

1 Jason S. Angell, Bar No. 221607
jangel@hopkinscarley.com
2 Christopher A. Hohn, Bar No. 271759
chohn@hopkinscarley.com
3 Cary Chien, Bar No. 274078
cchien@hopkinscarley.com
4 Robert K. Jain, Bar No. 309728
rjain@hopkinscarley.com
5 HOPKINS & CARLEY
A Law Corporation
6 The Letitia Building
70 South First Street
7 San Jose, CA 95113-2406

8 ***mailing address:***
P.O. Box 1469
9 San Jose, CA 95109-1469
Telephone: (408) 286-9800
10 Facsimile: (408) 998-4790

11 Attorneys for Plaintiffs
BROADCOM CORPORATION and AVAGO
12 TECHNOLOGIES INTERNATIONAL SALES
PTE. LIMITED.

13
14 UNITED STATES DISTRICT COURT
15 CENTRAL DISTRICT OF CALIFORNIA
16 SOUTHERN DIVISION

17 BROADCOM CORPORATION and
18 AVAGO TECHNOLOGIES
INTERNATIONAL SALES PTE.
19 LIMITED,

20 Plaintiffs,

21 v.

22 NETFLIX, INC.,

23 Defendant.

Case No. 8:20-cv-529

**COMPLAINT FOR PATENT
INFRINGEMENT**

DEMAND FOR JURY TRIAL

1 Plaintiffs Broadcom Corporation (“Broadcom Corp.”) and Avago
2 Technologies International Sales Pte. Limited (“Avago”) (collectively, the
3 “Broadcom Entities”) file this Complaint for Patent Infringement against Defendant
4 Netflix, Inc. (“Netflix”) and allege as follows:

5 **NATURE OF THIS ACTION**

6 1. This complaint alleges patent infringement. The Broadcom Entities
7 allege that Netflix has infringed and continues to infringe, directly and/or indirectly,
8 eight patents: U.S. Patent Nos. 7,266,079 (the “’079 Patent”); 8,259,121 (the “’121
9 Patent”); 8,959,245 (the “’245 Patent”); 8,270,992 (the “’992 Patent”); 6,341,375
10 (the “’375 Patent”); 8,572,138 (the “’138 Patent”); 6,744,387 (the “’387 Patent”);
11 6,982,663 (the “’663 Patent”); and 9,332,283 (the “’283 Patent”). Copies of these
12 patents (collectively, the “Patents-in-Suit”) are attached hereto as **Exhibits A-I**.

13 2. The Patents-in-Suit cover foundational technologies that are essential
14 to various aspects of Netflix’s video streaming service, and the systems that Netflix
15 uses to support this service.

16 3. Netflix directly infringes the Patents-in-Suit by making, using, offering
17 to sell, selling, and/or importing into the United States internet video streaming
18 technology, software, and services that practice the inventions claimed in the
19 Patents-in-Suit. Netflix directs and controls each relevant aspect of the accused
20 technology discussed herein, and benefits from the use of each feature that infringes
21 the Patents-in-Suit.

22 4. Netflix indirectly infringes the Patents-in-Suit by inducing its
23 consumer end-users to directly infringe these patents. For example, Netflix induces
24 infringement by providing software (e.g., the Netflix application) that, when used
25 by consumers or other content viewers to stream digital content to televisions,
26 personal computers, phones, tablets, and other devices, as directed and intended by
27 Netflix, causes those end-users to use and practice the inventions claimed in the
28 Patents-in-Suit.

5. The Broadcom Entities seek damages and other relief for Netflix's infringement of the Patents-in-Suit.

PARTIES

6. Plaintiff Broadcom Corporation is a California corporation headquartered at 1320 Ridder Park Drive, San Jose, California 95131. Broadcom Corp. maintains offices within the Central District of California at 15101 Alton Parkway, Irvine, California 92618. Broadcom Corp. is an indirect subsidiary of Broadcom, Inc.

7. Plaintiff Avago Technologies International Sales Pte. Ltd. is a corporation formed under the laws of Singapore with places of business at 1320 Ridder Park Dr., San Jose, California 95131 and 1 Yishun Avenue 7, Singapore 768923. Avago is also an indirect subsidiary of Broadcom, Inc.

8. Defendant Netflix, Inc. is a Delaware corporation that maintains its principal place of business and global headquarters at 100 Winchester Circle, Los Gatos, 95032.

9. Netflix maintains regular and established places of business in this District, including an office at 5808 Sunset Blvd., Los Angeles, CA 90028, where Netflix employs hundreds of people. According to Netflix’s website, the Los Angeles office “is the entertainment hub for Netflix with teams such as Content, Legal, Marketing & Publicity and is located on the Sunset Bronson Studio Lot where a variety of Netflix content is created.”¹

10. Netflix may be served through its registered agent for service of process in California: CT Corporation System, 818 W. Seventh St, Suite 930, Los Angeles, CA 90017.

11. Netflix claims to be a global leader in streaming digital video content. Netflix streams videos of various types, such as films and television series, to over

¹ <https://jobs.netflix.com/locations/los-angeles-california>

1 158 million paid members in over 190 countries. Upon information and belief,
2 Netflix designs, operates, tests, manufactures, uses, offers for sale, sells, and/or
3 imports into the United States—including into the Central District of California—
4 internet video streaming software, systems, and services that generate billions of
5 dollars of revenue for Netflix each year.

6 **JURISDICTION AND VENUE**

7 12. The Broadcom Entities bring this civil action for patent infringement
8 under the Patent Laws of the United States, 35 U.S.C. § 1 et. seq., including 35
9 U.S.C. §§ 271, 281-285. This Court has subject matter jurisdiction over this action
10 pursuant to 28 U.S.C. §§ 1331 and 1338.

11 13. The Broadcom Entities' claims for relief arise, at least in part, from
12 Netflix's business contacts and other activities in the State of California and in this
13 District. Upon information and belief, Netflix has committed acts of infringement
14 within this District and the State of California by making, using, selling, offering
15 for sale, and/or importing into the United States and this District products, systems,
16 and services that infringe one or more claims of the Patents-in-Suit as set forth
17 herein. Further, Netflix induces others within this District to infringe one or more
18 claims of the Patents-in-Suit.

19 14. Venue is proper in this district and division under 28 U.S.C. §§
20 1391(b)-(d) and 1400(b) because Netflix has committed acts of infringement in the
21 Central District of California and has a regular and established physical place of
22 business in Los Angeles, part of the Central District. Upon information and belief,
23 Netflix employs engineers and technical professionals of many disciplines at its Los
24 Angeles facility.

25 **FACTUAL BACKGROUND**

26 15. Henry Samueli and Henry Nicholas founded Broadcom in 1991 in Los
27 Angeles, California. Since then, Broadcom has grown to be a global technology
28 company that produces category-leading semiconductor and infrastructure software

1 solutions. Among other things, Broadcom provides one of the industry's broadest
2 portfolios of highly integrated semiconductor chips that seamlessly deliver voice,
3 video, data, and multimedia connectivity in the home, office, and mobile
4 environments. From its headquarters in San Jose, California, Broadcom has
5 expanded its footprint across the United States and around the world, employing
6 thousands of individuals globally and in the United States. An overview of
7 Broadcom's history can be found on its website at:

8 <https://www.broadcom.com/company/about-us/company-history/>.

9 16. Broadcom's continued success depends in substantial part upon its
10 constant attention to research and development. Broadcom and its subsidiaries
11 spend billions of dollars on research and development for their products each year.
12 Because of this focus, Broadcom has produced a wide range of novel technologies
13 and inventions that are directed to advancements in, among other things,
14 semiconductor design and digital communications, digital content distribution,
15 enterprise and data center networking, home connectivity, set top boxes,
16 infrastructure software, and other technologies integral to business and consumer
17 settings across the United States and throughout the world.

18 17. Broadcom relies on the patent system as an important part of its
19 intellectual property program to protect the valuable technology and inventions
20 resulting from its research and development efforts. The Broadcom Entities and
21 their related entities have tens of thousands of patents in the United States and
22 abroad.

23 18. In addition to their internally developed inventions and associated
24 intellectual property, the Broadcom group of companies have acquired technology
25 and intellectual property through mergers and acquisition with other major
26 technology companies, such as the Avago family of companies, LSI, Brocade, CA,
27 Inc. (formerly known as Computer Associates International, Inc.), and Symantec's
28 enterprise business.

1 19. As explained in detail below, Netflix has built its familiar video
2 streaming business, in part, on the Broadcom Entities' patented technology. Netflix
3 relies on this technology for crucial aspects of the Netflix streaming service. This
4 includes, for example, the Netflix systems used to ensure effective and reliable
5 delivery of streaming content with minimal interruptions, to ensure the efficient use
6 of Netflix server resources, and to encode Netflix streaming content in a format
7 compatible with a large percentage of the client devices (e.g., computers, smart
8 TVs) used to access the Netflix service.

9 20. In doing so, Netflix has caused, and continues to cause, substantial and
10 irreparable harm to the Broadcom Entities. For instance, the Broadcom Entities sell
11 semiconductor chips used in the set top boxes that enable traditional cable
12 television services. Upon information and belief, as a direct result of the on-
13 demand streaming services provided by Netflix, the market for traditional cable
14 services that require set top boxes has declined, and continues to decline, thereby
15 substantially reducing Broadcom's set top box business.

16 21. For instance, it is widely reported that the rise of on-demand video
17 streaming services such as Netflix has concurrently lead to a decrease in demand
18 for traditional cable services. As an example, *Variety* reported in February 2019
19 that "[t]he five biggest U.S. pay-television providers saw their traditional subscriber
20 rolls shrink 4.2% in 2018, as they collectively lost around 3.2 million customers for
21 the year. That's an acceleration from estimated sector-wide declines of 3.7% in
22 2017 and 2% in 2016." The article attributes the loss in part to a migration of
23 customers to Netflix.²

24 22. Upon information and belief, Netflix could not displace traditional
25 cable television services, or could not do so as effectively, without the use of the
26

27
28 ² <https://variety.com/2019/biz/news/cord-cutting-2018-accelerate-us-pay-tv-subscribers-1203138404/>

1 Broadcom Entities' patented technology, which—as explained above—enable
2 critical aspects of Netflix's systems.

3 23. Netflix is aware of the Broadcom Entities' patent portfolio, including
4 specifically nearly all the patents asserted in this Complaint, based on
5 communications between Netflix and the Broadcom Entities. Representatives of
6 the Broadcom Entities have repeatedly attempted to engage Netflix in licensing
7 discussions. As part of these attempts, the Broadcom Entities informed Netflix of
8 its infringement of the patents asserted in this Complaint (with the exception of the
9 '283 Patent) on or about September 26, 2019, and the parties engaged in in-person
10 discussions on October 24, 2019. Netflix did not dispute the infringement
11 presentations Broadcom provided to Netflix, or otherwise assert that it did and does
12 not infringe the patents identified to Netflix. Unfortunately, Netflix declined to
13 agree to terms for a license for its use of the Broadcom Entities' patents and
14 technology, and declined to present a counteroffer to license terms offered by the
15 Broadcom Entities.

16 24. Left with no other choice, the Broadcom Entities bring this action to
17 protect their rights and their investment in the research and development of novel
18 technologies.

19 **FIRST CLAIM FOR RELIEF**

20 **(Infringement of U.S. Patent No. 7,266,079)**

21 25. The Broadcom Entities reallege and incorporate by reference the
22 allegations of paragraphs 1-24 set forth above.

23 26. The '079 Patent, entitled "Dynamic Network Load Balancing Over
24 Heterogeneous Link Speed," was duly and legally issued on September 4, 2007
25 from a patent application filed on July 2, 2001, with Kan Frankie Fan as the named
26 inventor. A copy of the '079 Patent is attached hereto as **Exhibit A**.

27 27. The '079 Patent claims priority from U.S. Provisional Application No.
28 60/233,338, filed on September 18, 2000.

1 28. The '079 Patent was assigned to Avago, which currently holds all
2 substantial rights, title, and interest in and to the '079 Patent.

3 29. Pursuant to 35 U.S.C. § 282, the '079 Patent is presumed valid.

4 30. The '079 Patent is directed to an improvement in the functionality of
5 networked computer systems by “balancing data flow there through.”³ Specifically,
6 the '079 Patent’s claims describe a new approach for balancing transmission unit
7 traffic over the multiple heterogeneous links that often connect computing
8 platforms in a computer network.

9 31. The '079 Patent addresses a specific technical problem that arose in
10 the computer networking environment as the networks grew ever larger and more
11 complex, and as users sought to transmit ever greater volumes of data across these
12 networks. As the '079 Patent states, “[t]he present invention relates to
13 communications apparatus and methods, particularly to computer networking
14 apparatus and methods, and more particularly to computer networking apparatus
15 and methods for balancing data flow there through.”⁴

16 32. As the '079 Patent explains, “[a] common problem in communication
17 networks is maintaining efficient utilization of network resources, particularly with
18 regard to bandwidth, so that data traffic is efficiently distributed over the available
19 links between sources and destinations.”⁵ “Prior art solutions include apparatus and
20 methods that balance data traffic over homogeneous (same-speed) links between
21 heterogeneous or homogeneous computing platforms (servers, clients, etc.).”⁶
22 However, “[i]ncreasingly, high-performance computing platforms communicate
23 with other computers, routers, switches, and the like, using multiple links which,
24 for a variety of reasons, may operate at disparate link speeds.”⁷ “For
25

26 ³ '079 Patent, 1:17-21.

27 ⁴ *Id.*

28 ⁵ *Id.* at 1:23-27.

⁶ *Id.* at 1:27-30.

⁷ *Id.* at 1:30-34.

1 example...adverse network conditions may degrade the performance of one or
 2 more links, effectively presenting a heterogeneous-link-speed environment to the
 3 server and its link partner(s).”⁸

4 33. Accordingly, a need existed for a means to “dynamically balance
 5 transmission unit traffic in a heterogeneous-link-speed environment” in order to
 6 improve the functionality of computer networks.⁹

7 34. The ’079 Patent claims specific, novel ways to solve these technical
 8 problems by dynamically balancing data traffic in a computer networking
 9 environment with heterogeneous link speeds. The claims of the ’079 Patent are
 10 directed to new, improved methods and apparatuses for balancing transmission unit
 11 traffic over networks links.

12 35. The methods and apparatuses described in the ’079 Patent improve the
 13 functionality of a networked computer system by balancing the data traffic among
 14 network links having different speeds, capabilities, and congestion levels that
 15 connect the various networked elements, thereby improving the speed and
 16 efficiency of data transmission within the network.

17 36. Claim 1 of the ’079 Patent reads as follows:

18 A method for balancing transmission until traffic over
 19 network links, comprising:

- 20 a. disposing transmission units into flows;
- 21 b. grouping flows into first flow lists, each of the
- 22 first flow lists corresponding to a selected network
- 23 link;
- 24 c. determining a traffic metric representative of a
- 25 traffic load on the selected network link;
- 26 d. responsive to the traffic metric, regrouping flows
- 27 into second flow lists corresponding to the selected

28 ⁸ *Id.* at 1:34-40.

⁹ *Id.* at 1:40-43.

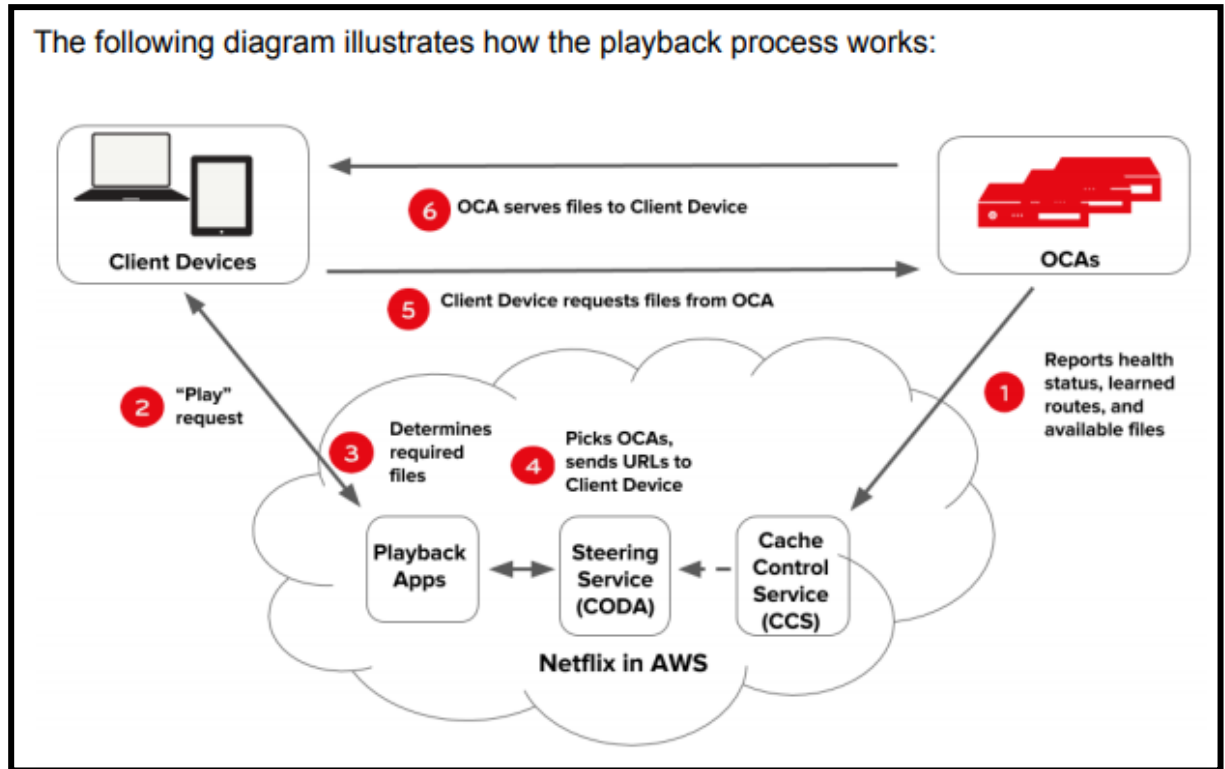
1 network link, the regrouping balancing the
2 transmission unit traffic among the network links;
3 and

4 e. transmitting the respective second flow list over
5 the respective selected network link.

6 37. Netflix directly infringes the '079 Patent by making, using, offering to
7 sell, and/or selling in the United States its Netflix service, which utilizes the
8 inventions claimed in the '079 Patent to balance traffic over Netflix's systems,
9 including its content delivery network ("CDN").

10 38. Netflix directly infringes at least independent claim 1 of the '079
11 Patent at least in the exemplary manner described below.

12 39. Netflix utilizes the claimed "method for balancing transmission unit
13 traffic over network links," including, for instance, in operating its CDN, which
14 Netflix uses to stream video content to its subscribers over the internet. The Netflix
15 CDN is illustrated in the following diagram.



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

40. The Netflix CDN comprises hardware that Netflix builds and operates. Additionally, for some aspects of its CDN, Netflix uses hardware produced and maintained by third parties, including cloud computing services purchased from third parties. However, all the infringing technologies discussed in this Complaint are developed and controlled by Netflix, including though the use of Netflix software that controls the relevant functions performed by the underlying hardware.

41. Notably, in apparent recognition that its cloud computing services could be used by a customer to infringe any number of patents involving a computer as part of a larger system or process, the AWS standard customer agreement makes clear that AWS has no liability for patent infringement claims arising from infringement by combinations of AWS's services with any other product, service, software, data, content or method.¹⁰

¹⁰ <https://aws.amazon.com/agreement/>

1 42. As part of its CDN, Netflix created, operates, uses, and maintains a
2 “global network of thousands of [Open Connect Appliances],” also known as
3 OCAs.¹¹

4 43. As Netflix explains, “[t]he building blocks of Open Connect are our
5 suite of purpose-built server appliances, called Open Connect Appliances (OCAs).
6 These appliances store and serve our video content, with the sole responsibility of
7 delivering playable bits to client devices as fast as possible.”¹²

8 44. Within the CDN, Netflix stores the TV programs and movies that it
9 offers as a series of files. Netflix’s customers access Netflix’s video content
10 through different types of client devices, including digital televisions, desktop
11 computers, laptop computers, tablet computers, and mobile phones. These client
12 devices are produced by many different manufacturers. Each device has certain
13 capabilities and features that require media to be delivered in a specific form or
14 format. In many cases, the media format used by one device cannot be used by
15 another. Thus, Netflix must make its content available to its users in many different
16 formats.

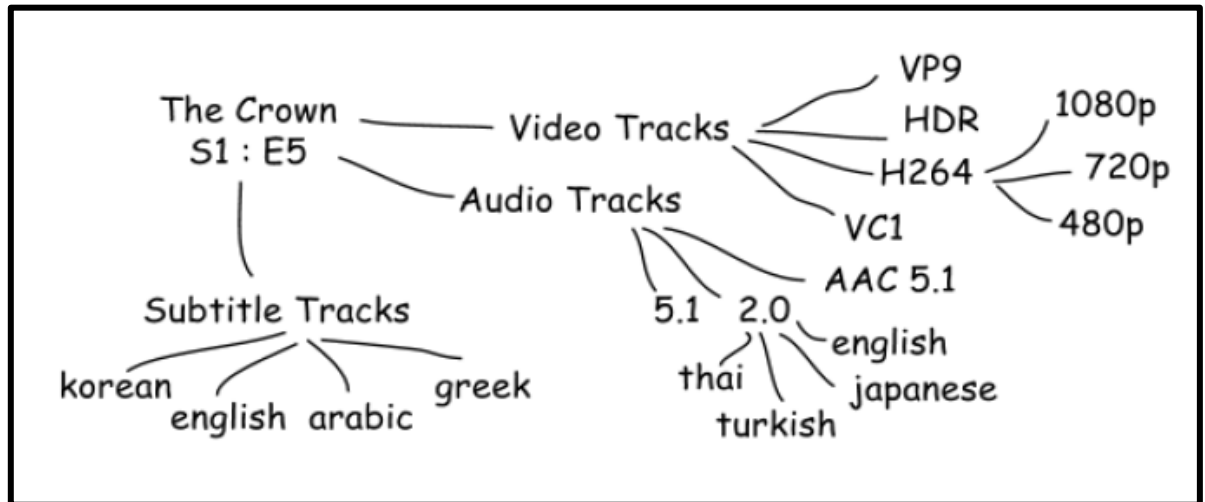
17 45. In Netflix’s words:

18 Every title is encoded in multiple formats, or *encoding*
19 *profiles*. For example, some profiles may be used by iOS
20 devices and others for a certain class of Smart TVs.
21 There are video profiles, audio profiles, and profiles that
22 contain subtitles. Each audio and video profile is encoded
23 into different levels of quality. For a given title, the
24 higher the number of bits used to encode a second of
25 content (bps), the higher the quality....Finally, we have
26 audio profiles and subtitles available in multiple
27 languages. So for each quadruple of (title, encoding
28 profile, bitrate, language), we need to cache one or more
files. As an example, for streaming one episode of The
Crown we store around 1,200 files!”¹³

11 <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

12 *Id.*

13 <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>



Source: <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>

46. Netflix divides these various content files “into 2 second intervals called *segments*.”¹⁴

47. When a Netflix subscriber initiates a playback session by selecting a movie or TV show and pressing “play,” the Netflix system “determine[s] which specific files are required to handle the playback request—taking individual client characteristics and current network conditions into account.” The Netflix system “pick[s] OCAs that the requested files should be served from [and] generates URLs for these OCAs...” The Netflix system then “hand[s] over URLs of the appropriate OCAs to the client device, and the OCA begins to serve the requested files.”

