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

| | |
|---|---|
| REALTIME ADAPTIVE STREAMING LLC,<br><br>Plaintiff,<br><br>v.<br><br>NETFLIX, INC. and<br>NETFLIX STREAMING SERVICES, INC.,<br><br>Defendants. | C.A. No. ____________________<br><br>**JURY TRIAL DEMANDED** |

## COMPLAINT FOR PATENT INFRINGEMENT

This is an action for Patent infringement arising under the Patent Laws of the United States of America, 35 U.S.C. § 1 *et seq.* in which Plaintiff Realtime Adaptive Streaming LLC ("Plaintiff" or "Realtime") makes the following allegations against Defendants Netflix, Inc. and Netflix Streaming Services, Inc. (collectively "Netflix" or "Defendants").

## PARTIES

1. Realtime is a Texas limited liability company. Realtime has a place of business at 1828 E.S.E. Loop 323, Tyler, Texas 75701. Realtime has researched and developed specific solutions for data compression, including, for example, those that increase the speeds at which data can be stored and accessed. As recognition of its innovations rooted in this technological field, Realtime holds multiple United States Patents and pending Patent applications.

2. Defendant Netflix, Inc. is a Delaware corporation, with its principal place of business at 100 Winchester Circle, Los Gatos, California 95032. Netflix, Inc. may be served with process by serving its registered agent, The Corporation Trust Company at

the Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801.

3. Defendant Netflix Streaming Services, Inc. is a Delaware corporation, with its principal place of business at 100 Winchester Circle, Los Gatos, California 95032. Netflix Streaming Services, Inc. may be served with process by serving its registered agent, The Corporation Trust Company at the Corporation Trust Center, 1209 Orange Street, Wilmington, Delaware 19801.

4. Defendants reside in this District because they are incorporated in Delaware. Defendants offer their products and/or services, including those accused herein of infringement, to customers and potential customers located in Delaware and in this District.

**JURISDICTION AND VENUE**

5. This action arises under the Patent laws of the United States, Title 35 of the United States Code. This Court has original subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

6. This Court has personal jurisdiction over Defendants in this action because Defendants have committed acts within the District of Delaware giving rise to this action and has established minimum contacts with this forum such that the exercise of jurisdiction over Defendants would not offend traditional notions of fair play and substantial justice. Defendants have also committed and continue to commit acts of infringement in this District by, among other things, offering to sell and selling products and/or services that infringe the asserted Patents.

7. Venue is proper in this district, *e.g.*, under 28 U.S.C. § 1400(b). Defendants reside in this District because they are incorporated in Delaware.

Furthermore, upon information and belief, Defendants have transacted business in the District of Delaware and have committed acts of direct and indirect infringement in the District of Delaware.

**THE PATENTS-IN-SUIT**

8. This action arises under 35 U.S.C. § 271 for Netflix's infringement of Realtime's United States Patent Nos. 7,386,046 (the "'046 Patent"), 8,634,462 (the "'462 Patent"), 8,934,535 (the "'535 Patent"), 9,578,298 (the "'298 Patent"), 9,762,907 (the "'907 Patent"), and 9,769,477 (the "'477 Patent") (collectively, the "Patents-In-Suit").

9. The '046 Patent, titled "Bandwidth Sensitive Data Compression and Decompression," was duly and properly issued by the United States Patent and Trademark Office ("USPTO") on June 10, 2008. A copy of the '046 Patent is attached hereto as Exhibit A. Realtime is the owner and assignee of the '046 Patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

10. The '462 Patent, titled "Quantization for Hybrid Video Coding," was duly and properly issued by the USPTO on January 21, 2014. A copy of the '462 Patent is attached hereto as Exhibit B. Realtime is the owner and assignee of the '462 Patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

11. The '535 Patent, titled "Systems and methods for video and audio data storage and distribution," was duly and properly issued by the USPTO on January 13, 2015. A copy of the '535 Patent is attached hereto as Exhibit C. Realtime is the owner and assignee of the '535 Patent and holds the right to sue for and recover all damages for

infringement thereof, including past infringement.

12. The ’298 Patent, titled “Method for Decoding 2D-Compatible Stereoscopic Video Flows,” was duly and properly issued by the USPTO on February 21, 2017. A copy of the ’298 Patent is attached hereto as Exhibit D. Realtime is the owner and assignee of the ’298 Patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

13. The ’907 Patent, titled “System and Methods for Video and Audio Data Distribution,” was duly and properly issued by the USPTO on September 12, 2017. A copy of the ’907 Patent is attached hereto as Exhibit E. Realtime is the owner and assignee of the ’907 Patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

14. The ’477 Patent, titled “Video data compression systems,” was duly and properly issued by the USPTO on September 19, 2017. A copy of the ’477 Patent is attached hereto as Exhibit F. Realtime is the owner and assignee of the ’477 Patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

**COUNT I**
**<u>INFRINGEMENT OF U.S. PATENT NO. 7,386,046</u>**

15. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

16. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ‘046 Patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix’s streaming video service; Netflix’s video encoders or codecs

including x264, x265 and libvpx; and Netflix's mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, and all versions and variations thereof since the issuance of the '046 Patent ("Accused Instrumentalities").

17. For example, an official website from Netflix known as "The Netflix Tech Blog" states that Netflix is "introducing two new types of mobile encodes – **AVCHi-Mobile and VP9-Mobile**…All the changes combined result in better video quality for the same bitrate compared to our current streams (**AVCMain**). Many Netflix-ready devices receive streams which are encoded using the **H.264/AVC Main profile (AVCMain**). This is a widely-used video compression format, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. However, newer formats are available that offer more sophisticated video coding tools. For our mobile bitstreams we adopt two compression formats: **H.264/AVC High profile and VP9 (profile 0)**. Similar to Main profile, **the High profile of H.264/AVC** enjoys broad decoder support. **VP9**, a royalty-free format developed by Google, is supported on the majority of Android devices, Chrome, and a growing number of consumer devices." (emphasis added). *See* https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

18. As confirmation, an article from Variety states that "Netflix has been using H.264/AVC almost exclusively" and "That's why Netflix is also encoding its downloadable videos with a different flavor of H.264/AVC, which is also known as a different profile. (For the technically inclined: Netflix's streams are encoded with **H.264/AVC Main,** whereas its downloads come in **H.264/AVC High**. The company

shared more details on its tech blog this week.) This isn't quite as effective as using VP9, but still allows Netflix to shave off some bits." *See* Janko Roettgers, *Variety*, December 2, 2016, "How Netflix Delivers Better-Looking Downloads Without Eating Up All Your Phone Storage," http://variety.com/2016/digital/news/netflix-offline-downloads-codecs-vp9-1201932502/.

19. In another entry of Netflix's "The Netflix Tech Blog," a test comparing several different encoders was described: "H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC…**x265** is an open-source HEVC encoder, originally ported from the x264 codebase. Concurrent to HEVC, Google developed VP9 as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

20. The Accused Instrumentalities determine a parameter of at least a portion of a video data block. As shown below, examples of such parameters include bitrate (or max video bitrate) and resolution parameters. Different parameters correspond with different end applications. H.264 provides for multiple different ranges of such parameters, each included in the "profiles" and "levels" as defined by the H.264 standard, from the below shown paragraphs from a white paper and Wikipedia. *See* http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf at 5:

### 4. H.264 profiles and levels

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

*See* https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC:

**Levels with maximum property values**

| Level | Max decoding speed | | Max frame size | | Max video bit rate for video coding layer (VCL) kbit/s | | | Examples for high resolution @ highest frame rate (max stored frames) Toggle additional details |
|---|---|---|---|---|---|---|---|---|
| | Luma samples/s | Macroblocks/s | Luma samples | Macroblocks | Baseline, Extended and Main Profiles | High Profile | High 10 Profile | |
| 1 | 380,160 | 1,485 | 25,344 | 99 | 64 | 80 | 192 | 176×144@15.0 (4) |
| 1b | 380,160 | 1,485 | 25,344 | 99 | 128 | 160 | 384 | 176×144@15.0 (4) |
| 1.1 | 768,000 | 3,000 | 101,376 | 396 | 192 | 240 | 576 | 352×288@7.5 (2) |
| 1.2 | 1,536,000 | 6,000 | 101,376 | 396 | 384 | 480 | 1,152 | 352×288@15.2 (6) |
| 1.3 | 3,041,280 | 11,880 | 101,376 | 396 | 768 | 960 | 2,304 | 352×288@30.0 (6) |
| 2 | 3,041,280 | 11,880 | 101,376 | 396 | 2,000 | 2,500 | 6,000 | 352×288@30.0 (6) |
| 2.1 | 5,068,800 | 19,800 | 202,752 | 792 | 4,000 | 5,000 | 12,000 | 352×576@25.0 (6) |
| 2.2 | 5,184,000 | 20,250 | 414,720 | 1,620 | 4,000 | 5,000 | 12,000 | 720×576@12.5 (5) |
| 3 | 10,368,000 | 40,500 | 414,720 | 1,620 | 10,000 | 12,500 | 30,000 | 720×576@25.0 (5) |
| 3.1 | 27,648,000 | 108,000 | 921,600 | 3,600 | 14,000 | 17,500 | 42,000 | 1,280×720@30.0 (5) |
| 3.2 | 55,296,000 | 216,000 | 1,310,720 | 5,120 | 20,000 | 25,000 | 60,000 | 1,280×1,024@42.2 (4) |
| 4 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 20,000 | 25,000 | 60,000 | 2,048×1,024@30.0 (4) |
| 4.1 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 50,000 | 62,500 | 150,000 | 2,048×1,024@30.0 (4) |
| 4.2 | 133,693,440 | 522,240 | 2,228,224 | 8,704 | 50,000 | 62,500 | 150,000 | 2,048×1,080@60.0 (4) |
| 5 | 150,994,944 | 589,824 | 5,652,480 | 22,080 | 135,000 | 168,750 | 405,000 | 3,672×1,536@26.7 (5) |
| 5.1 | 251,658,240 | 983,040 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@26.7 (5) |
| 5.2 | 530,841,600 | 2,073,600 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@56.3 (5) |

21. A video data block is organized by the group of pictures (GOP) structure, which is a "collection of successive pictures within a coded video stream." *See* https://en.wikipedia.org/wiki/Group_of_pictures. A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See* https://en.wikipedia.org/wiki/Video_compression_picture_types (for descriptions of I frames, P frames and B frames); https://en.wikipedia.org/wiki/MPEG-1#D-frames (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max video bitrate and resolution parameters) or even be considered as a parameter by itself.

