

1 Jason S. Angell, Bar No. 221607  
 jangell@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.

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

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

19 Plaintiffs,

20 v.

21 NETFLIX, INC.,

22 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.



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.<sup>2</sup>

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 <sup>2</sup> <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.”<sup>3</sup> 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.”<sup>4</sup>

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.”<sup>5</sup> “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.).”<sup>6</sup>  
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.”<sup>7</sup> “For  
25

---

26 <sup>3</sup> '079 Patent, 1:17-21.

27 <sup>4</sup> *Id.*

28 <sup>5</sup> *Id.* at 1:23-27.

<sup>6</sup> *Id.* at 1:27-30.

<sup>7</sup> *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).”<sup>8</sup>

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.<sup>9</sup>

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  
into second flow lists corresponding to the selected

27 \_\_\_\_\_  
28 <sup>8</sup> *Id.* at 1:34-40.

<sup>9</sup> *Id.* at 1:40-43.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

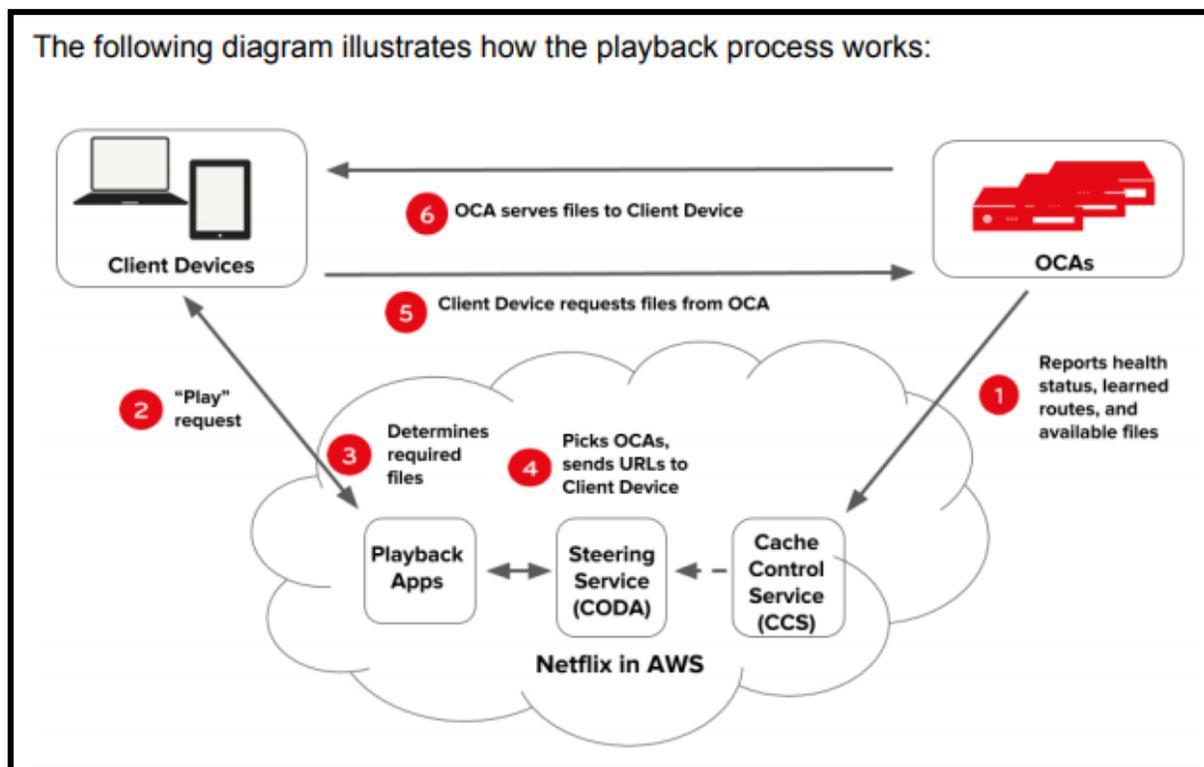
network link, the regrouping balancing the transmission unit traffic among the network links; and

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

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

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

39. Netflix utilizes the claimed "method for balancing transmission unit traffic over network links," including, for instance, in operating its CDN, which Netflix uses to stream video content to its subscribers over the internet. The Netflix 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.<sup>10</sup>

<sup>10</sup> <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.<sup>11</sup>

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.”<sup>12</sup>

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.

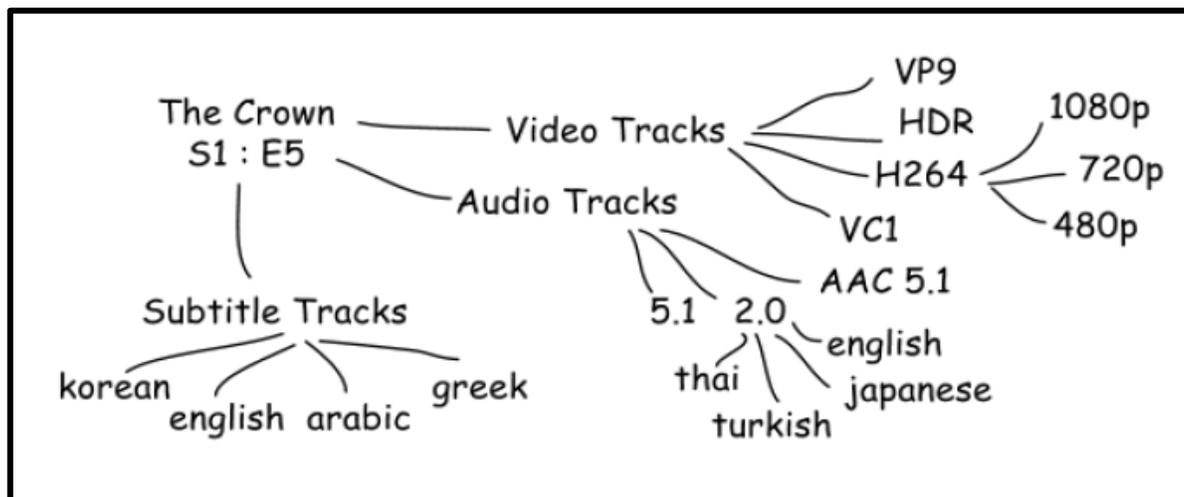
17           45. In Netflix’s words:

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!”<sup>13</sup>

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

12 *Id.*

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



9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

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*.”<sup>14</sup>

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.”

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

1  
2  
3  
4  
5

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

3  
4  
5

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.

6  
7  
8  
9

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.

10  
11

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.

12

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

13  
14  
15  
16  
17  
18  
19  
20  
21

48. Additionally, when a Netflix subscriber initiates a playback session, the Netflix system provides the “offsets of all segments.”<sup>15</sup> 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.”<sup>16</sup> 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).”<sup>17</sup> Thus, and as described further below, Netflix performs a “method for balancing transmission unit traffic over network links.”

22  
23  
24  
25  
26

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

27  
28

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

<sup>16</sup> *Id.*

<sup>17</sup> *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.”<sup>18</sup> “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.”<sup>19</sup> “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.”<sup>20</sup> 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

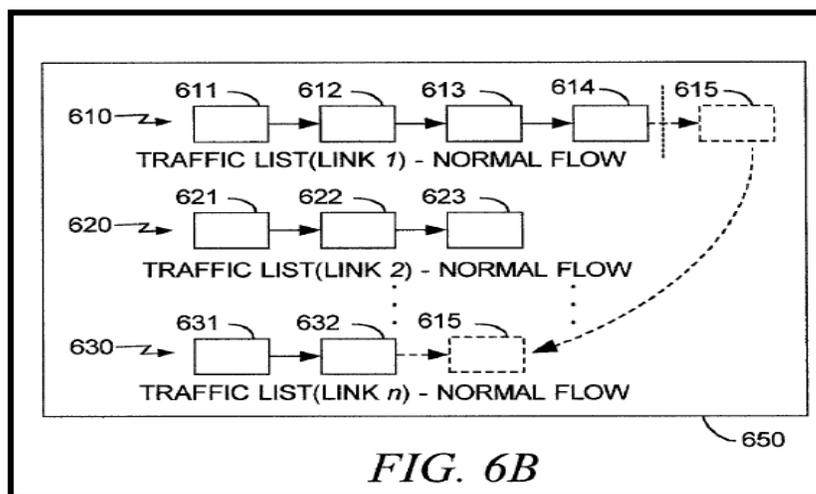
26 \_\_\_\_\_  
27 <sup>18</sup> <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>

28 <sup>19</sup> *Id.*

<sup>20</sup> *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 58. Netflix also has induced, and continues to induce, direct infringement  
2 by third-parties of at least claim 1 of the '079 Patent, at least in the exemplary  
3 manner described above, by actively encouraging their use of the Netflix system, in  
4 violation 35 U.S.C. § 271(b).

5 59. For example, Netflix has induced, and continues to induce, direct  
6 infringement of the '079 Patent by customers and/or end users of client devices  
7 enabled with the Netflix software application and service. In light of the notice the  
8 Broadcom Entities provided to Netflix of its infringement of the '079 Patent,  
9 Netflix knows that it provides and specifically intends to provide an application for  
10 use on client devices that, when used as intended with the Netflix streaming service,  
11 meets the limitations of claim 1 of the '079 Patent. Netflix knows and specifically  
12 intends that its end users practice the method recited in claim 1 of the '079 Patent,  
13 when using its application and service as intended.

14 60. Netflix's knowing and willful infringement of the '079 Patent has  
15 caused and continues to cause damage to Avago, and Avago is entitled to recover  
16 damages sustained as a result of Netflix's wrongful acts in an amount subject to  
17 proof at trial.

## 18 **SECOND CLAIM FOR RELIEF**

### 19 **(Infringement of U.S. Patent No. 8,259,121)**

20 61. The Broadcom Entities reallege and incorporate by reference the  
21 allegations of paragraphs 1-60 set forth above.

22 62. The '121 Patent, entitled "System and Method for Processing Data  
23 Using a Network," was duly and legally issued on September 4, 2012 from a patent  
24 application filed on December 9, 2002, with Patrick Law, Darren Neuman, and  
25 David Baer as the named inventors. A copy of the '121 Patent is attached hereto as  
26 **Exhibit B.**

27 63. The '121 Patent claims priority from U.S. Provisional Application No.  
28 60/420,151, filed on December 9, 2002.

