RUSS, AUGUST & KABAT
Marc A. Fenster, SBN 181067
Email: mfenster@raklaw.com
Reza Mirzaie (CA SBN 246953)
Email: rmirzaie@raklaw.com
Brian D. Ledahl (CA SBN 186579)
Email: bledahl@raklaw.com
Paul Kroeger (CA SBN 229074)
Email: pkroeger@raklaw.com
C. Jay Chung (CA SBN 252794)
Email: jchung@raklaw.com
Philip X. Wang (CA SBN 262239)
Email: pwang@raklaw.com
12424 Wilshire Boulevard, 12th Floor
Los Angeles, California 90025
Telephone: (310) 826-7474
Facsimile: (310) 826-6991

Attorneys for Plaintiff
REALTIME ADAPTIVE STREAMING LLC

UNITED STATES DISTRICT COURT

CENTRAL DISTRICT OF CALIFORNIA

WESTERN DIVISION

REALTIME ADAPTIVE STREAMING LLC, Plaintiff, vs. GOOGLE LLC, and YOUTUBE, LLC, Defendants.	Case No. 2:18-cv-03629-GW-JC **JURY TRIAL DEMANDED**

<u>SECOND AMENDED COMPLAINT FOR PATENT INFRINGEMENT</u>

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 Google LLC and YouTube, LLC. (collectively, "Defendants" or "YouTube").

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. As recognition of its innovations rooted in this technological field, Realtime holds multiple United States patents and pending patent applications.

2. Defendant Google LLC is a Delaware limited liability company with its principal place of business in Mountain View, California.

3. Defendant YouTube, LLC is a Delaware limited liability company with its principal place of business in San Bruno, California. YouTube, LLC is a wholly owned subsidiary of Google LLC.

4. Google LLC and YouTube, LLC have regular and established places of business in this District, including, e.g., at 340 Main St., Venice, CA 90291. They offer their products and/or services, including those accused herein of infringement, to customers and potential customers located in California and in this District. Google LLC and YouTube, LLC may be served with process through its registered agent for service at The Corporation Service Company (CSC- Lawyers Incorporating Service), 2710 Gateway Oaks Drive, Suite 150N, Sacramento, California 95833.

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 they have committed acts within the Central District of California 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 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 are registered to do business in California, and they have transacted business in the Central District of California and have committed acts of direct and indirect infringement in the Central District of California. Defendants have regular and established place(s) of business in this District, as set forth above.

THE PATENTS-IN-SUIT

8. This action arises under 35 U.S.C. § 271 for Defendants' infringement of Realtime's United States Patent Nos. 7,386,046 (the "'046 patent"), 8,934,535 (the "'535 patent"), 9,769,477 (the "'477 patent"), RE46,777 (the "'777 patent"), and 9,578,298 (the "'298 patent").

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 '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 B. 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.

11. 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 C. 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.

12. The ’777 patent, titled “Quantization for Hybrid Video Coding,” was duly and properly issued by the USPTO on April 3, 2018. The ‘777 patent is a reissue of U.S. Pat. No. 8,634,462, which was issued on January 21, 2014. A copy of the ‘777 patent is attached hereto as Exhibit D. Realtime is the owner and assignee of the ‘777 patent and holds the right to sue for and recover all damages for infringement thereof, including past infringement.

13. 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 E. 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.

<u>COUNT I</u>

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

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

15. On information and belief, Defendants have made, used, offered for sale, sold and/or imported into the United States products that infringe the '046 patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Defendants' streaming products/services such as, e.g., YouTube site, YouTube TV, YouTube Live Streaming Platform, YouTube App, etc., and all versions and variations thereof since the issuance of the '046 patent ("Accused Instrumentalities").

16. On information and belief, Defendants have directly infringed and continue to infringe the '046 patent, for example, through their sale, offer for sale, importation, use and testing of the Accused Instrumentalities, which practices the system claimed by, e.g., Claim 40 of the '046 patent, namely, a system, comprising: a data compression system for compressing and decompressing data input; a plurality of compression routines selectively utilized by the data compression system, wherein a first one of the plurality of compression routines includes a first compression algorithm and a second one of the plurality of compression routines includes a second compression algorithm; and a controller for tracking throughput and generating a control signal to select a compression routine based on the throughput, wherein said tracking throughput comprises tracking a number of pending access requests to a storage device; and wherein when the controller determines that the throughput falls below a predetermined throughput threshold, the controller commands the data compression engine to use one of the plurality of compression routines to provide a

faster rate of compression so as to increase the throughput. Upon information and belief, Defendants use 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 Defendants' customers. For example, the Accused Instrumentalities utilize H.264 video compression standard in delivering live video via HTTP Live Streaming (HLS) technology. For example, YouTube "transcodes your content into lower bit rates, including HLS and iOS." *See, e.g.,* https://support.google.com/youtube/answer/6251900. Moreover, YouTube also decompresses live streams "on game consoles and mobile devices via the YouTube app and m.youtube.com. *See, e.g.,* https://support.google.com/youtube/answer/2853702?hl=en. Furthermore, according to HLS "protocol specification does not limit the encoder selection. However, the current Apple implementation should interoperate with encoders that produce MPEG-2 Transport Streams containing H.264 video and AAC audio (HE-AAC or AAC-LC)." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html. As another example, HLS developer guide also states: "HTTP Live Streaming supports switching between streams dynamically if the available bandwidth changes. The client software uses heuristics to determine appropriate times to switch between the alternates. Currently, these heuristics are based on recent trends in measured network throughput." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html.