22. Based on the bitrate and/or resolution parameter identified (e.g. bitrate, max video bitrate, resolution, GOP structure or frame type within a GOP structure), any H.264-compliant system such as the Accused Instrumentalities would determine which profile (e.g., "baseline," "extended," "main", or "high") corresponds with that parameter, then select between at least two asymmetric compressors. If baseline or extended is the corresponding profile, then the system will select a Context-Adaptive Variable Length Coding ("CAVLC") entropy encoder. If main or high is the corresponding profile, then the system will select a Context-Adaptive Binary Arithmetic Coding ("CABAC") entropy encoder. Both encoders are asymmetric compressors because it takes a longer period of time for them to compress data than to decompress data. *See*

https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

| | Baseline | Extended | Main | High | High 10 |
|---|---|---|---|---|---|
| I and P Slices | Yes | Yes | Yes | Yes | Yes |
| B Slices | No | Yes | Yes | Yes | Yes |
| SI and SP Slices | No | Yes | No | No | No |
| Multiple Reference Frames | Yes | Yes | Yes | Yes | Yes |
| In-Loop Deblocking Filter | Yes | Yes | Yes | Yes | Yes |
| CAVLC Entropy Coding | Yes | Yes | Yes | Yes | Yes |
| CABAC Entropy Coding | No | No | Yes | Yes | Yes |
| Flexible Macroblock Ordering (FMO) | Yes | Yes | No | No | No |
| Arbitrary Slice Ordering (ASO) | Yes | Yes | No | No | No |
| Redundant Slices (RS) | Yes | Yes | No | No | No |
| Data Partitioning | No | Yes | No | No | No |
| Interlaced Coding (PicAFF, MBAFF) | No | Yes | Yes | Yes | Yes |
| 4:2:0 Chroma Format | Yes | Yes | Yes | Yes | Yes |
| Monochrome Video Format (4:0:0) | No | No | No | Yes | Yes |
| 4:2:2 Chroma Format | No | No | No | No | No |
| 4:4:4 Chroma Format | No | No | No | No | No |
| 8 Bit Sample Depth | Yes | Yes | Yes | Yes | Yes |
| 9 and 10 Bit Sample Depth | No | No | No | No | Yes |
| 11 to 14 Bit Sample Depth | No | No | No | No | No |
| 8×8 vs. 4×4 Transform Adaptivity | No | No | No | Yes | Yes |
| Quantization Scaling Matrices | No | No | No | Yes | Yes |
| Separate Cb and Cr QP control | No | No | No | Yes | Yes |
| Separate Color Plane Coding | No | No | No | No | No |
| Predictive Lossless Coding | No | No | No | No | No |

*See* http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

**H.264 Entropy Coding – Comparison of Approaches**

| Characteristics | Variable Length Coding (VLC) | Context Adaptive Binary Arithmetic Coding(CABAC) |
|---|---|---|
| • Where it is used | MPEG-2, MPEG-4 ASP | H.264/MPEG-4 AVC (high efficiency option) |
| • Probability distribution | **Static** - Probabilities never change | **Adaptive** - Adjusts probabilities based on actual data |
| • Leverages correlation between symbols | **No** - Conditional probabilities ignored | **Yes** - Exploits symbol correlations by using "contexts" |
| • Non-integer code words | **No** - Low coding efficiency forhigh probability symbols | **Yes** - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency |

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy_coding_mode_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

23. The Accused Instrumentalities compress the at least the portion of the data block with the selected one or more asymmetric compressors to provide one or more compressed data blocks, which can be organized in a GOP structure (see above). After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can also be organized in a GOP structure, as discussed previously above. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

**Entropy Coding**

For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic coding (CABAC) which are complex techniques to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. The use of CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total processing power to be accomplished.

*See*

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf at 13:

Typical compression ratios to maintain excellent quality are:

- 10:1 for general images using JPEG
- 30:1 for general video using H.263 and MPEG-2
- 60:1 for general video using H.264 and WMV9

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles includes the functionality of the Baseline Profile and add improvements to the predictions algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

*See* http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf at 2:

24. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the ’046 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 1 of the ’046 Patent, namely, a method comprising: compressing data using a first compression routine providing a first compression rate, wherein the first compression routine comprises a first compression algorithm; tracking the throughput of a data processing system to determine if the first compression rate provides a throughput that meets a predetermined throughput threshold, wherein said tracking throughput comprises tracking a number of pending requests for data transmission; and when the tracked

throughput does not meet the predetermined throughput threshold, compressing data using a second compression routine providing a second compression rate that is greater than the first compression rate, to increase the throughput of the data processing system to at least the predetermined throughput level, wherein the second compression routine comprises a second compression algorithm. Upon information and belief, Netflix uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

25. On information and belief, the Accused Instrumentalities store at least a portion of the one or more compressed data blocks in buffers, hard disk, or other forms of memory/storage.

26. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’046 Patent, for similar reasons as explained above with respect to Claim 1 of the ’046 Patent.

27. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the H.264 standard.

28. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the systems and/or methods claimed by the ’046 Patent.

29. On information and belief, Netflix has had knowledge of the ’046 Patent since at least the filing of this Complaint or shortly thereafter, and on information and

belief, Netflix knew of the ’046 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’046 Patent.

30. Upon information and belief, Netflix’s affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the ’046 Patent by practicing a method comprising: compressing data using a first compression routine providing a first compression rate, wherein the first compression routine comprises a first compression algorithm; tracking the throughput of a data processing system to determine if the first compression rate provides a throughput that meets a predetermined throughput threshold, wherein said tracking throughput comprises tracking a number of pending requests for data transmission; and when the tracked throughput does not meet the predetermined throughput threshold, compressing data using a second compression routine providing a second compression rate that is greater than the first compression rate, to increase the throughput of the data processing system to at least the predetermined throughput level, wherein the second compression routine comprises a second compression algorithm. For example, Netflix adopted H.264 as its video codec in its products/services, such as its streaming services, and uses H.264 as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused Instrumentalities to infringe other

claims of the ’046 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the ’046 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the ’046 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continues to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’046 Patent, knowing that such use constitutes infringement of the ’046 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’046 Patent, in violation of 35 U.S.C. § 271(b).

31. Netflix has also infringed, and continues to infringe, claims of the ‘046 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’046 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’046 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’046 Patent, in violation of 35 U.S.C. § 271(c).

32. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Netflix has injured Realtime and is

liable to Realtime for infringement of the ’046 Patent pursuant to 35 U.S.C. § 271.

33. As a result of Netflix’s infringement of the ’046 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix’s infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

**COUNT II**
**INFRINGEMENT OF U.S. PATENT NO. 8,634,462**

34. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

35. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ’462 Patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix’s streaming video service; Netflix’s video encoders or codecs including x264, x265 and libvpx; and Netflix’s mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, H.265/HEVC and all versions and variations thereof since the issuance of the ’462 Patent (“Accused Instrumentalities”).

36. For example, in an entry of Netflix’s “The Netflix Tech Blog,” a test comparing several different encoders was described: “H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC. HEVC is the successor to H.264/AVC and results reported from standardization showed about 50% bitrate savings for the same quality compared to H.264/AVC. **x265** is an open-source HEVC encoder, originally

ported from the x264 codebase. Concurrent to HEVC, Google developed VP9 as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." The results of that test were as follows: "Here's a snapshot: x265 and libvpx demonstrate superior compression performance compared to x264, with bitrate savings reaching up to 50% especially at the higher resolutions. x265 outperforms libvpx for almost all resolutions and quality metrics, but the performance gap narrows (or even reverses) at 1080p." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

37. In addition, another article on the website "Arstechnica" mentions that "There's also the matter of hardware decoding support for 10-bit HEVC, the 4K codec used by Netflix and other streaming services." *See* https://arstechnica.com/gadgets/2016/11/netflix-4k-streaming-pc-kaby-lake-cpu-windows-10-edge-browser/.

38. The Accused Instrumentalities performs a method for coding a video signal using hybrid coding. For example, the aim of the coding process is the production of a bitstream, as defined in definition 3.12 of the ITU-T H.265 Series H: Audiovisual and Multimedia Systems, "Infrastructure of audiovisual services – Coding of moving video" High efficiency video coding ("HEVC Spec"): "bitstream: A sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of coded pictures and associated data forming one or more coded video sequences (CVSs)." *See also, e.g.,* "Overview of the High Efficiency Video Coding (HEVC) Standard" by Gary J. Sullivan, Fellow, IEEE, Jens-Rainer Ohm, Member, IEEE, Woo-Jin Han, Member, IEEE, and Thomas Wiegand, Fellow, IEEE, published in IEEE TRANSACTIONS ON

CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, DECEMBER 2012 ("IEEE HEVC) ("The video coding layer of HEVC employs the same hybrid approach (inter-/intrapicture prediction and 2-D transform coding) used in all video compression standards since H.261"). *See also, e.g.*, HEVC Spec at 0.7 "Overview of the design characteristics."

39. The Accused Instrumentalities reduce temporal redundancy by block based motion compensated prediction in order to establish a prediction error signal. For example, clause 8.5.3 Decoding process for prediction units in inter prediction mode and the subclauses thereof of the HEVC Spec describe the block based motion compensation techniques used in the decoding process. *See also, e.g.*, IEEE HEVC at 1651-1652 6) Motion compensation: Quarter-sample precision is used for the MVs, and 7-tap or 8-tap filters are used for interpolation of fractional-sample positions (compared to six-tap filtering of half-sample positions followed by linear interpolation for quarter-sample positions in H.264/MPEG-4 AVC). Similar to H.264/MPEG-4 AVC, multiple reference pictures are used. For each PB, either one or two motion vectors can be transmitted, resulting either in unipredictive or bipredictive coding, respectively. As in H.264/MPEG-4 AVC, a scaling and offset operation may be applied to the prediction signal(s) in a manner known as weighted prediction.").

40. The Accused Instrumentalities perform quantization on samples of the prediction error signal or on coefficients resulting from a transformation of the prediction error signal into the frequency domain to obtain quantized values, representing quantized samples or quantized coefficients respectively. For example, the quantization parameter and the scaling (inverse quantization) are defined in definitions 3.112 (page 10) and

3.131 (page 11), respectively, the usage of the scaling process in the decoding being described in clause and 8.6 Scaling, transformation and array construction process prior to deblocking filter process of the HEVC Spec. *See also, e.g.*, IEEE HEVC at 1652 ("8) Quantization control: As in H.264/MPEG-4 AVC, uniform reconstruction quantization (URQ) is used in HEVC, with quantization scaling matrices supported for the various transform block sizes.").

41. The Accused Instrumentalities perform a method wherein the prediction error signal includes a plurality of subblocks each including a plurality of quantized values. For example, the quantized samples or transform coefficients from the subblock are scaled and transformed as described in above mentioned clause 8.6 of the HEVC Spec. *See also, e.g.*, IEEE HEVC at 1652 ("Prediction units and prediction blocks (PBs): The decision whether to code a picture area using interpicture or intrapicture prediction is made at the CU level. A PU partitioning structure has its root at the CU level. Depending on the basic prediction-type decision, the luma and chroma CBs can then be further split in size and predicted from luma and chroma prediction blocks (PBs). HEVC supports variable PB sizes from 64×64 down to 4×4 samples.").