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.”<sup>21</sup>

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.”<sup>22</sup>  
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.”<sup>23</sup> 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.”<sup>24</sup>

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 <sup>21</sup> '121 Patent, 1:42-45.

28 <sup>22</sup> *Id.* at 1:48-51.

<sup>23</sup> *Id.* at 1:51-55.

<sup>24</sup> *Id.* at 1:66-2:1.

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

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:

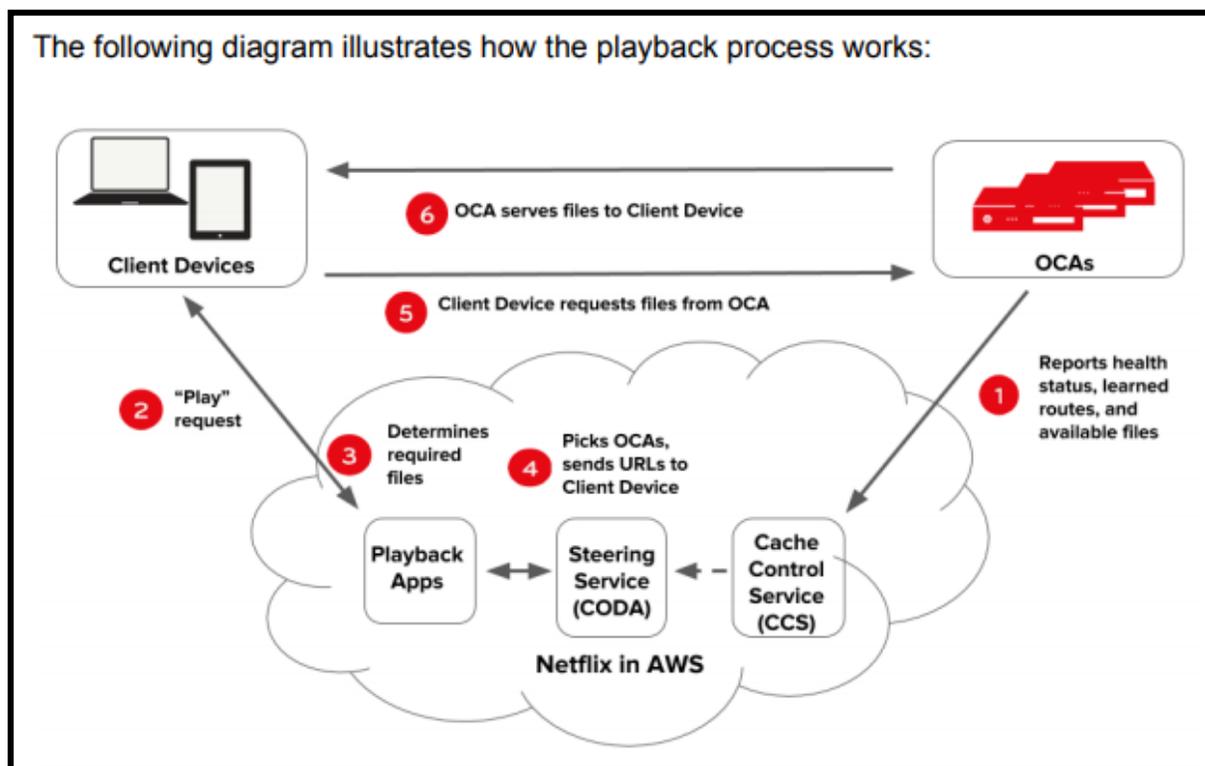
7 A network for processing data configured by a controller  
8 to form at least one display pipeline therein by  
9 dynamically selecting use of at least two selectable nodes  
10 from a plurality of selectable nodes and dynamically  
11 concatenating the selected at least two selectable nodes in  
the 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 <sup>25</sup> *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

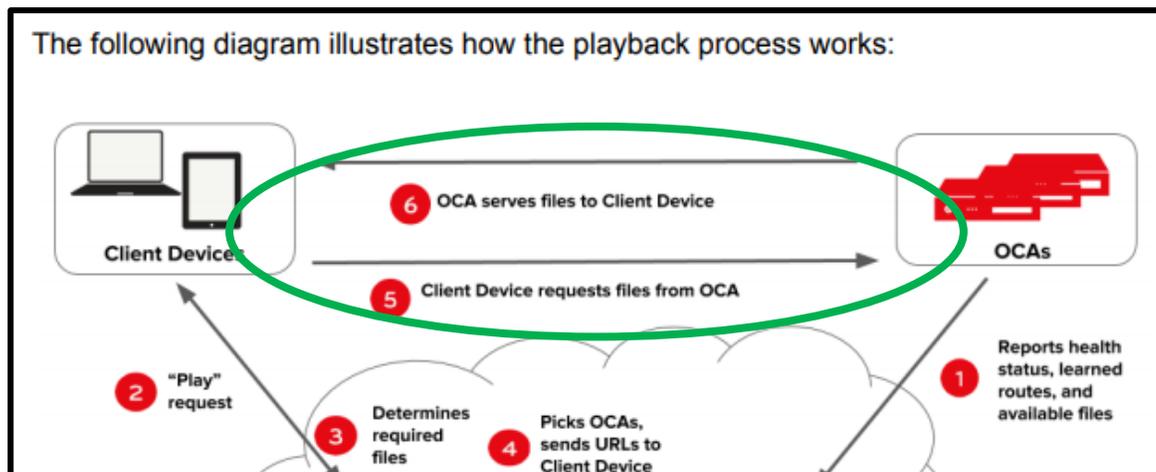
1 streamed; and (4) provide a manifest file to the client device containing URLs that  
 2 specify the OCAs and files needed for the playback process. Thus, Netflix  
 3 “controls” the path of the delivery of video content through the network to the  
 4 Netflix subscribers.

- 5 2. A user on a client device requests playback of a title (TV show or movie) from the Netflix application in AWS.
- 6
- 7 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.
- 8

- 9
- 10 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.
- 11
- 12
- 13 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.
- 14

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

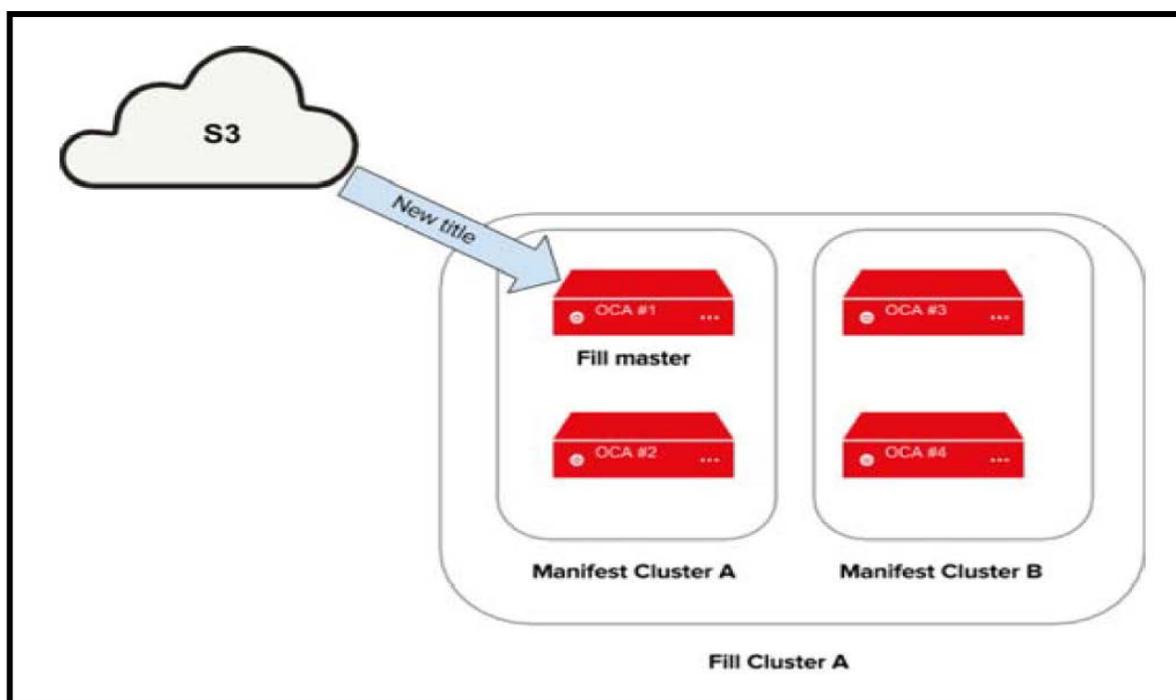
16 77. Through this process, Netflix’s content delivery network forms “at  
 17 least one display pipeline therein.” For example, the CDN establishes a link  
 18 between the client device and one or more Netflix OCAs through which the various  
 19 video, audio, and other files associated with the requested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



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

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

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



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

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

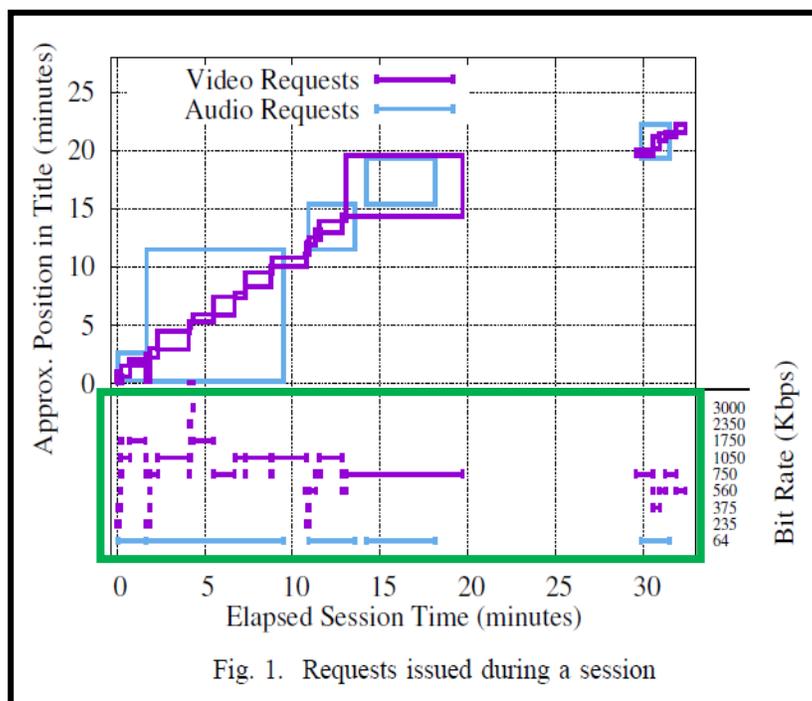
1           80. Upon receiving a request for content from a client device, the OCA  
2 identifies and sends the specific files requested, filtering out the requested data from  
3 the numerous content files contained on the OCA.