¹⁴ <https://cs.uwaterloo.ca/~brecht/papers/iiswc-netflix-wload-2016.pdf>

2. A user on a client device requests playback of a title (TV show or movie) from the Netflix application in AWS.

3. The playback application services in AWS check user authorization and licensing, then determine which specific files are required to handle the playback request - taking individual client characteristics and current network conditions into account.

4. The steering service in AWS uses the information stored by the cache control service to pick OCAs that the requested files should be served from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

5. The playback application services hand over URLs of the appropriate OCAs to the client device, and the OCA begins to serve the requested files.

Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

48. Additionally, when a Netflix subscriber initiates a playback session, the Netflix system provides the “offsets of all segments.”¹⁵ The client device, under the control of the Netflix application running on it, then “downloads segment offsets and content from multiple content files with different bit rates, then selects a starting bitrate that can be supported by available network bandwidth.”¹⁶ The client device “continue[s] to download segments sequentially from the same file unless network or server conditions change (which may result in switching to a different bit rate) or a user event occurs (e.g., stopping or skipping to a new title position).”¹⁷ Thus, and as described further below, Netflix performs a “method for balancing transmission unit traffic over network links.”

49. The Netflix CDN also “dispos[es]” the “transmission units into flows.” For example, each of the above-described “segments” of a given content file constitutes a “transmission unit.” The Netflix system “disposes” these segments

¹⁵ <https://cs.uwaterloo.ca/~brecht/papers/iiswc-netflix-wload-2016.pdf>

¹⁶ *Id.*

¹⁷ *Id.*

1 “into flows” by causing the transmission of a series of segments of various content
2 files—each a “flow”—to each of its many subscribers during the playback process.

3 50. Upon information and belief, the Netflix system also “group[s] flows
4 into first flow lists,” each of which “correspond[] to a selected network link.” For
5 example, each of the OCAs, and each of the file locations specified by a URL
6 within each OCA, constitutes a “network link.” As explained above, Netflix
7 “group[s] flows into first flow lists” by picking the OCAs, and the specific files
8 thereon, that are used to serve a subscriber’s playback request for each of the
9 thousands, or even millions, of subscribers streaming video content from Netflix at
10 any given moment.

11 51. On information and belief, the Netflix system “determin[es] a traffic
12 metric representative of a traffic load on the selected network link.”

13 52. For example, during the playback process, which is controlled by
14 Netflix software, the client device “intelligently selects which OCA to use.”¹⁸ “It
15 does this by testing the quality of the network connection to each OCA. It will
16 connect to the fastest, most reliable OCA first.”¹⁹ “The client keeps running these
17 tests throughout the video streaming process. The client probes to figure out the
18 best way to receive content from the OCA.”²⁰ Thus, through the playback process,
19 the Netflix system “determin[es] a traffic metric representative of a traffic load on
20 the selected network link.”

21 53. Upon information and belief, “responsive to the traffic metric,” the
22 Netflix system “regroup[s] flows into second flow lists corresponding to the
23 selected network link.” For example, over the course of the playback session, the
24 Netflix system redirects flows of content to different OCAs and to different URLs
25 on the same OCA (i.e., different “network links”), thereby regrouping the original

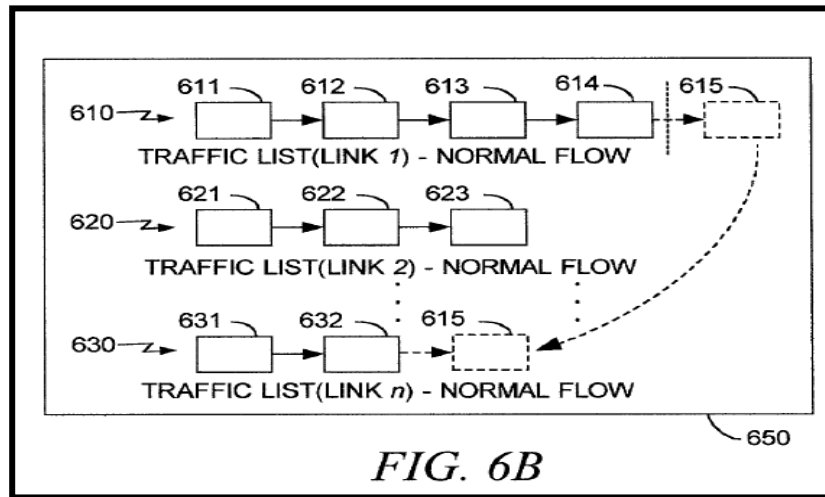
26 ¹⁸ <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

27 ¹⁹ *Id.*

28 ²⁰ *Id.*

1 set of flows from a given OCA or URL to various Netflix subscribers to a second
 2 flow list corresponding to a different OCA or URL.

3 54. The “regrouping balanc[es] the transmission unit traffic upon the
 4 network links.” This element is illustrated, for example, by Figure 6B of the ’079
 5 Patent.



14 Source: ’079 Patent Specification, Fig. 6B.

15 55. For instance, the Netflix system moves flows of content files from one
 16 OCA or URL to another less congested or otherwise more favorable OCA or URL.

17 56. Finally, the Netflix system “transmit[s] the respective second flow list
 18 over the respective selected network link.” For instance, after the Netflix system
 19 transitions a flow of a particular content files to a new OCA or URL, it continues to
 20 transmit all the flows then associated with the new OCA or URL (i.e., the “second
 21 flow list”) to the many Netflix subscribers who are streaming from that OCA or
 22 URL at that moment.

23 57. At least as of on or around September 26, 2019, when the Broadcom
 24 Entities informed Netflix of its infringement of the ’079 Patent, and by no later than
 25 the filing and service of this Complaint, Netflix has had knowledge of the ’079
 26 Patent and Netflix’s infringement thereof.

1 64. The '121 Patent is assigned to Avago, which currently holds all
2 substantial rights, title, and interest in and to the '121 Patent.

3 65. Pursuant to 35 U.S.C. § 282, the '121 Patent is presumed valid.

4 66. The claims in the '121 Patent are directed to an improved network for
5 processing audio and visual ("A/V") data. Specifically, the inventions described in
6 the '121 Patent "relate[] to a network environment in an A/V system using 'A/V
7 decoders,' where the A/V decoders are adapted to process, decode or decompress
8 one or more input data streams."²¹

9 67. The '121 Patent addresses a technical problem in a network processing
10 A/V data. The patent explains that, at the time, there was "no known
11 methodological way to connect video processing modules in A/V systems" and that
12 "[m]ost video processing modules are connected together in an ad hoc manner."²²
13 "As a result, such ad-hoc designs may become difficult to verify, maintain and
14 reuse. Furthermore, as more features are added to the A/V systems...it becomes
15 more difficult to design and integrate such features properly."²³ Thus, there was "a
16 need for an architecture or network that provides a general model illustrating how
17 various video processing modules behave in a network environment."²⁴

18 68. The '121 Patent describes and claims solutions to these technical
19 problems, including specific, novel networks with various features for processing
20 A/V data.

21 69. The inventions described and claimed in the '121 Patent have
22 applications both in the context of the ecosystem within a computer or device, and
23 in more complex computer networking environments. Indeed, as the patent
24 explains, "[m]any modifications and variations of the present invention are possible
25 in light of the above teachings. Thus, it is to be understood that, within the scope of
26

27 ²¹ '121 Patent, 1:42-45.

28 ²² *Id.* at 1:48-51.

²³ *Id.* at 1:51-55.

²⁴ *Id.* at 1:66-2:1.

1 the appended claims, the invention may be practiced otherwise than as described
2 hereinabove.”²⁵

3 70. The systems and methods described in the ’121 Patent improve the
4 functionality of computer networks used for processing A/V data by providing a
5 new, advantageous approach for those networks.

6 71. Claim 1 of the ’121 Patent is directed to:

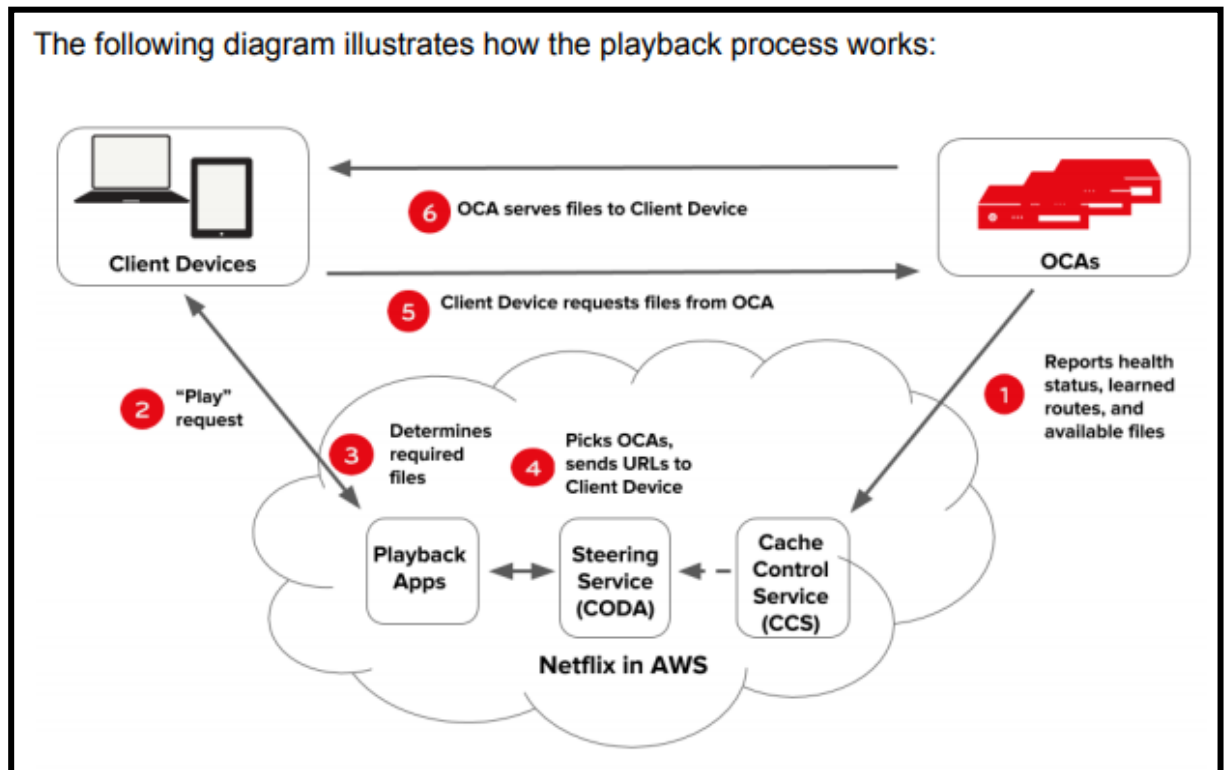
7 A network for processing data configured by a controller
8 to form at least one display pipeline therein by
9 dynamically selecting use of at least two selectable nodes
10 from a plurality of selectable nodes and dynamically
11 concatenating the selected at least two selectable nodes in
the network together, wherein said at least one display
pipeline has an independent data rate and a flow control
module enables said independent data rate.

12 72. Netflix directly infringes the ’121 Patent by making, using, offering to
13 sell, and/or selling in the United States its Netflix service, which utilizes Netflix’s
14 CDN to process audio and visual data in a manner that uses the inventions claimed
15 in the ’121 Patent.

16 73. Upon information and belief, Netflix directly infringes at least
17 independent claim 1 of the ’121 Patent at least in the exemplary manner described
18 below.

19 74. Netflix created, operates, and maintains a “network for processing
20 data,” namely, the CDN that Netflix uses to stream TV shows and movies to its
21 subscribers over the internet, as illustrated in the following diagram from Netflix’s
22 website.

23
24
25
26
27
28 ²⁵ *Id.* at 16:26-30.



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

75. As shown in this diagram, the Netflix CDN consists of three primary groups of systems: (1) the client devices (e.g., smart TVs, computers, mobile phones, etc.) that subscribers use to access the Netflix service with the help software applications that were developed, or partially developed, by Netflix; (2) the Netflix "backend" (referred to in the diagram as "Netflix in AWS"), which receives and processes requests for video content from subscribers; and (3) a network of OCAs, which deliver the video content to the client devices.

76. The CDN is "configured by a controller." For example, when a Netflix user requests playback of a particular title (e.g., a film or television program) using a client device, various computing resources in the Netflix backend, which are controlled by Netflix and run Netflix applications: (1) receive the request; (2) determine which specific streaming assets are required to handle the request, taking individual client characteristics and current network conditions into account; (3) pick OCAs from which the requested streaming assets should be

streamed; and (4) provide a manifest file to the client device containing URLs that specify the OCAs and files needed for the playback process. Thus, Netflix “controls” the path of the delivery of video content through the network to the Netflix subscribers.

2. A user on a client device requests playback of a title (TV show or movie) from the Netflix application in AWS.

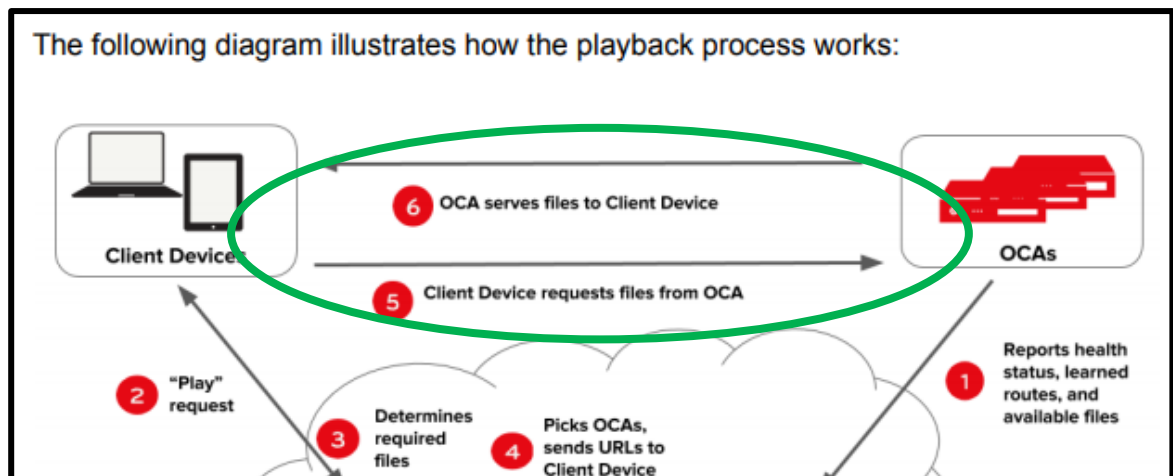
3. The playback application services in AWS check user authorization and licensing, then determine which specific files are required to handle the playback request - taking individual client characteristics and current network conditions into account.

4. The steering service in AWS uses the information stored by the cache control service to pick OCAs that the requested files should be served from, generates URLs for these OCAs, and hands the URLs over to the playback application services.

5. The playback application services hand over URLs of the appropriate OCAs to the client device, and the OCA begins to serve the requested files.

Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

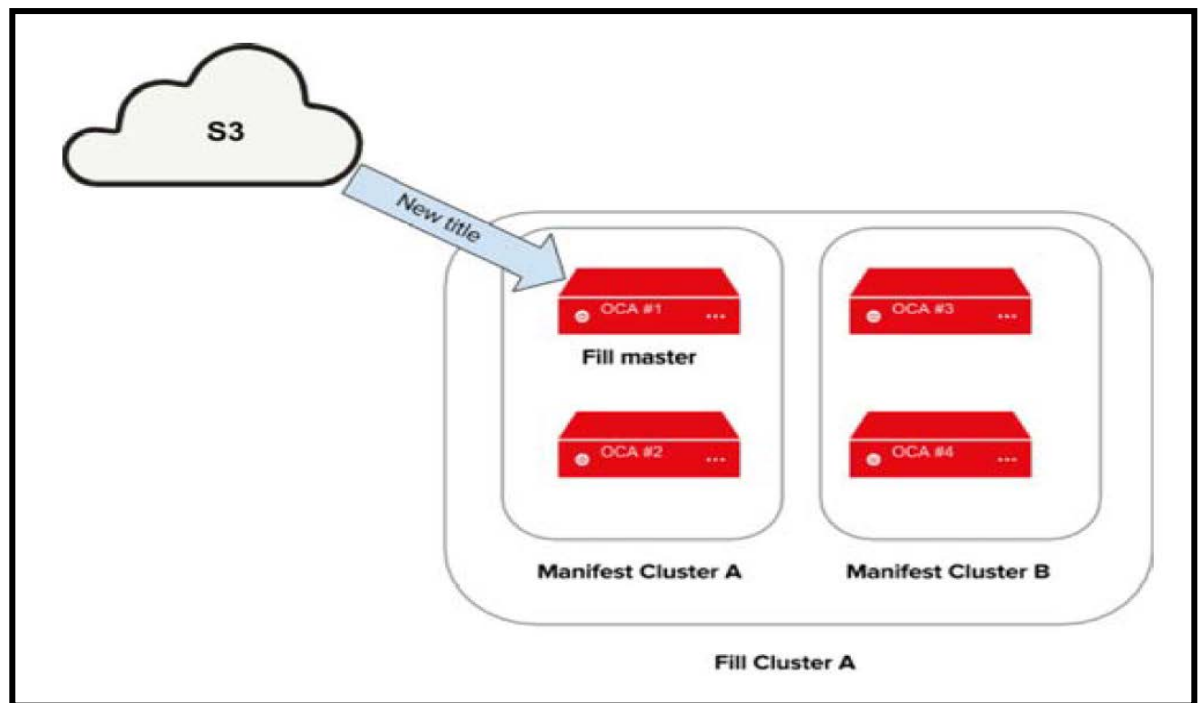
77. Through this process, Netflix’s content delivery network forms “at least one display pipeline therein.” For example, the CDN establishes a link between the client device and one or more Netflix OCAs through which the various video, audio, and other files associated with the requested title are streamed.



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

78. The “at least one display pipeline” within Netflix’s content delivery network is formed “by dynamically selecting use of at least two selectable nodes from a plurality of selectable nodes.” For instance, the content delivery network forms display pipelines between the millions of client devices streaming content from Netflix all over the world (“nodes”) and the numerous OCAs and other servers that Netflix uses to stream its content (also “nodes”).

79. The OCAs periodically receive new content files—which constitute video and other A/V data—from Netflix’s backend systems and from other OCAs.



Source: <https://medium.com/netflix-techblog/netflix-and-fill-c43a32b490c0>.

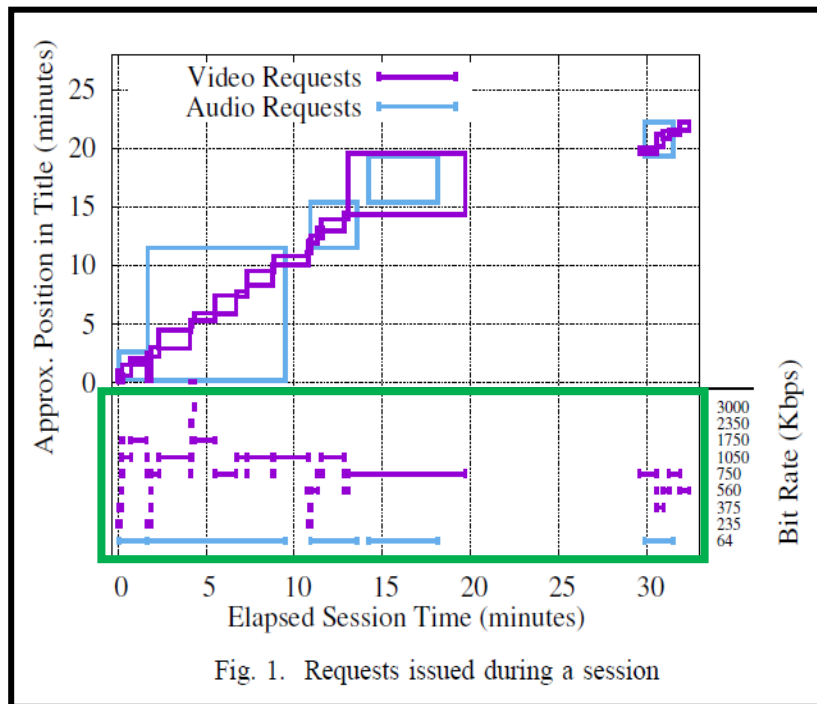
The above image, generated by Netflix, shows that the Netflix local OCAs are supplied with content by Netflix’s more remotely-located repository of content titles, which creates and stores copies of content that has been transcoded into various formats, as discussed above.

1 80. Upon receiving a request for content from a client device, the OCA
2 identifies and sends the specific files requested, filtering out the requested data from
3 the numerous content files contained on the OCA.

