Cardenas-Navia (*pro hac vice*)
jcardenas-navia@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
750 Third Avenue, Suite 2400
New York, NY 10017
Telephone: (212) 381-1965
Facsimile: (650) 623-1449

Courtland L. Reichman (*pro hac vice*)
creichman@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
100 Marine Parkway, Suite 300
Redwood Shores, California 94065
Telephone: (650) 623-1401
Fax: (650) 623-1449

Taylor N. Mauze (*pro hac vice*)
tmauze@reichmanjorgensen.com
REICHMAN JORGENSEN LEHMAN &
FELDBERG LLP
7500 Rialto Blvd., Ste. 1-250
Austin, Texas 78735
Telephone: (650) 623-1401
Facsimile: (650) 623-1449

***Attorneys for Plaintiffs VideoLabs, Inc. and
VL Collective IP LLC***