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.”<sup>26</sup>

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 <sup>26</sup> <https://openconnect.netflix.com/Open-Connect-Overview.pdf>.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28



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

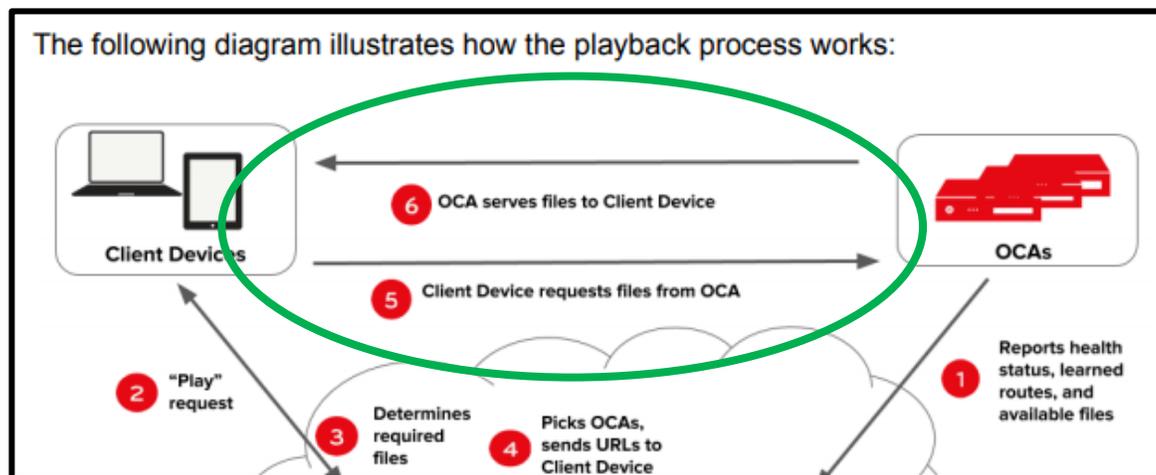
83. The “dynamic” selection has also been described by other third parties that have analyzed the Netflix system, further illustrating how the Netflix application on the client device—which Netflix controls—automatically switches between URLs for video files and OCAs over the course of the streaming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

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



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

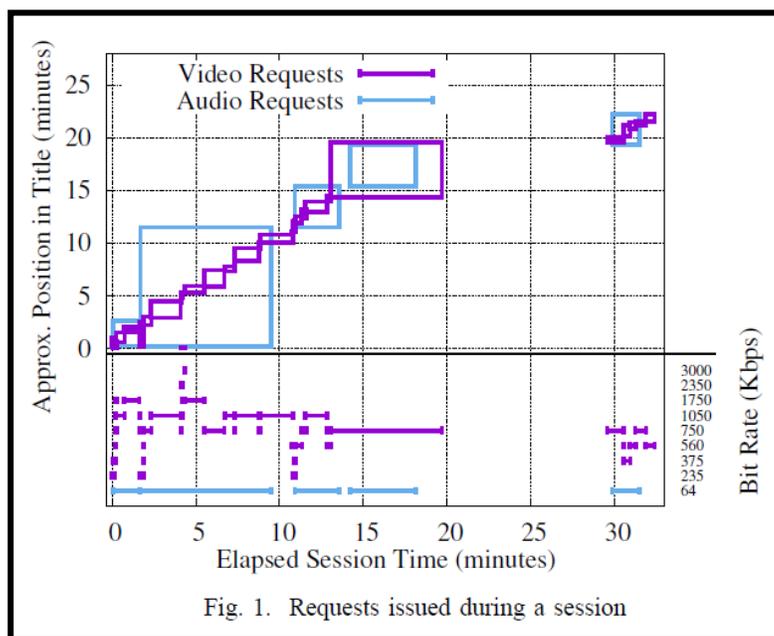
16 86. As another example of the dynamic nature of the concatenation, the  
 17 client devices within Netflix’s CDN continue to “adapt to the quality of the  
 18 network...If the network quality declines, the client lowers video quality to match.  
 19 The client will switch to another OCA when the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

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

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

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

1 module” by “pacing” the transmission of the streaming content from the Netflix  
 2 OCA. As illustrated below, the client device—under the control of the Netflix  
 3 application operating thereon—controls the pace at which it downloads content  
 4 files from Netflix to address network issues and user events, including by varying  
 5 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.”<sup>28</sup>

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 <sup>28</sup> [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.<sup>29</sup>

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 <sup>29</sup> ’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.<sup>30</sup>

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.<sup>31</sup>

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.<sup>32</sup>

27 <sup>30</sup> *Id.* at 1:19-40.

28 <sup>31</sup> *See id.* at 2:29-32, 2:41-45.

<sup>32</sup> *Id.* at 2:29-32.

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

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

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

15                   1. A method for communication, the method comprising:

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

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

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

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

---

<sup>33</sup> *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.

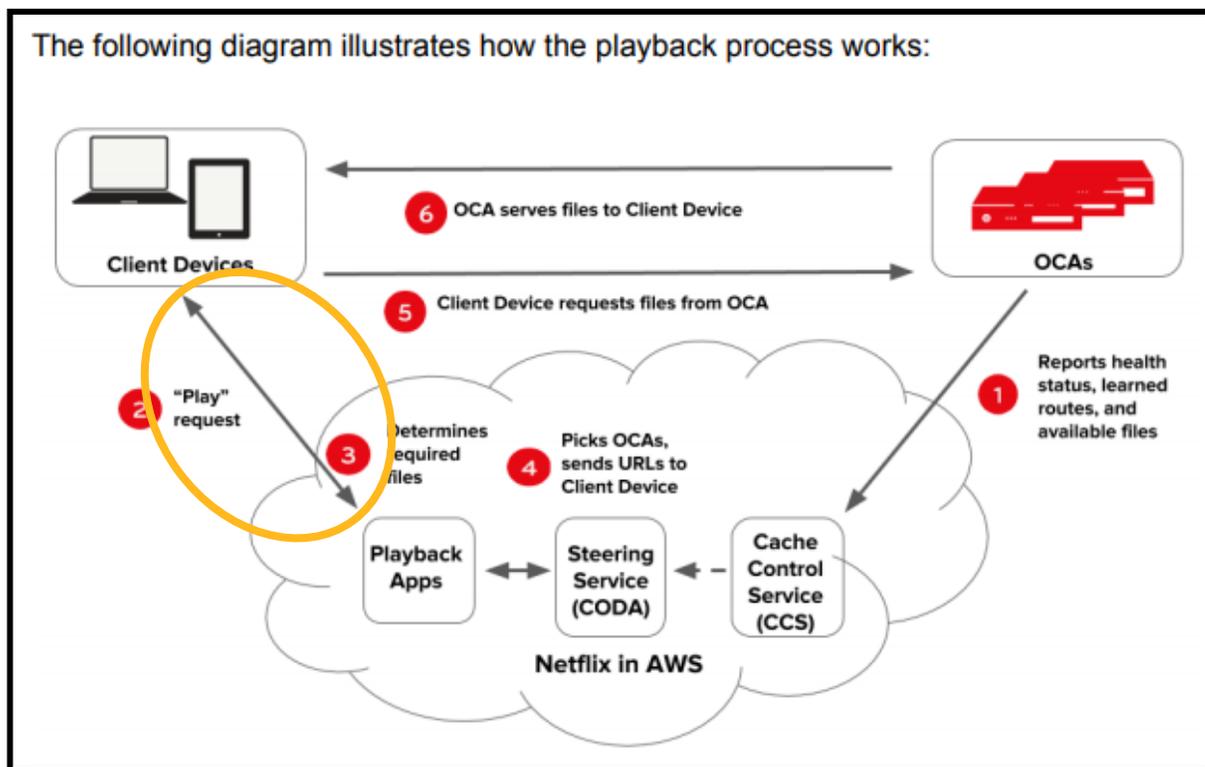
14 110. Thus, claims 3 and 6 further describe the invention's improved method  
15 whereby a network management server determines multiple routes for delivering  
16 content based on a provisioning profile in response to receiving a request for  
17 service from a user device, and then delivers that 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.

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

**Netflix Controls The Client**

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

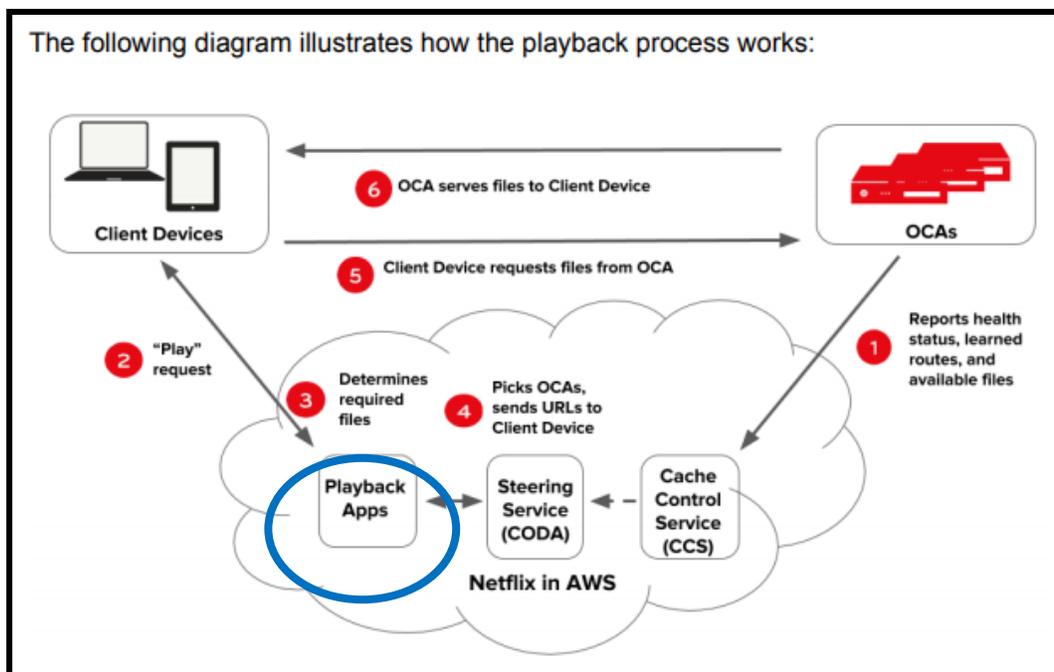
Netflix develops its Android and iOS apps themselves, so you might expect them to control those. But even on platforms like Smart TVs, where Netflix doesn’t build the client, Netflix still has control because it controls the *software development kit* (SDK).