4 81. Netflix's selection of nodes is "dynamic." For example, Netflix's
5 "control plane services in AWS take the data that the OCAs report and use it to
6 steer clients via URL to the most optimal OCAs given their file availability, health,
7 and network proximity to the client."²⁶

8 82. As another example of Netflix's "dynamic" selection of nodes, the
9 Netflix system switches between content files and OCAs during the playback
10 process in order to automatically adapt to network conditions and user behavior.
11 This "dynamic" selection is illustrated, for instance, in Figure 1 of *Characterizing*
12 *the Workload of a Netflix Streaming Video Server*, a technical paper published in
13 2016 by the Institute of Electrical and Electronics Engineers ("IEEE"). The portion
14 of this figure within the green box shows, for example, how the Netflix system
15 switches between different video quality levels, or bit rates (shown on the vertical
16 axis), over the course of a Netflix content streaming session (i.e., a Netflix user
17 watching a movie or television program). The horizontal axis shows the elapsed
18 time (in minutes) of that streaming session:

19
20
21
22
23
24
25
26
27
28 ²⁶ <https://openconnect.netflix.com/Open-Connect-Overview.pdf>.



Source: <https://cs.uwaterloo.ca/~brecht/papers/iiswc-netflix-wload-2016.pdf>

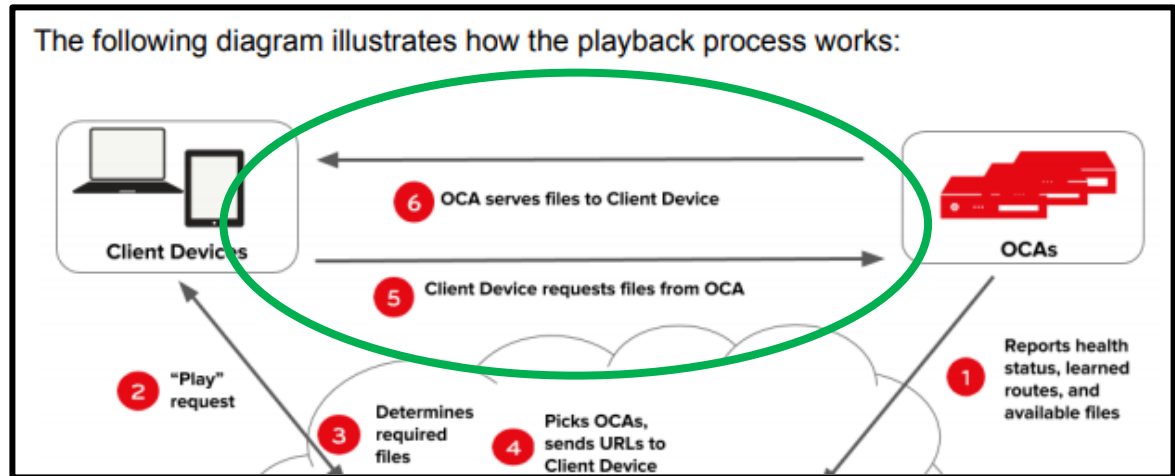
83. The “dynamic” selection has also been described by other third parties that have analyzed the Netflix system, further illustrating how the Netflix application on the client device—which Netflix controls—automatically switches between URLs for video files and OCAs over the course of the streaming session:

- Have you noticed when watching a video the picture quality varies? Sometimes it will look pixelated, and after awhile the picture snaps back to HD quality? That’s because the client is adapting to the quality of the network. If the network quality declines, the client lowers video quality to match. The client will switch to another OCA when the quality declines too much.

Source: <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

84. Thus, the “selection” of the “nodes” is “dynamic” because the node selection can be performed instantaneously, upon the occurrence of network events, and in response to changing network conditions.

85. The Netflix CDN “dynamically concatenate[s] the selected at least two selectable nodes in the network together.” The selectable nodes are “concatenated” when a connection is established between them. For example, as explained above, the network establishes a connection between the client device and the best suited OCA through which the client device, using instructions provided by Netflix, “requests files from the OCA” and the “OCA serves files to the Client Device.”



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

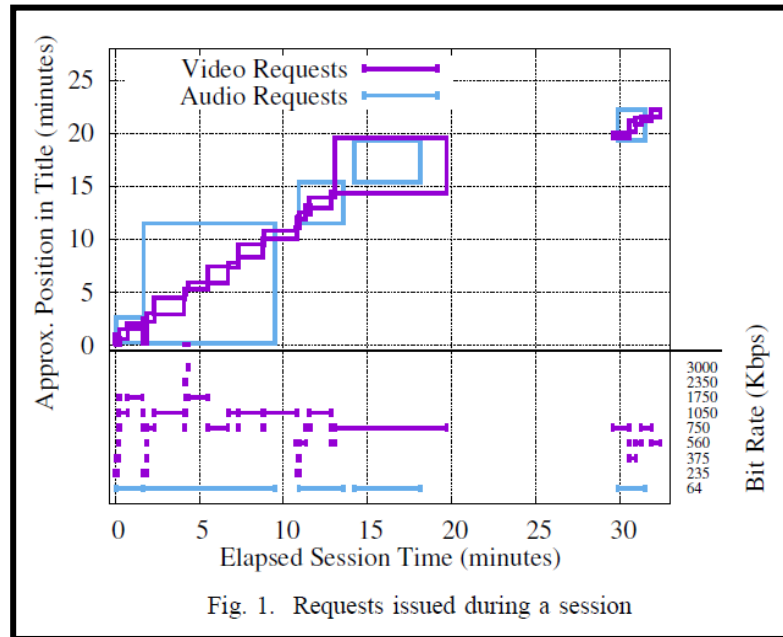
86. As another example of the dynamic nature of the concatenation, the client devices within Netflix’s CDN continue to “adapt to the quality of the network...If the network quality declines, the client lowers video quality to match. The client will switch to another OCA when the quality declines too much.”²⁷

87. The “display pipeline” within the Netflix CDN “has an independent data rate.” For example, on information and belief, the data rate of the transmission of streaming content between the OCA and a given client device within the Netflix system varies from and is not dependent upon the data rate of the transmission of streaming content to the other client devices on the system.

88. The Netflix CDN includes a “flow control module” that “enables said independent data rate.” For instance, the Netflix client serves as a “flow control

²⁷ <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

module” by “pacing” the transmission of the streaming content from the Netflix OCA. As illustrated below, the client device—under the control of the Netflix application operating thereon—controls the pace at which it downloads content files from Netflix to address network issues and user events, including by varying the bit rate of the streaming video content.



bandwidth. Clients continue to download segments sequentially from the same file unless network or server conditions change (which may result in switching to a different bit rate) or a user event occurs (e.g., stopping or skipping to a new title position). Clients that are in a steady state limit the number of segments they download ahead of the playback point to avoid waste when a user event occurs. This is called *pacing* [30] and is a defining feature of HTTP streaming video clients [3]. There is no simple relationship between segments

Source: <https://cs.uwaterloo.ca/~brecht/papers/iiswc-netflix-wload-2016.pdf>

89. Upon information and belief, Netflix controls the playback functionality of the client devices used by subscribers to access the Netflix service. As an example, for many client devices, Netflix develops the software applications used to access the Netflix service. Even in client devices for which Netflix does not develop the application, it “still has control because it controls the software

1 development kit (SDK).” The SDK “is a set of software development tools that
 2 allows the creation of applications. Every Netflix app...plays video using the
 3 SDK.” “By controlling the SDK, Netflix can adapt consistently and transparently
 4 to slow networks, failed OCAs, and any other problems that may arise.”²⁸

5 90. In Netflix’s words , “[t]he SDK provides a rendering engine,
 6 JavaScript runtime, networking, security, *video playback*, and other platform
 7 hooks.”

8 that runs on the metal, and a UI written in JavaScript. The SDK provides a
 9 rendering engine, JavaScript runtime, networking, security, video
 10 playback, and other platform hooks. Depending on the device, SDK

11 Source: [https://medium.com/netflix-techblog/building-the-new-netflix-
 12 experience-for-tv-920d71d875de](https://medium.com/netflix-techblog/building-the-new-netflix-experience-for-tv-920d71d875de)

13 91. At least as of on or around September 26, 2019, when the Broadcom
 14 Entities informed Netflix of its infringement of the ’121 Patent, and by no later than
 15 the date of this Complaint, Netflix has had knowledge of the ’121 Patent and that its
 16 video streaming service infringes the ’121 Patent.

17 92. In addition to direct infringement, Netflix indirectly infringes the ’121
 18 Patent in violation of 35 U.S.C. 271(b) by inducing third-parties to directly infringe
 19 at least claim 1 of the ’121 Patent, at least in the exemplary manner described
 20 above. Netflix has induced, and continues to induce, direct infringement of the
 21 ’121 Patent by customers and/or end users of infringing client devices enabled with
 22 the Netflix software application and service. Netflix knows that it provides and
 23 specifically intends to provide an application for use on client devices that, when
 24 used as intended with the Netflix streaming service, meets the limitations of claim 1
 25 of the ’121 Patent. Netflix knows and specifically intends that its customers and
 26 end users practice the system recited in claim 1 of the ’121 Patent, when using its

27
 28 ²⁸ [http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-
 press-play.html](http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html)

1 application and service as intended. As explained above, the relevant aspects of
 2 that use, including the “playback” process for streaming content, are controlled by
 3 Netflix software on the client devices, along with other Netflix systems.

4 93. Netflix’s knowing and willful infringement of the ’121 Patent has
 5 caused and continues to cause damage to Avago. Avago is entitled to recover
 6 damages sustained as a result of Netflix’s wrongful acts in an amount subject to
 7 proof at trial.

8 **THIRD CLAIM FOR RELIEF**

9 **(Infringement of U.S. Patent No. 8,959,245)**

10 94. The Broadcom Entities reallege and incorporate by reference the
 11 allegations of paragraphs 1-93 set forth above.

12 95. The ’245 Patent, entitled “Multiple Pathway Session Setup to Support
 13 QoS Services” was duly and legally issued on February 17, 2015, from a patent
 14 application filed on November 25, 2008, with Jeyhan Karaoguz and James Bennett
 15 as the named inventors. A copy of the ’245 Patent is attached hereto as **Exhibit C**.

16 96. The ’245 Patent is assigned to Avago, which currently holds all
 17 substantial rights, title, and interest in and to the ’245 Patent.

18 97. Pursuant to 35 U.S.C. § 282, the ’245 Patent is presumed valid.

19 98. The ’245 Patent is directed to an improvement in the functionality of a
 20 communication network, such as the internet.

21 99. Specifically, the ’245 Patent’s claims are directed to a novel system
 22 and method for delivering content to a user through a communication network, in
 23 which a network management server determines multiple routes for delivering the
 24 content based on a provisioning profile for the user device.²⁹

25 100. The inventions of the ’245 Patent resolve technical problems related to
 26 delivering content through the then-existing internet network architecture. At the
 27

28 ²⁹ ’245 Patent, Abstract, 1:44-51, 2:13-40.

1 time, the internet was becoming increasingly important as a commercial
 2 infrastructure and there was a growing need for “massive internet based services,”
 3 such as voice over internet protocol (“VoIP”), video-conferencing, and video
 4 streaming. Much of the then-current internet architecture was based on the “best
 5 effort” model. This architecture attempts to deliver all data traffic “as soon as
 6 possible within the limits of its abilities.” However, the data packets being
 7 transferred “can be dropped indiscriminately” in the event of network congestion.
 8 While this approach worked well for some less time-sensitive applications, such as
 9 email and FTP data transfer, it did not work as well for real-time multimedia
 10 applications, such as streaming video on demand.³⁰

11 101. In light of the growing use of high-bandwidth, time-sensitive
 12 applications, there was a need for technologies to improve the quality of service
 13 (“QoS”) of data transmissions over communication networks like the internet.

14 102. The inventions described in the ’245 Patent address technical problems
 15 associated with the conventional systems and methods for delivering high-quality
 16 video, and other data, including by utilizing multiple routes among the available
 17 routes in the communication network, thereby increasing the reliability of the data
 18 transmission.³¹

19 103. The ’245 Patent describes and claims specific ways to implement this
 20 solution using a network management server capable of determining multiple routes
 21 for delivering the data content based on a “provisioning profile,” thereby better
 22 ensuring delivery of content over the communication network. The ’245 Patent
 23 explains how the provisioning profile may contain information relevant to the
 24 delivery of the content, such as preferred service types, desired quality of service,
 25 client account information, and/or client credit information.³²

27 ³⁰ *Id.* at 1:19-40.

28 ³¹ *See id.* at 2:29-32, 2:41-45.

³² *Id.* at 2:29-32.

104. As the '245 Patent explains, different sets of packets associated with the data content may be transmitted over different routes amongst the multiple available routes to, for instance, take advantage of paths that have less usage and increase reliability. The network management server can manage and prioritize this allocation of routes, which may involve the use of a primary route and/or one or more secondary routes. The network management server can also enable a "handoff" between the routes, such as when QoS degrades on the primary route. The handoff can be seamless to the user, ensuring an uninterrupted user experience.³³

105. The systems and methods described and claimed in the '245 Patent improve the functionality of a computer network by providing a new, advantageous approach for delivering content through the network that enables higher QoS standards, such as those required for video-on-demand applications.

106. Claim 1 of the '245 Patent is directed to:

1. A method for communication, the method comprising:

receiving from a user device, by a network management server via a communication network, a request for a service;

determining multiple routes for delivering content associated with said requested service based on a provisioning profile for said user device;

and delivering said content associated with said requested service via said determined multiple routes.

107. Claim 1 thus claims a novel solution for transmitting digital media content over a communication network via multiple routes, using a network management server and a provisioning profile. This solution was not well-understood, routine, or conventional at the time of the '245 Patent because it claims

³³ *Id.* at 2:41-61.

1 a new and specific improvement over the prior art. The inventions claimed in the
 2 '245 Patent comprise a novel arrangement of streaming content equipment that
 3 results in a better experience for the content user with fewer interruptions.

4 108. Claim 3 of the '245 Patent, which depends from claim 1, is directed to:

5 3. The method according to claim 1, wherein said
 6 provisioning profile comprises preferred service types,
 7 desired QoS for one or more services, client account
 8 information, and/or client credit verification information.

9 109. Claim 6 of the '245 Patent, which also depends from claim 1, is
 10 directed to:

11 6. The method according to claim 1, comprising
 12 allocating via said network management server, one or
 13 more of said determined multiple routes based on priority.

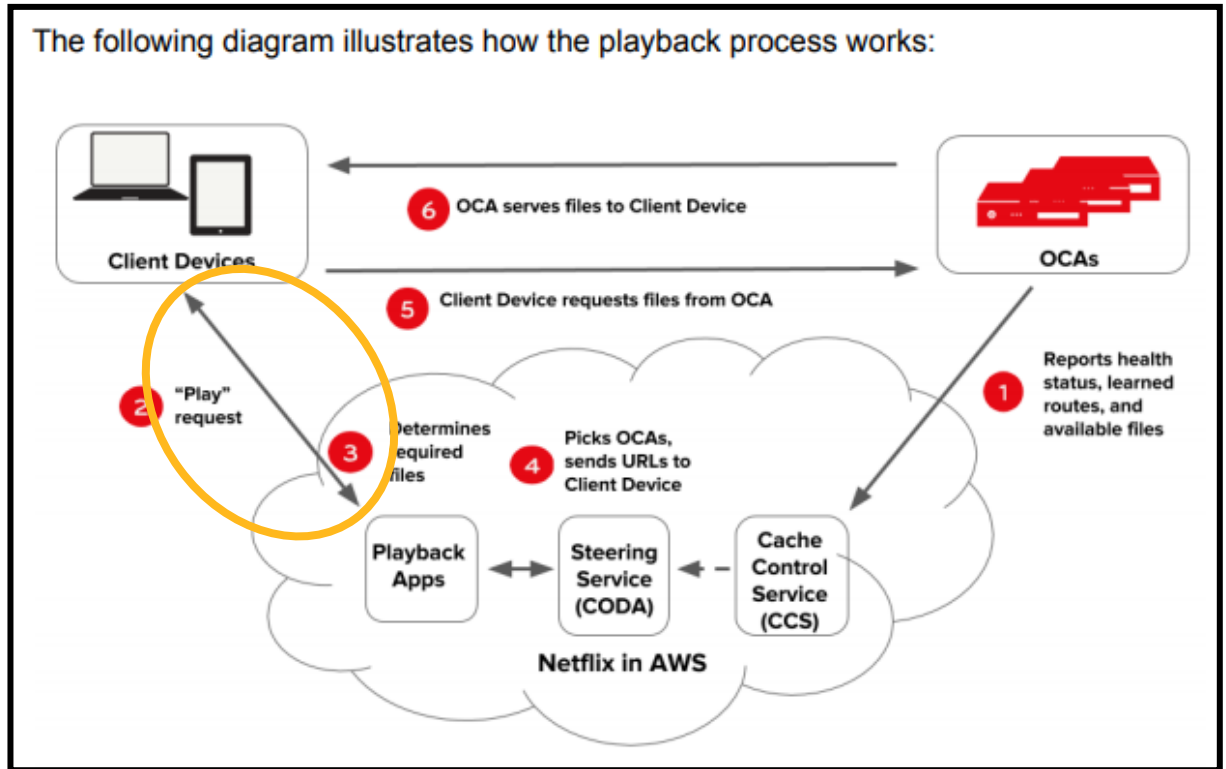
14 110. Thus, claims 3 and 6 further describe the invention's improved method
 15 whereby a network management server determines multiple routes for delivering
 16 content based on a provisioning profile in response to receiving a request for
 17 service from a user device, and then delivers that content via multiple routes based
 18 on priority. The ordered combination of elements in each of claims 3 and 6, in
 19 conjunction with the elements of the claims from which they depend, therefore
 20 recite unconventional, new, and improved digital media content delivery methods
 21 that were not well-understood at the time of the '245 Patent.

22 111. Netflix directly infringes the '245 Patent by making, using, offering to
 23 sell, and/or selling into the United States its Netflix service, which utilizes a
 24 playback system that practices the inventions claimed in the '245 Patent.

25 112. Upon information and belief, Netflix directly infringes at least claims
 26 1, 3, and 6 of the '245 Patent, at least in the exemplary manner described below.

27 113. Netflix practices a "method for communication" that involves
 28 "receiving from a user device, by a network management server via a
 communication network, a request for a service." For example, when a Netflix user

1 uses the Netflix application on their TV, computer, smartphone, or other client
 2 device to view a movie or television program, the Netflix application sends a “play”
 3 request to Netflix’s backend systems. This represents a “request for service” from a



17 “user device.” The Netflix backend systems then determine, using Netflix-
 18 developed solutions, the optimal manner in which to deliver the requested content.

19 Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

20 114. On information and belief, Netflix controls the playback process from
 21 the client side as well. For example, as explained above, Netflix develops many of
 22 the Netflix applications used by client devices to access the Netflix service,
 23 including the applications for Android and iOS devices. As for the client devices
 24 for which Netflix does not develop the application itself, it provides the software
 25 development kit, or SDK, used to create the Netflix application. On information
 26 and belief, the SDK includes the code responsible for the playback process.

115. Netflix's control of the playback process at the client side is confirmed by third party sources:

Netflix Controls The Client

Netflix handles failures gracefully because it controls the client on every device running Netflix.

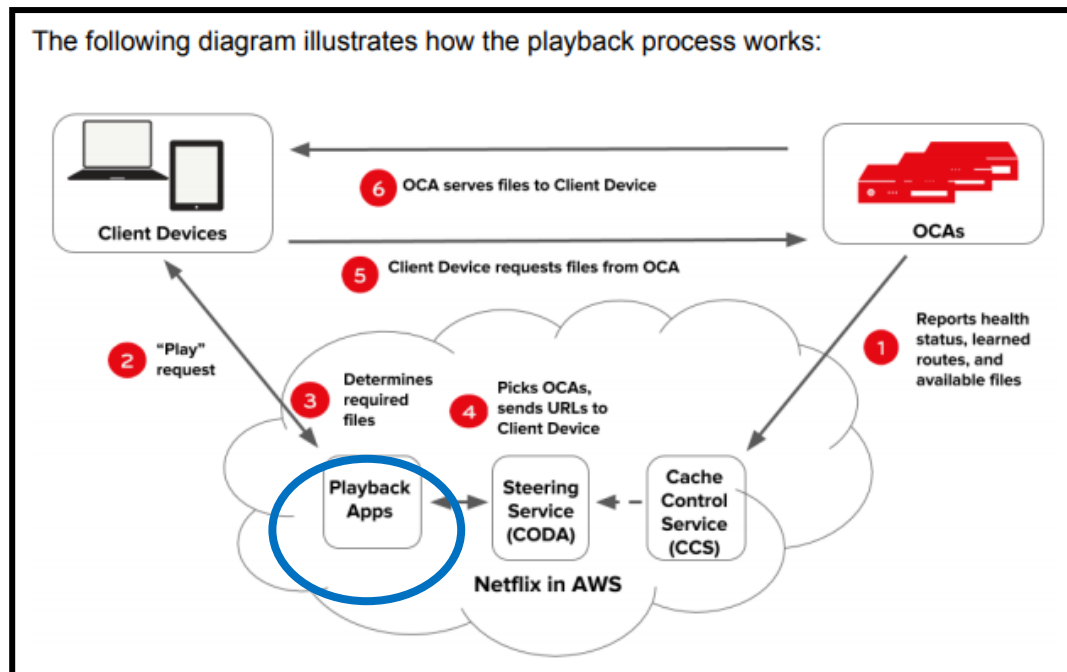
Netflix develops its Android and iOS apps themselves, so you might expect them to control those. But even on platforms like Smart TVs, where Netflix doesn't build the client, Netflix still has control because it controls the *software development kit* (SDK).

A SDK is *a set of software development tools that allows the creation of applications*. Every Netflix app makes requests to AWS and plays video using the SDK.

By controlling the SDK, Netflix can adapt consistently and transparently to slow networks, failed OCAs, and any other problems that might arise.

Source: <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

116. The request for service from the Netflix user's client device is received "by a network management server via a communication network." For example, as shown below, the playback request is received by the Playback Apps service within



1 the Netflix backend, which, on information and belief, operates on one or more
2 servers.

3 Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

4 117. The Netflix system also “determin[es] multiple routes for delivering
5 content associated with said requested service based on a provisioning profile for
6 said user device.” For instance, upon receiving a playback request for a specific
7 title from a Netflix user, the Netflix backend systems determine the video, audio,
8 and other files needed for playback and pick the OCAs from which these files
9 should be streamed to the client device. It then generates a manifest file containing
10 the URLs for the files and OCAs, which the Netflix system sends to the client
11 device.

12 118. As Netflix explains:

- 13 4. The steering service in AWS uses the information stored by the cache control
14 service to pick OCAs that the requested files should be served from, generates
15 URLs for these OCAs, and hands the URLs over to the playback application
16 services.