42. The Accused Instrumentalities perform a method of calculating a first quantization efficiency for the quantized values of at least one subblock of the plurality of subblocks; setting the quantized values of the at least one subblock to all zeroes; calculating a second quantization efficiency for the at least one subblock while all of the quantized values are zeroes; selecting which of the first and second quantization efficiencies is a higher efficiency; and selecting, for further proceeding, the at least one subblock with the quantized values prior to setting the quantized values of the at least one

subblock to all zeroes if the first quantization efficiency is higher and selecting the at least one subblock with the quantized values set to zero, for further proceeding, if the second quantization efficiency is higher. For example, the bitstream resulting from the encoding as described in this last item of the claim contains all the relevant information as needed by the decoder for proper decoding. If the coefficients of the subblock are set to zero as a consequence of the efficiency calculation, the coded_sub_block_flag, as described in clause 7.4.9.11 Residual coding semantics, HEVC Spec, is set to 0, indicating that all the 16 coefficients of the coded sub block have been set to 0: "coded_sub_block_flag[ xS ][ yS ] specifies the following for the sub-block at location ( xS, yS ) within the current transform block, where a sub-block is a (4x4) array of 16 transform coefficient levels: – If coded_sub_block_flag[ xS ][ yS ] is equal to 0, the 16 transform coefficient levels of the sub-block at location ( xS, yS ) are inferred to be equal to 0."

43. When coded_sub_block_flag[ xS ][ yS ] has not been set equal to 0, the position in the array of non 0 coefficients can be determined as follows:

– Otherwise (coded_sub_block_flag[ xS ][ yS ] is equal to 1), the following applies:

> – If ( xS, yS ) is equal to ( 0, 0 ) and ( LastSignificantCoeffX, LastSignificantCoeffY ) is not equal to ( 0, 0 ), at least one of the 16 sig_coeff_flag syntax elements is present for the sub-block at location ( xS, yS ) .
>
> – Otherwise, at least one of the 16 transform coefficient levels of the sub-block at location ( xS, yS ) has a non zero value.
>
> When coded_sub_block_flag[ xS ][ yS ] is not present, it is inferred as follows:

– If one or more of the following conditions are true, coded_sub_block_flag[ xS ][ yS ] is inferred to be equal to 1:

– ( xS, yS ) is equal to ( 0, 0 )

– ( xS, yS ) is equal to ( LastSignificantCoeffX >> 2 , LastSignificantCoeffY >> 2 )

– Otherwise, coded_sub_block_flag[ xS ][ yS ] is inferred to be equal to 0.

HEVC Spec at 7.4.9.11 Residual coding semantics. Therefore, even though the coding algorithms than can be used for reaching specific efficiency targets are not specified by the HEVC Spec (as stated in clause 0.7), this particular combination of choices produces a valid bitstream that has to be decoded by a conformant decoder.

44. The infringement of the Accused Instrumentalities is also shown by way of considering the reference software (*see*, *e.g.*, https://hevc.hhi.fraunhofer.de/). Setting the flag RDOQ=true in the encoder configuration file enables rate-distortion-optimized quantization for transformed TUs. This feature is implemented in the HM reference software as function xRateDistOptQuant in file TComTrQuant.cpp. In the function xRateDistOptQuant, the efficiency for setting all quantized values to zero is calculated and stored in the variable d64BestCost. In the variable iBestLastIdxP1, a 0 is stored indicating that all values starting from the 0th position are set to zero. Afterwards, the efficiency for keeping quantized values unequal to zero is calculated and stored in the variable totalCost. The variable iBestLastIdxP1 is adjusted correspondingly to values unequal to 0. The two efficiencies d64BestCost and totalCost are compared, and selecting for further proceeding either quantized values, which are all set to zero or quantized values, which are not all set to zero. All values starting from the position

defined by the variable iBestLastIdxP1 are set to zero.

```
Double  d64BestCost         = 0;
Int     ui16CtxCbf          = 0;
Int     iBestLastIdxP1      = 0;
if( !pcCU->isIntra( uiAbsPartIdx ) && isLuma(compID) && pcCU->getTransformIdx( uiAbsPartIdx ) == 0 )
{
  ui16CtxCbf    = 0;
  d64BestCost   = d64BlockUncodedCost + xGetICost( m_pcEstBitsSbac->blockRootCbpBits[ ui16CtxCbf ][ 0 ] );
  d64BaseCost  += xGetICost( m_pcEstBitsSbac->blockRootCbpBits[ ui16CtxCbf ][ 1 ] );
}
else
{
  ui16CtxCbf    = pcCU->getCtxQtCbf( rTu, channelType );
  ui16CtxCbf   += getCBFContextOffset(compID);
  d64BestCost   = d64BlockUncodedCost + xGetICost( m_pcEstBitsSbac->blockCbpBits[ ui16CtxCbf ][ 0 ] );
  d64BaseCost  += xGetICost( m_pcEstBitsSbac->blockCbpBits[ ui16CtxCbf ][ 1 ] );
}
```

45. Calculation of the efficiency for setting all quantized values to zero and storing the result in the variable d64BestCost:

HEVC Reference Software (https://hevc.hhi.fraunhofer.de/).

46. Calculating the efficiency for keeping quantized values unequal to zero and storing the result in the variable totalCost:

```
Bool bFoundLast = false;
for (Int iCGScanPos = iCGLastScanPos; iCGScanPos >= 0; iCGScanPos--)
{
  UInt uiCGBlkPos = codingParameters.scanCG[ iCGScanPos ];

  d64BaseCost -= pdCostCoeffGroupSig [ iCGScanPos ];
  if (uiSigCoeffGroupFlag[ uiCGBlkPos ])
  {
    for (Int iScanPosinCG = uiCGSize-1; iScanPosinCG >= 0; iScanPosinCG--)
    {
      iScanPos = iCGScanPos*uiCGSize + iScanPosinCG;

      if (iScanPos > iLastScanPos) continue;
      UInt   uiBlkPos     = codingParameters.scan[iScanPos];

      if( piDstCoeff[ uiBlkPos ] )
      {
        UInt   uiPosY       = uiBlkPos >> uiLog2BlockWidth;
        UInt   uiPosX       = uiBlkPos - ( uiPosY << uiLog2BlockWidth );

        Double d64CostLast= codingParameters.scanType == SCAN_VER ? xGetRateLast( uiPosY, uiPosX, compID ) :
                                                                    xGetRateLast( uiPosX, uiPosY, compID );
        Double totalCost = d64BaseCost + d64CostLast - pdCostSig[ iScanPos ];
```

HEVC Reference Software (https://hevc.hhi.fraunhofer.de/).

47. Comparing the two efficiencies d64BestCost and totalCost:

```
if( totalCost < d64BestCost )
{
  iBestLastIdxP1  = iScanPos + 1;
  d64BestCost     = totalCost;
}
```

HEVC Reference Software (https://hevc.hhi.fraunhofer.de/).

48. Selecting for further proceeding either quantized values, which are all set

```
//===== clean uncoded coefficients =====
for ( Int scanPos = iBestLastIdxP1; scanPos <= iLastScanPos; scanPos++ )
{
  piDstCoeff[ codingParameters.scan[ scanPos ] ] = 0;
}
```

to zero or quantized values, which are not all set to zero:

HEVC Reference Software (https://hevc.hhi.fraunhofer.de/).

49. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the ’462 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 1 of the ‘462 Patent, namely, a method for coding a video signal using hybrid coding, comprising: reducing temporal redundancy by block based motion compensated prediction in order to establish a prediction error signal; performing quantization on samples of the prediction error signal or on coefficients resulting from a transformation of the prediction error signal into the frequency domain to obtain quantized values, representing quantized samples or quantized coefficients respectively, wherein the prediction error signal includes a plurality of subblocks each including a plurality of quantized values; calculating a first quantization efficiency for the quantized values of at least one subblock of the plurality of subblocks; setting the quantized values of the at least one subblock to all zeroes; calculating a second quantization efficiency for the at least one subblock while all of the quantized values are zeroes; selecting which of the first and second quantization efficiencies is a higher efficiency; and selecting, for further proceeding, the at least one subblock with the quantized values prior to setting the

quantized values of the at least one subblock to all zeroes if the first quantization efficiency is higher and selecting the at least one subblock with the quantized values set to zero, for further proceeding, if the second quantization efficiency is higher. Upon information and belief, Netflix uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

50. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’462 Patent, for similar reasons as explained above with respect to Claim 1 of the ’462 Patent.

51. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the HEVC (or H.265) standard.

52. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods and/or systems claimed by the ’462 Patent.

53. On information and belief, Netflix has had knowledge of the ’462 Patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Netflix knew of the ’462 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’462 Patent.

54. Upon information and belief, Netflix’s affirmative acts of making, using,

and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the ’462 Patent by practicing a method for coding a video signal using hybrid coding, comprising: reducing temporal redundancy by block based motion compensated prediction in order to establish a prediction error signal; performing quantization on samples of the prediction error signal or on coefficients resulting from a transformation of the prediction error signal into the frequency domain to obtain quantized values, representing quantized samples or quantized coefficients respectively, wherein the prediction error signal includes a plurality of subblocks each including a plurality of quantized values; calculating a first quantization efficiency for the quantized values of at least one subblock of the plurality of subblocks; setting the quantized values of the at least one subblock to all zeroes; calculating a second quantization efficiency for the at least one subblock while all of the quantized values are zeroes; selecting which of the first and second quantization efficiencies is a higher efficiency; and selecting, for further proceeding, the at least one subblock with the quantized values prior to setting the quantized values of the at least one subblock to all zeroes if the first quantization efficiency is higher and selecting the at least one subblock with the quantized values set to zero, for further proceeding, if the second quantization efficiency is higher. For example, Netflix adopted HEVC (or H.265) as its video codec in its products/services, such as its streaming services, and uses HEVC (or H.265) as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused

Instrumentalities to infringe other claims of the ’462 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the ’462 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the ’462 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’462 Patent, knowing that such use constitutes infringement of the ’462 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’462 Patent, in violation of 35 U.S.C. § 271(b).

55. Netflix has also infringed, and continues to infringe, claims of the ’462 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’462 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’462 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’462 Patent, in violation of 35 U.S.C. § 271(c).

56. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Netflix has injured Realtime and is

liable to Realtime for infringement of the ’462 Patent pursuant to 35 U.S.C. § 271.

57. As a result of Netflix’s infringement of the ’462 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix’s infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

**COUNT III**
**INFRINGEMENT OF U.S. PATENT NO. 8,934,535**

58. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

59. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ’535 Patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix’s streaming video service; Netflix’s video encoders or codecs including x264, x265 and libvpx; and Netflix’s mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, and all versions and variations thereof since the issuance of the ’535 Patent (“Accused Instrumentalities”).