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>

14 116. The request for service from the Netflix user’s client device is received  
 15 “by a network management server via a communication network.” For example, as  
 16 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:

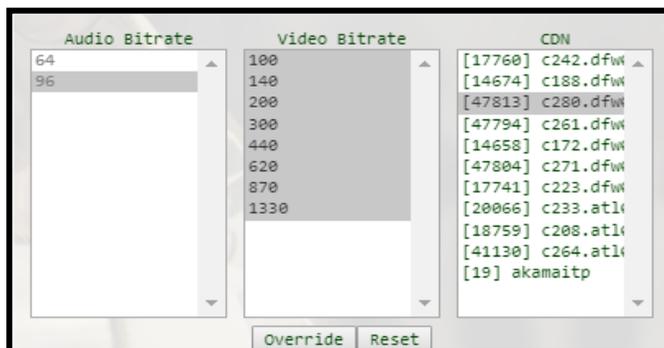
- 13
- 14 4. The steering service in AWS uses the information stored by the cache control  
15 service to pick OCAs that the requested files should be served from, generates  
16 URLs for these OCAs, and hands the URLs over to the playback application  
17 services.
- 18 5. The playback application services hand over URLs of the appropriate OCAs to  
19 the client device, and the OCA begins to serve the requested files.

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

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

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

28 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.

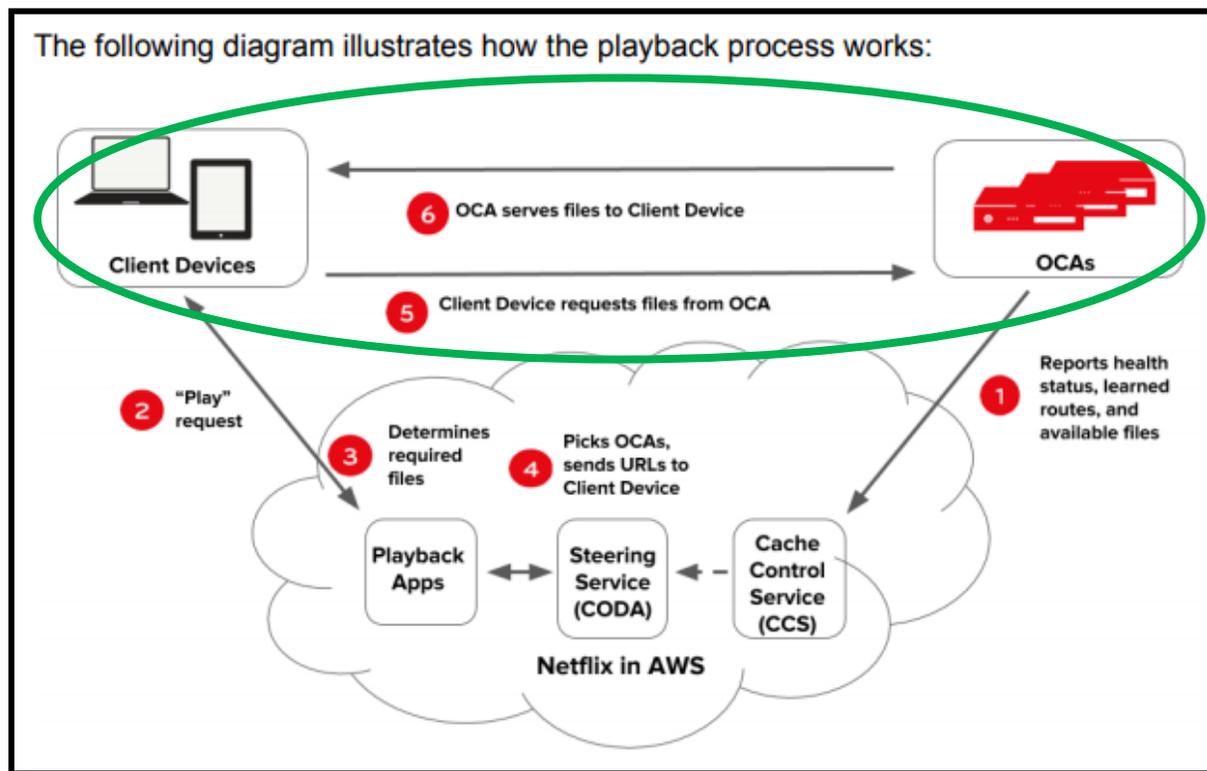
2. A user on a client device requests playback of a title from the Netflix application.
3. The playback application services check user authorization and licensing, then determine which specific streaming assets are required to handle the playback request - taking individual client characteristics and 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.”<sup>34</sup>

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.



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

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

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

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 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.

9 Source: [http://highscalability.com/blog/2017/12/11/netflix-what-happens-](http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html)  
10 [when-you-press-play.html](https://openconnect.netflix.com/Open-Connect-Overview.pdf); [https://openconnect.netflix.com/Open-Connect-](https://openconnect.netflix.com/Open-Connect-Overview.pdf)  
11 [Overview.pdf](https://openconnect.netflix.com/Open-Connect-Overview.pdf)

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

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

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

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

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

7 128. In addition to direct infringement, Netflix has induced, and continues  
8 to induce, direct infringement by third-parties of at least claims 1, 3, and 6 of the  
9 '245 Patent, at least in the exemplary manner described above, by actively  
10 encouraging their use of the Netflix system, in violation 35 U.S.C. § 271(b).  
11 Netflix has induced, and continues to induce, direct infringement of the '245 Patent  
12 by customers and/or end users of playback devices enabled with the Netflix  
13 software application and service. Netflix knows that it provides and specifically  
14 intends to provide an application for use on playback devices that, when used as  
15 intended with the Netflix streaming service, meets the limitations of claims 1, 3,  
16 and 6 of the '245 Patent. Netflix knows and specifically intends that its end users  
17 practice the method recited in claims 1, 3, and 6 of the '245 Patent, when using its  
18 application and service as intended.

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

#### 22 **FOURTH CLAIM FOR RELIEF**

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

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

26 131. The '992 Patent, entitled "Automatic Quality of Service Based  
27 Resource Allocation," was duly and legally issued on September 18, 2012, from a  
28 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.<sup>36</sup>

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 <sup>36</sup> 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.<sup>37</sup>

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.<sup>38</sup> 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 <sup>37</sup> *Id.* at 1:39-42 (describing delivery of high quality audio), 1:42-49 (describing  
delivery of high quality video).

28 <sup>38</sup> *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.<sup>39</sup>

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  
9 level to a user;  
10 determining that a network connection with a second  
11 system is available and is characterized by a  
12 communication bandwidth that is high enough to provide  
13 the digital media content to the user at a quality level  
14 higher than the current quality level;  
15 using the network connection to obtain the digital media  
16 content at the higher quality level from the second  
17 system; and  
18 delivering the digital media content at the higher quality  
19 level to the user instead of the digital media content at the  
20 current quality level.

21 143. Claim 1 thus recites a novel solution of determining, accessing, and  
22 using the resources of another system on a dynamic network environment in order  
23 to improve digital media content quality delivered to the user by obtaining the  
24 higher quality content from the second system in a manner that was not well-  
25 understood, routine, or conventional at the time of the '992 Patent. In the prior art,  
26 among other things, the source of the content remained the same between the two  
27 systems, whereas claim 1 claims obtaining the digital media content at a higher  
28 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 <sup>39</sup> *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.”<sup>40</sup>

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 <sup>40</sup> <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.<sup>41</sup>

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 <sup>41</sup> <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.”<sup>42</sup> 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 <sup>42</sup> *Id.*



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

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.<sup>45</sup> 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.<sup>46</sup> In prior art systems, each user has a  
16 dedicated video system, such as a DVD player, and decoder at the user's location.<sup>47</sup>

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

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 <sup>43</sup> '375 Patent at 1:56-60.

<sup>44</sup> *Id.* at 1:60-63.

<sup>45</sup> *Id.* at 5:38-43.

<sup>46</sup> *Id.* at 1:14-41.

<sup>47</sup> *Id.* at 1:26-27, 33-35.

<sup>48</sup> *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.

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

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

27  
28

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.

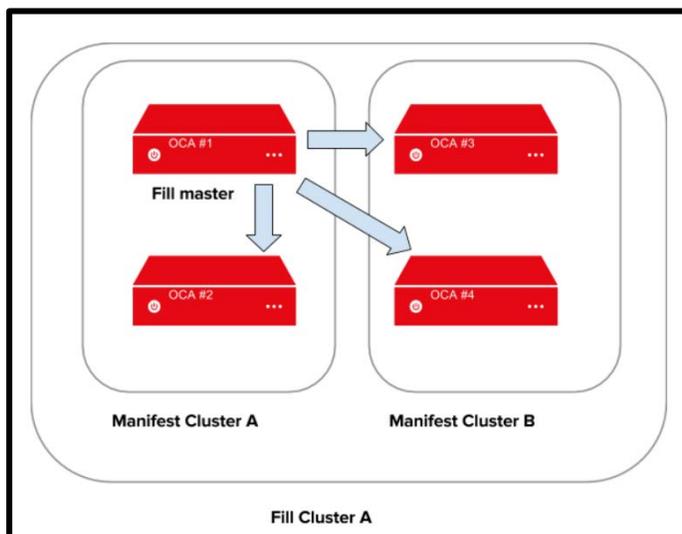
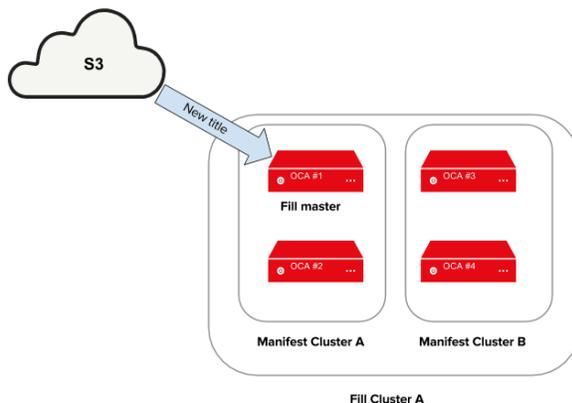
4           175. Netflix practices a “method for distributing video” and “presenting a  
5 plurality of compressed data streams with a drive server to a control server in  
6 response to one or more first control signals.” The Netflix streaming service is a  
7 streaming entertainment service that delivers 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.”<sup>49</sup>