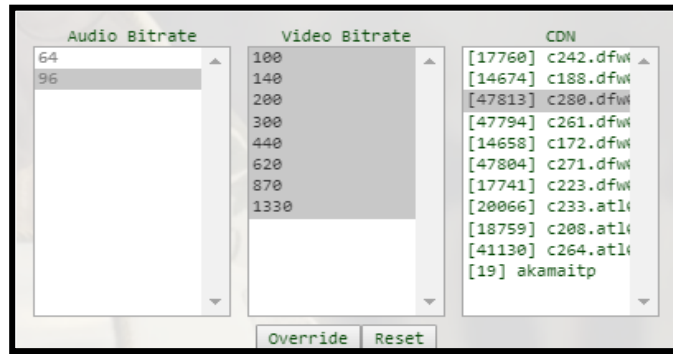
17 5. The playback application services hand over URLs of the appropriate OCAs to
the client device, and the OCA begins to serve the requested files.

18 Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

19 119. Through this process, the Netflix system will select as many as ten
20 different OCAs from which the requested content may be streamed:

- 21 ■ Taking into account all the relevant information, the Playback Apps service returns
22 URLs for up to ten different OCA servers. These are the same sort of URLs you use all
23 the time in your web browser. Netflix uses your IP address and information from ISPs to
identify which OCA clusters are best for you to use.

24 Source: <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>



Source: Screenshot from Netflix Application

120. On information and belief, Netflix bases its selection of OCAs and URLs, at least in part, on a “provisioning profile” for the user device used to access the Netflix service. For instance, upon receiving a playback request for a specific title, the Netflix system “checks user authentication and licensing” and takes “individual client characteristics into account” in selecting the “specific streaming assets” required to handle the playback request and the OCAs from which they should be streamed.

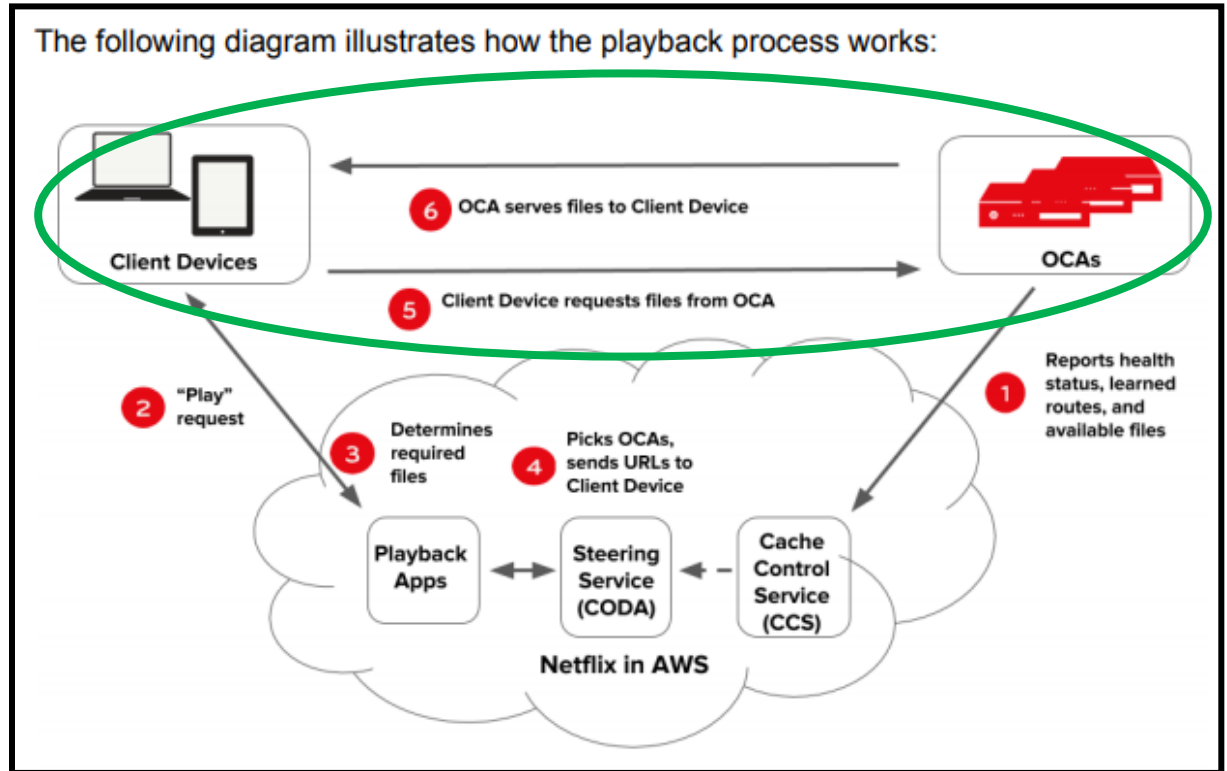
2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and current network conditions into account.
4. The steering service uses the information stored by the cache control service to pick OCAs that the requested video assets should be streamed from, generates URLs for these OCAs, and hands the URLs over to the playback application services.
5. The playback application services hand over URLs of the appropriate OCAs to the client device, and video streaming starts.

Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

121. As further explained in a study published by the IEEE, the client device—under the control of the Netflix application—“indicates the formats of the content it can play. Netflix server then sends back a manifest file based upon the client request. For instance, Netflix client running on an older computer (Thinkpad T60 with Windows XP) and a newer computer (MacBook with Snow Leopard)

1 have different capabilities and received different video downloading format and bit
2 rates.”³⁴

3 122. The Netflix system “deliver[s] said content associated with said
4 requested service via said determined multiple routes.” For example, the Netflix
5 service connects to one or more of the numerous OCAs within Netflix’s CDN and
6 streams video, audio, and other content to the user’s computer.



19 Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

20 123. In doing so, the Netflix application on the client device intelligently
21 selects which of the specific OCAs identified by Netflix to use by evaluating
22 various factors, including the quality of the network connection to each OCA. The
23 client device connects to the fastest, most reliable OCA first, but will switch to
24 another OCA if the video quality declines too much.

28 ³⁴ <https://www.moritzsteiner.de/papers/netflix-hulu.pdf>

- The client intelligently selects which OCA to use. It does this by testing the quality of the network connection to each OCA. It will connect to the fastest, most reliable OCA first. The client keeps running these tests throughout the video streaming process.
- The client probes to figure out the best way to receive content from the OCA.
- The client connects to the OCA and starts streaming video to your device.
- Have you noticed when watching a video the picture quality varies? Sometimes it will look pixelated, and after awhile the picture snaps back to HD quality? That's because the client is adapting to the quality of the network. If the network quality declines, the client lowers video quality to match. The client will switch to another OCA when the quality declines too much.

Source: <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>; <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

124. With regard to claim 3 of the '245 Patent, the Netflix system utilizes a “provisioning profile” comprising “preferred service types, desired QoS for one or more services, client account information, and/or client credit verification information.” For example, the Netflix system maintains account information for its users. This includes, amongst other thing, preferred playback settings, plan details (e.g., standard or HD), user profile information, parental control information, and credit card payment and billing information. On information and belief, some or all of this account information is used by the Netflix system in selecting the content files and OCAs for use in delivering the title selected by the Netflix user.

125. With regard to claim 6, the Netflix system also allocates multiple routes for delivering the streamed content “based on priority.” For instance, on information and belief, the list of OCAs that Netflix provides to the client device via the manifest file at the start of the playback session is ranked according to priority rules established by Netflix.³⁵

³⁵ See, e.g., <http://oc.nflxvideo.net/docs/OpenConnect-Deployment-Guide.pdf>; <https://www.moritzsteiner.de/papers/netflix-hulu.pdf>

126. As alleged above, Netflix directly infringes at least claims 1, 3, and 6 of the '245 Patent.

127. At least as of on or around September 26, 2019, when the Broadcom Entities informed Netflix of its infringement of the '245 Patent, and by no later than the date of this Complaint, Netflix has had knowledge of the '245 Patent and its infringement thereof.

128. In addition to direct infringement, Netflix has induced, and continues to induce, direct infringement by third-parties of at least claims 1, 3, and 6 of the '245 Patent, at least in the exemplary manner described above, by actively encouraging their use of the Netflix system, in violation 35 U.S.C. § 271(b). Netflix has induced, and continues to induce, direct infringement of the '245 Patent by customers and/or end users of playback devices enabled with the Netflix software application and service. Netflix knows that it provides and specifically intends to provide an application for use on playback devices that, when used as intended with the Netflix streaming service, meets the limitations of claims 1, 3, and 6 of the '245 Patent. Netflix knows and specifically intends that its end users practice the method recited in claims 1, 3, and 6 of the '245 Patent, when using its application and service as intended.

129. Netflix's knowing and willful infringement has caused and continues to cause damage to Avago, and Avago is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

FOURTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 8,270,992)

130. The Broadcom Entities reallege and incorporate by reference the allegations of paragraphs 1-129 set forth above.

131. The '992 Patent, entitled "Automatic Quality of Service Based Resource Allocation," was duly and legally issued on September 18, 2012, from a patent application filed on July 18, 2011, with Jeyhan Karaoguz and James D.

1 Bennett as the named inventors. The '992 Patent claims priority to U.S. Provisional
2 Application No. 60/504,876, filed on September 22, 2003. A copy of the '992
3 Patent is attached hereto as **Exhibit D**.

4 132. The '992 Patent is assigned to Avago, which holds all substantial
5 rights, title, and interest in and to and '992 Patent.

6 133. Pursuant to 35 U.S.C. § 282, the '992 Patent is presumed valid.

7 134. The '992 Patent is directed to an improvement in the functionality of a
8 communication network used to provide a digital media service, such as video
9 streaming over the internet.

10 135. Specifically, the '992 Patent describes and claims a new system and
11 method for delivering digital media service to users in a dynamic communication
12 network. The invention allocates and utilizes resources from other systems on the
13 network in order to provide the user with the digital media service at a higher
14 quality level than the quality level they are currently experiencing. As explained in
15 the specification, the resource allocation and utilization can be determined by
16 quality control modules and communication modules. These modules
17 communicate various capability information about the network, such as processing
18 capability, communication capability, and information access capability. The
19 quality control modules can determine whether utilizing resources of another
20 system will provide the service to the user at a higher quality level. If it so
21 determines, a distributing processing module can manage the resource allocation.³⁶

22 136. The inventions of the '992 Patent address technical problems related to
23 delivering content over unstable network environments, an issue with prior art
24 system existing at the time. In a dynamic and unstable network environment,
25 processing resources continuously toggle between available and unavailable. These
26 processing resources may offer service capabilities that are superior or inferior to
27

28 ³⁶ See '992 Patent, 2:8-32.

1 other resources present in the network. For example, a system providing low
2 quality audio or video service may communicate with a second system capable of
3 providing a higher quality audio or video service.³⁷

4 137. The first system providing the lower quality service needs a way of
5 both accessing the system with superior resources, utilizing those resources, and
6 ultimately delivering content from the superior resource in order to provide higher
7 quality content to the user.

8 138. The '992 Patent, therefore, addresses the technical problem of ensuring
9 delivery of content at the highest quality level in an unstable network environment
10 by utilizing a quality-of-service based network resource allocation delivery system.

11 139. The '992 Patent claims specific ways to solve these technical problems
12 with a digital media delivery system that is capable of automatically determining
13 bandwidth capability information in multiple systems, using that information to
14 obtain digital media content at a higher quality level than the current quality, and
15 then delivering that higher quality digital media content to the user.³⁸ In this
16 manner, it improves the functionality of computer networks used to deliver media
17 content by providing a new, advantageous approach for those networks.

18 140. The digital media delivery system disclosed and claimed in the '992
19 Patent uses novel quality control modules, resource allocation modules, and
20 distributing processing modules to assess, manage, and use resources in a network
21 environment where the availability of resources may vary between systems in the
22 network.

23 141. In a two-system network environment, for example, if it is determined
24 that the second system has access to higher quality audio or video content than the
25 first system, then the first system may receive higher quality data from the second

26 _____
27 ³⁷ *Id.* at 1:39-42 (describing delivery of high quality audio), 1:42-49 (describing
delivery of high quality video).

28 ³⁸ *Id.* at 3:1-9:16 (describing Figure 1, the method for quality of service based
resource allocation and utilization in a dynamic wireless network).

1 system for further processing. The first system can also receive higher quality data
 2 from the system for immediate delivery and presentation to the user. Alternatively,
 3 the first system may direct the second system to provide the higher quality data
 4 directly to the user.³⁹

5 142. Claim 1 of the '992 Patent reads as follows:

6 1. In a portable system, a method for providing a digital
 7 media service to a user, the method comprising:

8 delivering digital media content having a current quality
 level to a user;

9 determining that a network connection with a second
 10 system is available and is characterized by a
 11 communication bandwidth that is high enough to provide
 the digital media content to the user at a quality level
 higher than the current quality level;

12 using the network connection to obtain the digital media
 13 content at the higher quality level from the second
 system; and

14 delivering the digital media content at the higher quality
 15 level to the user instead of the digital media content at the
 current quality level.

16 143. Claim 1 thus recites a novel solution of determining, accessing, and
 17 using the resources of another system on a dynamic network environment in order
 18 to improve digital media content quality delivered to the user by obtaining the
 19 higher quality content from the second system in a manner that was not well-
 20 understood, routine, or conventional at the time of the '992 Patent. In the prior art,
 21 among other things, the source of the content remained the same between the two
 22 systems, whereas claim 1 claims obtaining the digital media content at a higher
 23 quality level from another source, namely the second system.

24 144. Claims 2, 3 and 5 of the '992 Patent depend from claim 1.

25 145. Claim 2 of the '992 Patent reads as follows:

26 2. The method of claim 1, where the digital media
 27 content is video media.

28 ³⁹ *Id.* at 14:42-53.

1 146. Claim 3 of the '992 Patent reads as follows:

2 3. The method of claim 1, where the digital media content
3 is audio media.

4 147. Claim 5 of the '992 Patent reads as follows:

5 5. The method of claim 1, where the portable system
6 automatically performs, without user interaction, said
7 determining, said using, and said delivering the digital
8 media content at the higher quality level to the user.

9 148. Netflix directly infringes the '992 Patent by making, using, selling, and
10 offering to sell the Netflix service, which practices the patented invention.

11 149. Upon information and belief, Netflix directly infringes at least claims
12 1, 2, 3, and 5 of the '992 Patent, at least in the exemplary manner described below.

13 150. Netflix practices a “method for providing a digital media service to a
14 user” “[i]n a portable system.”

15 151. For example, the Netflix system provides a streaming entertainment
16 service that delivers digital video content such as TV series, documentaries, and
17 feature films to a wide variety of internet-connected devices, including mobile
18 devices such as laptops, tablets, and mobile phones.

19 152. The Netflix system “deliver[s] digital media content having a current
20 quality level to a user.” For example, the Netflix system delivers the “best video
21 quality stream” to its users “tailored to the member’s available bandwidth and view
22 device capability.” To account for variable network conditions, the Netflix
23 streaming service encodes video titles at different bit rates so that video can be
24 delivered at different quality levels. The Netflix streaming service “pre-encode[s]
25 streams at various bitrates applying optimized encoding recipes.”⁴⁰

26 153. The Netflix system “determin[es] that a network connection with a
27 second system is available and is characterized by a communication bandwidth that

28 ⁴⁰ <https://medium.com/netflix-techblog/per-title-encode-optimization-7e99442b62a2>

1 is high enough to provide the digital media content to the user at a quality level
 2 higher than the current quality level.” For instance, the Netflix streaming service is
 3 designed to adapt to changing network conditions so that the content can be
 4 delivered and viewed at high levels of quality even when network conditions
 5 become constrained. Throughout a user’s playback session, the Netflix streaming
 6 service continuously monitors the network to evaluate changing conditions and
 7 makes adjustments to the video that is being delivered.

8 154. The client device running the Netflix application, which is being
 9 directed and controlled by Netflix code and other instructions, “intelligently selects
 10 which OCA to use. It does this by testing the quality of the network connection to
 11 each OCA. It will connect to the fastest, most reliable OCA first.” The client
 12 device “keeps running these tests throughout the video streaming process.” “If the
 13 network quality declines, the client lowers video quality to match. The client will
 14 switch to another OCA when the quality declines too much.” The Netflix streaming
 15 service uses the user’s IP address and information from internet Service Providers
 16 to adapt to the quality of the network. The Netflix streaming service also identifies
 17 which Open Connect Appliance (OCA) clusters are best for the user’s client to use.
 18 It selects which OCA to use by testing the quality of the network connection to
 19 each OCA and will connect to the fastest, most reliable OCA first. This process is
 20 repeated throughout the user’s video streaming experience.⁴¹

21 155. Netflix “us[es] the network connection to obtain the digital media
 22 content at the higher quality level from the second system” and “deliver[s] the
 23 digital media content at the higher quality level to the user instead of the digital
 24 media content at the current quality level.” For example, the client device
 25 continues to test the quality of the network connection at each OCA throughout the
 26 video streaming process. “If the network quality declines, the client lowers video

27 _____
 28 ⁴¹ <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

1 quality to match.” The client will switch to another OCA when the quality declines
 2 too much. “The Netflix streaming service adapts to the quality of the network and
 3 will adjust the video quality by switching to another OCA when the quality declines
 4 too much.”⁴² After establishing a connection to another OCA capable of streaming
 5 content at a higher quality level—for example, streaming video at a higher
 6 resolution—the Netflix system will provide that content at the higher quality level.

7 156. With regard to claims 2 and 3, the Netflix system provides “digital
 8 media content” in the form of “video media” and “audio media.” For example,
 9 during the playback process, the Netflix system streams the video and audio files
 10 associated with the title being viewed.

11 157. With regard to claim 5, within the Netflix system, the “portable system
 12 automatically performs, without user interaction, said determining, said using, and
 13 said delivering the digital media content at the higher quality level to the user.” For
 14 example, Netflix’s process of identifying and using the best OCA and bitrate is
 15 generally performed by the Netflix system automatically and without user
 16 interaction. As Netflix has explained, the Netflix application on the client device
 17 “runs adaptive streaming algorithms which instantaneously select the best encode to
 18 maximize video quality while avoiding playback interruptions due to rebuffers.”

19 158. Netflix has infringed, and continues to infringe, at least claim 1 of the
 20 ’992 Patent in the United States by making, using, offering for sale, selling, and/or
 21 importing the Netflix streaming service, in violation of 35 U.S.C. § 271(a).

22 159. At least as of on or around September 26, 2019 when the Broadcom
 23 Entities informed Netflix of its infringement of the ’992 Patent, and by no later than
 24 the date of this Complaint, Netflix has had knowledge of the ’992 Patent and that its
 25 video streaming service infringes the ’992 Patent.

26
 27
 28 ⁴² *Id.*

1 160. In addition to its own direct infringement, Netflix has induced, and
 2 continues to induce, direct infringement by third-parties of at least claims 1, 2, 3,
 3 and 5 of the '992 Patent, at least in the exemplary manner described above, by
 4 actively encouraging their use of the Netflix system, in violation 35 U.S.C. §
 5 271(b). For example, Netflix has induced, and continues to induce, direct
 6 infringement of the '992 Patent by customers and/or end users of infringing
 7 playback devices enabled with the Netflix software application and service. Netflix
 8 knows that it provides and specifically intends to provide an application for use on
 9 playback devices that, when used as intended with Netflix streaming service, meets
 10 the limitations of claims 1, 2, 3, and 5 of the '992 Patent. Netflix knows and
 11 specifically intends that its end users practice the method recited in claims 1, 2, 3,
 12 and 5 of the '992 Patent, when using its application and service as intended.

13 161. Netflix's knowing and willful infringement has caused and continues
 14 to cause damage to Broadcom, and Broadcom is entitled to recover damages
 15 sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

16 **FIFTH CLAIM FOR RELIEF**

17 **(Infringement of U.S. Patent No. 6,341,375)**

18 162. The Broadcom Entities reallege and incorporates by reference the
 19 allegations of paragraphs 1-161 set forth above.

20 163. The '375 patent, entitled "Video on demand DVD system" was duly
 21 and legally issued on January 22, 2002, from a patent application filed on July 14,
 22 1999, with Daniel Watkins as the named inventor. A copy of the '375 Patent is
 23 attached hereto as **Exhibit E**.

24 164. The '375 Patent is assigned to Broadcom Corp., which holds all
 25 substantial rights, title, and interest in and to the '375 Patent.

26 165. Pursuant to 35 U.S.C. § 282, the '375 Patent is presumed valid.

27 166. The '375 Patent claims are directed to a new method for distributing
 28 video. The '375 Patent describes systems and methods in which the video is

1 delivered to the user on-demand using a drive server, a control server, and one or
 2 more decoder devices.⁴³ The system can process multiple compressed video
 3 streams in response to multiple user requests for video content.⁴⁴

4 167. The '375 Patent includes embodiments described in the specification
 5 referring to delivery of compressed video data originating from DVD
 6 media. However, the systems and methods are not limited to compressed video
 7 from a particular source and, as the patent explains, a person having ordinary skill
 8 in the art would recognize that the systems and methods described and claimed in
 9 the '375 Patent can be applied to other video distribution models.⁴⁵ The Netflix
 10 streaming service is an example of an application of the systems and methods
 11 described in the claims.

12 168. The inventions of the '375 Patent resolve technical problems related to
 13 conventional video-on-demand systems that require the use of physical connections
 14 and short distances between the sources of video and the decoders or players on
 15 which the end-user views video content.⁴⁶ In prior art systems, each user has a
 16 dedicated video system, such as a DVD player, and decoder at the user's location.⁴⁷

17 169. The '375 Patent, therefore, addresses the technical problem of ensuring
 18 delivery of compressed video content to multiple remote end user locations.⁴⁸

19 170. The '375 Patent claims specific ways to solve these technical problems
 20 with a video on demand system that is centrally managed and implemented by a
 21 drive server, a control server, and one or more decoder devices. Each of these
 22 servers can process one or more compressed video streams in response to one or
 23 more request signals initiated by a user requesting a video.

24 171. Claim 15 claims an improved method of distributing video:

25 _____
 26 ⁴³ '375 Patent at 1:56-60.

⁴⁴ *Id.* at 1:60-63.

⁴⁵ *Id.* at 5:38-43.

⁴⁶ *Id.* at 1:14-41.

⁴⁷ *Id.* at 1:26-27, 33-35.

⁴⁸ *See id.* at 1:56-2:8.