60. For example, an official website from Netflix known as “The Netflix Tech Blog” states that Netflix is “introducing two new types of mobile encodes – **AVCHi-Mobile and VP9-Mobile**…All the changes combined result in better video quality for the same bitrate compared to our current streams (**AVCMain**). Many Netflix-ready devices receive streams which are encoded using the **H.264/AVC Main profile (AVCMain**). This is a widely-used video compression format, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. However,

newer formats are available that offer more sophisticated video coding tools. For our mobile bitstreams we adopt two compression formats: **H.264/AVC High profile and VP9 (profile 0)**. Similar to Main profile, **the High profile of H.264/AVC** enjoys broad decoder support. **VP9**, a royalty-free format developed by Google, is supported on the majority of Android devices, Chrome, and a growing number of consumer devices." (emphasis added). *See* https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

61. As confirmation, an article from Variety states that "Netflix has been using H.264/AVC almost exclusively" and "That's why Netflix is also encoding its downloadable videos with a different flavor of H.264/AVC, which is also known as a different profile. (For the technically inclined: Netflix's streams are encoded with **H.264/AVC Main,** whereas its downloads come in **H.264/AVC High**. The company shared more details on its tech blog this week.) This isn't quite as effective as using VP9, but still allows Netflix to shave off some bits." *See* Janko Roettgers, *Variety*, December 2, 2016, "How Netflix Delivers Better-Looking Downloads Without Eating Up All Your Phone Storage," http://variety.com/2016/digital/news/netflix-offline-downloads-codecs-vp9-1201932502/.

62. In another entry of Netflix's "The Netflix Tech Blog," a test comparing several different encoders was described: "H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC…**x265** is an open-source HEVC encoder, originally ported from the x264 codebase. Concurrent to HEVC, Google developed VP9

as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

63. The Accused Instrumentalities determine a parameter of at least a portion of a video data block. As shown below, examples of such parameters include bitrate (or max video bitrate) and resolution parameters. Different parameters correspond with different end applications. H.264 provides for multiple different ranges of such parameters, each included in the "profiles" and "levels" as defined by the H.264 standard, from the below shown paragraphs from a white paper and Wikipedia. *See* http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf at 5:

**4. H.264 profiles and levels**

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

*See* https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC:

**Levels with maximum property values**

| Level | Max decoding speed | | Max frame size | | Max video bit rate for video coding layer (VCL) kbit/s | | | Examples for high resolution @ highest frame rate (max stored frames) Toggle additional details |
|---|---|---|---|---|---|---|---|---|
| | Luma samples/s | Macroblocks/s | Luma samples | Macroblocks | Baseline, Extended and Main Profiles | High Profile | High 10 Profile | |
| 1 | 380,160 | 1,485 | 25,344 | 99 | 64 | 80 | 192 | 176×144@15.0 (4) |
| 1b | 380,160 | 1,485 | 25,344 | 99 | 128 | 160 | 384 | 176×144@15.0 (4) |
| 1.1 | 768,000 | 3,000 | 101,376 | 396 | 192 | 240 | 576 | 352×288@7.5 (2) |
| 1.2 | 1,536,000 | 6,000 | 101,376 | 396 | 384 | 480 | 1,152 | 352×288@15.2 (6) |
| 1.3 | 3,041,280 | 11,880 | 101,376 | 396 | 768 | 960 | 2,304 | 352×288@30.0 (6) |
| 2 | 3,041,280 | 11,880 | 101,376 | 396 | 2,000 | 2,500 | 6,000 | 352×288@30.0 (6) |
| 2.1 | 5,068,800 | 19,800 | 202,752 | 792 | 4,000 | 5,000 | 12,000 | 352×576@25.0 (6) |
| 2.2 | 5,184,000 | 20,250 | 414,720 | 1,620 | 4,000 | 5,000 | 12,000 | 720×576@12.5 (5) |
| 3 | 10,368,000 | 40,500 | 414,720 | 1,620 | 10,000 | 12,500 | 30,000 | 720×576@25.0 (5) |
| 3.1 | 27,648,000 | 108,000 | 921,600 | 3,600 | 14,000 | 17,500 | 42,000 | 1,280×720@30.0 (5) |
| 3.2 | 55,296,000 | 216,000 | 1,310,720 | 5,120 | 20,000 | 25,000 | 60,000 | 1,280×1,024@42.2 (4) |
| 4 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 20,000 | 25,000 | 60,000 | 2,048×1,024@30.0 (4) |
| 4.1 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 50,000 | 62,500 | 150,000 | 2,048×1,024@30.0 (4) |
| 4.2 | 133,693,440 | 522,240 | 2,228,224 | 8,704 | 50,000 | 62,500 | 150,000 | 2,048×1,080@60.0 (4) |
| 5 | 150,994,944 | 589,824 | 5,652,480 | 22,080 | 135,000 | 168,750 | 405,000 | 3,672×1,536@26.7 (5) |
| 5.1 | 251,658,240 | 983,040 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@26.7 (5) |
| 5.2 | 530,841,600 | 2,073,600 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@56.3 (5) |

64. A video data block is organized by the group of pictures (GOP) structure, which is a "collection of successive pictures within a coded video stream." *See* https://en.wikipedia.org/wiki/Group_of_pictures. A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See* https://en.wikipedia.org/wiki/Video_compression_picture_types (for descriptions of I frames, P frames and B frames); https://en.wikipedia.org/wiki/MPEG-1#D-frames (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP structure can

be controlled and used to fine-tune other parameters (e.g. bitrate, max video bitrate and resolution parameters) or even be considered as a parameter by itself.

65. Based on the bitrate and/or resolution parameter identified (e.g. bitrate, max video bitrate, resolution, GOP structure or frame type within a GOP structure), any H.264-compliant system such as the Accused Instrumentalities would determine which profile (e.g., "baseline," "extended," "main", or "high") corresponds with that parameter, then select between at least two asymmetric compressors. If baseline or extended is the corresponding profile, then the system will select a Context-Adaptive Variable Length Coding ("CAVLC") entropy encoder. If main or high is the corresponding profile, then the system will select a Context-Adaptive Binary Arithmetic Coding ("CABAC") entropy encoder. Both encoders are asymmetric compressors because it takes a longer period of time for them to compress data than to decompress data. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

| | Baseline | Extended | Main | High | High 10 |
|---|---|---|---|---|---|
| **I and P Slices** | Yes | Yes | Yes | Yes | Yes |
| **B Slices** | No | Yes | Yes | Yes | Yes |
| **SI and SP Slices** | No | Yes | No | No | No |
| **Multiple Reference Frames** | Yes | Yes | Yes | Yes | Yes |
| **In-Loop Deblocking Filter** | Yes | Yes | Yes | Yes | Yes |
| **CAVLC Entropy Coding** | Yes | Yes | Yes | Yes | Yes |
| **CABAC Entropy Coding** | No | No | Yes | Yes | Yes |
| **Flexible Macroblock Ordering (FMO)** | Yes | Yes | No | No | No |
| **Arbitrary Slice Ordering (ASO)** | Yes | Yes | No | No | No |
| **Redundant Slices (RS)** | Yes | Yes | No | No | No |
| **Data Partitioning** | No | Yes | No | No | No |
| **Interlaced Coding (PicAFF, MBAFF)** | No | Yes | Yes | Yes | Yes |
| **4:2:0 Chroma Format** | Yes | Yes | Yes | Yes | Yes |
| **Monochrome Video Format (4:0:0)** | No | No | No | Yes | Yes |
| **4:2:2 Chroma Format** | No | No | No | No | No |
| **4:4:4 Chroma Format** | No | No | No | No | No |
| **8 Bit Sample Depth** | Yes | Yes | Yes | Yes | Yes |
| **9 and 10 Bit Sample Depth** | No | No | No | No | Yes |
| **11 to 14 Bit Sample Depth** | No | No | No | No | No |
| **8×8 vs. 4×4 Transform Adaptivity** | No | No | No | Yes | Yes |
| **Quantization Scaling Matrices** | No | No | No | Yes | Yes |
| **Separate Cb and Cr QP control** | No | No | No | Yes | Yes |
| **Separate Color Plane Coding** | No | No | No | No | No |
| **Predictive Lossless Coding** | No | No | No | No | No |

*See* http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

**H.264 Entropy Coding – Comparison of Approaches**

| Characteristics | Variable Length Coding (VLC) | Context Adaptive Binary Arithmetic Coding(CABAC) |
|---|---|---|
| • Where it is used | MPEG-2, MPEG-4 ASP | H.264/MPEG-4 AVC (high efficiency option) |
| • Probability distribution | **Static** - Probabilities never change | **Adaptive** - Adjusts probabilities based on actual data |
| • Leverages correlation between symbols | **No** - Conditional probabilities ignored | **Yes** - Exploits symbol correlations by using "contexts" |
| • Non-integer code words | **No** - Low coding efficiency forhigh probability symbols | **Yes** - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency |

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy_coding_mode_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

66. The Accused Instrumentalities compress the at least the portion of the data block with the selected one or more asymmetric compressors to provide one or more compressed data blocks, which can be organized in a GOP structure (see above). After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can also be organized in a GOP structure, as discussed previously above. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

**Entropy Coding**

For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic coding (CABAC) which are complex techniques to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. The use of CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total processing power to be accomplished.

*See*

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf at 13:

Typical compression ratios to maintain excellent quality are:

- 10:1 for general images using JPEG
- 30:1 for general video using H.263 and MPEG-2
- 60:1 for general video using H.264 and WMV9

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles includes the functionality of the Baseline Profile and add improvements to the predictions algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

*See* http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf at 2:

67. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the ’535 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 15 of the ’535 Patent, namely, a method, comprising: determining a parameter of at least a portion of a data block; selecting one or more asymmetric compressors from among a plurality of compressors based upon the determined parameter or attribute; compressing the at least the portion of the data block with the selected one or more asymmetric compressors to provide one or more compressed data blocks; and storing at least a portion of the one or more compressed data blocks. Upon information and belief, Netflix

uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

68. On information and belief, the Accused Instrumentalities store at least a portion of the one or more compressed data blocks in buffers, hard disk, or other forms of memory/storage.

69. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’535 Patent, for similar reasons as explained above with respect to Claim 15 of the ’535 Patent.

70. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the H.264 standard.

71. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the systems and/or methods claimed by the ’535 Patent.

72. On information and belief, Netflix has had knowledge of the ’535 Patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Netflix knew of the ’535 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’535 Patent.

73. Upon information and belief, Netflix’s affirmative acts of making, using,

and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the ’535 Patent by practicing a method, comprising: determining a parameter of at least a portion of a data block; selecting one or more asymmetric compressors from among a plurality of compressors based upon the determined parameter or attribute; compressing the at least the portion of the data block with the selected one or more asymmetric compressors to provide one or more compressed data blocks; and storing at least a portion of the one or more compressed data blocks. For example, Netflix adopted H.264 as its video codec in its products/services, such as its streaming services, and uses H.264 as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused Instrumentalities to infringe other claims of the ’535 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the ’535 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the ‘535 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continues to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’535 Patent, knowing that such use constitutes infringement of the ’535 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’535 Patent, in violation of 35 U.S.C. § 271(b).

74. Netflix has also infringed, and continues to infringe, claims of the ‘535 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’535 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’535 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’535 Patent, in violation of 35 U.S.C. § 271(c).

75. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Netflix has injured Realtime and is liable to Realtime for infringement of the ’535 Patent pursuant to 35 U.S.C. § 271.

76. As a result of Netflix’s infringement of the ’535 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix’s infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

## COUNT IV
## INFRINGEMENT OF U.S. PATENT NO. 9,578,298

77. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

78. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ’298 Patent, and

continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix's streaming video service; Netflix's video encoders or codecs including x264, x265 and libvpx; and Netflix's mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, H.265/HEVC and all versions and variations thereof since the issuance of the '298 Patent ("Accused Instrumentalities").

79. For example, in an entry of Netflix's "The Netflix Tech Blog," a test comparing several different encoders was described: "H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC. HEVC is the successor to H.264/AVC and results reported from standardization showed about 50% bitrate savings for the same quality compared to H.264/AVC. **x265** is an open-source HEVC encoder, originally ported from the x264 codebase. Concurrent to HEVC, Google developed VP9 as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." The results of that test were as follows: "Here's a snapshot: x265 and libvpx demonstrate superior compression performance compared to x264, with bitrate savings reaching up to 50% especially at the higher resolutions. x265 outperforms libvpx for almost all resolutions and quality metrics, but the performance gap narrows (or even reverses) at 1080p." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

80. In addition, another article on the website "Arstechnica" mentions that "There's also the matter of hardware decoding support for 10-bit HEVC,

the 4K codec used by Netflix and other streaming services." *See* https://arstechnica.com/gadgets/2016/11/netflix-4k-streaming-pc-kaby-lake-cpu-windows-10-edge-browser/.

81. The Accused Instrumentalities receive the video stream which comprises at least one composite frame (FC), each composite frame containing a pair of stereoscopic digital images (L,R) according to a predetermined frame packing format. For example, the coded bitstream when it contains a stereoscopic video in one of the frame packing arrangements such as side-by-side or top-and-bottom or segmented rectangular frame packing format as defined in the following sections of the ITU-T H.265 Series H: Audiovisual and Multimedia Systems, "Infrastructure of audiovisual services – Coding of moving video" High efficiency video coding ("HEVC Spec"): D.2.16 Frame packing arrangement SEI message syntax, D.3.16 Frame packing arrangement SEI message semantics, D.2.29 Segmented rectangular frame packing arrangement SEI message syntax, D.3.29 Segmented rectangular frame packing arrangement SEI message semantics.

82. The Accused Instrumentalities generate an output video stream which can be reproduced on a visualization apparatus. For example, the output of the decoding process as defined above is a sequence of decoded pictures. *See, e.g.*, HEVC Spec at 3.39 ("3.39 decoded picture: A decoded picture is derived by decoding a coded picture"). Decoded pictures are the input of the display process. *Id.* at 3.47 ("3.47 display process: A process not specified in this Specification having, as its input, the cropped decoded pictures that are the output of the decoding process.").

83. The Accused Instrumentalities receive metadata which determine an area

occupied by one of the two images within said composite frame, said metadata indicating either a geometry of the frame packing format or a frame packing type of said composite frame. For example, the HEVC spec provides the default display window parameter to support 2D compatible decoding of stereo formats. *See*, *e.g.*, HEVC Spec ("NOTE 9 – The default display window parameters in the VUI parameters of the SPS can be used by an encoder to indicate to a decoder that does not interpret the frame packing arrangement SEI message that the default display window is an area within only one of the two constituent frames.").

84. The Accused Instrumentalities determine the area in the composite frame (FC) which is occupied by said one image of the stereoscopic pair within the composite frame based on said metadata. For example, the default display window parameter has been defined to support this application. The parameter syntax is defined in clause E.2.1 VUI parameters syntax, the semantics thereof being described in clause E.3.1 VUI parameters semantics. The usage of the Default Display Window for signaling the 2D single view in a stereoscopic frame packing format is illustrated in Note 9 of clause D.3.16 and Note 3 in Clause D.3.29 cited above.

85. The Accused Instrumentalities decode only that part of the composite frame which contains said one image to be displayed. For example, tiles are intended to support independent decoding of different picture regions. Clause 7.4.3.2.1 cited above illustrates the process to convert CTB picture scan in CTB tile scan to enable independent decoding of the tile. *See also* HEVC Spec:

> **row_height_minus1**[ i ] plus 1 specifies the height of the i-th tile row in units of coding tree blocks.
>
> The following variables are derived by invoking the coding tree block raster and tile scanning conversion process as specified in clause 6.5.1:
>
> – The list CtbAddrRsToTs[ ctbAddrRs ] for ctbAddrRs ranging from 0 to PicSizeInCtbsY − 1, inclusive, specifying the conversion from a CTB address in the CTB raster scan of a picture to a CTB address in the tile scan,
> – the list CtbAddrTsToRs[ ctbAddrTs ] for ctbAddrTs ranging from 0 to PicSizeInCtbsY − 1, inclusive, specifying the conversion from a CTB address in the tile scan to a CTB address in the CTB raster scan of a picture,
> – the list TileId[ ctbAddrTs ] for ctbAddrTs ranging from 0 to PicSizeInCtbsY − 1, inclusive, specifying the conversion from a CTB address in tile scan to a tile ID,
> – the list ColumnWidthInLumaSamples[ i ] for i ranging from 0 to num_tile_columns_minus1, inclusive, specifying the width of the i-th tile column in units of luma samples,
> – the list RowHeightInLumaSamples[ j ] for j ranging from 0 to num_tile_rows_minus1, inclusive, specifying the height of the j-th tile row in units of luma samples.
>
> The values of ColumnWidthInLumaSamples[ i ] for i ranging from 0 to num_tile_columns_minus1, inclusive, and RowHeightInLumaSamples[ j ] for j ranging from 0 to num_tile_rows_minus1, inclusive, shall all be greater than 0.
>
> The array MinTbAddrZs with elements MinTbAddrZs[ x ][ y ] for x ranging from 0 to ( PicWidthInCtbsY << ( CtbLog2SizeY − MinTbLog2SizeY ) ) − 1, inclusive, and y ranging from 0 to ( PicHeightInCtbsY << ( CtbLog2SizeY − MinTbLog2SizeY ) ) − 1, inclusive, specifying the conversion from a location ( x, y ) in units of minimum transform blocks to a transform block address in z-scan order, is derived by invoking the z-scan order array initialization process as specified in clause 6.5.2.

86. The Accused Instrumentalities generate an output frame containing said extracted image. For example, there is an output of the tile decoding process. *See, e.g.*, HEVC Spec at 8.1.1 ("8.1.1 General…Input to this process is a bitstream. Output of this process is a list of decoded pictures.").

87. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the '298 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 1 of the '298 Patent, namely, a method for processing a video stream of digital images, the method comprising the steps of: receiving the video stream which comprises at least one composite frame (FC), each composite frame containing a pair of stereoscopic digital images (L,R) according to a predetermined frame packing format; generating an output video stream which can be reproduced on a visualization apparatus, receiving metadata which determine an area occupied by one of the two images within said composite frame (FC), said metadata indicating either a geometry of the frame packing format or a frame

packing type of said composite frame (FC); determining the area in the composite frame (FC) which is occupied by said one image of the stereoscopic pair within the composite frame based on said metadata; decoding only that part of the composite frame (FC) which contains said one image to be displayed, and generating an output frame containing said decoded image. Upon information and belief, Netflix uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

88. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’298 Patent, for similar reasons as explained above with respect to Claim 1 of the ’298 Patent.

89. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the HEVC (or H.265) standard.

90. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods claimed by the ’298 Patent.

91. On information and belief, Netflix has had knowledge of the ’298 Patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Netflix knew of the ’298 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’298 Patent.

92. Upon information and belief, Netflix's affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the '298 by practicing a method for processing a video stream of digital images, the method comprising the steps of: receiving the video stream which comprises at least one composite frame (FC), each composite frame containing a pair of stereoscopic digital images (L,R) according to a predetermined frame packing format; generating an output video stream which can be reproduced on a visualization apparatus, receiving metadata which determine an area occupied by one of the two images within said composite frame (FC), said metadata indicating either a geometry of the frame packing format or a frame packing type of said composite frame (FC); determining the area in the composite frame (FC) which is occupied by said one image of the stereoscopic pair within the composite frame based on said metadata; decoding only that part of the composite frame (FC) which contains said one image to be displayed, and generating an output frame containing said decoded image. For example, Netflix adopted HEVC (or H.265) as its video codec in its products/services, such as its streaming services, and uses HEVC (or H.265) as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused Instrumentalities to infringe other claims of the '298 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the '298 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the

knowledge of the ’298 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’298 Patent, knowing that such use constitutes infringement of the ’298 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’298 Patent, in violation of 35 U.S.C. § 271(b).

93. Netflix has also infringed, and continues to infringe, claims of the ’298 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’298 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’298 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’298 Patent, in violation of 35 U.S.C. § 271(c).

94. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Netflix has injured Realtime and is liable to Realtime for infringement of the ’298 Patent pursuant to 35 U.S.C. § 271.

95. As a result of Netflix’s infringement of the ’298 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix’s

infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

**COUNT V**
**INFRINGEMENT OF U.S. PATENT NO. 9,762,907**

77. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

78. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ’907 Patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix’s streaming video service; Netflix’s video encoders or codecs including x264, x265 and libvpx; and Netflix’s mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, and all versions and variations thereof since the issuance of the ’907 Patent (“Accused Instrumentalities”).

79. For example, an official website from Netflix known as “The Netflix Tech Blog” states that Netflix is “introducing two new types of mobile encodes – **AVCHi-Mobile and VP9-Mobile**…All the changes combined result in better video quality for the same bitrate compared to our current streams (**AVCMain**). Many Netflix-ready devices receive streams which are encoded using the **H.264/AVC Main profile (AVCMain**). This is a widely-used video compression format, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. However, newer formats are available that offer more sophisticated video coding tools. For our mobile bitstreams we adopt two compression formats: **H.264/AVC High profile and VP9 (profile 0)**. Similar to Main profile, **the High profile of H.264/AVC** enjoys broad

decoder support. **VP9**, a royalty-free format developed by Google, is supported on the majority of Android devices, Chrome, and a growing number of consumer devices." (emphasis added). *See* https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

80. As confirmation, an article from Variety states that "Netflix has been using H.264/AVC almost exclusively" and "That's why Netflix is also encoding its downloadable videos with a different flavor of H.264/AVC, which is also known as a different profile. (For the technically inclined: Netflix's streams are encoded with **H.264/AVC Main,** whereas its downloads come in **H.264/AVC High**. The company shared more details on its tech blog this week.) This isn't quite as effective as using VP9, but still allows Netflix to shave off some bits." *See* Janko Roettgers, *Variety*, December 2, 2016, "How Netflix Delivers Better-Looking Downloads Without Eating Up All Your Phone Storage," http://variety.com/2016/digital/news/netflix-offline-downloads-codecs-vp9-1201932502/.