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

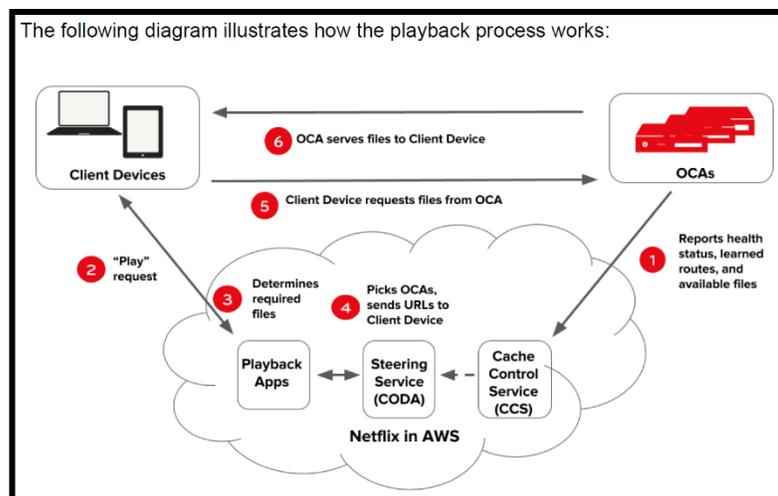
### Example Scenario

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”).

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

### 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.

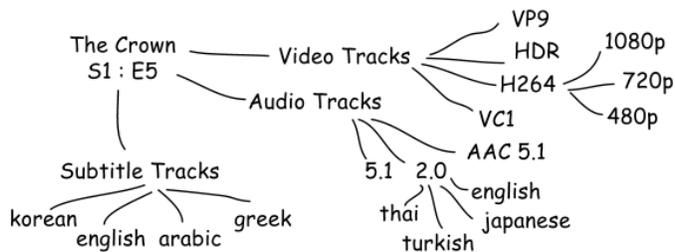
By controlling the SDK, Netflix can adapt consistently and transparently to slow networks, failed OCAs, and any other problems that might arise.

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

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

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

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



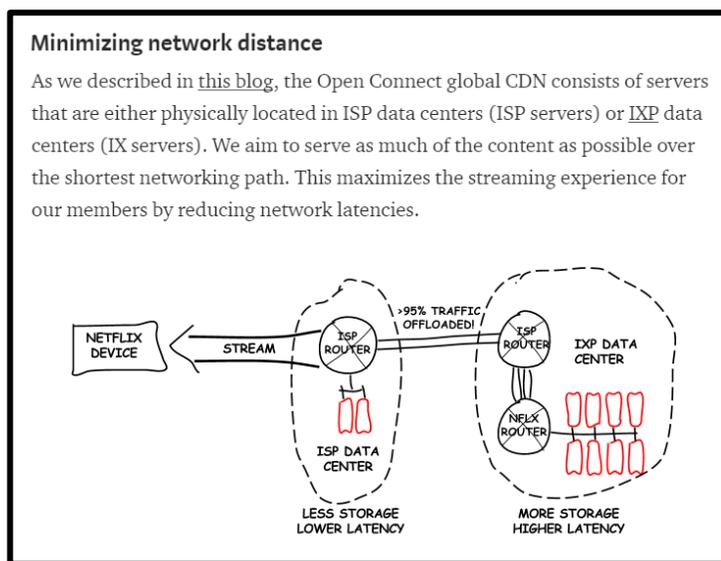
21 Source: <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>

22

23

24 182. Further, in the Netflix system, “at least one of said one or more  
25 decoders is disposed in a separate room from said control server and said drive  
26 server.” For example, the Netflix CDN processes video requests and audio requests  
27 using OCAs that Netflix strategically positions all over the world based on the  
28 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.<sup>50</sup> 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).

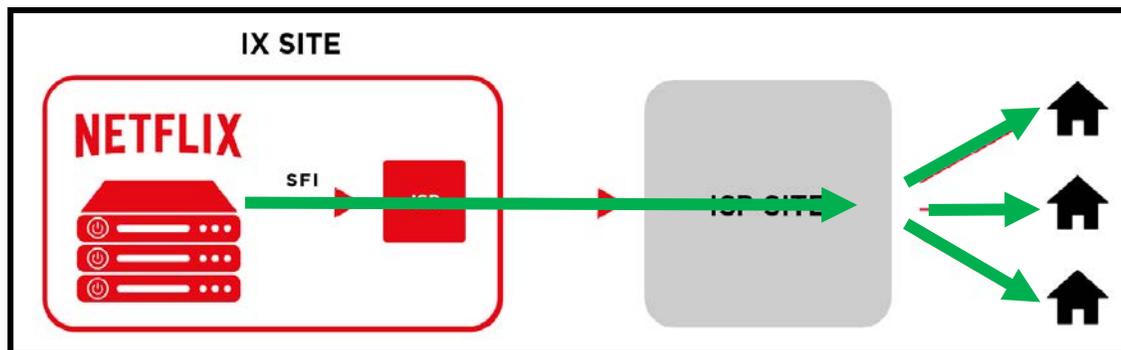


Source: <https://medium.com/netflix-techblog/content-popularity-for-open-connect-b86d56f613b>

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

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

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



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

10 (arrows added).

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

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

17 **SIXTH CLAIM FOR RELIEF**

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

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

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

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

28 189. 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.<sup>51</sup>

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.<sup>52</sup>

27 \_\_\_\_\_  
28 <sup>51</sup> '138 Patent at 1:16-33.

<sup>52</sup> *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.

29 195. Netflix developed the Titus Container Management Platform internally  
30 and uses it in production to “power Netflix streaming, recommendations, and  
31 content systems.”<sup>53</sup>

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

36  
37  
38 <sup>53</sup> <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 **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

5

6

7

8

9

10

11

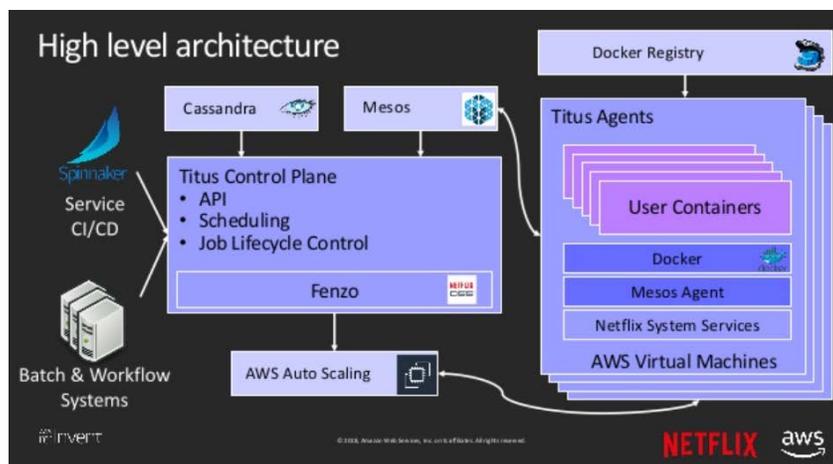
12 Source: [https://www.slideshare.net/aspyster/qconsf18-disenchantment-netflix-titus-its-feisty-team-and-daemons?next\\_slideshow=1](https://www.slideshare.net/aspyster/qconsf18-disenchantment-netflix-titus-its-feisty-team-and-daemons?next_slideshow=1).

13

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:





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

12 Source: [https://www.slideshare.net/aspkyer/cmp376-another-week-another-](https://www.slideshare.net/aspkyer/cmp376-another-week-another-million-containers-on-amazon-ec2)  
13 [million-containers-on-amazon-ec2](https://www.slideshare.net/aspkyer/cmp376-another-week-another-million-containers-on-amazon-ec2); [https://medium.com/netflix-](https://medium.com/netflix-techblog/auto-scaling-production-services-on-titus-1f3cd49f5cd7)  
14 [techblog/auto-scaling-production-services-on-titus-1f3cd49f5cd7](https://medium.com/netflix-techblog/auto-scaling-production-services-on-titus-1f3cd49f5cd7)

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

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

24 201. For example, as Netflix explains, the Titus Agents set up and manage  
25 “virtual machine” containers that are able to independently carry out discrete  
26 computing tasks to ensure reliable and efficient delivery of Netflix’s streaming  
27

28 <sup>54</sup> <https://netflix.github.io/titus>.

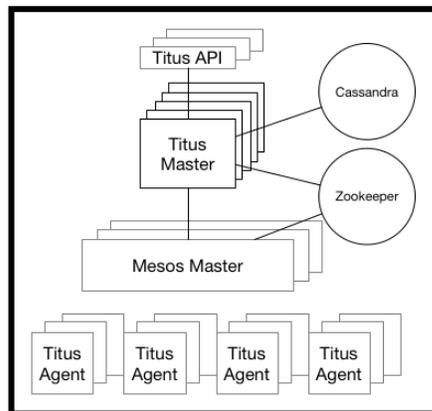
1 services.<sup>55</sup> As in the '138 Patent, each of these virtual machine containers emulates  
2 a computer platform and is able to execute various software applications under the  
3 supervision of the Titus Agents.

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.<sup>56</sup> 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 <sup>55</sup> *Id.*

27 <sup>56</sup> <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.”<sup>57</sup>

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.<sup>58</sup> 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.”<sup>59</sup> 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.<sup>60</sup> Storing and transmitting only the changes,  
17 instead of entire pictures, results in considerable savings in data transmission.<sup>61</sup>

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.”<sup>62</sup> The  
21 '387 Patent refers to the resulting transformed and quantized coefficients as  
22 “symbols.”<sup>63</sup> Second, the transformed-quantized symbols are “binarized” to create  
23 binary representations of each symbol in the form of a “codeword.”<sup>64</sup> Third, the

24 \_\_\_\_\_  
25 <sup>57</sup> '387 Patent, Abstract; *see also id.* at 1:7-11.

<sup>58</sup> *Id.* at 1:15-30.

<sup>59</sup> *Id.* at 1:36-39.

<sup>60</sup> *Id.* at 3:29-37.

<sup>61</sup> *Id.* at 1:51-54, 3:34-37.