As another example, YouTube specifies 'a plurality of compression routines selectively utilized by the data compression system" as follows:

Video codec:	H.264, 4.1 for up to 1080p 30 FPS
	H.264, 4.2 for 1080p 60 FPS
	H.264, 5.0 for 1440p 30 FPS
	H.264, 5.1 for 1440p 60 FPS
	H.264, 5.1 for 2160p 30 FPS
	H.264, 5.2 for 2160p 60 FPS

See, e.g., https://support.google.com/youtube/answer/2853702?hl=en

Moreover, YouTube "will automatically transcode your live stream to create many different output formats so all of your viewers on all of their devices and networks can watch!" *See, e.g.* https://support.google.com/youtube/answer/2853702?hl=en.

Furthermore, HLS developer guide states: "The current implementation of the client observes the effective bandwidth while playing a stream. If a higher-quality stream is available and the bandwidth appears sufficient to support it, the client switches to a higher quality. If a lower-quality stream is available and the current bandwidth appears insufficient to support the current stream, the client switches to a lower quality." *See,* *e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html.

17. The Accused Instrumentalities include a data compression system for compressing and decompressing data input. YouTube "transcodes your content into lower bit rates, including HLS and iOS." *See, e.g.,* https://support.google.com/youtube/answer/6251900. For example, YouTube's streaming products/services utilizes H.264 compression standard. As another example, YouTube's live streaming platform utilizes H.264 compression standard.

Video codec:	H.264, 4.1 for up to 1080p 30 FPS
	H.264, 4.2 for 1080p 60 FPS
	H.264, 5.0 for 1440p 30 FPS
	H.264, 5.1 for 1440p 60 FPS
	H.264, 5.1 for 2160p 30 FPS
	H.264, 5.2 for 2160p 60 FPS

See, e.g., https://support.google.com/youtube/answer/2853702?hl=en.

18. The Accused Instrumentalities include a plurality of compression routines selectively utilized by the data compression system, wherein a first one of the plurality of compression routines includes a first compression algorithm and a second one of the plurality of compression routines includes a second compression algorithm. For example, the Accused Instrumentalities utilize H.264, which include, e.g.,

Context-Adaptive Variable Length Coding ("CAVLC") entropy encoder and Context-Adaptive Binary Arithmetic Coding ("CABAC") entropy encoder. H.264 provides for multiple different ranges of parameters (e.g., bitrate, resolution parameters, etc.), each included in the "profiles" and "levels" defined by the H.264 standard. *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)

19. 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.

20. 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), a 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. *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).

21. After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can be organized in a GOP structure (see 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.

22. *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

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

> 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.

23. The Accused Instrumentalities includes a controller for tracking throughput and generating a control signal to select a compression routine based on the throughput, wherein said tracking throughput comprises tracking a number of pending access requests to a storage device, and a controller where, when the controller determines that the throughput falls below a predetermined throughput threshold, the controller commands the data compression engine to use one of the plurality of compression routines to provide a faster rate of compression so as to

increase the throughput. For example, YouTube “will automatically transcode your live stream to create many different output formats so all of your viewers on all of their devices and networks can watch!”

See, e.g. https://support.google.com/youtube/answer/2853702?hl=en.

In this regard, YouTube “transcodes your content into lower bit rates, including HLS and iOS.” *See, e.g.,* https://support.google.com/youtube/answer/6251900. As such, HLS (HTTP Live Streaming) “supports switching between streams dynamically if the available bandwidth changes. The client software uses heuristics to determine appropriate times to switch between the alternates. Currently, these heuristics are based on recent trends in measured network throughput.” *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html. “The current implementation of the client observes the effective bandwidth while playing a stream. If a higher-quality stream is available and the bandwidth appears sufficient to support it, the client switches to a higher quality. If a lower-quality stream is available and the current bandwidth appears insufficient to support the current stream, the client switches to a lower quality.” *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html. The controller in the Accused Instrumentalities decides which compression (e.g., CABAC, CAVLC, etc.) to use at a point in time based on parameters, for example, e.g., current or anticipated throughput. For example, when a low bandwidth is present, the Accused Instrumentalities select lower quality stream using a particular compression technique. As another example, when a high bandwidth is present, the Accused Instrumentalities select higher quality stream using another particular compression technique. For example, the Accused Instrumentalities' use of HTTP Live Streaming is directed to this selection. As another example, the Accused Instrumentalities' use of different "Profiles" of H.264 is directed to selecting lower quality stream using a particular compression technique (e.g., CABAC or CAVLC, etc.) for lower anticipated bandwidth situations, and selecting higher quality stream using a higher compression technique (e.g., CABAC or CAVLC, etc.) for higher anticipated bandwidth situations.