1 A method for distributing video comprising the steps of:

2 (A) presenting a plurality of compressed data streams
3 with a drive server to a control server in response to one
4 or more first control signals;

5 (B) distributing said one or more compressed data streams
6 received from said drive server with said control server to
7 one or more decoder devices in response to one or more
8 request signals;

9 (C) decoding at least one of said one or more compressed
10 data streams with said one or more decoders in response
11 to receiving said one or more compressed data streams
12 from said control server; and

13 (D) presenting at least one signal selected from a
14 decoded video signal and a decoded audio signal in
15 response to decoding said at least one of said one or more
16 compressed data streams, wherein at least one of said one
17 or more decoders is disposed in a separate room from said
18 control server and said driver server, wherein a first
19 portion of a selected one of said compressed data streams
20 is presented to one of said decoder devices and a second
21 portion of said selected compressed data stream is
22 presented to another of said decoder devices.

23 172. Claim 15 thus recites a novel solution of a drive server presenting
24 multiple compressed video streams and delivering those streams to multiple
25 decoder devices in a remote location in a manner that was not well-understood,
26 routine, or conventional at the time of the '375 Patent, resulting in a better video on
27 demand system.

28 173. Netflix directly infringed the '375 Patent by making, using, offering to
sell, and/or selling in the United States its Netflix service, which utilizes the
inventions claimed in the '375 Patent to deliver streaming video content. Although
the '375 Patent is presently expired, Netflix infringed the '375 Patent prior to its
expiration as described below. Avago thus is entitled to damages for Netflix's
unauthorized use of the inventions described in the '375 Patent prior to its
expiration.

1 174. Upon information and belief, the Netflix streaming service directly
2 infringed at least claim 15 of the '375 Patent, at least in the exemplary manner
3 described below.

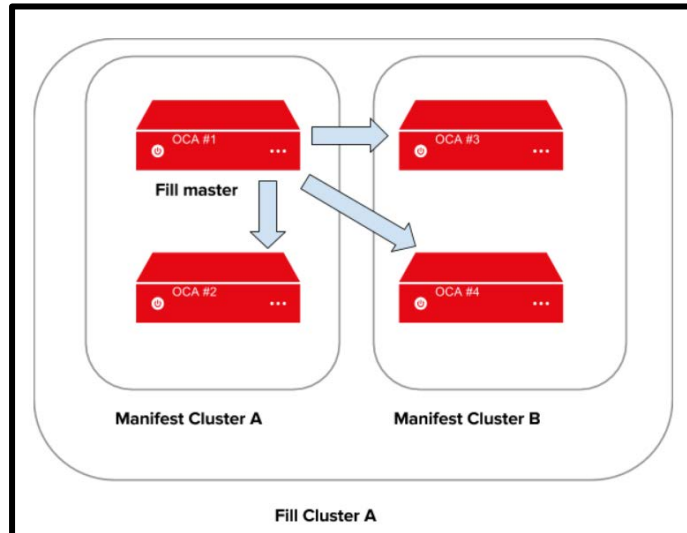
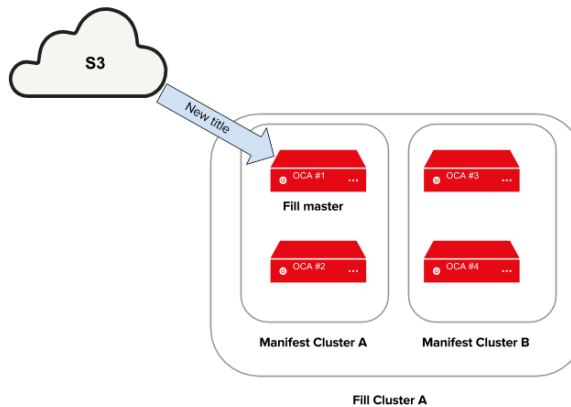
4 175. Netflix practices a “method for distributing video” and “presenting a
5 plurality of compressed data streams with a drive server to a control server in
6 response to one or more first control signals.” The Netflix streaming service is a
7 streaming entertainment service that delivers video content using the Netflix CDN,
8 OCAs, and S3 servers. For example, the Netflix video titles (“data streams”) are
9 presented from an S3 server or an OCA (“drive server”) and sent to other OCAs
10 (“control server”), which store and serve video content.

11 176. The Netflix system presents the compressed data streams to OCAs in
12 response to one or more control signals. For example, as Netflix describes, “OCAs
13 communicate at regular intervals with the control plane services, requesting (among
14 other things) a manifest file that contains the list of titles they should be storing and
15 serving to members. If there is a delta between the list of titles in the manifest file
16 and what they are currently storing, each OCA will send a request, during its
17 configured fill window, that includes a list of the new or update titles that it needs.
18 The response from the control plan in AWS is a ranked list of potential download
19 locations, a.k.a. *fill sources*, for each title.”⁴⁹

20
21
22
23
24
25
26
27
28 ⁴⁹ <https://netflixtechblog.com/netflix-and-fill-c43a32b490c0>

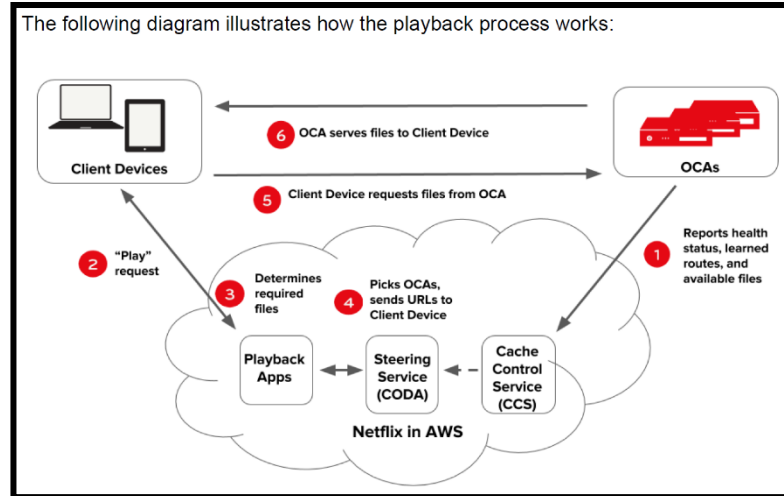
Example Scenario

After the fill master OCA has completed its S3 download, it reports back to the control plane that it now has the title stored. The next time the other OCAs communicate with the control plane to request a fill source for this title, they are given the option to fill from the fill master.



Source: <https://medium.com/netflix-techblog/netflix-and-fill-c43a32b490c0>

177. Netflix performs the step of “distributing said one or more compressed data streams received from said drive server with said control server to one or more decoder devices in response to one or more request signals.” For example, in response to a “Play” request, the Netflix CDN delivers content to Netflix client applications (“Netflix Playback Apps”), which are installed on Netflix’s customers’ client devices, including digital televisions, desktop computers, laptop computers, tablet computers, and mobile phones (“decoder devices”). The Netflix CDN is illustrated in the following diagram:



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

178. Netflix practices “decoding at least one of said one or more compressed data streams with said one or more decoders in response to receiving said one or more compressed data streams from said control server.” For instance, upon receipt of the compressed video, audio, and other content from the Netflix CDN at the client device, the content is decoded so that it can be viewed. On information and belief, the Netflix Playback Apps decode, or cause the decoding of, the content. In Netflix’s own words, “[t]he app uses several approaches for video playback on Android such as hardware decoder, software decoder, OMX-AL, iOMX”).

Device Diversity

To put device diversity in context, we see almost around 1000 different devices streaming Netflix on Android every day. We had to figure out how to categorize these devices in buckets so that we can be reasonably sure that we are releasing something that will work properly on these devices. So the devices we choose to participate in our continuous integration system are based on the following criteria.

- We have at least one device for each playback pipeline architecture we support (The app uses several approaches for video playback on Android such as hardware decoder, software decoder, OMX-AL, iOMX).
- We choose devices with high and low end processors as well as devices with different memory capabilities.
- We have representatives that support each major operating system by make in addition to supporting custom ROMs (most notably CM7, CM9).
- We choose devices that are most heavily used by Netflix Subscribers.

Source: <http://techblog.netflix.com/2012/03/testing-netflix-on-android.html>

179. On information and belief, Netflix directs and controls the playback process associated with the Netflix service from the client side through its control of the Netflix Playback Apps. Netflix develops its own Android and iOS Netflix Playback Apps for Android and Apple devices. Netflix also develops its own software development kit (SDK) to control third party development of Netflix applications on platforms like Smart TVs.

Netflix Controls The Client

Netflix handles failures gracefully because it controls the client on every device running Netflix.

Netflix develops its Android and iOS apps themselves, so you might expect them to control those. But even on platforms like Smart TVs, where Netflix doesn't build the client, Netflix still has control because it controls the *software development kit* (SDK).

A SDK is a *set of software development tools that allows the creation of applications*. Every Netflix app makes requests to AWS and plays video using the SDK.

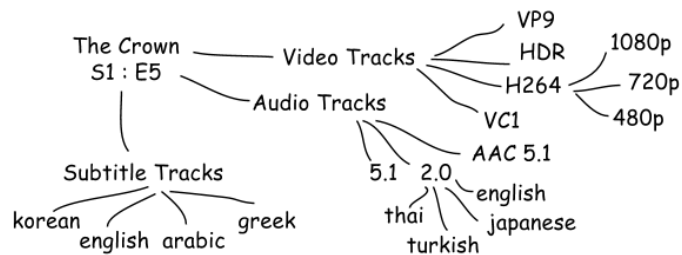
By controlling the SDK, Netflix can adapt consistently and transparently to slow networks, failed OCAs, and any other problems that might arise.

Source: <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

180. Netflix practices “presenting at least one signal selected from a decoded video signal and a decoded audio signal in response to decoding said at least one of said one or more compressed data streams, wherein at least one of said one or more decoders is disposed in a separate room from said control server and said driver server, wherein a first portion of a selected one of said compressed data streams is presented to one of said decoder devices and a second portion of said selected compressed data stream is presented to another of said decoder devices.”

181. For example, a Netflix client—under the control of the Netflix Playback App—presents “at least one signal selected from a decoded video signal and a decoded audio signal in response to decoding said at least one of said one or more compressed data streams” by playing the decoded video, audio, and other Netflix content on the client device. As illustrated below, Netflix content consists of video and audio data, in addition to other data types.

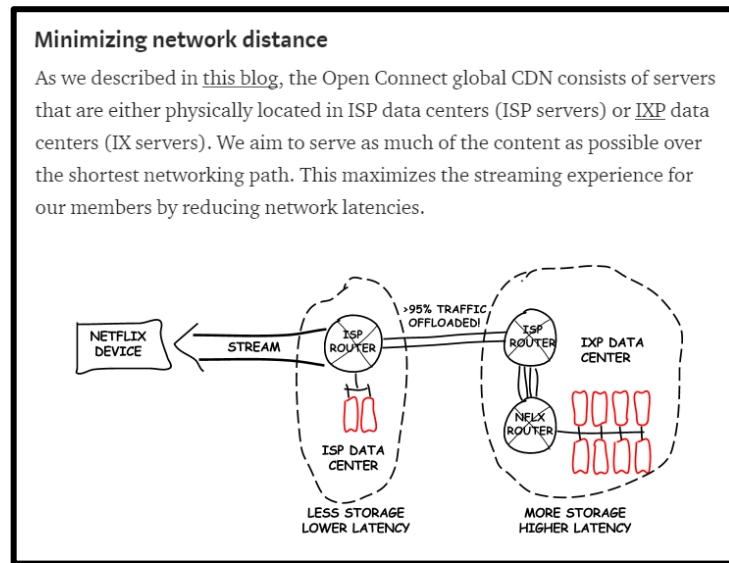
So for each quadruple of (title, encoding profile, bitrate, language), we need to cache one or more files. As an example, for streaming one episode of The Crown we store around 1,200 files!



Source: <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>

182. Further, in the Netflix system, “at least one of said one or more decoders is disposed in a separate room from said control server and said drive server.” For example, the Netflix CDN processes video requests and audio requests using OCAs that Netflix strategically positions all over the world based on the location of Netflix’s users. The OCAs are located within internet exchange points

1 in significant Netflix markets and are interconnected with internet service
 2 providers.⁵⁰ Similarly, the S3 is located on one or more servers in data centers that
 3 are, generally speaking, far removed from the users streaming Netflix content. In
 4 contrast, Netflix users can stream Netflix content virtually anywhere with an
 5 internet connection in numerous countries around the world. Thus, the “one or
 6 more decoders” (in the client devices) “is disposed in a separate room” from the
 7 “control server” (the OCA) and the “drive server” (the S3 or OCA).

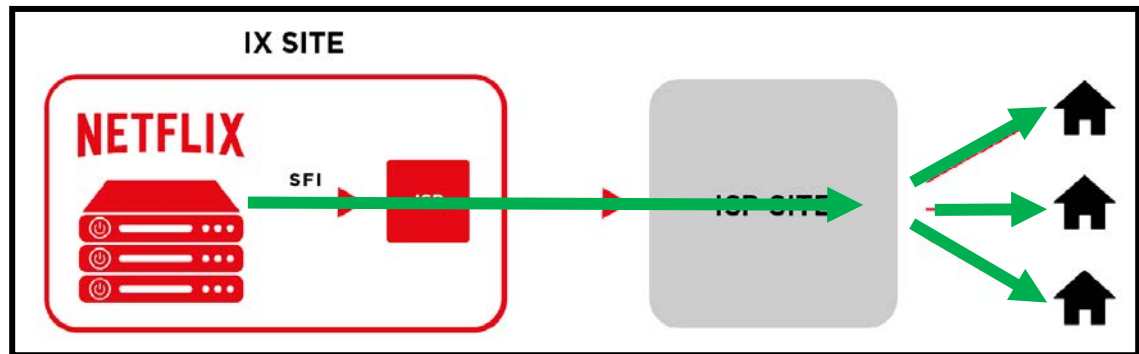


Source: <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>

183. Finally, within the Netflix system, “a first portion of a selected one of
 said compressed data streams is presented to one of said decoder devices and a
 second portion of said selected compressed data stream is presented to another of
 said decoder devices.” For example, on information and belief, when video, audio,
 and other data associated with Netflix content is streamed from an OCA to the
 numerous client devices obtaining Netflix content from that OCA, the data travels
 through the internet infrastructure in a compressed data stream. The relevant

⁵⁰ <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

portions of that stream are then routed and presented to the appropriate client device.



Source: <https://openconnect.netflix.com/Open-Connect-Overview.pdf>

(arrows added).

184. Netflix directly infringed claim 15, at least as described, when the Netflix Playback Apps, Netflix CDN, and Netflix OCAs are used to stream video content to multiple users in remote and different locations.

185. Netflix's infringement caused damage to Broadcom Corp., and Broadcom Corp. is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

SIXTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 8,572,138)

186. The Broadcom Entities reallege and incorporate by reference the allegations of paragraphs 1-185 set forth above.

187. The '138 Patent, entitled "Distributed Computing System Having Autonomic Deployment of Virtual Machine Disk Images," was duly and legally issued on October 29, 2013, from a patent application filed March 30, 2007, with Jagane Sundar, Sanjay Radia, and David A. Henseler as the named inventors. A copy of the '138 Patent is attached hereto as **Exhibit F**.

188. The '138 Patent is assigned to Avago, which holds all substantial rights, title, and interest in and to the '138 Patent.

189. Pursuant to 35 U.S.C. § 282, the '138 Patent is presumed valid.

1 190. The '138 Patent is directed to an improvement in the functionality of a
2 complex distributed computing system. Specifically, the '138 Patent claims a new
3 method and system for a distributed computing environment that conforms to a
4 multi-level, hierarchical organizational model.

5 191. Traditional distributed computing systems faced the significant
6 challenge of providing an organizational structure that could handle the deployment
7 and administration of thousands of virtual computing resources that could carry out
8 millions of operations simultaneously. As the '138 Patent explains, an enterprise
9 environment—such as a large business organization—often includes several
10 business groups, and each group may have competing and variable computing
11 requirements that necessitate separate, independent computing devices connected to
12 each other on the network. However, the diversity of competing and variable
13 computing requirements increases the cost of distributed computing systems
14 because of the increased time and expense associated with the management of
15 resources that need to be customized to the unique computing requirements of each
16 business group.⁵¹

17 192. The '138 invention solved the technical problems presented by such
18 traditional distributed computing systems by developing an infrastructure
19 management facility (“IMF”) that guarantees reliable and efficient application
20 service delivery independent of the computational infrastructure. The IMF includes
21 the implementation of virtual machine managers, or “container services,” capable
22 of managing other container services and virtual machines (“VMs”). The virtual
23 machines, managed by the VM managers, then appear on the network as available
24 resources as if they were independent computing resources that can be accessed by
25 various groups and utilized to suit their highly-diverse and specialized computing
26 needs.⁵²

27
28 ⁵¹ '138 Patent at 1:16-33.

⁵² *Id.* at 8:32-36, 9:65-67, 32:65-33:6.

1 193. Claim 1 recites an improved method of distributing software “images”
 2 (i.e. computer programs) via a number of virtual machines to application nodes:

3 A distributed computing system comprising:

4 a software image repository comprising non-transitory,
 5 computer-readable media operable to store: (i) a plurality
 6 of image instances of a virtual machine manager that is
 7 executable on a plurality of application nodes, wherein
 8 when executed on the applications nodes, the image
 9 instances of the virtual machine manager provide a
 10 plurality of virtual machines, each of the plurality of
 11 virtual machine operable to provide an environment that
 12 emulates a computer platform, and (ii) a plurality of
 13 image instances of a plurality of software applications that
 14 are executable on the plurality of virtual machines; and

15 a control node that comprises an automation infrastructure
 16 to provide autonomic deployment of the plurality of
 17 image instances of the virtual machine manager on the
 18 application nodes by causing the plurality of image
 19 instances of the virtual machine manager to be copied
 20 from the software image repository to the application
 21 nodes and to provide autonomic deployment of the
 22 plurality of image instances of the software applications
 23 on the virtual machines by causing the plurality of image
 24 instances of the software applications to be copied from
 25 the software image repository to the application nodes.

26 194. Netflix directly infringes the ’138 Patent by making, using, offering to
 27 sell, and/or selling in the United States its Netflix service, which utilizes Netflix’s
 28 Titus Container Management Platform to deploy and manage virtual computing
 resources in a manner that practices the inventions claimed in the ’138 Patent.

20 195. Netflix developed the Titus Container Management Platform internally
 21 and uses it in production to “power Netflix streaming, recommendations, and
 22 content systems.”⁵³

23 196. Using the Titus Container Management Platform, Netflix is able to
 24 more efficiently manage the deployment and administration of the hundreds or
 25 thousands of VMs necessary for it to provide reliable, high-quality streaming
 26 services to its millions of customers. For example, in 2018 Netflix reportedly
 27

28 ⁵³ <https://netflix.github.io/titus/overview/>

1 launched approximately thousands of VM managers and hundreds of thousands of
2 VM containers each day.

3

4

5

6

7

8

9

10

11

Q4 2018 Container Usage NETFLIX

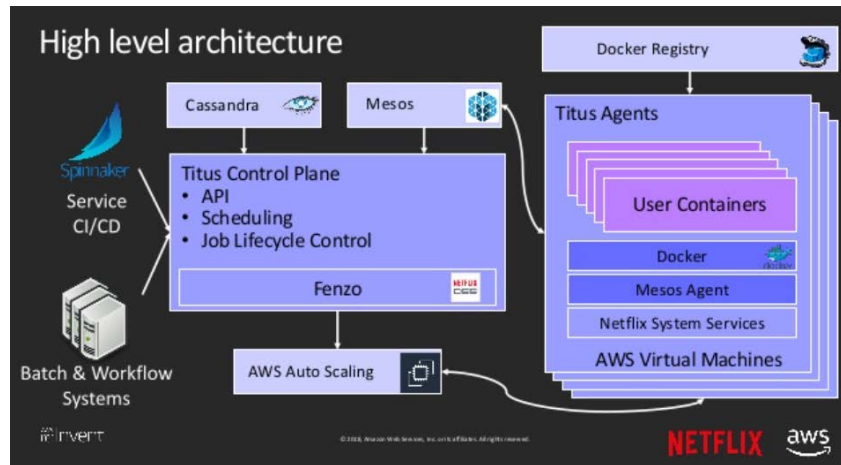
Common	
Jobs Launched	255K jobs / day
Different applications	1K+ different images
Isolated Titus deployments	7
Services	
Single App Cluster Size	5K (real), 12K containers (benchmark)
Hosts managed	7K VMs (435,000 CPUs)
Batch	
Containers launched	450K / day (750K / day peak)
Hosts managed (autoscaled)	55K / month

12 Source: [https://www.slideshare.net/aspyker/qconsf18-disenchantment-](https://www.slideshare.net/aspyker/qconsf18-disenchantment-netflix-titus-its-feisty-team-and-daemons?next_slideshow=1)
13 [netflix-titus-its-feisty-team-and-daemons?next_slideshow=1](https://www.slideshare.net/aspyker/qconsf18-disenchantment-netflix-titus-its-feisty-team-and-daemons?next_slideshow=1).

14 197. Upon information and belief, Netflix directly infringes at least
15 independent claim 1 of the '138 Patent at least in the exemplary manner described
16 below.

17 198. Netflix utilizes a hierarchical “distributed computing system,” as
18 taught by the '138 Patent, to provide a streaming entertainment service that delivers
19 video content and other content—such as a customer’s browsing experience,
20 content recommendations, and payment information—over a wide geographic area
21 through a distributed network of virtual machines. The Titus architecture is
22 illustrated in the following diagram and description:





Titus schedules application containers to be run across a fleet of thousands of Amazon EC2 instances.

Source: <https://www.slideshare.net/aspyker/cmp376-another-week-another-million-containers-on-amazon-ec2>; <https://medium.com/netflix-techblog/auto-scaling-production-services-on-titus-1f3cd49f5cd7>

199. The Netflix system uses the Netflix Titus Container Management Platform (“Titus”) to manage the distributed computing environment that powers its streaming, recommendation, and content delivery systems.⁵⁴ The Titus system integrates a “software image repository,” such as Docker Registry, to store various operating system and application files (i.e., “image instances”) needed to initialize new instances of virtual machine managers, which Netflix refers to as “Titus Agents.”

200. The Titus Agent image instances developed by Netflix are executed on the “application nodes”—i.e., “cloud” servers.

201. For example, as Netflix explains, the Titus Agents set up and manage “virtual machine” containers that are able to independently carry out discrete computing tasks to ensure reliable and efficient delivery of Netflix’s streaming

⁵⁴ <https://netflix.github.io/titus>.