<sup>62</sup> *Id.* at 1:55-58, 3:40-43.

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

<sup>64</sup> *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.”<sup>65</sup> The patent explains  
3 that several types of codecs<sup>66</sup> exist for performing the entropy encoding; “[o]ne of  
4 the most efficient of which is the family of binary arithmetic encoders (BACs).”<sup>67</sup>  
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.<sup>68</sup>

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

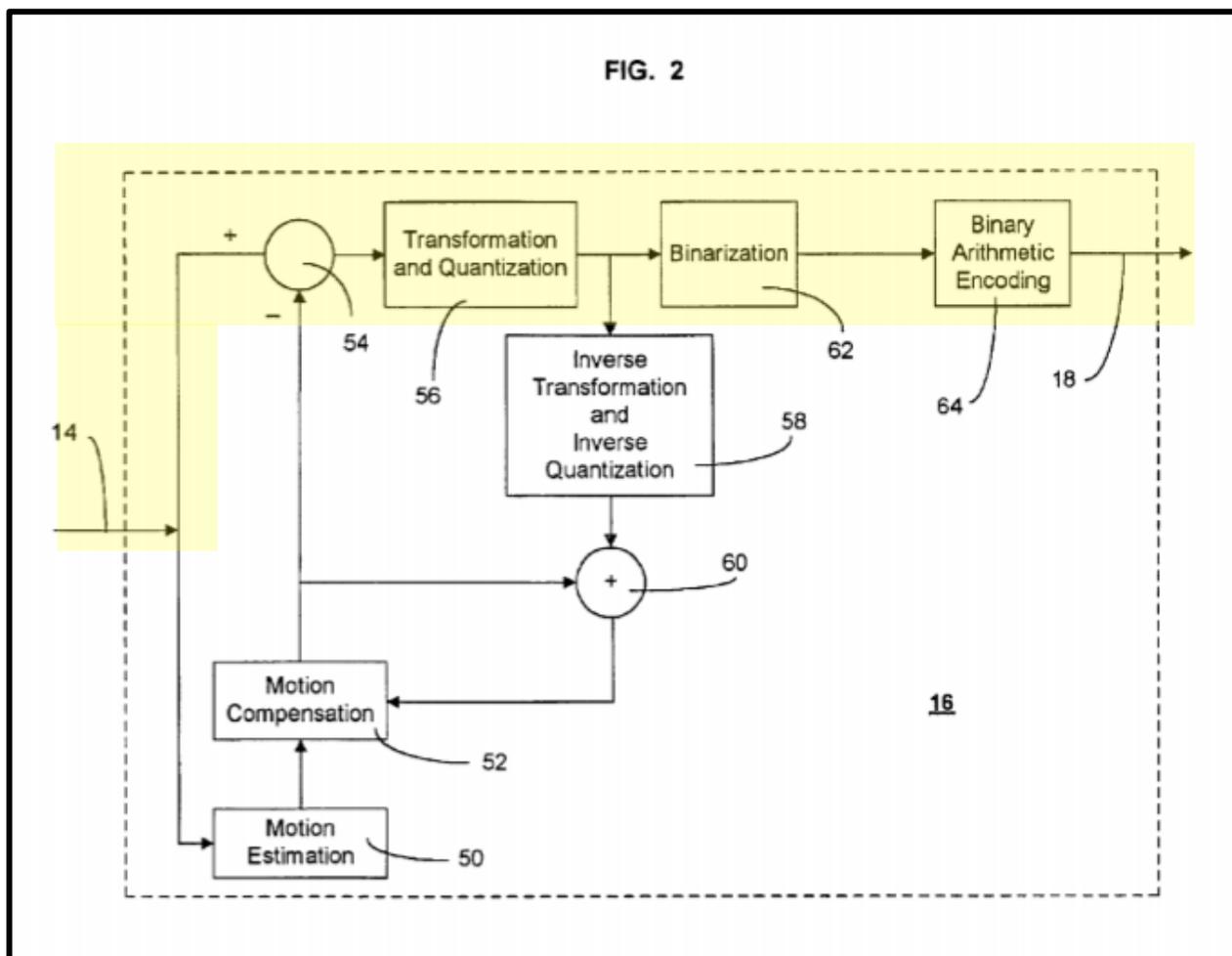
---

26 <sup>65</sup> *Id.* at 1:59-63, 4:26-36.

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

28 <sup>67</sup> ’387 Patent, 4:37-39.

<sup>68</sup> *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.<sup>69</sup> The output of the combination module passes to the next module (56), where it is transformed and quantized.<sup>70</sup> Symbols created by module 56 pass to the binarization module (62), which creates binary codewords that represent the symbols. The codewords next

<sup>69</sup> *Id.* at 3:49-53.

<sup>70</sup> *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.<sup>71</sup>

3 215. The '387 Patent is directed to the step in this process that occurs in the  
4 binarization module 62.<sup>72</sup> In the prior art, several techniques were available for  
5 binarizing the symbols, including, for example: unary, binary, Golomb, and exp-  
6 Golomb binarization.<sup>73</sup> 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....”<sup>74</sup> 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.<sup>75</sup>

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.”<sup>76</sup>

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.<sup>77</sup> 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 <sup>71</sup> *Id.* at 3:57-4:8.

26 <sup>72</sup> *Id.* at 4:1-2.

27 <sup>73</sup> *Id.* at 4:50-51.

28 <sup>74</sup> *Id.* at 4:52-54.

<sup>75</sup> *Id.* at 5:22-5:45.

<sup>76</sup> *Id.* at 6:19-24.

<sup>77</sup> *Id.* at 2:2-12.

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

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

9 A binarization system comprising:

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

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

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

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

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

25  
 26  
 27 <sup>78</sup> *Id.* at 6:26-36.

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

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

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

### 7 **Pipeline in the Cloud**

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

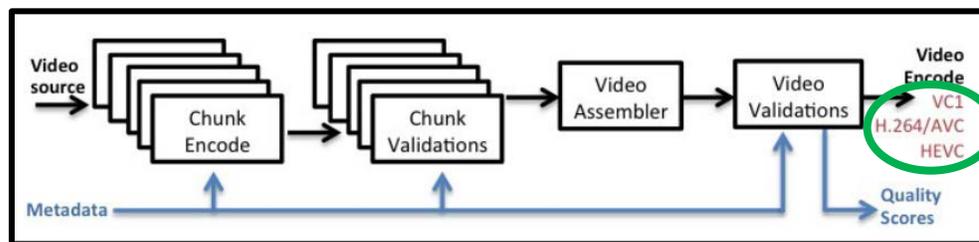
13 Source: [https://medium.com/netflix-techblog/high-quality-video-encoding-at-  
14 scale-d159db052746](https://medium.com/netflix-techblog/high-quality-video-encoding-at-scale-d159db052746)

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

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

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

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

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

```

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).

```

5 static void unary_exp_golomb_mv_encode(EncodingEnvironmentPtr eep_dp,
6                                     unsigned int symbol,
7                                     BiContextTypePtr ctx,
8                                     unsigned int max_bin)
9 {
10     if (symbol==0)
11     {
12         biari_encode_symbol(eep_dp, 0, ctx );
13         return;
14     }
15     else
16     {
17         unsigned int bin = 1;
18         unsigned int l = symbol, k = 1;
19         biari_encode_symbol(eep_dp, 1, ctx++ );
20
21         while (((--l)>0) && (++k <= 8))
22         {
23             biari_encode_symbol(eep_dp, 1, ctx );
24             if ((++bin) == 2)
25                 ++ctx;
26             if (bin == max_bin)
27                 ++ctx;
28         }
29         if (symbol < 8)
30             biari_encode_symbol(eep_dp, 0, ctx);
31         else
32             exp_golomb_encode_eq_prob(eep_dp, symbol - 8, 3);
33     }
34 }
  
```

17 Source: unary\_exp\_golomb\_mv\_encode() function

```

18 static void unary_exp_golomb_level_encode( EncodingEnvironmentPtr eep_dp,
19                                           unsigned int symbol,
20                                           BiContextTypePtr ctx)
21 {
22     if (symbol==0)
23     {
24         biari_encode_symbol(eep_dp, 0, ctx );
25         return;
26     }
27     else
28     {
29         unsigned int l=symbol;
30         unsigned int k = 1;
31
32         biari_encode_symbol(eep_dp, 1, ctx );
33         while (((--l)>0) && (++k <= 13))
34             biari_encode_symbol(eep_dp, 1, ctx);
35         if (symbol < 13)
36             biari_encode_symbol(eep_dp, 0, ctx);
37         else
38             exp_golomb_encode_eq_prob(eep_dp, symbol - 13, 0);
39     }
40 }
  
```

27 Source: unary\_exp\_golomb\_level\_encode() function

1           228. On information and belief, the Netflix video encoding pipeline also  
 2 includes a “means for constructing a codeword using an exp-Golomb binarization if  
 3 said code symbol index value is not less than said threshold value.” For instance,  
 4 functions unary\_exp\_golomb\_mv\_encode() and unary\_exp\_golomb\_level\_encode()  
 5 in the H.264.2 reference software both call the function  
 6 exp\_golomb\_encode\_eq\_prob() when the code symbol value is not less than the  
 7 above-described threshold in order to construct the remainder of the codeword  
 8 using exp-Golomb binarization.

```

  9
 10
 11
 12
 13
 14
 15
 16
 17
 18
 19
 20
 21
 22
 23
 24
 25
 26
 27
 28
  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

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

26           230. At least as of on or around September 26, 2019, when the Broadcom  
 27 Entities informed Netflix of its infringement of the ’387 Patent, and by no later than  
 28

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."<sup>81</sup>

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.<sup>82</sup> The  
26 MPEG formats make it possible to "represent[] a video signal with data roughly

27 \_\_\_\_\_  
28 <sup>81</sup> '663 Patent, Abstract; *see also id.* at 1:7-11.

<sup>82</sup> *Id.* at 1:15-30.