24. On information and belief, Defendants also directly infringe and continue to infringe other claims of the '046 patent.

25. 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.

26. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods and systems claimed by the '046 patent.

27. On information and belief, Defendants have had knowledge of the '046 patent since at least the filing of this Complaint or shortly thereafter, and on

information and belief, Defendants knew of the ‘046 patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Defendants 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.

28. Upon information and belief, Defendants’ 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. For example, Defendants adopted H.264 as its video codec in its YouTube TV, YouTube Live Streaming Platform, and YouTube App. As another example, YouTube on its help webpages states that “[A]n **encoder** compresses audio and video into a format that can be delivered to the YouTube platform. YouTube transcodes your content into lower bit rates, including HLS and iOS. This means your content will be available to as wide an audience as possible on desktop, mobile and tablet.”

See, e.g. https://support.google.com/youtube/answer/6251900. For similar reasons, Defendants also induce their customers to use the Accused Instrumentalities to infringe other claims of the ‘046 patent. Defendants specifically intended and was aware that these normal and customary activities would infringe the ‘046 patent.

Defendants 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. For example, since filing of this action, Defendants know that the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—in the Accused Instrumentalities infringes the patent but nevertheless continues to promote HTTP Live Streaming to customers. The only reasonable inference is that Defendants specifically intend the users to infringe the patent. On information and belief, Defendants engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Defendants have induced and continue 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, Defendants have been (as of filing of the original complaint), and currently are, inducing infringement of the '046 patent, in violation of 35 U.S.C. § 271(b).

29. Defendants have also infringed, and continue 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. Defendants know 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. For example, the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—infringes the patent, and as such, is especially adapted for use in infringement. Moreover, there is no substantial noninfringing use, as HTTP Live Streaming is directed to choosing different compression techniques based on current or anticipated throughput. Accordingly, Defendants have been (as of filing of the original complaint), and currently is, contributorily infringing the '046 patent, in violation of 35 U.S.C. § 271(c).

30. 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, Defendants have injured Realtime and is liable to Realtime for infringement of the '046 patent pursuant to 35 U.S.C. § 271.

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

COUNT II

INFRINGEMENT OF U.S. PATENT NO. 8,934,535

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

33. On information and belief, Defendants have made, used, offered for sale, sold and/or imported into the United States products that infringe the '535 patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, YouTube's streaming products/services such as, e.g., YouTube site, YouTube TV, YouTube Live Streaming Platform, YouTube App, etc., and all versions and variations thereof since the issuance of the '535 patent ("Accused Instrumentalities").

34. On information and belief, Defendants have 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 method claimed by Claim 1 of the '535 patent, namely, a method, comprising: determining a parameter or attribute of at least a portion of a data block having audio or video data; selecting an access profile from among a plurality of access profiles based upon the determined parameter or attribute; and compressing the at least the portion of the data block with one or more compressors using asymmetric data compression and information from the selected access profile to create one or more compressed data blocks, the information being indicative of the one or more compressors to apply to the at least the portion of the data block. Upon information and belief, Defendants use 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 Defendants' customers.

35. For example, the Accused Instrumentalities utilize H.264 video compression standard in delivering live video via HTTP Live Streaming (HLS) technology. For example, YouTube "transcodes your content into lower bit rates, including HLS and iOS." *See, e.g.,* https://support.google.com/youtube/answer/6251900. Furthermore, according to HLS "protocol specification does not limit the encoder selection. However, the current Apple implementation should interoperate with encoders that produce MPEG-2 Transport Streams containing H.264 video and AAC audio (HE-AAC or AAC-LC)." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html. As another example, HLS developer guide also states: "HTTP Live Streaming supports switching between streams dynamically if the available bandwidth changes. The client software uses heuristics to determine appropriate times to switch between the alternates. Currently, these heuristics are based on recent trends in measured network throughput." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.

html. Moreover, HLS developer guide states: "The current implementation of the client observes the effective bandwidth while playing a stream. If a higher-quality stream is available and the bandwidth appears sufficient to support it, the client switches to a higher quality. If a lower-quality stream is available and the current bandwidth appears insufficient to support the current stream, the client switches to a lower quality." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html.

36. The Accused Instrumentalities determine a parameter or attribute of at least a portion of a data block having audio or video data. For example, YouTube "automatically detects the stream resolution and frame rate. When you are live, we'll transcode to lower resolutions so all of your fans can enjoy your stream no matter the quality of their Internet connection." *See, e.g.* https://support.google.com/youtube/answer/2853700?hl=en&ref_topic=6136989 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" defined by the H.264 standard. *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)

37. 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.

38. The Accused Instrumentalities selects an access profile from among a plurality of access profiles based upon the determined parameter or attribute. For example, 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).