81. In another entry of Netflix's "The Netflix Tech Blog," a test comparing several different encoders was described: "H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC…**x265** is an open-source HEVC encoder, originally ported from the x264 codebase. Concurrent to HEVC, Google developed VP9 as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

82. The Accused Instrumentalities determine a parameter of at least a portion of a video data block. As shown below, examples of such parameters include bitrate (or max video bitrate) and resolution parameters. Different parameters correspond with different end applications. H.264 provides for multiple different ranges of such parameters, each included in the "profiles" and "levels" as defined by the H.264 standard, from the below shown paragraphs from a white paper and Wikipedia. *See* http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf at 5:

**4. H.264 profiles and levels**

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

*See* https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC:

Levels with maximum property values

| Level | Max decoding speed | | Max frame size | | Max video bit rate for video coding layer (VCL) kbit/s | | | Examples for high resolution @ highest frame rate (max stored frames) Toggle additional details |
|---|---|---|---|---|---|---|---|---|
| | Luma samples/s | Macroblocks/s | Luma samples | Macroblocks | Baseline, Extended and Main Profiles | High Profile | High 10 Profile | |
| 1 | 380,160 | 1,485 | 25,344 | 99 | 64 | 80 | 192 | 176×144@15.0 (4) |
| 1b | 380,160 | 1,485 | 25,344 | 99 | 128 | 160 | 384 | 176×144@15.0 (4) |
| 1.1 | 768,000 | 3,000 | 101,376 | 396 | 192 | 240 | 576 | 352×288@7.5 (2) |
| 1.2 | 1,536,000 | 6,000 | 101,376 | 396 | 384 | 480 | 1,152 | 352×288@15.2 (6) |
| 1.3 | 3,041,280 | 11,880 | 101,376 | 396 | 768 | 960 | 2,304 | 352×288@30.0 (6) |
| 2 | 3,041,280 | 11,880 | 101,376 | 396 | 2,000 | 2,500 | 6,000 | 352×288@30.0 (6) |
| 2.1 | 5,068,800 | 19,800 | 202,752 | 792 | 4,000 | 5,000 | 12,000 | 352×576@25.0 (6) |
| 2.2 | 5,184,000 | 20,250 | 414,720 | 1,620 | 4,000 | 5,000 | 12,000 | 720×576@12.5 (5) |
| 3 | 10,368,000 | 40,500 | 414,720 | 1,620 | 10,000 | 12,500 | 30,000 | 720×576@25.0 (5) |
| 3.1 | 27,648,000 | 108,000 | 921,600 | 3,600 | 14,000 | 17,500 | 42,000 | 1,280×720@30.0 (5) |
| 3.2 | 55,296,000 | 216,000 | 1,310,720 | 5,120 | 20,000 | 25,000 | 60,000 | 1,280×1,024@42.2 (4) |
| 4 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 20,000 | 25,000 | 60,000 | 2,048×1,024@30.0 (4) |
| 4.1 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 50,000 | 62,500 | 150,000 | 2,048×1,024@30.0 (4) |
| 4.2 | 133,693,440 | 522,240 | 2,228,224 | 8,704 | 50,000 | 62,500 | 150,000 | 2,048×1,080@60.0 (4) |
| 5 | 150,994,944 | 589,824 | 5,652,480 | 22,080 | 135,000 | 168,750 | 405,000 | 3,672×1,536@26.7 (5) |
| 5.1 | 251,658,240 | 983,040 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@26.7 (5) |
| 5.2 | 530,841,600 | 2,073,600 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@56.3 (5) |

83. A video data block is organized by the group of pictures (GOP) structure, which is a "collection of successive pictures within a coded video stream." *See* https://en.wikipedia.org/wiki/Group_of_pictures. A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See* https://en.wikipedia.org/wiki/Video_compression_picture_types (for descriptions of I frames, P frames and B frames); https://en.wikipedia.org/wiki/MPEG-1#D-frames (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP structure can

be controlled and used to fine-tune other parameters (e.g. bitrate, max video bitrate and resolution parameters) or even be considered as a parameter by itself.

84. Based on the bitrate and/or resolution parameter identified (e.g. bitrate, max video bitrate, resolution, GOP structure or frame type within a GOP structure), any H.264-compliant system such as the Accused Instrumentalities would determine which profile (e.g., "baseline," "extended," "main", or "high") corresponds with that parameter, then select between at least two asymmetric compressors. If baseline or extended is the corresponding profile, then the system will select a Context-Adaptive Variable Length Coding ("CAVLC") entropy encoder. If main or high is the corresponding profile, then the system will select a Context-Adaptive Binary Arithmetic Coding ("CABAC") entropy encoder. Both encoders are asymmetric compressors because it takes a longer period of time for them to compress data than to decompress data. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

| | Baseline | Extended | Main | High | High 10 |
|---|---|---|---|---|---|
| I and P Slices | Yes | Yes | Yes | Yes | Yes |
| B Slices | No | Yes | Yes | Yes | Yes |
| SI and SP Slices | No | Yes | No | No | No |
| Multiple Reference Frames | Yes | Yes | Yes | Yes | Yes |
| In-Loop Deblocking Filter | Yes | Yes | Yes | Yes | Yes |
| CAVLC Entropy Coding | Yes | Yes | Yes | Yes | Yes |
| CABAC Entropy Coding | No | No | Yes | Yes | Yes |
| Flexible Macroblock Ordering (FMO) | Yes | Yes | No | No | No |
| Arbitrary Slice Ordering (ASO) | Yes | Yes | No | No | No |
| Redundant Slices (RS) | Yes | Yes | No | No | No |
| Data Partitioning | No | Yes | No | No | No |
| Interlaced Coding (PicAFF, MBAFF) | No | Yes | Yes | Yes | Yes |
| 4:2:0 Chroma Format | Yes | Yes | Yes | Yes | Yes |
| Monochrome Video Format (4:0:0) | No | No | No | Yes | Yes |
| 4:2:2 Chroma Format | No | No | No | No | No |
| 4:4:4 Chroma Format | No | No | No | No | No |
| 8 Bit Sample Depth | Yes | Yes | Yes | Yes | Yes |
| 9 and 10 Bit Sample Depth | No | No | No | No | Yes |
| 11 to 14 Bit Sample Depth | No | No | No | No | No |
| 8×8 vs. 4×4 Transform Adaptivity | No | No | No | Yes | Yes |
| Quantization Scaling Matrices | No | No | No | Yes | Yes |
| Separate Cb and Cr QP control | No | No | No | Yes | Yes |
| Separate Color Plane Coding | No | No | No | No | No |
| Predictive Lossless Coding | No | No | No | No | No |

*See* http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

**H.264 Entropy Coding – Comparison of Approaches**

| Characteristics | Variable Length Coding (VLC) | Context Adaptive Binary Arithmetic Coding(CABAC) |
|---|---|---|
| • Where it is used | MPEG-2, MPEG-4 ASP | H.264/MPEG-4 AVC (high efficiency option) |
| • Probability distribution | **Static** - Probabilities never change | **Adaptive** - Adjusts probabilities based on actual data |
| • Leverages correlation between symbols | **No** - Conditional probabilities ignored | **Yes** - Exploits symbol correlations by using "contexts" |
| • Non-integer code words | **No** - Low coding efficiency forhigh probability symbols | **Yes** - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency |

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy_coding_mode_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

85. The Accused Instrumentalities compress the at least the portion of the data

block with the selected one or more asymmetric compressors to provide one or more compressed data blocks, which can be organized in a GOP structure (see above). After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can also be organized in a GOP structure, as discussed previously above. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

> **Entropy Coding**
> For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic coding (CABAC) which are complex techniques to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. The use of CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total processing power to be accomplished.

*See*

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf at 13:

> Typical compression ratios to maintain excellent quality are:
> - 10:1 for general images using JPEG
> - 30:1 for general video using H.263 and MPEG-2
> - 60:1 for general video using H.264 and WMV9

Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.

The Extended and Main Profiles includes the functionality of the Baseline Profile and add improvements to the predictions algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.

The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.

The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

*See* http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf at 2:

86. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the ’907 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 1 of the ’907 Patent, namely, a system comprising: one or more different asymmetric data compression algorithms, wherein each algorithm of the one or more different asymmetric data compression algorithms utilizes one or more asymmetric data compression routines of a plurality of different asymmetric data compression routines, wherein a first asymmetric data compression routine of the plurality of different asymmetric data compression routines is configured to produce compressed data with a higher data rate

for a given data throughput than a second asymmetric data compression routine of the plurality of different asymmetric data compression routines; and a processor configured: to analyze one or more data parameters from one or more data blocks containing video data, wherein at least one data parameter relates to an expected or anticipated throughput of a communications channel; and to select two or more different data compression routines from among a plurality of different data compression routines based upon, at least in part, the one or more data parameters relating to the expected or anticipated throughput of the communications channel. Upon information and belief, Netflix uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

87. On information and belief, the Accused Instrumentalities store at least a portion of the one or more compressed data blocks in buffers, hard disk, or other forms of memory/storage.

88. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’907 Patent, for similar reasons as explained above with respect to Claim 1 of the ’907 Patent.

89. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the H.264 standard.

90. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the systems and/or methods

claimed by the ’907 Patent.

91. On information and belief, Netflix has had knowledge of the ‘907 Patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Netflix knew of the ’907 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’907 Patent.

92. Upon information and belief, Netflix’s affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the ’907 Patent by practicing a system comprising: one or more different asymmetric data compression algorithms, wherein each algorithm of the one or more different asymmetric data compression algorithms utilizes one or more asymmetric data compression routines of a plurality of different asymmetric data compression routines, wherein a first asymmetric data compression routine of the plurality of different asymmetric data compression routines is configured to produce compressed data with a higher data rate for a given data throughput than a second asymmetric data compression routine of the plurality of different asymmetric data compression routines; and a processor configured: to analyze one or more data parameters from one or more data blocks containing video data, wherein at least one data parameter relates to an expected or anticipated throughput of a communications channel;

and to select two or more different data compression routines from among a plurality of different data compression routines based upon, at least in part, the one or more data parameters relating to the expected or anticipated throughput of the communications channel. For example, Netflix adopted H.264 as its video codec in its products/services, such as its streaming services, and uses H.264 as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused Instrumentalities to infringe other claims of the ’907 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the ’907 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the ’907 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continues to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’907 Patent, knowing that such use constitutes infringement of the ’907 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’907 Patent, in violation of 35 U.S.C. § 271(b).

93. Netflix has also infringed, and continues to infringe, claims of the ’907 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’907 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’907 Patent, not a staple article, and not

a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’907 Patent, in violation of 35 U.S.C. § 271(c).

94. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities’ compression features, Netflix has injured Realtime and is liable to Realtime for infringement of the ’907 Patent pursuant to 35 U.S.C. § 271.