1 1/50th the size of the original uncompressed video, while still maintaining good  
 2 visual quality.”<sup>83</sup> 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.<sup>84</sup> Storing and transmitting only the changes,  
 6 instead of entire pictures, results in considerable savings in data transmission.<sup>85</sup>

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.”<sup>86</sup> The  
 10 ’663 Patent refers to the resulting transformed and quantized coefficients as  
 11 “symbols.”<sup>87</sup> Second, the transformed-quantized symbols are “binarized” to create  
 12 binary representations of each symbol in the form of a “codeword.”<sup>88</sup> Third, the  
 13 binarized codewords are entropy encoded “to reduce the number of bits per symbol  
 14 without introducing any additional video signal distortion.”<sup>89</sup> The patent explains  
 15 that several types of codecs<sup>90</sup> exist for performing the entropy encoding, “[o]ne of  
 16 the most efficient of which is the family of binary arithmetic encoders (BACs).”<sup>91</sup>  
 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.<sup>92</sup>

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 <sup>83</sup> *Id.* at 1:38-41.

<sup>84</sup> *Id.* at 3:31-36.

24 <sup>85</sup> *Id.* at 1:51-54; 3:35-39.

<sup>86</sup> *Id.* at 1:55-58; 3:41-44.

25 <sup>87</sup> *See, e.g., id.* at 1:55-64, 4:1-4, 4:45-54, 5:10-46, Table 1.

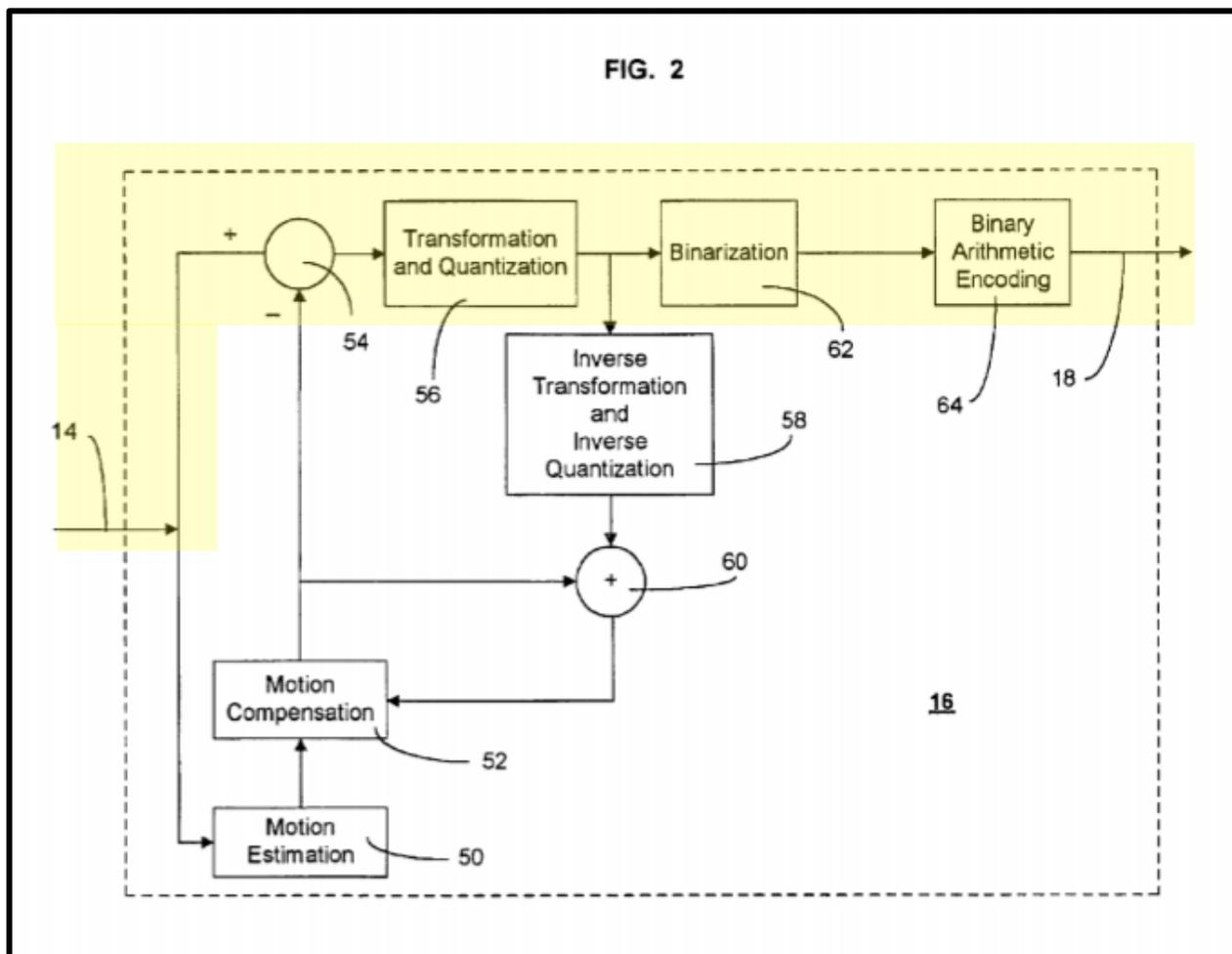
<sup>88</sup> *Id.* at 4:1-4.

26 <sup>89</sup> *Id.* at 1:59-63, 4:26-36.

27 <sup>90</sup> A codec is a device or computer program for encoding or decoding a digital  
 stream or signal.

<sup>91</sup> ’663 Patent, 4:34-36.

28 <sup>92</sup> *Id.* at 4:42-47.



Source: '663 Patent, Fig. 2

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

27 <sup>93</sup> *Id.* at 3:50-58.

28 <sup>94</sup> *Id.* at 3:57-4:8.

1           241. The '663 Patent is directed, in part, to the step in this process that  
 2 occurs in the binarization module 62.<sup>95</sup> In the prior art, several techniques were  
 3 available for binarizing the symbols, including, for example: unary, binary,  
 4 Golomb, and exp-Golomb binarization.<sup>96</sup> The patent explains that those techniques  
 5 each have certain strengths and weaknesses. Unary binarization, for example,  
 6 generates codewords that are distinguishable from one another but that can be  
 7 exceptionally long. Specifically, “[u]nary binarization consists of a number of  
 8 binary 1s equal to an index for a symbol followed by a zero....”<sup>97</sup> Thus, a symbol  
 9 index value of “1” results in a 2-bit codeword, namely “10.” A symbol index value  
 10 of “2” results in the 3-bit codeword “110,” and so on. Thus, each codeword is  
 11 easily distinguishable from the others, but the number of binary values can be quite  
 12 large—encoding a large symbol index may require tens of thousands of bits.<sup>98</sup>

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

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

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

25 \_\_\_\_\_  
 26 <sup>95</sup> *Id.* at 4:1-2.

<sup>96</sup> *Id.* at 4:46-47.

<sup>97</sup> *Id.* at 4:48-49.

<sup>98</sup> *Id.* at 5:22-5:45.

<sup>99</sup> *Id.* at 6:14-17.

<sup>100</sup> *Id.* at 2:1-12.

1 arithmetic encoding for entropy encoding.<sup>101</sup>

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  
9 great as a threshold;

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

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

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

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

24 247. Upon information and belief, Netflix directly infringes at least claim  
25 12 of the '663 Patent at least in the exemplary manner described below.

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

---

<sup>101</sup> *Id.* at 6:19-28.

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

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

7

8 **Table 3 - Motion vector magnitude residual binarization.**

Index	Unary Prefix	exp-Golomb Suffix
0	0	
1	10	
2	110	
...		
63	1...10	
64	1...110	0
65	1...110	1
66	1...1110	00
67	1...1110	01
68	1...1110	10
69	1...1110	11
70	1...11110	000
71	1...11110	001
72	1...11110	010
73	1...11110	011
74	1...11110	100
75	1...11110	101
...		

9

10

11

12

13

14

15

16

17

18

19

20

21

22 **FIG. 5**

23

24 Source: '663 Patent, Fig. 5

25

26

27 <sup>103</sup> '663 Patent, 6:35-40.

28 <sup>104</sup> 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.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

```

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

1           252. On information and belief, the Netflix video encoding system also  
 2 “generat[es] a second pattern in a second portion of said codeword following said  
 3 first portion representing an offset of said index value above said threshold.” For  
 4 example, both the unary\_exp\_golomb\_mv\_encode() and  
 5 unary\_exp\_golomb\_level\_encode() functions call the function  
 6 exp\_golomb\_encode\_eq\_prob() when the index value exceeds the thresholds  
 7 described above. The exp\_golomb\_encode\_eq\_prob() function generates a second  
 8 portion of the codeword—the unary representation appended to the initial prefix to  
 9 form the unary prefix—when the index value exceeds the thresholds.

```

10 static void exp_golomb_encode_eq_prob( EncodingEnvironmentPtr eep_dp,
11                                     unsigned int symbol,
12                                     int k)
13 {
14     for(;;)
15     {
16         if (symbol >= (unsigned int)(1<<k))
17         {
18             biari_encode_symbol_eq_prob(eep_dp, 1); //first unary part
19             symbol = symbol - (1<<k);
20             k++;
21         }
22         else
23         {
24             biari_encode_symbol_eq_prob(eep_dp, 0); //now terminated zero of unary part
25             while (k--) //next binary part
26                 biari_encode_symbol_eq_prob(eep_dp, ((symbol>>k)&1));
27             break;
28         }
29     }
30 }

```

Source: exp\_golomb\_encode\_eq\_prob() function

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

```

8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
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.<sup>105</sup> 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.”<sup>106</sup> Subsequently,  
14 “prediction units” (PU) can be encoded for different “partition modes,” to be used  
15 in the intra- or inter-prediction processing.