The Accused Instrumentalities compresses the at least the portion of the data block with one or more compressors using asymmetric data compression and information from the selected access profile to create one or more compressed data blocks, the information being indicative of the one or more compressors to apply to the at least the portion of the data block. The controller in the Accused Instrumentalities decides which compression (e.g., CABAC, CAVLC, etc.) to use at a point in time based on parameters, for example, e.g., current or anticipated throughput. For example, when a low bandwidth is present, the Accused Instrumentalities select lower quality stream using a particular compression technique. As another example, when a high bandwidth is present, the Accused Instrumentalities select higher quality stream using another particular compression technique. For example, the Accused Instrumentalities' use of HTTP Live Streaming is directed to this selection. As another example, the Accused Instrumentalities' use of different "Profiles" of H.264 is directed to selecting lower quality stream using a particular compression technique (e.g., CABAC or CAVLC, etc.) for lower anticipated bandwidth situations, and selecting higher quality stream using a higher compression technique (e.g., CABAC or CAVLC, etc.) for higher anticipated bandwidth situations.

39. Moreover, compression techniques utilized in H.264 standard are asymmetric.

Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.4, No.2, 2014

www.iiste.org

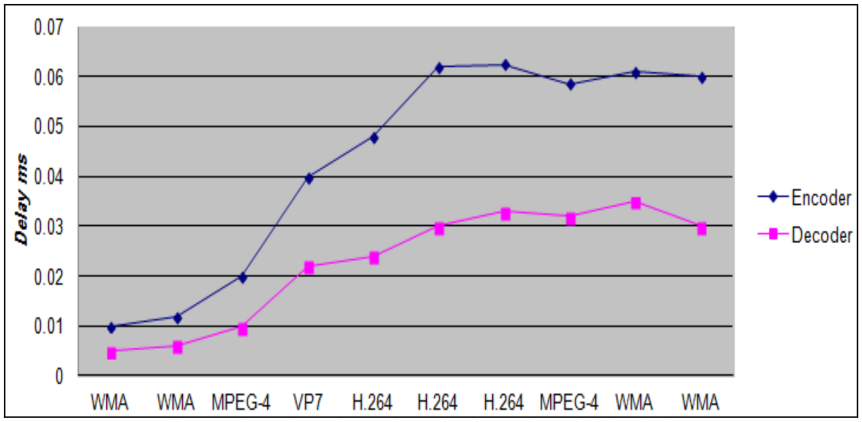

Figure 4: Encoder and Decoder Delay for Variety of Video Codec's

It is clear from figure 4, decoder delay in the most of video codec's can be assumed half of the encode delay.

See e.g., http://www.iiste.org/Journals/index.php/NCS/article/viewFile/11072/11373

40. YouTube "will automatically transcode your live stream to create many different output formats so all of your viewers on all of their devices and networks can watch!" *See, e.g.* https://support.google.com/youtube/answer/2853702?hl=en. In particular, 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. *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

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

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.

41. YouTube will "automatically archive the event up to 12 hours and make it available in the Video Manager." *See, e.g.* https://support.google.com/youtube/answer/2853700?hl=en&ref_topic=6136989

Thus, 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.

42. On information and belief, Defendants also directly infringe and continue to infringe other claims of the ‘535 patent.

43. 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.

44. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods and systems claimed by the ‘535 patent.

45. On information and belief, Defendants have had knowledge of the ‘535 patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Defendants knew of the ‘535 patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Defendants 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.

46. Upon information and belief, Defendants’ 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, Defendants adopted H.264 as its video codec in its YouTube TV, YouTube Live Streaming Platform, and YouTube App. As another example, YouTube on its help webpages states that "[A]n **encoder** compresses audio and video into a format that can be delivered to the YouTube platform. YouTube transcodes your content into lower bit rates, including HLS and iOS. This means your content will be available to as wide an audience as possible on desktop, mobile and tablet."

See, e.g. https://support.google.com/youtube/answer/6251900. For similar reasons, Defendants also induce their customers to use the Accused Instrumentalities to infringe other claims of the ‘535 patent. Defendants specifically intended and was aware that these normal and customary activities would infringe the ‘535 patent. Defendants 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. For example, since filing of this action, Defendants know that the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—in the Accused Instrumentalities infringes the patent but nevertheless continues to promote HTTP Live Streaming to customers. The only reasonable inference is that Defendants specifically intend the users to infringe the patent. On information and belief, Defendants engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Defendants have induced and continue 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, Defendants have been (as of filing of the original complaint), and currently is, inducing infringement of the '535 patent, in violation of 35 U.S.C. § 271(b).

47. Defendants have 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. Defendants know 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. For example, the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—infringes the patent, and as such, is especially adapted for use in infringement. Moreover, there is no substantial noninfringing use, as HTTP Live Streaming is directed to choosing different compression techniques based on current or anticipated throughput. Accordingly, Defendants have been (as of filing of the original complaint), and currently is, contributorily infringing the '535 patent, in violation of 35 U.S.C. § 271(c).

48. 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, Defendants have injured Realtime and is liable to Realtime for infringement of the '535 patent pursuant to 35 U.S.C. § 271.

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

COUNT III

INFRINGEMENT OF U.S. PATENT NO. 9,769,477

50. Plaintiff re-alleges and incorporates by reference the foregoing

paragraphs, as if fully set forth herein.