95. As a result of Netflix’s infringement of the ’907 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix’s infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

**COUNT VI**
**INFRINGEMENT OF U.S. PATENT NO. 9,769,477**

96. Plaintiff re-alleges and incorporates by reference the foregoing paragraphs, as if fully set forth herein.

97. On information and belief, Netflix has made, used, offered for sale, sold and/or imported into the United States Netflix products that infringe the ’477 Patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Netflix’s streaming video service; Netflix’s video encoders or codecs including x264, x265 and libvpx; and Netflix’s mobile encoders, encoders or codecs including AVCMain (H.264/AVC Main), H.264/AVC High, VP9, AVCHi-Mobile and VP9-Mobile, and all versions and variations thereof since the issuance of the ’477 Patent (“Accused Instrumentalities”).

98. For example, an official website from Netflix known as “The Netflix Tech

Blog" states that Netflix is "introducing two new types of mobile encodes – **AVCHi-Mobile and VP9-Mobile**…All the changes combined result in better video quality for the same bitrate compared to our current streams (**AVCMain**). Many Netflix-ready devices receive streams which are encoded using the **H.264/AVC Main profile (AVCMain**). This is a widely-used video compression format, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. However, newer formats are available that offer more sophisticated video coding tools. For our mobile bitstreams we adopt two compression formats: **H.264/AVC High profile and VP9 (profile 0)**. Similar to Main profile, **the High profile of H.264/AVC** enjoys broad decoder support. **VP9**, a royalty-free format developed by Google, is supported on the majority of Android devices, Chrome, and a growing number of consumer devices." (emphasis added). *See* https://medium.com/netflix-techblog/more-efficient-mobile-encodes-for-netflix-downloads-625d7b082909.

99. As confirmation, an article from Variety states that "Netflix has been using H.264/AVC almost exclusively" and "That's why Netflix is also encoding its downloadable videos with a different flavor of H.264/AVC, which is also known as a different profile. (For the technically inclined: Netflix's streams are encoded with **H.264/AVC Main,** whereas its downloads come in **H.264/AVC High**. The company shared more details on its tech blog this week.) This isn't quite as effective as using VP9, but still allows Netflix to shave off some bits." *See* Janko Roettgers, *Variety*, December 2, 2016, "How Netflix Delivers Better-Looking Downloads Without Eating Up All Your Phone Storage," http://variety.com/2016/digital/news/netflix-offline-downloads-codecs-vp9-1201932502/.

100. In another entry of Netflix's "The Netflix Tech Blog," a test comparing several different encoders was described: "H.264/AVC is a very widely-used video compression standard on the Internet, with ubiquitous decoder support on web browsers, TVs, mobile devices, and other consumer devices. **x264** is the most established open-source software encoder for H.264/AVC…**x265** is an open-source HEVC encoder, originally ported from the x264 codebase. Concurrent to HEVC, Google developed VP9 as a royalty-free video compression format and released **libvpx** as an open-source software library for encoding VP9." *See* https://medium.com/netflix-techblog/a-large-scale-comparison-of-x264-x265-and-libvpx-a-sneak-peek-2e81e88f8b0f.

101. The Accused Instrumentalities determine a parameter of at least a portion of a video data block. As shown below, examples of such parameters include bitrate (or max video bitrate) and resolution parameters. Different parameters correspond with different end applications. H.264 provides for multiple different ranges of such parameters, each included in the "profiles" and "levels" as defined by the H.264 standard, from the below shown paragraphs from a white paper and Wikipedia. *See* http://www.axis.com/files/whitepaper/wp_h264_31669_en_0803_lo.pdf at 5:

### 4. H.264 profiles and levels

The joint group involved in defining H.264 focused on creating a simple and clean solution, limiting options and features to a minimum. An important aspect of the standard, as with other video standards, is providing the capabilities in profiles (sets of algorithmic features) and levels (performance classes) that optimally support popular productions and common formats.

H.264 has seven profiles, each targeting a specific class of applications. Each profile defines what feature set the encoder may use and limits the decoder implementation complexity.

Network cameras and video encoders will most likely use a profile called the baseline profile, which is intended primarily for applications with limited computing resources. The baseline profile is the most suitable given the available performance in a real-time encoder that is embedded in a network video product. The profile also enables low latency, which is an important requirement of surveillance video and also particularly important in enabling real-time, pan/tilt/zoom (PTZ) control in PTZ network cameras.

H.264 has 11 levels or degree of capability to limit performance, bandwidth and memory requirements. Each level defines the bit rate and the encoding rate in macroblock per second for resolutions ranging from QCIF to HDTV and beyond. The higher the resolution, the higher the level required.

*See* https://en.wikipedia.org/wiki/H.264/MPEG-4_AVC:

**Levels with maximum property values**

| Level | Max decoding speed | | Max frame size | | Max video bit rate for video coding layer (VCL) kbit/s | | | Examples for high resolution @ highest frame rate (max stored frames) Toggle additional details |
|---|---|---|---|---|---|---|---|---|
| | Luma samples/s | Macroblocks/s | Luma samples | Macroblocks | Baseline, Extended and Main Profiles | High Profile | High 10 Profile | |
| 1 | 380,160 | 1,485 | 25,344 | 99 | 64 | 80 | 192 | 176×144@15.0 (4) |
| 1b | 380,160 | 1,485 | 25,344 | 99 | 128 | 160 | 384 | 176×144@15.0 (4) |
| 1.1 | 768,000 | 3,000 | 101,376 | 396 | 192 | 240 | 576 | 352×288@7.5 (2) |
| 1.2 | 1,536,000 | 6,000 | 101,376 | 396 | 384 | 480 | 1,152 | 352×288@15.2 (6) |
| 1.3 | 3,041,280 | 11,880 | 101,376 | 396 | 768 | 960 | 2,304 | 352×288@30.0 (6) |
| 2 | 3,041,280 | 11,880 | 101,376 | 396 | 2,000 | 2,500 | 6,000 | 352×288@30.0 (6) |
| 2.1 | 5,068,800 | 19,800 | 202,752 | 792 | 4,000 | 5,000 | 12,000 | 352×576@25.0 (6) |
| 2.2 | 5,184,000 | 20,250 | 414,720 | 1,620 | 4,000 | 5,000 | 12,000 | 720×576@12.5 (5) |
| 3 | 10,368,000 | 40,500 | 414,720 | 1,620 | 10,000 | 12,500 | 30,000 | 720×576@25.0 (5) |
| 3.1 | 27,648,000 | 108,000 | 921,600 | 3,600 | 14,000 | 17,500 | 42,000 | 1,280×720@30.0 (5) |
| 3.2 | 55,296,000 | 216,000 | 1,310,720 | 5,120 | 20,000 | 25,000 | 60,000 | 1,280×1,024@42.2 (4) |
| 4 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 20,000 | 25,000 | 60,000 | 2,048×1,024@30.0 (4) |
| 4.1 | 62,914,560 | 245,760 | 2,097,152 | 8,192 | 50,000 | 62,500 | 150,000 | 2,048×1,024@30.0 (4) |
| 4.2 | 133,693,440 | 522,240 | 2,228,224 | 8,704 | 50,000 | 62,500 | 150,000 | 2,048×1,080@60.0 (4) |
| 5 | 150,994,944 | 589,824 | 5,652,480 | 22,080 | 135,000 | 168,750 | 405,000 | 3,672×1,536@26.7 (5) |
| 5.1 | 251,658,240 | 983,040 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@26.7 (5) |
| 5.2 | 530,841,600 | 2,073,600 | 9,437,184 | 36,864 | 240,000 | 300,000 | 720,000 | 4,096×2,304@56.3 (5) |

102. A video data block is organized by the group of pictures (GOP) structure, which is a "collection of successive pictures within a coded video stream." *See* https://en.wikipedia.org/wiki/Group_of_pictures. A GOP structure can contain intra coded pictures (I picture or I frame), predictive coded pictures (P picture or P frame), bipredictive coded pictures (B picture or B frame) and direct coded pictures (D picture or D frames, or DC direct coded pictures which are used only in MPEG-1 video). *See* https://en.wikipedia.org/wiki/Video_compression_picture_types (for descriptions of I frames, P frames and B frames); https://en.wikipedia.org/wiki/MPEG-1#D-frames (for descriptions of D frames). Thus, at least a portion of a video data block would also make up a GOP structure and could also contain I frames, P frames, B frames and/or D frames. The GOP structure also reflects the size of a video data block, and the GOP structure can be controlled and used to fine-tune other parameters (e.g. bitrate, max video bitrate and resolution parameters) or even be considered as a parameter by itself.

103. Based on the bitrate and/or resolution parameter identified (e.g. bitrate, max video bitrate, resolution, GOP structure or frame type within a GOP structure), any H.264-compliant system such as the Accused Instrumentalities would determine which profile (e.g., "baseline," "extended," "main", or "high") corresponds with that parameter, then select between at least two asymmetric compressors. If baseline or extended is the corresponding profile, then the system will select a Context-Adaptive Variable Length Coding ("CAVLC") entropy encoder. If main or high is the corresponding profile, then the system will select a Context-Adaptive Binary Arithmetic Coding ("CABAC") entropy encoder. Both encoders are asymmetric compressors because it takes a longer period of

time for them to compress data than to decompress data. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

| | Baseline | Extended | Main | High | High 10 |
|---|---|---|---|---|---|
| I and P Slices | Yes | Yes | Yes | Yes | Yes |
| B Slices | No | Yes | Yes | Yes | Yes |
| SI and SP Slices | No | Yes | No | No | No |
| Multiple Reference Frames | Yes | Yes | Yes | Yes | Yes |
| In-Loop Deblocking Filter | Yes | Yes | Yes | Yes | Yes |
| CAVLC Entropy Coding | Yes | Yes | Yes | Yes | Yes |
| CABAC Entropy Coding | No | No | Yes | Yes | Yes |
| Flexible Macroblock Ordering (FMO) | Yes | Yes | No | No | No |
| Arbitrary Slice Ordering (ASO) | Yes | Yes | No | No | No |
| Redundant Slices (RS) | Yes | Yes | No | No | No |
| Data Partitioning | No | Yes | No | No | No |
| Interlaced Coding (PicAFF, MBAFF) | No | Yes | Yes | Yes | Yes |
| 4:2:0 Chroma Format | Yes | Yes | Yes | Yes | Yes |
| Monochrome Video Format (4:0:0) | No | No | No | Yes | Yes |
| 4:2:2 Chroma Format | No | No | No | No | No |
| 4:4:4 Chroma Format | No | No | No | No | No |
| 8 Bit Sample Depth | Yes | Yes | Yes | Yes | Yes |
| 9 and 10 Bit Sample Depth | No | No | No | No | Yes |
| 11 to 14 Bit Sample Depth | No | No | No | No | No |
| 8×8 vs. 4×4 Transform Adaptivity | No | No | No | Yes | Yes |
| Quantization Scaling Matrices | No | No | No | Yes | Yes |
| Separate Cb and Cr QP control | No | No | No | Yes | Yes |
| Separate Color Plane Coding | No | No | No | No | No |
| Predictive Lossless Coding | No | No | No | No | No |

*See* http://web.cs.ucla.edu/classes/fall03/cs218/paper/H.264_MPEG4_Tutorial.pdf at 7:

The following table summarizes the two major types of entropy coding: Variable Length Coding (VLC) and Context Adaptive Binary Arithmetic Coding (CABAC). CABAC offers superior coding efficiency over VLC by adapting to the changing probability distribution of symbols, by exploiting correlation between symbols, and by adaptively exploiting bit correlations using arithmetic coding. H.264 also supports Context Adaptive Variable Length Coding (CAVLC) which offers superior entropy coding over VLC without the full cost of CABAC.