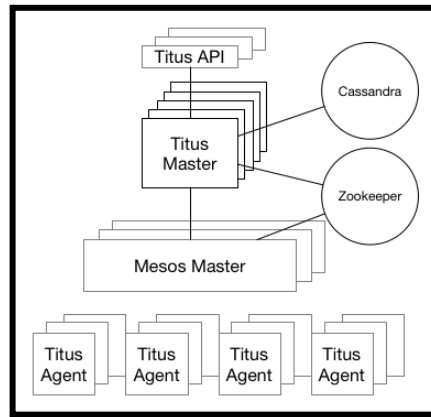
1 services.⁵⁵ As in the '138 Patent, each of these virtual machine containers emulates
 2 a computer platform and is able to execute various software applications under the
 3 supervision of the Titus Agents.

4 202. The Netflix video distribution system also includes “a control node
 5 that comprises an automation infrastructure to provide autonomic deployment of
 6 the plurality of image instances of the virtual machine manager on the application
 7 nodes by causing the plurality of image instances of the virtual machine manager to
 8 be copied from the software image repository to the application nodes and to
 9 provide autonomic deployment of the plurality of image instances of the software
 10 applications on the virtual machines by causing the plurality of image instances of
 11 the software applications to be copied from the software image repository to the
 12 application nodes.”

13 203. For example, the Netflix Titus system employs an automation
 14 infrastructure known as “Titus Master” to be the “control node” for Netflix’s
 15 distributed computing system. Like the control node of the '138 invention, the
 16 Titus Master provides “autonomic deployment” for new instances of the Titus
 17 Agents (i.e., the “virtual machine managers”) on the “cloud.” The Titus Master is
 18 responsible for persisting job and task information, scheduling tasks, and managing
 19 the pool of Titus Agents and can scale the pool of Agents up or down in response to
 20 demand.⁵⁶ In doing so, the Titus Master causes the appropriate “image instances”
 21 (i.e. operating system and software application packages) to be copied from the
 22 software image repository to the Titus Agents. The Titus Master is illustrated in the
 23 following diagram and description:

24
 25
 26 ⁵⁵ *Id.*

27 ⁵⁶ <https://medium.com/netflix-techblog/the-evolutionof-container-usage-at-netflix-3abfc096781b>;
 28 <https://medium.com/netflix-techblog/titusthe-netflix-container-management-platformis-now-open-source-f868c9fb5436>.



Titus Master

The Titus Master is responsible for persisting job and task information, scheduling tasks, and managing the pool of EC2 Agents. The Master receives requests from Gateway instances and creates and persists job and task info in response. The Master schedules tasks onto Agents with available resources and scales the pool of Titus Agents up or down in response to demand.

Source: <https://medium.com/netflix-techblog/auto-scaling-production-services-on-titus-1f3cd49f5cd7>; <https://netflix.github.io/titus/overview/>

204. At least as of on or around September 26, 2019, when the Broadcom Entities provided Netflix with an exemplary infringement chart for the '138 Patent, and by no later than the date of this Complaint, Netflix has had knowledge of the '138 Patent and its infringement thereof.

205. Netflix's knowing and willful infringement has caused and continues to cause damage to Avago, and Avago is entitled to recover damages sustained as a result of Netflix's wrongful acts in an amount subject to proof at trial.

SEVENTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 6,744,387)

206. The Broadcom Entities reallege and incorporate by reference the allegations of paragraphs 1-205 set forth above.

207. The '387 Patent, entitled "Method and System for Symbol Binarization," was duly and legally issued on June 1, 2004 from a patent application filed on July 10, 2002, with Lowell Winger as the named inventor. A copy of the '387 Patent is attached hereto as **Exhibit G**.

1 208. The '387 Patent is assigned to Broadcom Corp., which holds all
2 substantial rights, title, and interest in and to the '387 Patent.

3 209. Pursuant to 35 U.S.C. § 282, the '387 Patent is presumed valid.

4 210. The '387 Patent generally concerns an improvement in the way a
5 computer system compresses visual and audio data. Specifically, the patent is
6 “directed to an improved method for the binarization of data in an MPEG data
7 stream.”⁵⁷

8 211. As the patent explains, MPEG refers to a family of international
9 standards developed by the Motion Picture Expert Group that specify how to
10 represent visual and audio information in a compressed digital format.⁵⁸ The
11 MPEG formats make it possible to “represent[] a video signal with data roughly
12 1/50th the size of the original uncompressed video, while still maintaining good
13 visual quality.”⁵⁹ The MPEG formats achieve such high compression by taking
14 advantage of the fact that many images in a video stream do not change
15 significantly from picture to picture, and if they do change, the differences from one
16 picture to the next are often simple.⁶⁰ Storing and transmitting only the changes,
17 instead of entire pictures, results in considerable savings in data transmission.⁶¹

18 212. In practice, this compression technique is accomplished in a number of
19 steps. First, pixel differences between the pictures are “transformed into frequency
20 coefficients, and then quantized to further reduce the data transmission.”⁶² The
21 '387 Patent refers to the resulting transformed and quantized coefficients as
22 “symbols.”⁶³ Second, the transformed-quantized symbols are “binarized” to create
23 binary representations of each symbol in the form of a “codeword.”⁶⁴ Third, the

24 _____
25 ⁵⁷ '387 Patent, Abstract; *see also id.* at 1:7-11.

26 ⁵⁸ *Id.* at 1:15-30.

27 ⁵⁹ *Id.* at 1:36-39.

28 ⁶⁰ *Id.* at 3:29-37.

⁶¹ *Id.* at 1:51-54, 3:34-37.

⁶² *Id.* at 1:55-58, 3:40-43.

⁶³ *See, e.g., id.* at 1:55-64, 4:1-4, 4:45-54, 5:10-46, Table 1.

⁶⁴ *Id.* at 4:1-4.

1 binarized codewords are entropy encoded “to reduce the number of bits per symbol
2 without introducing any additional video signal distortion.”⁶⁵ The patent explains
3 that several types of codecs⁶⁶ exist for performing the entropy encoding; “[o]ne of
4 the most efficient of which is the family of binary arithmetic encoders (BACs).”⁶⁷
5 As the name implies, BACs operate only on binary valued data, which is why the
6 symbols must be binarized before they can be entropy encoded.⁶⁸

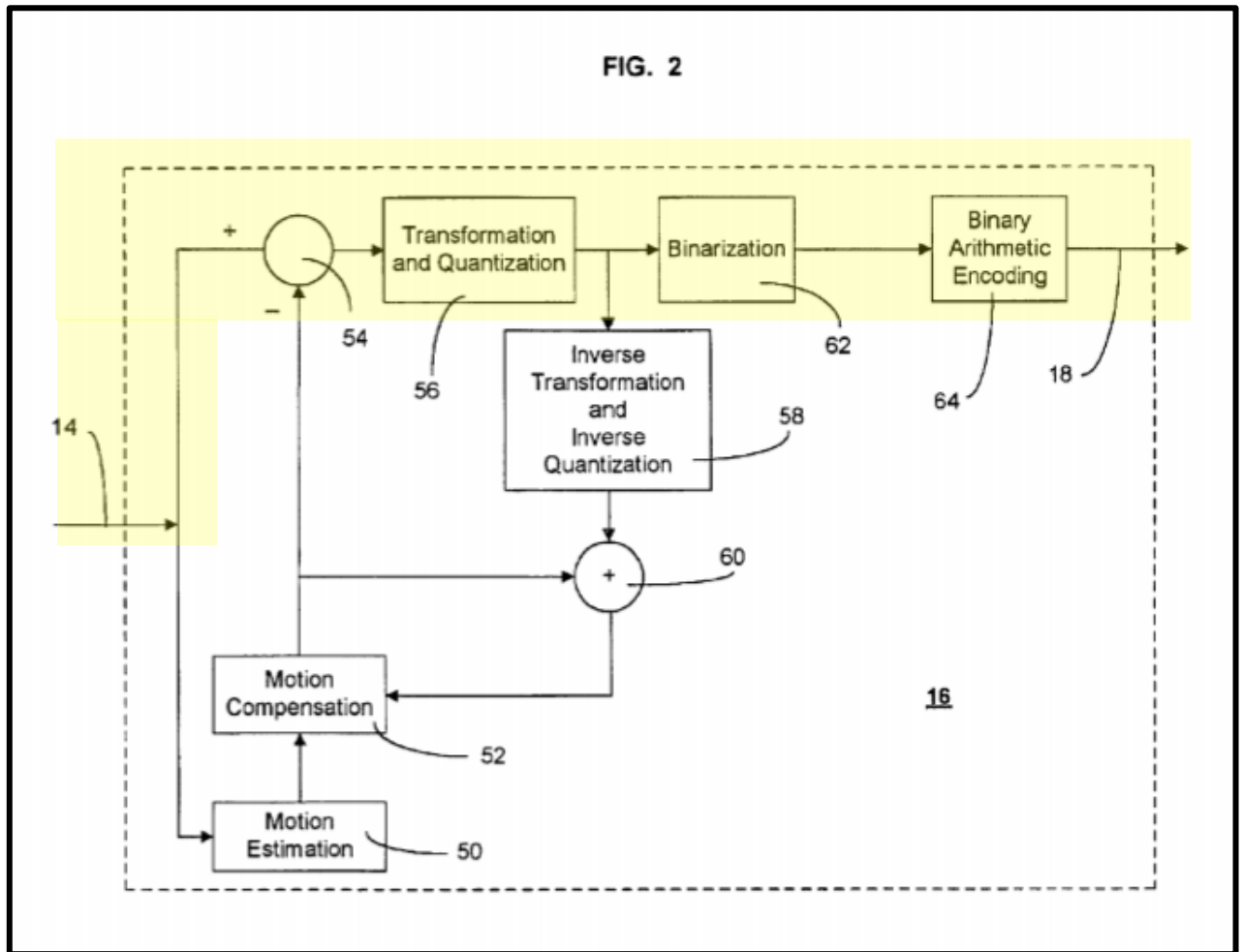
7 213. Figure 2 of the ’387 Patent—the relevant portion of which is
8 highlighted below—is a block diagram of an encoder that carries out this encoding
9 process.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26 ⁶⁵ *Id.* at 1:59-63, 4:26-36.

27 ⁶⁶ A codec is a device or computer program for encoding or decoding a digital
stream or signal.

28 ⁶⁷ ’387 Patent, 4:37-39.

⁶⁸ *Id.* at 4:45-49.



Source: '387 Patent, Fig. 2

214. Figure 2 depicts that a source video stream (14) enters the encoder (16) at the left and passes to a “combination” module (54), which assembles data related to pixel differences between pictures in the video stream.⁶⁹ The output of the combination module passes to the next module (56), where it is transformed and quantized.⁷⁰ Symbols created by module 56 pass to the binarization module (62), which creates binary codewords that represent the symbols. The codewords next

⁶⁹ *Id.* at 3:49-53.

⁷⁰ *Id.* at 3:54-57.

1 pass to binary arithmetic encoding module (64), where they are entropy encoded.
 2 And finally, the encoded bitstream (18) exits the encoder at the right.⁷¹

3 215. The '387 Patent is directed to the step in this process that occurs in the
 4 binarization module 62.⁷² In the prior art, several techniques were available for
 5 binarizing the symbols, including, for example: unary, binary, Golomb, and exp-
 6 Golomb binarization.⁷³ The patent explains that those techniques each have certain
 7 strengths and weaknesses. Unary binarization, for example, generates codewords
 8 that are more easily distinguishable from one another but that can be exceptionally
 9 long. Specifically, "[u]nary binarization consists of a number of binary 1s equal to
 10 an index for a symbol followed by a zero...."⁷⁴ Thus, a symbol index value of "1"
 11 results in a 2-bit codeword, namely "10." A symbol index value of "2" results in
 12 the 3-bit codeword "110," and so on. Thus, each codeword is easily distinguishable
 13 from the others, but the number of binary values can be quite large—encoding a
 14 large symbol index may require tens of thousands of bits.⁷⁵

15 216. Exp-Golomb binarization, on the other hand, greatly reduces the
 16 maximum possible size of the codewords, but "it does not permit codewords with a
 17 small symbol index (other than index 0) to be uniquely distinguished from
 18 codewords with larger symbol indices."⁷⁶

19 217. Thus, there was a need for a method and system that could exploit the
 20 most valuable properties of the unary and exp-Golomb binarizations.⁷⁷ The
 21 invention described in the '387 Patent meets that need:

22 The present invention provides a binarization that retains
 23 the most valuable properties of the unary and exp-Golomb
 24 binarizations. That is, small codewords are
 distinguishable as with a unary binarization, while large

25 ⁷¹ *Id.* at 3:57-4:8.

26 ⁷² *Id.* at 4:1-2.

27 ⁷³ *Id.* at 4:50-51.

28 ⁷⁴ *Id.* at 4:52-54.

⁷⁵ *Id.* at 5:22-5:45.

⁷⁶ *Id.* at 6:19-24.

⁷⁷ *Id.* at 2:2-12.

codewords have their binarization limited to a reasonable length. By doing so, the present invention provides a binarization that reduces the complexity and the bitrate/size for compressing and decompressing video, images, and signals that are compressed using binary arithmetic encoding for entropy encoding.⁷⁸

218. The '387 Patent claims methods and systems for constructing binarized codewords for digital video data (i.e., encoding) based on the index values of symbols produced by the transformation-and-quantization module of an encoder. Independent claim 3, for example, recites:

A binarization system comprising:

means for determining if a code symbol index value is less than a threshold value

means for constructing a codeword using a unary binarization if said code symbol index value is less than said threshold value; and

means for constructing a codeword using a exp-Golomb binarization if said code symbol index value is not less than said threshold value.

219. The methods and systems described in the '387 Patent improve the functionality of computer systems by improving the way they compress and process video and audio data.

220. Notably, the Hon. James V. Selna of this District previously held that the claims of the '387 Patent are patent-eligible under 35 U.S.C. § 101. In doing so, Judge Selna concluded that these claims “do not simply use general computers to perform abstract ideas; instead, the mathematical formula attempts to improve the functioning of compressing and decompressing video, images, and signals. Therefore, an inventive concept sufficiently transforms the nature of the claims...into patent-eligible inventions.”⁷⁹

⁷⁸ *Id.* at 6:26-36.

⁷⁹ *Broadcom Corp., et al. v. Sony Corp., et al.*, SAVC 16-1052 JVS (JCGx), p. 12 (C.D.C.A. Oct. 5, 2016).

221. Netflix directly infringes at least claim 3 of the '387 Patent at least in the exemplary manner described below.

222. Netflix developed, operates, and uses a “video encoding pipeline”, i.e., a series of video processing applications that operate “in the cloud.” Netflix claims that it has developed, and that it uses, its own proprietary video encoding software in this pipeline.

Pipeline in the Cloud

The video encoding pipeline runs EC2 Linux cloud instances. The elasticity of the cloud enables us to seamlessly scale up when more titles need to be processed, and scale down to free up resources. Our video processing applications don't require any special hardware and can run on a number of EC2 instance types. Long processing jobs are divided into smaller tasks and

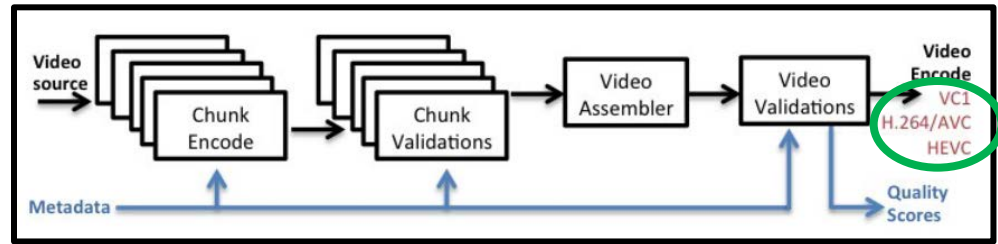
Source: <https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746>

223. Netflix uses its video encoding pipeline to generate encoded video files in a variety of formats, which it then uses to stream movie and TV content to its subscribers. As Netflix explains:

We ingest high quality video sources and generate video encodes of various codec profiles, at multiple quality representations per profile. The encodes are packaged and then deployed to a content delivery network for streaming. During a streaming session, the client requests the encodes it can play and adaptively switches among quality levels based on network conditions.⁸⁰

224. Among other encodes, Netflix uses its video encoding pipeline to generate content files in the ITU-T H.264 format, also known as Advanced Video Coding or “AVC” (“H.264”), and in the ITU-T H.265 format, also known as High Efficiency Video Coding or “HEVC” (“H.265”).

⁸⁰ <https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746>.



Source: <https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746>

225. The Netflix video encoding pipeline includes a “binarization system.” For example, with regard to the H.264 format used by Netflix, the H.264 documentation explains how the format employs a “concatenated unary/k-th order Exp-Golomb (UEGk) binarization process.” On information and belief, Netflix uses the method for UEG(k) encoding set forth in the H.264.2 reference software, which serves as an aid for the study and implementation of H.264 video coding.

226. On information and belief, the Netflix video encoding pipeline includes a “means for determining if a code symbol index value is less than a threshold value.” For instance, the H.264.2 reference software features the functions unary_exp_golomb_mv_encode() and unary_exp_golomb_level_encode(), which are responsible for performing unary exp-golomb encoding for various syntax elements. Both of these functions employ code that determines if a symbol index value is less than a threshold value.

```

static void unary_exp_golomb_mv_encode(EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     BiContextTypePtr ctx,
                                     unsigned int max_bin)
{
    if (symbol==0)
    {
        biari_encode_symbol(eep_dp, 0, ctx );
        return;
    }
    else
    {
        unsigned int bin = 1;
        unsigned int l = symbol, k = 1;
        biari_encode_symbol(eep_dp, 1, ctx++ );

        while (((--l)>0) && (++k <= 8))
        {
            biari_encode_symbol(eep_dp, 1, ctx );
            if ((++bin) == 2)
                ++ctx;
            if (bin == max_bin)
                ++ctx;
        }
        if (symbol < 8)
            biari_encode_symbol(eep_dp, 0, ctx);
        else
            exp_golomb_encode_eq_prob(eep_dp, symbol - 8, 3);
    }
}

```

Source: unary_exp_golomb_mv_encode() function

```

static void unary_exp_golomb_level_encode( EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     BiContextTypePtr ctx)
{
    if (symbol==0)
    {
        biari_encode_symbol(eep_dp, 0, ctx );
        return;
    }
    else
    {
        unsigned int l=symbol;
        unsigned int k = 1;

        biari_encode_symbol(eep_dp, 1, ctx );
        while (((--l)>0) && (++k <= 13))
            biari_encode_symbol(eep_dp, 1, ctx);
        if (symbol < 13)
            biari_encode_symbol(eep_dp, 0, ctx);
        else
            exp_golomb_encode_eq_prob(eep_dp, symbol - 13, 0);
    }
}

```

Source: unary_exp_golomb_level_encode() function

227. On information and belief, the Netflix video encoding pipeline includes a “means for constructing a codeword using a unary binarization if said code symbol index value is less than said threshold value.” For example, in the

1 H.264.2 reference software, the functions unary_exp_golomb_mv_encode() and
 2 unary_exp_golomb_level_encode() perform unary binarization of various syntax
 3 elements when the symbol index value is less than a threshold value (said threshold
 4 value is 8 and 13, respectively, in the excerpts below).

```
static void unary_exp_golomb_mv_encode(EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     BiContextTypePtr ctx,
                                     unsigned int max_bin)
{
  if (symbol==0)
  {
    biari_encode_symbol(eep_dp, 0, ctx );
    return;
  }
  else
  {
    unsigned int bin = 1;
    unsigned int l = symbol, k = 1;
    biari_encode_symbol(eep_dp, 1, ctx++ );

    while (((--l)>0) && (++k <= 8))
    {
      biari_encode_symbol(eep_dp, 1, ctx );
      if ((++bin) == 2)
        ++ctx;
      if (bin == max_bin)
        ++ctx;
    }
    if (symbol < 8)
      biari_encode_symbol(eep_dp, 0, ctx);
    else
      exp_golomb_encode_eq_prob(eep_dp, symbol - 8, 3);
  }
}
```

Source: unary_exp_golomb_mv_encode() function

```
static void unary_exp_golomb_level_encode( EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     BiContextTypePtr ctx)
{
  if (symbol==0)
  {
    biari_encode_symbol(eep_dp, 0, ctx );
    return;
  }
  else
  {
    unsigned int l=symbol;
    unsigned int k = 1;

    biari_encode_symbol(eep_dp, 1, ctx );
    while (((--l)>0) && (++k <= 13))
      biari_encode_symbol(eep_dp, 1, ctx);
    if (symbol < 13)
      biari_encode_symbol(eep_dp, 0, ctx);
    else
      exp_golomb_encode_eq_prob(eep_dp, symbol - 13, 0);
  }
}
```

Source: unary_exp_golomb_level_encode() function

228. On information and belief, the Netflix video encoding pipeline also includes a “means for constructing a codeword using an exp-Golomb binarization if said code symbol index value is not less than said threshold value.” For instance, functions unary_exp_golomb_mv_encode() and unary_exp_golomb_level_encode() in the H.264.2 reference software both call the function exp_golomb_encode_eq_prob() when the code symbol value is not less than the above-described threshold in order to construct the remainder of the codeword using exp-Golomb binarization.

```
static void exp_golomb_encode_eq_prob( EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     int k)
{
    for(;;)
    {
        if (symbol >= (unsigned int)(1<<k))
        {
            biari_encode_symbol_eq_prob(eep_dp, 1); //first unary part
            symbol = symbol - (1<<k);
            k++;
        }
        else
        {
            biari_encode_symbol_eq_prob(eep_dp, 0); //now terminated zero of unary part
            while (k--) //next binary part
                biari_encode_symbol_eq_prob(eep_dp, ((symbol>>k)&1));
            break;
        }
    }
}
```

Source: exp_golomb_encode_eq_prob() function

229. On information and belief, Netflix’s video encoding pipeline also infringes claim 3 of the ’387 Patent through its use of H.265 encoding. The reference software associated with that format proposes a binarization process that operates in substantially the same way as described above with regard to the H.264.2 reference software. On information and belief, Netflix uses the approach to binarization proposed by the H.265.2 reference software in encoding content files to the H.265 format.