51. On information and belief, Defendants have made, used, offered for sale, sold and/or imported into the United States products that infringe the ‘477 patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, YouTube’s streaming products/services such as, e.g., YouTube site, YouTube TV, YouTube Live Streaming Platform, YouTube App, etc., and all versions and variations thereof since the issuance of the ‘477 patent (“Accused Instrumentalities”).

52. On information and belief, Defendants have directly infringed and continues to infringe the ‘477 patent, for example, through its sale, offer for sale, importation, use and testing of the Accused Instrumentalities that practice Claim 1 of the ‘477 patent, namely, a system, comprising: 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.

53. For example, the Accused Instrumentalities utilize H.264 video compression standard in delivering live video via HTTP Live Streaming (HLS) technology. For example, YouTube "transcodes your content into lower bit rates, including HLS and iOS." *See, e.g.,* https://support.google.com/youtube/answer/6251900. Furthermore, according to HLS "protocol specification does not limit the encoder selection. However, the current Apple implementation should interoperate with encoders that produce MPEG-2 Transport Streams containing H.264 video and AAC audio (HE-AAC or AAC-LC)." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html. As another example, HLS developer guide also states: "HTTP Live Streaming supports switching between streams dynamically if the available bandwidth changes. The client software uses heuristics to determine appropriate times to switch between the alternates. Currently, these heuristics are based on recent trends in measured network throughput." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conc

eptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html. Moreover, HLS developer guide states: "The current implementation of the client observes the effective bandwidth while playing a stream. If a higher-quality stream is available and the bandwidth appears sufficient to support it, the client switches to a higher quality. If a lower-quality stream is available and the current bandwidth appears insufficient to support the current stream, the client switches to a lower quality." *See, e.g.,* https://developer.Apple.com/library/content/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/FrequentlyAskedQuestions/FrequentlyAskedQuestions.html.

54. The Accused Instrumentalities include 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. H.264 provides for multiple different ranges of parameters (e.g., bitrate, max video bitrate, resolution parameters, etc.), each included in the "profiles" and "levels" defined by the H.264 standard. *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.

55. *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)

56. 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.

57. Moreover, compression algorithms utilized in H.264 standard are asymmetric.

Network and Complex Systems www.iiste.org
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.4, No.2, 2014

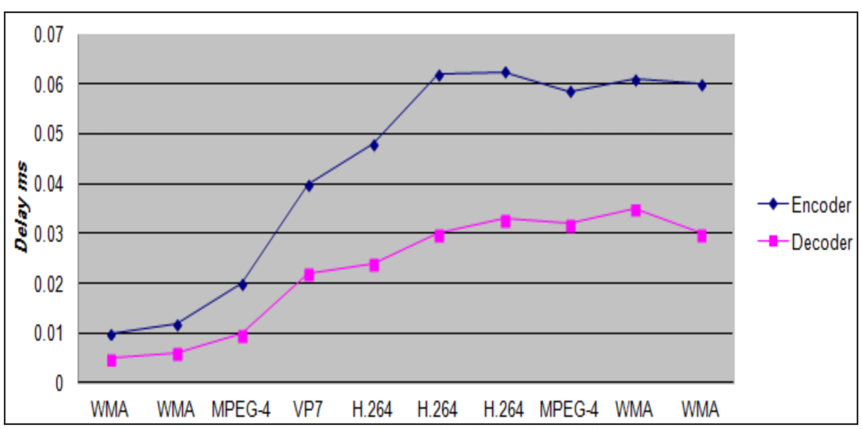

Figure 4: Encoder and Decoder Delay for Variety of Video Codec's

It is clear from figure 4, decoder delay in the most of video codec's can be assumed half of the encode delay.

See e.g., http://www.iiste.org/Journals/index.php/NCS/article/viewFile/11072/11373

58. The Accused Instrumentalities include 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, YouTube "automatically detects the stream resolution and frame rate. When you are live, we'll transcode to lower resolutions so all of your fans can enjoy your stream no matter the quality of their Internet connection." *See, e.g.* https://support.google.com/youtube/answer/2853700?hl=en&ref_topic=6136989 Moreover, 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 e.g., http://www.iiste.org/Journals/index.php/NCS/article/viewFile/11072/11373

Network and Complex Systems
ISSN 2224-610X (Paper) ISSN 2225-0603 (Online)
Vol.4, No.2, 2014
www.iiste.org

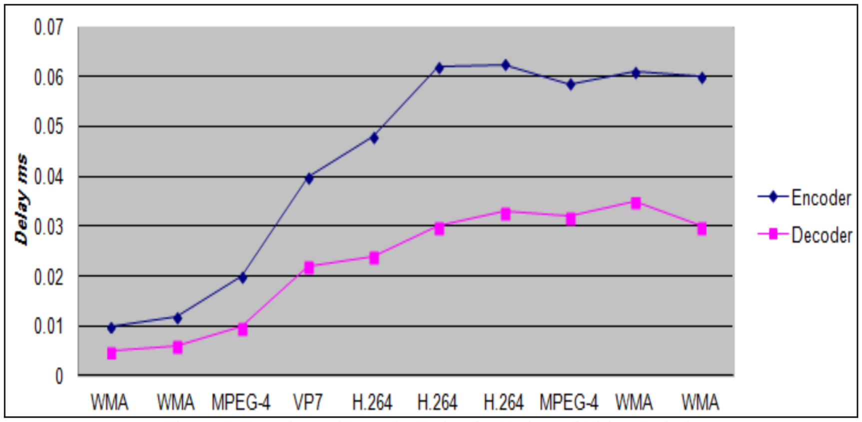

Figure 4: Encoder and Decoder Delay for Variety of Video Codec's

It is clear from figure 4, decoder delay in the most of video codec's can be assumed half of the encode delay.