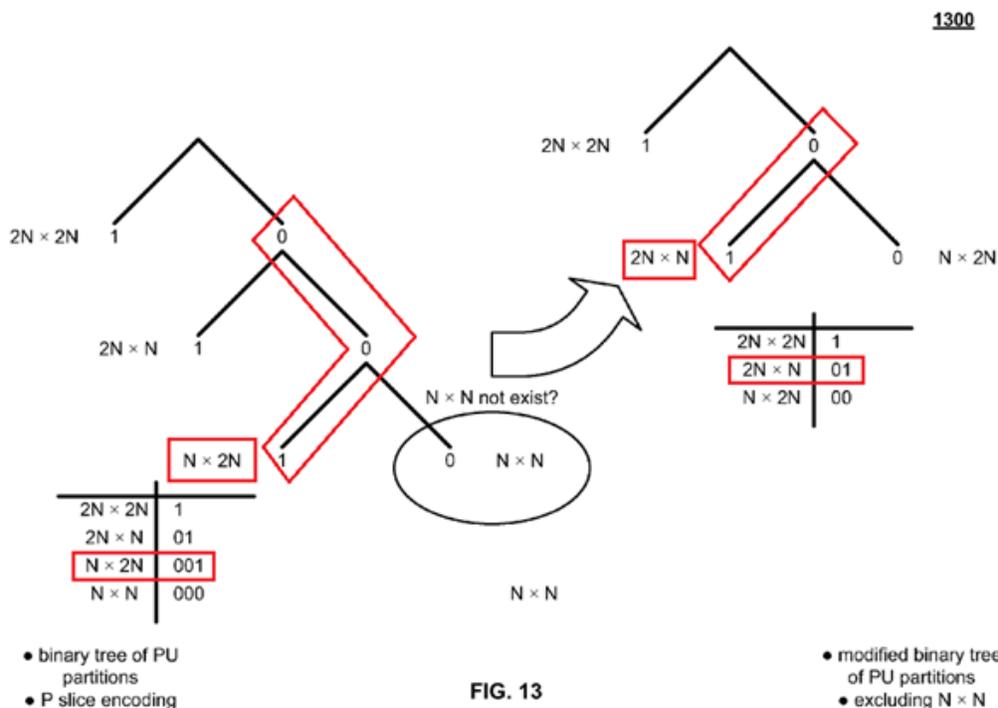
16 265. Generally speaking, prediction and partition modes are encoded using  
17 binary “codewords.” These codewords can be generated using a “binary tree.” In  
18 general terms, a binary tree is a data structure that can be used to represent data and  
19 associate it with a corresponding bit sequence 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.<sup>107</sup> Figure 13 of the ’283 Patent, for example, represents  
25 binary trees as follows:

26  
27 <sup>105</sup> ’283 Patent, at 17:24-48.

28 <sup>106</sup> *See, e.g., id.* at 17:9-12.

<sup>107</sup> *See id.* at Figs. 13, 14, 15.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28



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

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

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

<sup>108</sup> *Id.* at Fig. 13 (annotations added).

<sup>109</sup> *Id.* at 19:21-35; 20:19-30.

<sup>110</sup> ’283 Patent at 18:55-19:12.

<sup>111</sup> *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

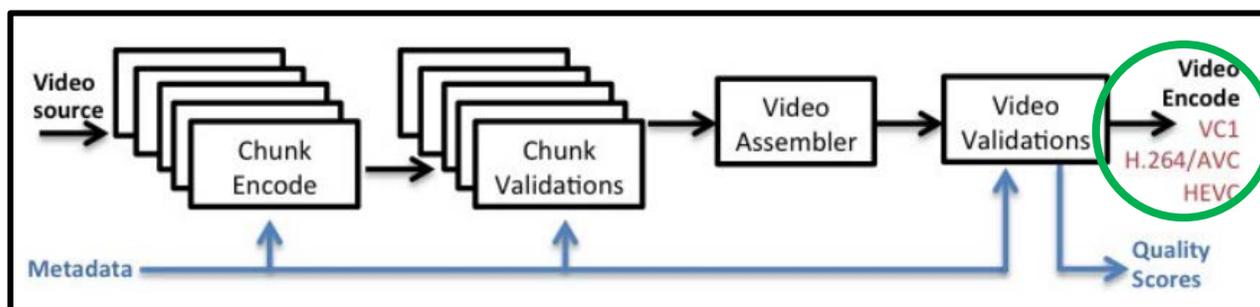
1 CUs and is based on a different size PU than the size  $N \times N$   
 2 PU when the selected CU is another CU than the SCU of  
 the plurality of CUs, wherein  $N$  is a positive integer.

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  
 representations per profile. The encodes are packaged  
 11 and then deployed to a content delivery network for  
 streaming. During a streaming session, the client requests  
 12 the encodes it can play and adaptively switches among  
 quality levels based on network conditions.<sup>112</sup>

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 <sup>112</sup> [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.<sup>113</sup>

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.<sup>114</sup>

```
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 };
```

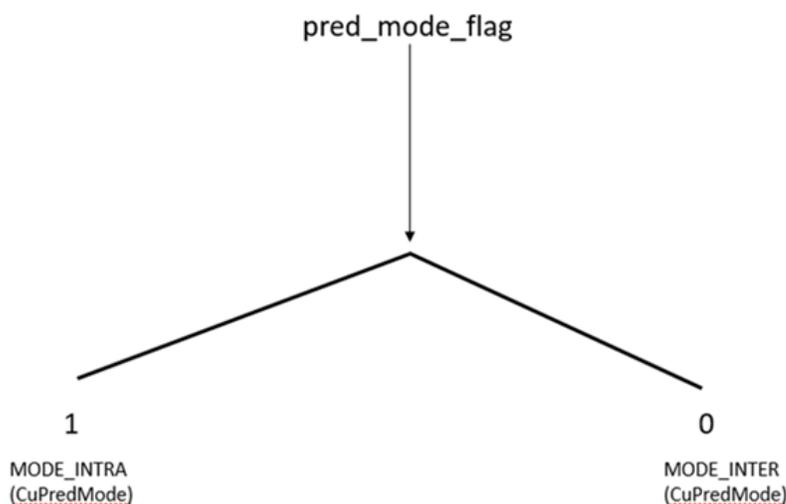
```
21 Void TEncSbac::codePartSize( TComDataCU* pcCU, UInt
22 uiAbsPartIdx, UInt uiDepth )
23 {
24     PartSize eSize          = pcCU->getPartitionSize(
25     uiAbsPartIdx );
26     ...
27     const UInt log2DiffMaxMinCodingBlockSize = pcCU-
28     >getSlice()->getSPS()-
29     >getLog2DiffMaxMinCodingBlockSize();
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 <sup>113</sup> 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>)  
28 (“H.265 Recommendation”) (describing binary tree coding syntax using pseudo  
code).

<sup>114</sup> See also, H.265 Recommendation at § 7.4.9.5.

1  
2 Source: H.265.2: Reference software for ITU-T H.265 High Efficiency  
3 Video Coding (December 2016) (available at [https://www.itu.int/rec/T-REC-](https://www.itu.int/rec/T-REC-H.265.2)  
4 [H.265.2](https://www.itu.int/rec/T-REC-H.265.2)).

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



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

27  
28

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

```

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

```

```

Void TEncSbac::codePartSize( TComDataCU* pcCU, UInt
uiAbsPartIdx, UInt uiDepth )
{
    PartSize eSize = pcCU->getPartitionSize(
uiAbsPartIdx );
    ..|
    switch(eSize)
    {
        case SIZE_2Nx2N:
        {
            m_pcBinIf->encodeBin( 1, m_cCUPartSizeSCModel.get(
0, 0, 0) );
            break;
        }
        ...
    }
}

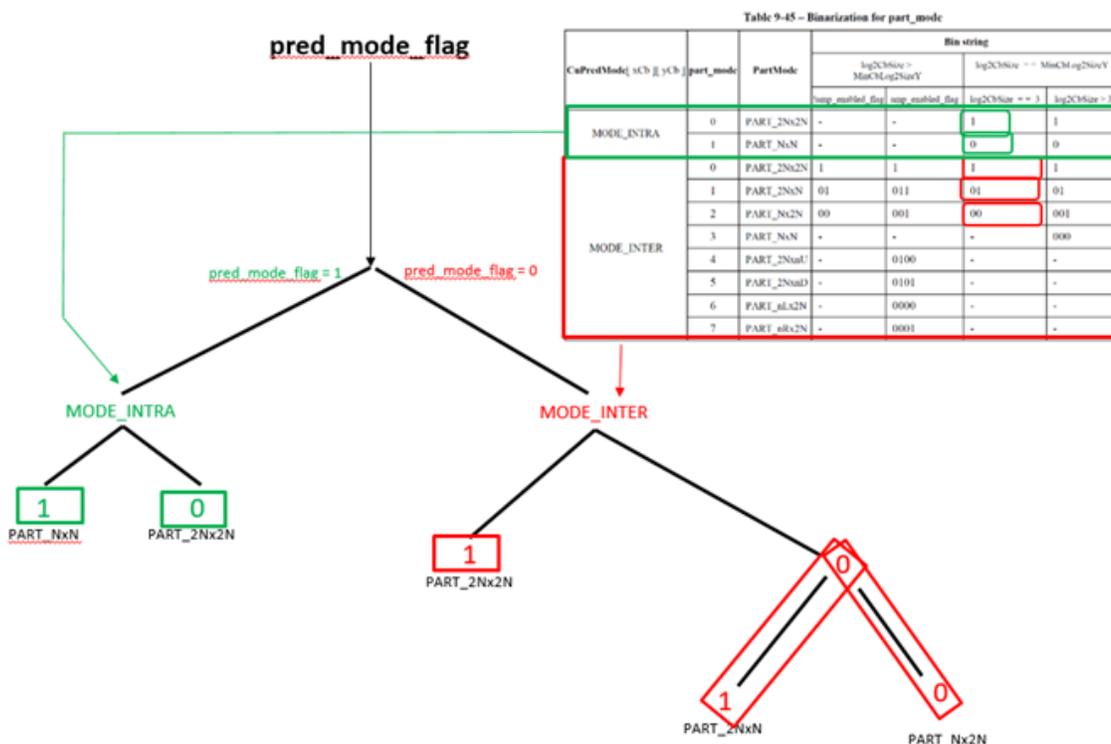
```

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.<sup>115</sup> 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.

<sup>115</sup> See H.265 Recommendation, at Table 9-45.

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28



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.

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

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

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

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

24  
25  
26 <sup>116</sup> 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).

27 <sup>117</sup> See Table 9-45 at column “PartMode.”

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

**PRAYER FOR RELIEF**

1  
2 WHEREFORE, the Broadcom Entities respectfully request that the Court  
3 enter a judgment in their favor and against Netflix:

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

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

8 3. Declaring that Netflix's infringement of the '079, '121, '245, '992,  
9 '138, '387, '663, and '283 Patents is willful and deliberate pursuant to 35 U.S.C. §  
10 284;

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

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

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

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

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

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

**JURY DEMAND**

25  
26 The Broadcom Entities demand a trial by jury of all claims in this action so  
27 triable.

28

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28

Dated: March 13, 2020

HOPKINS & CARLEY  
A Law Corporation

By: /s/ Jason S. Angell

Jason S. Angell  
Christopher A. Hohn  
Cary Chien  
Robert K. Jain  
Attorneys for Plaintiff  
BROADCOM CORPORATION,  
AVAGO TECHNOLOGIES  
INTERNATIONAL SALES PTE.  
LIMITED