230. At least as of on or around September 26, 2019, when the Broadcom Entities informed Netflix of its infringement of the ’387 Patent, and by no later than

1 the date of this Complaint, Netflix has had knowledge of the '387 Patent and the
2 infringement thereof by its video encoding pipeline.

3 231. Netflix's knowing and willful infringement of the '387 Patent has
4 caused and continues to cause damage to Broadcom Corp., and Broadcom Corp. is
5 entitled to recover damages sustained as a result of Netflix's wrongful acts in an
6 amount subject to proof at trial.

7 **EIGHTH CLAIM FOR RELIEF**

8 **(Infringement of U.S. Patent No. 6,982,663)**

9 232. The Broadcom Entities reallege and incorporate by reference the
10 allegations of paragraphs 1-231 set forth above.

11 233. The '663 Patent, entitled "Method and System for Symbol
12 Binarization," was duly and legally issued on January 3, 2006 from a patent
13 application filed on Jul. 10, 2002, naming Lowell Winger as the inventor. A copy
14 of the '663 Patent is attached hereto as **Exhibit H**.

15 234. The '663 Patent is assigned to Broadcom Corp., which holds all
16 substantial rights, title, and interest in and to the '663 Patent.

17 235. Pursuant to 35 U.S.C. § 282, the '663 Patent is presumed valid.

18 236. The '663 Patent originates from the same specification as the '387
19 Patent. Like the '387 Patent, the '663 Patent generally concerns an improvement in
20 the way a computer system compresses visual and audio data. Specifically, the
21 patent is "directed to an improved method for the binarization of data in an MPEG
22 data stream."⁸¹

23 237. As the '663 Patent explains, MPEG refers to a family of international
24 standards developed by the Motion Picture Expert Group that specify how to
25 represent visual and audio information in a compressed digital format.⁸² The
26 MPEG formats make it possible to "represent[] a video signal with data roughly

27 _____
28 ⁸¹ '663 Patent, Abstract; *see also id.* at 1:7-11.

⁸² *Id.* at 1:15-30.

1 1/50th the size of the original uncompressed video, while still maintaining good
 2 visual quality.”⁸³ The MPEG formats achieve such high compression by taking
 3 advantage of the fact that many images in a video stream do not change
 4 significantly from picture to picture, and if they do change, the differences from one
 5 picture to the next are often simple.⁸⁴ Storing and transmitting only the changes,
 6 instead of entire pictures, results in considerable savings in data transmission.⁸⁵

7 238. In practice, this compression technique is accomplished in a number of
 8 steps. First, pixel differences between the pictures are “transformed into frequency
 9 coefficients, and then quantized to further reduce the data transmission.”⁸⁶ The
 10 ’663 Patent refers to the resulting transformed and quantized coefficients as
 11 “symbols.”⁸⁷ Second, the transformed-quantized symbols are “binarized” to create
 12 binary representations of each symbol in the form of a “codeword.”⁸⁸ Third, the
 13 binarized codewords are entropy encoded “to reduce the number of bits per symbol
 14 without introducing any additional video signal distortion.”⁸⁹ The patent explains
 15 that several types of codecs⁹⁰ exist for performing the entropy encoding, “[o]ne of
 16 the most efficient of which is the family of binary arithmetic encoders (BACs).”⁹¹
 17 As the name implies, BACs operate only on binary valued data, which is why the
 18 symbols must be binarized before they can be entropy encoded.⁹²

19 239. Figure 2 of the ’663 Patent—the relevant portion of which is
 20 highlighted below—is a block diagram of an encoder that carries out this encoding
 21 process.

22
 23 ⁸³ *Id.* at 1:38-41.

24 ⁸⁴ *Id.* at 3:31-36.

25 ⁸⁵ *Id.* at 1:51-54; 3:35-39.

26 ⁸⁶ *Id.* at 1:55-58; 3:41-44.

27 ⁸⁷ *See, e.g., id.* at 1:55-64, 4:1-4, 4:45-54, 5:10-46, Table 1.

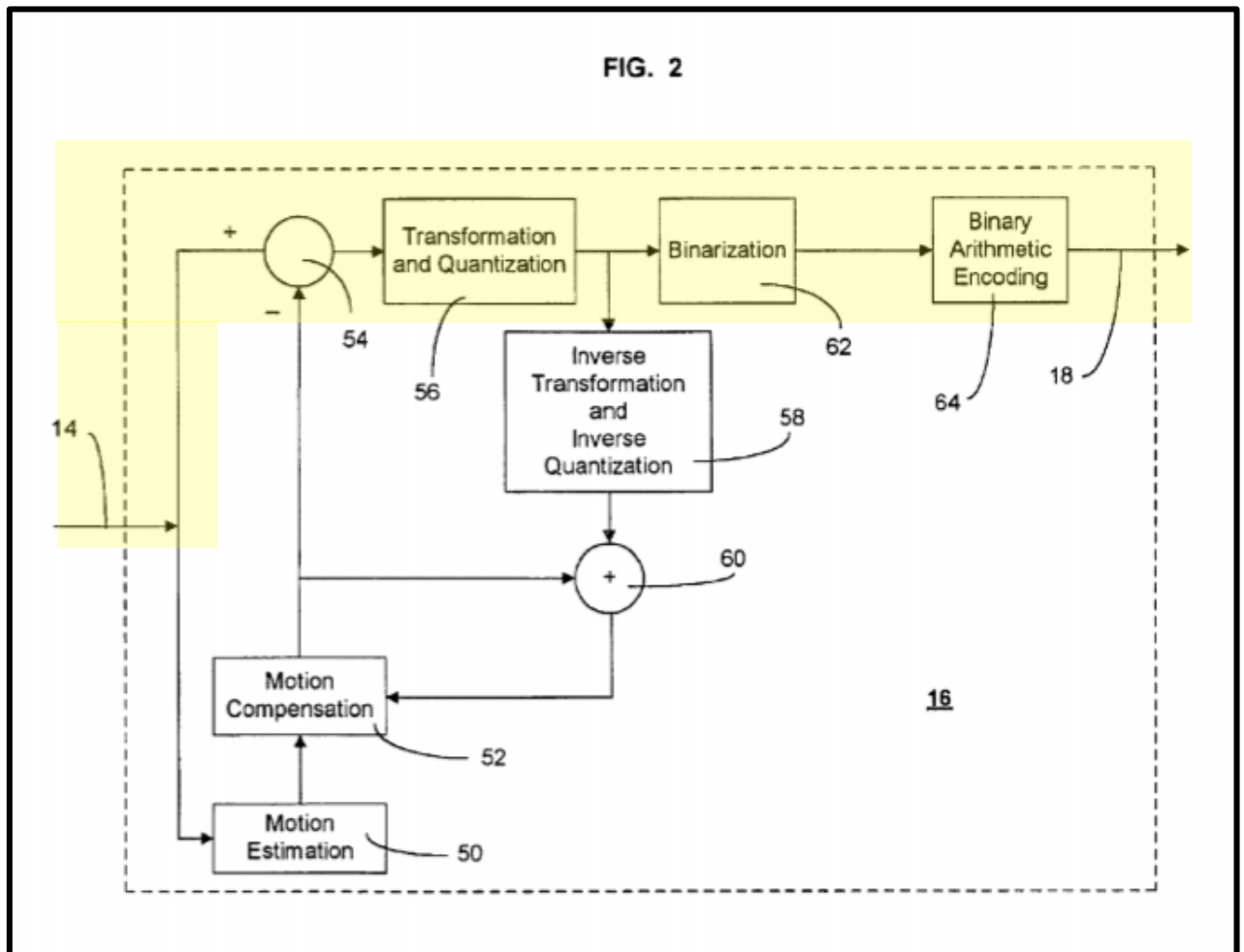
28 ⁸⁸ *Id.* at 4:1-4.

⁸⁹ *Id.* at 1:59-63, 4:26-36.

⁹⁰ A codec is a device or computer program for encoding or decoding a digital stream or signal.

⁹¹ ’663 Patent, 4:34-36.

⁹² *Id.* at 4:42-47.



Source: '663 Patent, Fig. 2

240. Figure 2 depicts that a source video stream (14) enters the encoder (16) at the left and passes to a “combination” module (54), which assembles data related to pixel differences between pictures in the video stream.⁹³ The output of the combination module passes to the next module (56), where it is transformed and quantized. Symbols created by module (56) pass to the binarization module (62), which creates binary codewords that represent the symbols. The codewords next pass to binary arithmetic encoding module (64), where they are entropy encoded. And finally, the encoded bitstream (18) exits the encoder at the right.⁹⁴

⁹³ *Id.* at 3:50-58.

⁹⁴ *Id.* at 3:57-4:8.

241. The '663 Patent is directed, in part, to the step in this process that occurs in the binarization module 62.⁹⁵ In the prior art, several techniques were available for binarizing the symbols, including, for example: unary, binary, Golomb, and exp-Golomb binarization.⁹⁶ The patent explains that those techniques each have certain strengths and weaknesses. Unary binarization, for example, generates codewords that are distinguishable from one another but that can be exceptionally long. Specifically, “[u]nary binarization consists of a number of binary 1s equal to an index for a symbol followed by a zero....”⁹⁷ Thus, a symbol index value of “1” results in a 2-bit codeword, namely “10.” A symbol index value of “2” results in the 3-bit codeword “110,” and so on. Thus, each codeword is easily distinguishable from the others, but the number of binary values can be quite large—encoding a large symbol index may require tens of thousands of bits.⁹⁸

242. Exp-Golomb binarization, on the other hand, greatly reduces the maximum possible size of the codewords, but “it does not permit codewords with a small symbol index (other than index 0) to be uniquely distinguished from codewords with larger symbol indices.”⁹⁹

243. Thus, there was a need for a method and system that could exploit the most valuable properties of the unary and exp-Golomb binarizations.¹⁰⁰ The invention described in the '663 Patent meets that need:

The present invention provides a binarization that retains the most valuable properties of the unary and exp-Golomb binarizations. That is, small codewords are distinguishable as with a unary binarization, while large codewords have their binarization limited to a reasonable length. By doing so, the present invention provides a binarization that reduces the complexity and the bitrate/size for compressing and decompressing video, images, and signals that are compressed using binary

⁹⁵ *Id.* at 4:1-2.

⁹⁶ *Id.* at 4:46-47.

⁹⁷ *Id.* at 4:48-49.

⁹⁸ *Id.* at 5:22-5:45.

⁹⁹ *Id.* at 6:14-17.

¹⁰⁰ *Id.* at 2:1-12.

1 arithmetic encoding for entropy encoding.¹⁰¹

2 244. The '663 Patent claims methods and systems utilizing a combination
3 of unary and exp-Golomb binarization for encoding and decoding digital video
4 data. Independent claim 12, for example, recites:

5 A method for generating a codeword from an index value
6 for digital video encoding, comprising the steps of:

7 (A) generating a first pattern in a first portion of said
8 codeword in response to said index value being at least as
great as a threshold;

9 (B) generating a second pattern in a second portion of said
10 codeword following said first portion representing an
offset of said index value above said threshold; and

11 (C) generating a third pattern in a third portion of said
12 codeword following said second portion representing a
value of said index value above said offset.

13 245. As in the '387 Patent, the methods and systems described in the '663
14 Patent improve the functionality of computer systems by improving the way they
15 compress and process video and audio data.

16 246. Notably, the Hon. James V. Selna of this District previously held that
17 the claims of the '663 Patent are patent-eligible under 35 U.S.C. § 101. In doing
18 so, Judge Selna concluded that claim 12, and others in the '663 Patent, are “directed
19 to improving digital video decoding” and, thus, “are not directed to abstract
20 ideas.”¹⁰²

21 247. Upon information and belief, Netflix directly infringes at least claim
22 12 of the '663 Patent at least in the exemplary manner described below.

23 248. Figure 5 of the '663 Patent, which depicts Table 3, demonstrates a
24 particular instance of the hybrid unary-exp-Golomb codes described in that patent.
25 As the '663 Patent explains, “Table 3 illustrates a binarization that is particularly
26

27 ¹⁰¹ *Id.* at 6:19-28.

28 ¹⁰² *Broadcom Corp., et al. v. Sony Corp., et al.*, SAVC 16-1052 JVS (JCGx), p. 9
(C.D.C.A. Oct. 5, 2016).

appropriate for the binarization of quarter pixel motion vector residual magnitudes of MPEG-AVC/H.264.”¹⁰³ The patent describes how, upon reaching the threshold at which unary to exp-Golomb switching occurs (N=64 for Table 3), the index comprises three parts: (1) the initial prefix (highlighted in blue below); (2) a unary representation appended to the initial prefix to form the unary prefix (highlighted in red below); and (3) the exp-Golomb suffix (highlighted in green below).¹⁰⁴

Table 3 - Motion vector magnitude residual binarization.

Index	Unary Prefix	exp-Golomb Suffix
0	0	
1	10	
2	110	
...		
63	1...1 0	
64	1...1 10	0
65	1...1 10	1
66	1...1 110	00
67	1...1 110	01
68	1...1 110	10
69	1...1 110	11
70	1...1 1110	000
71	1...1 1110	001
72	1...1 1110	010
73	1...1 1110	011
74	1...1 1110	100
75	1...1 1110	101
...		

FIG. 5

Source: '663 Patent, Fig. 5

¹⁰³ '663 Patent, 6:35-40.

¹⁰⁴ See *id.* at 6:44-63.

1 249. As explained above, Netflix developed, operates, and uses a video
2 encoding pipeline to encode its film and TV content in a variety of digital formats,
3 including H.264 and H.265.

4 250. The Netflix video encoding system practices a “method for generating
5 a codeword from an index value for digital video encoding.” For instance, with
6 regard to the H.264 format used by Netflix, the H.264 documentation explains how
7 the format employs a “concatenated unary/k-th order Exp-Golomb (UEGk)
8 binarization process.” This process generates codewords from index values. On
9 information and belief, Netflix uses the method for UEG(k) encoding set forth in
10 the H.264.2 reference software, which serves as an aid for the study and
11 implementation of H.264 video coding.

12 251. On information and belief, the Netflix video encoding pipeline
13 practices the step of “generating a first pattern in a first portion of said codeword in
14 response to said index value being at least as great as a threshold.” For instance,
15 the H.264.2 reference software features the functions
16 unary_exp_golomb_mv_encode() and unary_exp_golomb_level_encode(), which
17 are responsible—in part—for generating codewords from index values during the
18 encoding process. Where the index value meets or exceeds a certain threshold (8
19 for unary_exp_golomb_mv_encode() and 13 for
20 unary_exp_golomb_level_encode()), these functions populate the first portion of
21 the codeword with a pattern representing an initial prefix.

```

static void unary_exp_golomb_mv_encode(EncodingEnvironmentPtr eep_dp,
                                       unsigned int symbol,
                                       BiContextTypePtr ctx,
                                       unsigned int max_bin)
{
    if (symbol==0)
    {
        biari_encode_symbol(eep_dp, 0, ctx );
        return;
    }
    else
    {
        unsigned int bin = 1;
        unsigned int l = symbol, k = 1;
        biari_encode_symbol(eep_dp, 1, ctx++ );

        while (((--l)>0) && (++k <= 8))
        {
            biari_encode_symbol(eep_dp, 1, ctx );
            if ((++bin) == 2)
                ++ctx;
            if (bin == max_bin)
                ++ctx;
        }
        if (symbol < 8)
            biari_encode_symbol(eep_dp, 0, ctx);
        else
            exp_golomb_encode_eq_prob(eep_dp, symbol - 8, 3);
    }
}

```

Source: unary_exp_golomb_mv_encode() function

```

static void unary_exp_golomb_level_encode( EncodingEnvironmentPtr eep_dp,
                                           unsigned int symbol,
                                           BiContextTypePtr ctx)
{
    if (symbol==0)
    {
        biari_encode_symbol(eep_dp, 0, ctx );
        return;
    }
    else
    {
        unsigned int l=symbol;
        unsigned int k = 1;

        biari_encode_symbol(eep_dp, 1, ctx );
        while (((--l)>0) && (++k <= 13))
            biari_encode_symbol(eep_dp, 1, ctx);
        if (symbol < 13)
            biari_encode_symbol(eep_dp, 0, ctx);
        else
            exp_golomb_encode_eq_prob(eep_dp, symbol - 13, 0);
    }
}

```

Source: unary_exp_golomb_level_encode() function

252. On information and belief, the Netflix video encoding system also “generat[es] a second pattern in a second portion of said codeword following said first portion representing an offset of said index value above said threshold.” For example, both the unary_exp_golomb_mv_encode() and unary_exp_golomb_level_encode() functions call the function exp_golomb_encode_eq_prob() when the index value exceeds the thresholds described above. The exp_golomb_encode_eq_prob() function generates a second portion of the codeword—the unary representation appended to the initial prefix to form the unary prefix—when the index value exceeds the thresholds.

```
static void exp_golomb_encode_eq_prob( EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     int k)
{
    for(;;)
    {
        if (symbol >= (unsigned int)(1<<k))
        {
            biari_encode_symbol_eq_prob(eep_dp, 1); //first unary part
            symbol = symbol - (1<<k);
            k++;
        }
        else
        {
            biari_encode_symbol_eq_prob(eep_dp, 0); //now terminated zero of unary part
            while (k--) //next binary part
                biari_encode_symbol_eq_prob(eep_dp, ((symbol>>k)&1));
            break;
        }
    }
}
```

Source: exp_golomb_encode_eq_prob() function

253. Further, on information and belief, the Netflix video encoding system “generat[es] a third pattern in a third portion of said codeword following said second portion representing a value of said index value above said offset.” For example, after the unary prefix described above, the exp_golomb_encode_eq_prob() function generates a third portion of the codeword—the exp-Golomb suffix—which captures the value of the index above the offset described above.

```
static void exp_golomb_encode_eq_prob( EncodingEnvironmentPtr eep_dp,
                                     unsigned int symbol,
                                     int k)
{
    for(;;)
    {
        if (symbol >= (unsigned int)(1<<k))
        {
            biari_encode_symbol_eq_prob(eep_dp, 1); //first unary part
            symbol = symbol - (1<<k);
            k++;
        }
        else
        {
            biari_encode_symbol_eq_prob(eep_dp, 0); //now terminated zero of unary part
            while (k--) //next binary part
                biari_encode_symbol_eq_prob(eep_dp, ((symbol>>k)&1));
            break;
        }
    }
}
```

Source: exp_golomb_encode_eq_prob() function

254. On information and belief, Netflix’s video encoding pipeline also infringes claim 12 of the ’663 Patent through its use of H.265 encoding. The reference software associated with that format proposes a binarization process that operates in substantially the same way as described above with regard to the H.264.2 reference software. On information and belief, Netflix uses the approach to binarization proposed by the H.265.2 reference software in encoding content files to the H.265 format.

255. At least as of on or around September 26, 2019, when the Broadcom Entities informed Netflix of its infringement of the ’663 Patent, and by no later than

1 the date of this Complaint, Netflix has had knowledge of the '663 Patent and the
2 infringement thereof by its encoding pipeline.

3 256. Netflix's knowing and willful infringement of the '663 Patent has
4 caused and continues to cause damage to Broadcom Corp., and Broadcom Corp. is
5 entitled to recover damages sustained as a result of Netflix's wrongful acts in an
6 amount subject to proof at trial.

7 **NINTH CLAIM FOR RELIEF**

8 **(Infringement of U.S. Patent No. 9,332,283)**

9 257. The Broadcom Entities reallege and incorporate by reference the
10 allegations of paragraphs 1-256 set forth above.

11 258. The '283 Patent, entitled "Signaling of prediction size unit in
12 accordance with video coding," was duly and legally issued on May 3, 2016 from a
13 patent application filed on June 14, 2012, with Peisong Chen, Brian Heng, and
14 Wade Wan as the named inventors. A copy of the '283 Patent is attached hereto as
15 **Exhibit I.**

16 259. The '283 Patent claims priority from U.S. Provisional Application No.
17 60/539,948, filed on September 27, 2011.

18 260. The '283 Patent is assigned to Broadcom Corp., which holds all
19 substantial rights, title, and interest in and to the '283 Patent.

20 261. Pursuant to 35 U.S.C. § 282, the '283 Patent is presumed valid.

21 262. The '283 Patent generally concerns an improved system for encoding
22 and decoding video content that ensures a high quality output when transmitting
23 that content to viewers. Specifically, the patent relates to an improved method for
24 encoding video content using a process known as binarization in order to transmit
25 that information to users more efficiently.

26 263. At the time of the inventions claimed in the '283 Patent, persons
27 engaged in developing next-generation video encoding technologies were looking
28 to take advantage of ongoing innovations in parallel processing power and

1 increased video resolutions, which could make performing complex encoding
 2 operations more efficient and lead to improved user experiences. Video encoding is
 3 a multi-step process that begins with an input video signal, and results in an output
 4 bitstream of encoded video data. As discussed above, between the input of the
 5 video signal and the output of the encoded video data the encoding process includes
 6 various operations that are performed on constituent portions of the input video
 7 signal to create a video stream according to a particular encoding format.