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

59. 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).

60. The processor in the Accused Instrumentalities decides which compression (e.g., CABAC, CAVLC, etc.) to use at a point in time based on parameters, for example, e.g., current or anticipated throughput. For example, when a low bandwidth is present, the Accused Instrumentalities select lower quality stream using a particular compression technique. As another example, when a high

bandwidth is present, the Accused Instrumentalities select higher quality stream using another particular compression technique. For example, the Accused Instrumentalities' use of HTTP Live Streaming is directed to this selection. As another example, the Accused Instrumentalities' use of different "Profiles" of H.264 is directed to selecting lower quality stream using a particular compression technique (e.g., CABAC or CAVLC, etc.) for lower anticipated bandwidth situations, and selecting higher quality stream using a higher compression technique (e.g., CABAC or CAVLC, etc.) for higher anticipated bandwidth situations.

61. After its selection, the asymmetric compressor (CAVLC or CABAC) will compress the video data to provide various compressed data blocks, which can be organized in a GOP structure (see 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

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

> 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.

62. On information and belief, Defendants also directly infringe and continue to infringe other claims of the ’477 patent.

63. 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.

64. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods and systems claimed by the ‘477 patent.

65. On information and belief, Defendants have had knowledge of the ‘477

patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Defendants knew of the ’477 patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Defendants 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.

66. Upon information and belief, Defendants’ 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 using a system comprising: 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, Defendants adopted H.264 as its video codec in its YouTube TV, YouTube Live Streaming Platform, and YouTube App. As another example, YouTube on its help webpages states that "[A]n **encoder** compresses audio and video into a format that can be delivered to the YouTube platform. YouTube transcodes your content into lower bit rates, including HLS and iOS. This means your content will be available to as wide an audience as possible on desktop, mobile and tablet." *See, e.g.* https://support.google.com/youtube/answer/6251900. For similar reasons, Defendants also induce their customers to use the Accused Instrumentalities to infringe other claims of the '477 patent. Defendants specifically intended and was aware that these normal and customary activities would infringe the '477 patent. Defendants 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. For example, since filing of this action, Defendants know that the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—in the Accused Instrumentalities infringes the patent but nevertheless continues to promote HTTP

Live Streaming to customers. The only reasonable inference is that Defendants specifically intend the users to infringe the patent. On information and belief, Defendants engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Defendants have induced and continue 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, Defendants have been (as of filing of the original complaint), and currently are, inducing infringement of the ’477 patent, in violation of 35 U.S.C. § 271(b).

67. Defendants have 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. Defendants know 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. For example, the ordinary way of using HTTP Live Streaming—which is directed to choosing different compression techniques based on current or anticipated throughput—infringes the patent, and as such, is especially adapted for use in infringement. Moreover, there is no substantial noninfringing use, as HTTP Live Streaming is directed to choosing different compression techniques based on current

or anticipated throughput. Accordingly, Defendants have been (as of filing of the original complaint), and currently is, contributorily infringing the ’477 patent, in violation of 35 U.S.C. § 271(c).

68. 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, Defendants have injured Realtime and is liable to Realtime for infringement of the ’477 patent pursuant to 35 U.S.C. § 271.

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

COUNT IV

INFRINGEMENT OF U.S. PATENT NO. RE46,777

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

71. On information and belief, Defendants have made, used, offered for sale, sold and/or imported into the United States products that infringe the ‘777 patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Defendants’ products and services that implement the High

Efficiency Video Coding (HEVC; also known as H.265) standard (e.g., YouTube, Google Photos, Chromecast Ultra, Google Duo, etc.), and all versions and variations thereof since the issuance of the '777 patent ("Accused Instrumentalities").

72. On information and belief, Defendants have directly infringed and continues to infringe the '777 patent, for example, through its sale, offer for sale, importation, use and testing of the Accused Instrumentalities, which practices the method claimed by Claim 1 of the '777 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, Defendants use 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 Defendants' customers.

73. For example, the Accused Instrumentalities utilize the HEVC standard. *See, e.g.*, https://support.google.com/youtube/troubleshooter/2888402?hl=en&vid=0-315731740723-1524058922197 ("Supported YouTube file formats … HEVC (h265)"); https://plus.google.com/+PeggyKTC/posts/VTD6DM5Rxvx ("Google Photos now lets iOS11 users back up HEIF photos and HEVC videos"); https://developers.google.com/cast/docs/media#image-formats ("Supported Media for Google Cast … Video codecs (Chromecast Ultra) … HEVC / H.265"); https://wccftech.com/google-duo-v26-brings-support-h-265/ ("Google Duo V26 Brings Support for H.265").