**H.264 Entropy Coding – Comparison of Approaches**

| Characteristics | Variable Length Coding (VLC) | Context Adaptive Binary Arithmetic Coding(CABAC) |
|---|---|---|
| • Where it is used | MPEG-2, MPEG-4 ASP | H.264/MPEG-4 AVC (high efficiency option) |
| • Probability distribution | **Static** - Probabilities never change | **Adaptive** - Adjusts probabilities based on actual data |
| • Leverages correlation between symbols | **No** - Conditional probabilities ignored | **Yes** - Exploits symbol correlations by using "contexts" |
| • Non-integer code words | **No** - Low coding efficiency forhigh probability symbols | **Yes** - Exploits "arithmetic coding" which generates non-integer code words for higher efficiency |

Moreover, the H.264 Standard requires a bit-flag descriptor, which is set to determine the correct decoder for the corresponding encoder. As shown below, if the flag = 0, then CAVLC must have been selected as the encoder; if the flag = 1, then CABAC must have been selected as the encoder. *See* https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-201304-S!!PDF-E&type=items (Rec. ITU-T H.264 (04/2013)) at 80:

**entropy_coding_mode_flag** selects the entropy decoding method to be applied for the syntax elements for which two descriptors appear in the syntax tables as follows:

– If entropy_coding_mode_flag is equal to 0, the method specified by the left descriptor in the syntax table is applied (Exp-Golomb coded, see clause 9.1 or CAVLC, see clause 9.2).

– Otherwise (entropy_coding_mode_flag is equal to 1), the method specified by the right descriptor in the syntax table is applied (CABAC, see clause 9.3).

104. The Accused Instrumentalities compress the at least the portion of the data

block with the selected one or more asymmetric compressors to provide one or more compressed data blocks, which can be organized in a GOP structure (see above). After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can also be organized in a GOP structure, as discussed previously above. *See* https://sonnati.wordpress.com/2007/10/29/how-h-264-works-part-ii/:

**Entropy Coding**

For entropy coding, H.264 may use an enhanced VLC, a more complex context-adaptive variable-length coding (CAVLC) or an ever more complex Context-adaptive binary-arithmetic coding (CABAC) which are complex techniques to losslessly compress syntax elements in the video stream knowing the probabilities of syntax elements in a given context. The use of CABAC can improve the compression of around 5-7%. CABAC may requires a 30-40% of total processing power to be accomplished.

*See*

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.602.1581&rep=rep1&type=pdf at 13:

Typical compression ratios to maintain excellent quality are:

- 10:1 for general images using JPEG
- 30:1 for general video using H.263 and MPEG-2
- 60:1 for general video using H.264 and WMV9

> Most visual communication systems today use Baseline Profile. Baseline is the simplest H.264 profile and defines, for example, zigzag scanning of the picture and using 4:2:0 (YUV video formats) chrominance sampling. In Baseline Profile, the picture is split in blocks consisting of 4x4 pixels, and each block is processed separately. Another important element of the Baseline Profile is the use of Universal Variable Length Coding (UVLC) and Context Adaptive Variable Length Coding (CAVLC) entropy coding techniques.
>
> The Extended and Main Profiles includes the functionality of the Baseline Profile and add improvements to the predictions algorithms. Since transmitting every single frame (think 30 frames per second for good quality video) is not feasible if you are trying to reduce the bit rate 1000-2000 times, temporal and motion prediction are heavily used in H.264, and allow transmitting only the difference between one frame and the previous frames. The result is spectacular efficiency gain, especially for scenes with little change and motion.
>
> The High Profile is the most powerful profile in H.264, and it allows most efficient coding of video. For example, large coding gain achieved through the use of Context Adaptive Binary Arithmetic Coding (CABAC) encoding which is more efficient than the UVLC/CAVLC used in Baseline Profile.
>
> The High Profile also uses adaptive transform that decides on the fly if 4x4 or 8x8-pixel blocks should be used. For example, 4x4 blocks are used for the parts of the picture that are dense with detail, while parts that have little detail are transformed using 8x8 blocks.

*See* http://www.ijera.com/papers/Vol3_issue4/BM34399403.pdf at 2:

105. Therefore, from at least the above, Netflix has directly infringed and continues to infringe the ’477 Patent, for example, through its own use and testing of the Accused Instrumentalities, which when used, practices the system claimed by Claim 1 of the ’477 Patent, namely, a plurality of different asymmetric data compression encoders, wherein each asymmetric data compression encoder of the plurality of different asymmetric data compression encoders is configured to utilize one or more data compression algorithms, and wherein a first asymmetric data compression encoder of the plurality of different asymmetric data compression encoders is configured to compress data blocks containing video or image data at a higher data compression rate than a

second asymmetric data compression encoder of the plurality of different asymmetric data compression encoders; and one or more processors configured to: determine one or more data parameters, at least one of the determined one or more data parameters relating to a throughput of a communications channel measured in bits per second; and select one or more asymmetric data compression encoders from among the plurality of different asymmetric data compression encoders based upon, at least in part, the determined one or more data parameters. Upon information and belief, Netflix uses the Accused Instrumentalities to practice infringing methods for its own internal non-testing business purposes, while testing the Accused Instrumentalities, and while providing technical support and repair services for the Accused Instrumentalities to their customers.

106. On information and belief, the Accused Instrumentalities store at least a portion of the one or more compressed data blocks in buffers, hard disk, or other forms of memory/storage.

107. On information and belief, Netflix also directly infringes and continues to infringe other claims of the ’477 Patent, for similar reasons as explained above with respect to Claim 1 of the ’477 Patent.

108. On information and belief, all of the Accused Instrumentalities perform the claimed methods in substantially the same way, e.g., in the manner specified in the H.264 standard.

109. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the systems and/or methods claimed by the ’477 Patent.

110. On information and belief, Netflix has had knowledge of the ’477 Patent

since at least the filing of this Complaint or shortly thereafter, and on information and belief, Netflix knew of the ’477 Patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Netflix will have known and intended (since receiving such notice) that its continued actions would actively induce and contribute to the infringement of the claims of the ’477 Patent.

111. Upon information and belief, Netflix’s affirmative acts of making, using, and selling the Accused Instrumentalities, and providing implementation services and technical support to users of the Accused Instrumentalities, including, e.g., through training, demonstrations, brochures, installation and user guides, have induced and continue to induce users of the Accused Instrumentalities to use them in their normal and customary way to infringe the ’477 Patent by practicing a plurality of different asymmetric data compression encoders, wherein each asymmetric data compression encoder of the plurality of different asymmetric data compression encoders is configured to utilize one or more data compression algorithms, and wherein a first asymmetric data compression encoder of the plurality of different asymmetric data compression encoders is configured to compress data blocks containing video or image data at a higher data compression rate than a second asymmetric data compression encoder of the plurality of different asymmetric data compression encoders; and one or more processors configured to: determine one or more data parameters, at least one of the determined one or more data parameters relating to a throughput of a communications channel measured in bits per second; and select one or more asymmetric data compression encoders from among the plurality of different asymmetric data compression encoders based upon, at least in part, the determined one or more data parameters. For example, Netflix adopted H.264 as

its video codec in its products/services, such as its streaming services, and uses H.264 as an encoder, encode or codec. For similar reasons, Netflix also induces its customers to use the Accused Instrumentalities to infringe other claims of the ’477 Patent. Netflix specifically intended and was aware that these normal and customary activities would infringe the ’477 Patent. Netflix performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the ’477 Patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. On information and belief, Netflix engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Netflix has induced and continues to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the ’477 Patent, knowing that such use constitutes infringement of the ’477 Patent. Accordingly, Netflix has been, and currently is, inducing infringement of the ’477 Patent, in violation of 35 U.S.C. § 271(b).

112. Netflix has also infringed, and continues to infringe, claims of the ’477 Patent by offering to commercially distribute, commercially distributing, making, and/or importing the Accused Instrumentalities, which are used in practicing the process, or using the systems, of the ’477 Patent, and constitute a material part of the invention. Netflix knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the ’477 Patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. Accordingly, Netflix has been, and currently is, contributorily infringing the ’477 Patent, in violation of 35 U.S.C. § 271(c).

113. By making, using, offering for sale, selling and/or importing into the United States the Accused Instrumentalities, and touting the benefits of using the Accused Instrumentalities' compression features, Netflix has injured Realtime and is liable to Realtime for infringement of the '477 Patent pursuant to 35 U.S.C. § 271.

114. As a result of Netflix's infringement of the '477 Patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Netflix's infringement, but in no event less than a reasonable royalty for the use made of the invention by Netflix, together with interest and costs as fixed by the Court.

## PRAYER FOR RELIEF

WHEREFORE, Plaintiff Realtime respectfully requests that this Court enter:

a. A judgment in favor of Plaintiff that Netflix has infringed, literally and/or under the doctrine of equivalents, the '046, '462, '535, '298, '907, and '477 Patents;

b. A judgment and order requiring Netflix to pay Plaintiff its damages, costs, expenses, and prejudgment and post-judgment interest for its infringement of the Patents-in-Suit, as provided under 35 U.S.C. § 284;

c. A judgment and order requiring Netflix to provide an accounting and to pay supplemental damages to Realtime, including without limitation, prejudgment and post-judgment interest;

d. A judgment and order finding that this is an exceptional case within the meaning of 35 U.S.C. § 285 and awarding to Plaintiff its reasonable attorneys' fees against Netflix; and

e. Any and all other relief as the Court may deem appropriate and just under the circumstances.

**DEMAND FOR JURY TRIAL**

Plaintiff, under Rule 38 of the Federal Rules of Civil Procedure, requests a trial by jury of any issues so triable by right.

November 21, 2017

BAYARD, P.A.

*/s/ Stephen B. Brauerman*

Stephen B. Brauerman (No. 4952)
Sara E. Bussiere (No. 5725)
222 Delaware Avenue, Suite 900
Wilmington, DE 19801
(302) 655-5000
sbrauerman@bayardlaw.com
sbussiere@bayardlaw.com

*Attorneys for Plaintiff*
*Realtime Adaptive Streaming LLC*

OF COUNSEL:

Mark A. Fenster
Reza Mirzaie
Brian D. Ledahl
C. Jay Chung
Philip X. Wang
Timothy T. Hsieh
RUSS, AUGUST & KABAT
12424 Wilshire Boulevard, 12th Floor
(310) 826-7474
Los Angeles, CA 90025-1031
mfenster@raklaw.com
rmirzaie@raklaw.com
bledahl@raklaw.com
jchung@raklaw.com
pwang@raklaw.com
thsieh@raklaw.com