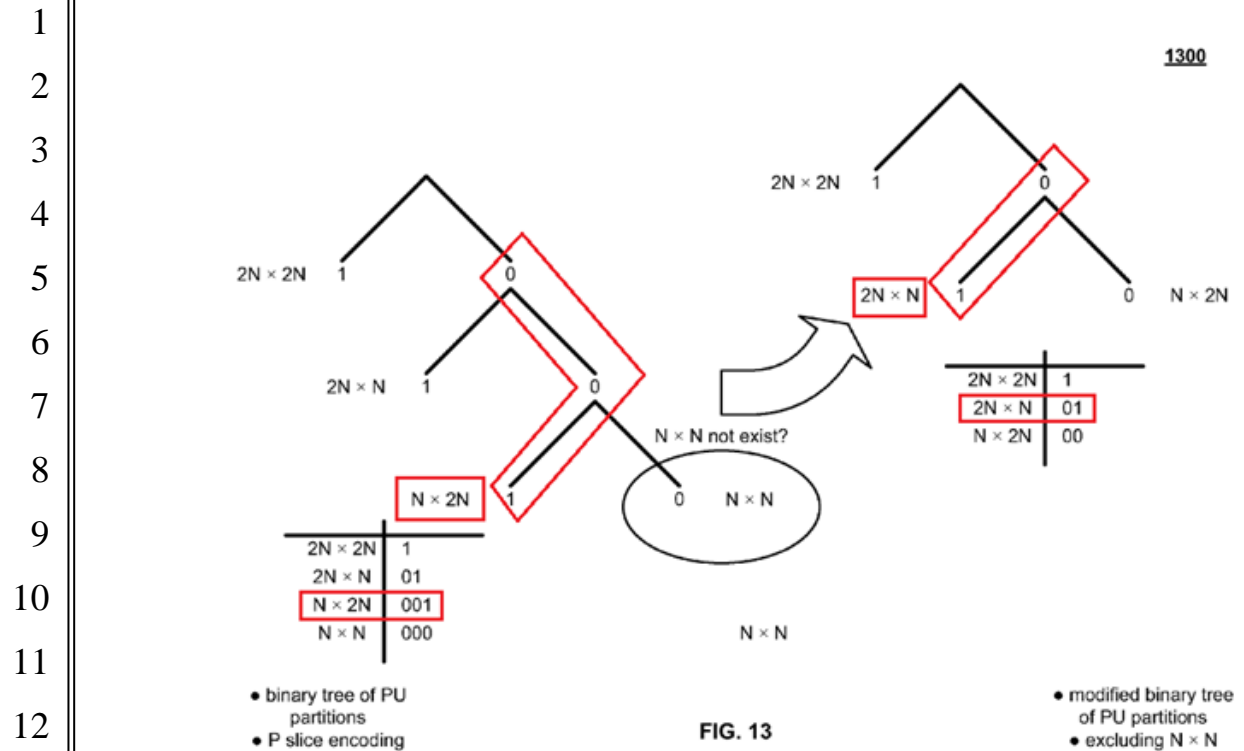
8 264. As the patent explains, among the advances at the time was the use of
 9 “predictive” (P) slices and “bi-predictive” (B) slices as components of the video
 10 data being processed.¹⁰⁵ P slices and B slices, in turn, are comprised of smaller
 11 components, including “coding tree units” and “coding units.” As described in the
 12 ’283 Patent, coding units can be “encoded” for different types of “prediction”
 13 processing, namely, “inter-prediction” or “intra-prediction.”¹⁰⁶ Subsequently,
 14 “prediction units” (PU) can be encoded for different “partition modes,” to be used
 15 in the intra- or inter-prediction processing.

16 265. Generally speaking, prediction and partition modes are encoded using
 17 binary “codewords.” These codewords can be generated using a “binary tree.” In
 18 general terms, a binary tree is a data structure that can be used to represent data and
 19 associate it with a corresponding bit sequence and vice versa. In this context, a
 20 “binary tree” data structure is used to create a sequence of binary numbers based on
 21 the selection of a “1” or a “0” at different positions (sometimes referred to as
 22 “nodes”) in the binary tree, starting at the beginning (“root”). Traversing the binary
 23 tree from the root to a “leaf” (an endpoint) results in a bit sequence that corresponds
 24 to a specific encoding.¹⁰⁷ Figure 13 of the ’283 Patent, for example, represents
 25 binary trees as follows:

26
 27 ¹⁰⁵ ’283 Patent, at 17:24-48.

28 ¹⁰⁶ *See, e.g., id.* at 17:9-12.

¹⁰⁷ *See id.* at Figs. 13, 14, 15.



14 In the left side of the exemplary image above, the annotated binary tree generates a
15 codeword of “001,” which is associated with a partition mode of “ $N \times 2N$.”¹⁰⁸

16 266. The use of P slices and B slices in an encoding protocol can introduce
17 inefficiencies when encoding coding units of the different slice types. This is
18 because at the time of the invention of the ’283 Patent, the encoding of P and B
19 slices required separate codewords generated using different binary trees, which
20 resulted in extra burdens on the encoding system and higher overhead.¹⁰⁹

21 267. Thus, there was a need for a method and system that could reduce this
22 inefficiency.¹¹⁰ The invention described in the ’283 Patent meets that need.
23 Specifically, the ’283 Patent’s system and method for using “only a singular
24 codebook . . . for both processing of the B slices and P slices,” may be employed to
25 provide a “very efficient implementation.”¹¹¹ The methods and systems described

26 ¹⁰⁸ *Id.* at Fig. 13 (annotations added).

27 ¹⁰⁹ *Id.* at 19:21-35; 20:19-30.

28 ¹¹⁰ ’283 Patent at 18:55-19:12.

¹¹¹ *Id.* at 18:60-63.

1 in the '283 Patent thus improve the functionality of computer systems by improving
2 the way they compress and process video and audio data.

3 268. The inventions described and claimed in the '283 Patent include
4 encoding using two syntax elements derived from a single binary tree that can be
5 applied to both P slices and B slices to indicate (1) whether inter-prediction or intra-
6 prediction applies to a selected coding unit, and (2) the prediction unit partition
7 mode that applies based on whether or not the selected coding unit is the smallest
8 coding unit (SCU) having a prediction unit size $N \times N$.

9 269. The '283 Patent claims methods and systems that use a single binary
10 tree to encode coding unit (CU) prediction when processing P slices and B slices
11 for digital video data. Independent claim 1, for example, recites:

12 A video processing device comprising:

13 a video encoder configured to:

14 encode an input video signal to generate an output
15 bitstream;

16 employ a single binary tree when processing at least one P
17 slice and at least one B slice to generate the output
18 bitstream, wherein the at least one P slice is used for
19 unidirectional prediction forward or behind in at least one
20 frame sequence, and wherein the at least one B slice is
21 used for bidirectional prediction both forward and behind
22 in the at least one frame sequence;

23 employ the single binary tree to encode coding unit (CU)
24 prediction based on a selected CU that is selected from a
25 plurality of CUs when generating a first syntax element
26 for both the at least one P slice and the at least one B slice
27 that undergo entropy encoding to generate the output
28 bitstream, wherein the first syntax element specifies intra-
prediction processing or inter-prediction processing for
the selected CU; and

employ the single binary tree to encode prediction unit
(PU) partition mode based on the selected CU when
generating a second syntax element for both the at least
one P slice and the at least one B slice that undergo the
entropy encoding to generate to generate [sic] the output
bitstream, wherein the second syntax element specifies
the PU partition mode for the selected CU, wherein the
PU partition mode is based on a size $N \times N$ PU when the
selected CU is a smallest CU (SCU) of the plurality of

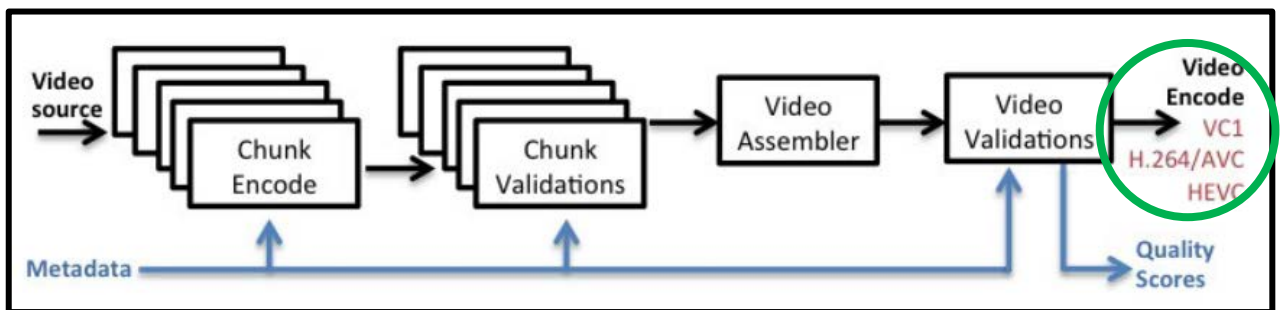
1 CUs and is based on a different size PU than the size $N \times N$
 2 PU when the selected CU is another CU than the SCU of
 the plurality of CUs, wherein N is a positive integer.

3 270. Netflix directly infringes at least claim 1 of the '283 Patent at least in
 4 the exemplary manner described below.

5 271. Netflix developed, operates, and uses a "video encoding pipeline", *i.e.*,
 6 a series of video processing applications. Netflix uses its video encoding pipeline to
 7 generate encoded video files in a variety of formats, which it then uses to stream
 8 movie and TV content to its subscribers. As Netflix explains:

9 We ingest high quality video sources and generate video
 10 encodes of various codec profiles, at multiple quality
 11 representations per profile. The encodes are packaged
 and then deployed to a content delivery network for
 12 streaming. During a streaming session, the client requests
 the encodes it can play and adaptively switches among
 quality levels based on network conditions.¹¹²

13 272. Among other encoding formats, Netflix uses its video encoding
 14 pipeline to generate content in the H.265 format, also known as High Efficiency
 15 Video Coding or "HEVC" ("H.265").



21 Source: [https://medium.com/netflix-techblog/high-quality-video-encoding-at-](https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746)
 22 [scale-d159db052746](https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746)

23 273. On information and belief, the Netflix video encoder includes a
 24 binarization system that employs a "single binary tree." For example, the H.265
 25 format employs a "binary tree" in the processing (*i.e.*, encoding) of P slices and B
 26 slices. Specifically, the H.265 format utilizes coding schemes that define a unique

27
 28 ¹¹² [https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-](https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746)
[d159db052746](https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746).

1 mapping of syntax element values to sequences of binary symbols, which are
 2 interpreted in terms of a binary code tree.¹¹³

3 274. On information and belief, the Netflix video encoder employs the
 4 binary tree to generate a “first syntax element” of a coding unit, wherein the “first
 5 syntax element” specifies intra-prediction processing or inter-prediction processing
 6 for the selected coding unit. For example, H.265 reference software features the
 7 parameter PredMode which specifies the “prediction mode” (i.e., intra-prediction
 8 processing or inter-prediction processing) of a selected coding unit. The function
 9 pcCU->isIntra() then returns either 1 or 0 depending on whether intra- or inter-
 10 prediction mode is selected for the particular coding unit. If the selected coding
 11 unit pcCU was encoded using intra-prediction mode (i.e., pcCU->isIntra() is
 12 TRUE) the encodeBin() function codes 1. If inter-prediction mode is selected, the
 13 encodeBin() codes 0.¹¹⁴

```
14  /// supported prediction type
15  enum PredMode
16  {
17      MODE_INTER = 0,      ///< inter-prediction mode
18      MODE_INTRA = 1,      ///< intra-prediction mode
19      NUMBER_OF_PREDICTION_MODES = 2,
20  };
```

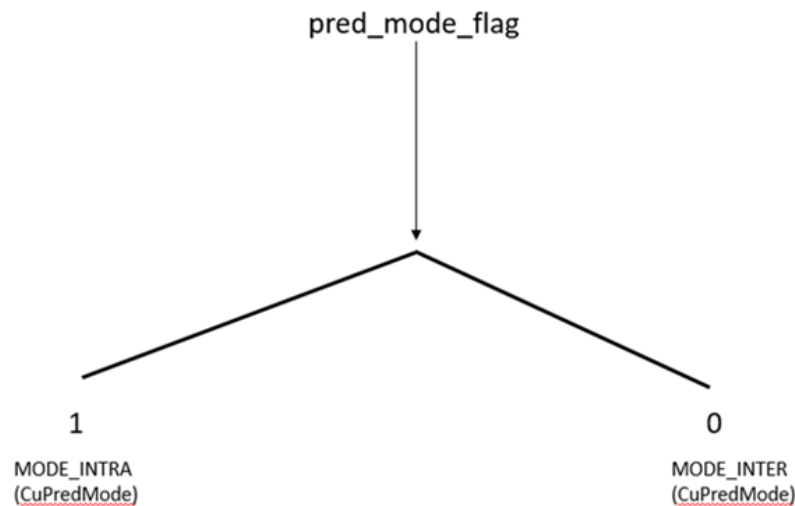
```
21  Void TEncSbac::codePartSize( TComDataCU* pcCU, UInt
22  uiAbsPartIdx, UInt uiDepth )
23  {
24      PartSize eSize          = pcCU->getPartitionSize(
25      uiAbsPartIdx );
26      ...
27      const UInt log2DiffMaxMinCodingBlockSize = pcCU-
28      >getSlice()->getSPS()-
29      >getLog2DiffMaxMinCodingBlockSize();
30
31      if ( pcCU->isIntra( uiAbsPartIdx ) )
32      {
33          if( uiDepth == log2DiffMaxMinCodingBlockSize )
34          {
35              m_pcBinIf->encodeBin( eSize == SIZE_2Nx2N? 1 : 0,
36              m_cCUPartSizeSCModel.get( 0, 0, 0 ) );
37          }
38          return;
39      }
40      ...
41      ...
42  }
```

26 ¹¹³ See, e.g., *ITU-T Recommendation H.265: High Efficiency Video Coding*, at §
 27 7.3.8.5 (November 2019) (available at <https://www.itu.int/rec/T-REC-H.265>)
 (“H.265 Recommendation”) (describing binary tree coding syntax using pseudo
 28 code).

¹¹⁴ See also, H.265 Recommendation at § 7.4.9.5.

Source: H.265.2: Reference software for ITU-T H.265 High Efficiency Video Coding (December 2016) (available at <https://www.itu.int/rec/T-REC-H.265.2>).

275. The binary tree that the Netflix video encoder employs to encode a coding unit prediction mode using a first syntax element for coding units for both P slices and B slices thus may be represented as follows, in which “1” codes for intra-prediction, and “0” codes for inter-prediction:



276. The Netflix video encoder is configured to employ the single binary tree to encode prediction unit (“PU”) partition mode based on the selected coding unit “when generating a second syntax element for both the at least one P slice and the at least one B slice,” wherein the second syntax element specifies the PU partition mode for the selected CU. For example, the H.265 reference software specifies the parameter “PartSize” and the function “getPartitionSize” which identifies the partition mode for the selected CU (“pcCU”). This particular example shows how in the case of a $2N \times 2N$ partition, the function encodeBin() generates the binary code “1”.

```

1  /// supported partition shape
2  enum PartSize
3  {
4  SIZE_2Nx2N = 0, ///< symmetric motion partition, 2Nx2N
5  SIZE_2NxN  = 1, ///< symmetric motion partition, 2Nx N
6  SIZE_Nx2N  = 2, ///< symmetric motion partition, Nx2N
7  SIZE_NxN   = 3, ///< symmetric motion partition, Nx N
8  SIZE_2NxnU = 4, ///< asymmetric motion partition, 2Nx(
   N/2) + 2Nx(3N/2)
9  SIZE_2NxD  = 5, ///< asymmetric motion partition,
   2Nx(3N/2) + 2Nx( N/2)
10 SIZE_nLx2N = 6, ///< asymmetric motion partition,
   (N/2)x2N + (3N/2)x2N
11 SIZE_nRx2N = 7, ///< asymmetric motion partition,
   (3N/2)x2N + ( N/2)x2N
12 NUMBER_OF_PART_SIZES = 8
13 };

```

```

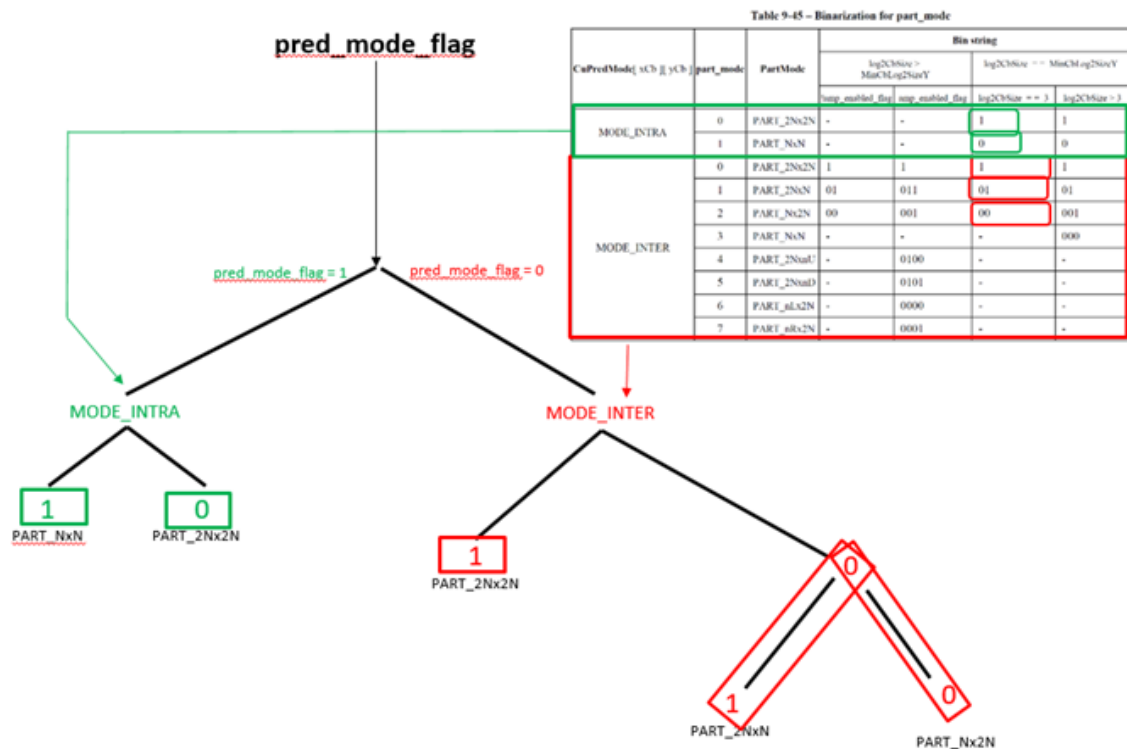
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

Source: H.265.2: Reference software for ITU-T H.265 High Efficiency Video Coding (December 2016) (available at <https://www.itu.int/rec/T-REC-H.265.2>).

277. As indicated in the Table 9-45 of the H.265 Recommendation, the partition mode for a particular coding unit is dependent on the prediction mode (CuPredMode) of the coding unit.¹¹⁵ Table 9-45 depicts the first syntax element (CuPredMode) and the second syntax element (encoding for the partition mode) for different size coding unit cases. The illustration below, in conjunction with Table 9-45, depicts the single binary tree encoding both the prediction mode and the partition mode.

¹¹⁵ See H.265 Recommendation, at Table 9-45.



Source: H.265 Recommendation at Table 9-45 (annotations added).

278. Thus, on information and belief, the Netflix video encoder employs the single binary tree to encode prediction unit partition mode when generating a second syntax element, wherein the second syntax element specifies the partition mode for the selected coding unit. In the table above, the coding unit example is of a size “log2CbSize” equals “3”.

279. As indicated in the image above, when the prediction mode is intra-prediction processing, the single binary tree encodes a second syntax element of either 1 or 0, which indicates a partition mode (PartMode (column 3)) of PART_2Nx2N or PART_NxN, respectively. Similarly, when the prediction mode is inter-prediction processing, the single binary tree for the example coding unit size specifies a second syntax element of 1, 01, or 00, which indicates a partition mode of PART_2Nx2N, PART_2NxN, or PART_Nx2N, respectively.

280. On information and belief, the Netflix video encoder employs a PU partition mode that is “based on a size $N \times N$ PU when the selected CU is a smallest CU (SCU)” and “is based on a different size PU than the size $N \times N$ PU when the selected CU is another CU than the SCU.” The H.265 format supports variable prediction block sizes from 64×64 down to 4×4 samples. The minimum size CU is assigned a size $N \times N$ Prediction Unit (PU) as a special case.¹¹⁶ Thus, the prediction unit partition mode is based on a coding unit size of $N \times N$ when the coding unit is the smallest coding unit, and the prediction unit partition mode is based on a different size prediction unit when the coding unit is a size other than the smallest coding unit.¹¹⁷ Thus, the Netflix video encoding system, as described above, practices at least claim 1 of the ’283 Patent.

281. Indeed, Netflix has published studies in which it has quantified the benefits of H.265 encoding. Through its own testing, Netflix determined that H.265 encoders can achieve equivalent subjective reproduction quality as encoders that conform to H.264/MPEG-4 AVC while using approximately 50% less bit rate (i.e., the amount of data “bits” transmitted per second).¹¹⁸ Thus, Netflix has specifically recognized benefits achieved by the inventions of the ’283 Patent.

282. By no later than the date of this Complaint, Netflix has had knowledge of the ’283 Patent and the infringement thereof by its encoding system.

283. Netflix’s infringement of the ’283 Patent, which is knowing and willful at least as of the filing of this Complaint, has caused and continues to cause damage to Broadcom Corp., and Broadcom Corp. is entitled to recover damages sustained as a result of Netflix’s wrongful acts in an amount subject to proof at trial.

¹¹⁶ See e.g., *Overview of the High Efficiency Video Coding (HEVC) Standard*, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 22, No. 12, (December 2012).

¹¹⁷ See Table 9-45 at column “PartMode.”

¹¹⁸ <https://netflixtechblog.com/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f>.

PRAYER FOR RELIEF

WHEREFORE, the Broadcom Entities respectfully request that the Court enter a judgment in their favor and against Netflix:

1. Declaring that Netflix has directly infringed one or more claims of the Patents-in-Suit in violation of 35 U.S.C. § 271;

2. Declaring that Netflix has induced infringement of one or more claims of the '079, '121, '245, and '992 Patents in violation of 35 U.S.C. § 271(b);

3. Declaring that Netflix's infringement of the '079, '121, '245, '992, '138, '387, '663, and '283 Patents is willful and deliberate pursuant to 35 U.S.C. § 284;

4. Enjoining Netflix from further infringing the '079, '121, '245, '992, and '138 Patents;

5. Ordering that the Broadcom Entities be awarded damages in an amount no less than a reasonable royalty for each asserted patent arising out of Netflix's infringement of the Patents-in-Suit, together with any other monetary amounts recoverable, such as treble damages;

6. Declaring that this is an exceptional case under 35 U.S.C. § 285 and awarding the Broadcom Entities their attorneys' fees and costs;

7. Ordering that Netflix is required to pay exemplary damages pursuant to 35 U.S.C. § 284;

8. Awarding pre-judgment and post-judgment interest and costs against Netflix; and

9. Awarding the Broadcom Entities such other and further relief as the Court deems just and proper.

JURY DEMAND

The Broadcom Entities demand a trial by jury of all claims in this action so triable.

1 Dated: March 13, 2020

HOPKINS & CARLEY
A Law Corporation

2
3 By: /s/ Jason S. Angell

4 Jason S. Angell
5 Christopher A. Hohn
6 Cary Chien
7 Robert K. Jain
8 Attorneys for Plaintiff
9 BROADCOM CORPORATION,
10 AVAGO TECHNOLOGIES
11 INTERNATIONAL SALES PTE.
12 LIMITED
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28