74. The HEVC Specification (e.g., ITU-T H.265 Series H: Audiovisual and Multimedia Systems, "Infrastructure of audiovisual services – Coding of moving video" High efficiency video coding) ("HEVC Spec") sets forth standard that is followed by HEVC compliant devices, and is relevant to both decoding and encoding that are performed pursuant to the HEVC standard.

75. The Accused Instrumentalities performs a method for coding a video

signal using hybrid coding. For instance, Accused Instrumentalities performs a method for coding a video signal using hybrid coding when performing coding using HEVC. 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.”

76. The Accused Instrumentalities reduce temporal redundancy by block based motion compensated prediction in order to establish a prediction error signal. For instance, the Accused Instrumentalities reduce temporal redundancy by block based motion compensated prediction in order to establish a prediction error signal when performing HEVC encoding. 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, which indicate that block based motion compensation prediction is used in encoding to reduce temporal redundancy and establish a prediction error signal. *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.").

77. 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 instance, 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 when performing HEVC coding. 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, which indicate quantization as claimed when doing HEVC encoding. *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.").

78. The Accused Instrumentalities perform a method wherein the prediction error signal includes a plurality of subblocks each including a plurality of quantized values. For instance, the Accused Instrumentalities perform a method wherein the prediction error signal includes a plurality of subblocks each including a plurality of quantized values when performing HEVC encoding. 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, indicating prediction error signal as claimed when doing HEVC encoding. *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.”).

79. 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 instance, 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 when performing HEVC

encoding. 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."

80. 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.

81. The infringement of the Accused Instrumentalities is also shown by way of considering the reference software (*see, e.g.,* https://hevc.hhi.fraunhofer.de/). The Accused Instrumentalities implement HEVC encoding in the same or substantially the same manner as the reference software, i.e., in infringing manner. 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.

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

```
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 ] );
}
```

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

83. 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/).

84. Comparing the two efficiencies d64BestCost and totalCost:

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

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

85. Selecting for further proceeding either quantized values, which are all set to zero or quantized values, which are not all set to zero:

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

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

86. On information and belief, Defendants also directly infringe and continue to infringe other claims of the '777 patent.

87. 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 standard.

88. On information and belief, use of the Accused Instrumentalities in their ordinary and customary fashion results in infringement of the methods claimed by the '777 patent.

89. On information and belief, Defendants have had knowledge of the '777 patent since at least the filing of this Complaint or shortly thereafter, and on

information and belief, Defendants knew of the '777 patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Defendants 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 '777 patent.

90. Upon information and belief, Defendants' 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 '777. For example, Defendants adopted HEVC as its video codec in its products and services. For similar reasons, Defendants also induce customers to use the Accused Instrumentalities to infringe other claims of the '777 patent. Defendants specifically intended and was aware that these normal and customary activities would infringe the '777 patent. Defendants performed the acts that constitute induced infringement, and would induce actual infringement, with the knowledge of the '777 patent and with the knowledge, or willful blindness to the probability, that the induced acts would constitute infringement. For example, since filing of this action, Defendants know that the ordinary way of using HEVC in the Accused Instrumentalities infringes the patent but nevertheless continues to promote HEVC to customers. The only reasonable

inference is that Defendants specifically intends the users to infringe the patent. On information and belief, Defendants engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Defendants have induced and continue to induce users of the Accused Instrumentalities to use the Accused Instrumentalities in their ordinary and customary way to infringe the '777 patent, knowing that such use constitutes infringement of the '777 patent. Accordingly, Defendants have been (as of filing of the original complaint), and currently are, inducing infringement of the '777 patent, in violation of 35 U.S.C. § 271(b).

91. Defendants have also infringed, and continues to infringe, claims of the '777 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 '777 patent, and constitute a material part of the invention. Defendants knows the components in the Accused Instrumentalities to be especially made or especially adapted for use in infringement of the '777 patent, not a staple article, and not a commodity of commerce suitable for substantial noninfringing use. For example, the ordinary way of using HEVC infringes the patent, and as such, is especially adapted for use in infringement with no substantial noninfringing use. Accordingly, Defendants have been (as of filing of the original complaint), and currently are, contributorily infringing the '777 patent, in violation of 35 U.S.C. § 271(c).

92. 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, Defendants have injured Realtime and is liable to Realtime for infringement of the '777 patent pursuant to 35 U.S.C. § 271.

93. As a result of Defendants' infringement of the '777 patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Defendants' infringement, but in no event less than a reasonable royalty for the use made of the invention by Defendants, together with interest and costs as fixed by the Court.

COUNT V

INFRINGEMENT OF U.S. PATENT NO. 9,578,298

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

95. On information and belief, Defendants have made, used, offered for sale, sold and/or imported into the United States products that infringe the '298 patent, and continues to do so. By way of illustrative example, these infringing products include, without limitation, Defendants' products and services that implement the High Efficiency Video Coding (HEVC; also known as H.265) standard (YouTube, Google Photos, Chromecast Ultra, Google Duo, Android 5.0+, etc.), and all versions and variations thereof since the issuance of the '298 patent ("Accused Instrumentalities").

96. On information and belief, Defendants have directly infringed and

continues to infringe the ‘298 patent, for example, through its sale, offer for sale, importation, use and testing of the Accused Instrumentalities, which practices the method 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, Defendants use 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 Defendants’ customers.

97. For example, the Accused Instrumentalities utilize the HEVC standard. *See, e.g.,* https://support.google.com/youtube/troubleshooter/2888402?hl=en&vid=0-

315731740723-1524058922197 ("Supported YouTube file formats … HEVC (h265)"); https://plus.google.com/+PeggyKTC/posts/VTD6DM5Rxvx ("Google Photos now lets iOS11 users back up HEIF photos and HEVC videos"); https://developers.google.com/cast/docs/media#image-formats ("Supported Media for Google Cast … Video codecs (Chromecast Ultra) … HEVC / H.265"); https://wccftech.com/google-duo-v26-brings-support-h-265/ ("Google Duo V26 Brings Support for H.265"); https://developer.android.com/guide/topics/media/media-formats.html ("Video formats and codecs … H.265 HEVC … (Android 5.0+)").

98. 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, in the Accused Instrumentalities, 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. Annex C, Annex D (Supplemental enhancement information), and Annex E (Video usability

information) are each "an integral part of the Recommendation," i.e., the HEVC Spec. HEVC Spec at Annex C, D, E. The Accused Instrumentalities implement SEI, VUI, and other items as specified in the HEVC Spec, and as stated herein.

99. The Accused Instrumentalities generate an output video stream which can be reproduced on a visualization apparatus. For example, in the Accused Instrumentalities, 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.").

100. 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, in the Accused Instrumentalities, 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.").

101. 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, in the Accused Instrumentalities, 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.

102. The Accused Instrumentalities decode only that part of the composite frame which contains said one image to be displayed. For example, in the Accused Instrumentalities, 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.

103. The Accused Instrumentalities generate an output frame containing said extracted image. For example, in the Accused Instrumentalities, 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.").

104. On information and belief, Defendants also directly infringes and continues to infringe other claims of the '298 patent.

105. 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 standard.

106. 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.

107. On information and belief, Defendants have had knowledge of the '298 patent since at least the filing of this Complaint or shortly thereafter, and on information and belief, Defendants knew of the '298 patent and knew of its infringement, including by way of this lawsuit. By the time of trial, Defendants 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.

108. Upon information and belief, Defendants' 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. For example, Defendants adopted HEVC as its video codec in its products and services. For similar reasons, Defendants also induces its customers to use the Accused Instrumentalities to infringe other claims of the '298 patent. Defendants specifically intended and was aware that these normal and customary activities would infringe the '298 patent. Defendants 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. For example, since filing of this action, Defendants know that the ordinary way of using HEVC in the Accused Instrumentalities infringes the patent but nevertheless continues to promote HEVC to customers. The only reasonable inference is that Defendants specifically intend the users to infringe the patent. On information and belief, Defendants engaged in such inducement to promote the sales of the Accused Instrumentalities. Accordingly, Defendants have 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, Defendants have been (as of filing of the original complaint), and currently is, inducing infringement of the '298 patent, in violation of 35 U.S.C. § 271(b).

109. Defendants have also infringed, and continue 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. Defendants know 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. For example, the ordinary way of using HEVC infringes the patent, and as such, is especially adapted for use in infringement with no substantial noninfringing use. Accordingly, Defendants have been (as of filing of the original complaint), and currently are, contributorily infringing the '298 patent, in violation of 35 U.S.C. § 271(c).

110. 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, Defendants have injured Realtime and is liable to Realtime for infringement of the '298 patent pursuant to 35 U.S.C. § 271.

111. As a result of Defendants' infringement of the '298 patent, Plaintiff Realtime is entitled to monetary damages in an amount adequate to compensate for Defendants' infringement, but in no event less than a reasonable royalty for the use

made of the invention by Defendants, 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 Defendants have infringed, literally and/or under the doctrine of equivalents the ’046, ’535, ’477, ‘777, and ‘298 patents (the “asserted patents”);

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

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

d. A permanent injunction prohibiting Defendants from further acts of infringement of the asserted patents;

e. 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 YouTube; and

f. 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.

Respectfully Submitted,

Dated: November 8, 2018 /s/ Marc A. Fenster

RUSS AUGUST & KABAT
Marc A. Fenster, SBN 181067
Email: mfenster@raklaw.com
Reza Mirzaie (CA SBN 246953)
Email: rmirzaie@raklaw.com
Brian D. Ledahl (CA SBN 186579)
Email: bledahl@raklaw.com
Paul Kroeger (CA SBN 229074)
Email: pkroeger@raklaw.com
C. Jay Chung (CA SBN 252794)
Email: jchung@raklaw.com
Philip X. Wang (CA SBN 262239)
Email: pwang@raklaw.com

Attorneys for Plaintiff
REALTIME ADAPTIVE STREAMING LLC

CERTIFICATE OF SERVICE

I hereby certify that the foregoing document was served on all counsel of record via electronic service on November 8, 2018.

/s/ Marc A. Fenster