

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PELOTON INTERACTIVE, INC.,
Petitioner,

v.

NEC CORPORATION,
Patent Owner.

Case IPR2023-01239
U.S. Patent No. 8,752,101

PETITIONER'S SUR-SUR-REPLY TO PATENT OWNER'S SUR-REPLY

TABLE OF CONTENTS

EXHIBIT LIST	ii
I. The Petition Demonstrated That Chou Satisfies Limitation 1(D).....	1
II. Peloton’s Reply Does Not Advance a “New Theory”	2
III. NEC’s “Encoder Parameters” Theory of Chou Fails for Other Reasons.....	3
A. Chou Uses “Encoder Parameters” <i>During Startup</i> , But Determines Code Rate Based on Decoder Status Thereafter.....	3
B. Chou Does Not Use “ $g^{\text{new}}(n)$ ” to Determine Code Rate	4
C. Chou’s Encoder <i>and</i> Decoder Both Simulate Leaky Bucket Models, and Their Buffers Each Have Their Own “Gaps”	4
CERTIFICATE OF SERVICE	7

EXHIBIT LIST

Exhibit No.	Description
1001	U.S. Patent No. 8,752,101 B2 (“the ’101 patent”)
1002	File History of U.S. Patent No. 8,752,101 B2 (U.S. Application No. 13/508,025) (“’101 FH”)
1003	Lee, Chiang, and Zhang, <i>Scalable Rate Control For MPEG-4 Video</i> , IEEE Transactions on Circuits and Systems for Video Technology, Vol. 10, No. 6, pp. 878 – 894 (Sept. 2000)
1004	U.S. Publication No. US 2006/0143678 A1 (“Chou”)
1005	File History of corresponding European Patent No. 2498492
1006	SEALED Revised Dr. Clifford Reader Declaration (“Reader Decl.”)
1007	International Publication No. WO 2006/108435 (“Kampmann”)
1008	Japanese Unexamined Patent Application Publication No. 2005-191786
1009	Japanese Unexamined Patent Application Publication No. 2007-13705
1010	U.S. Patent No. 8,909,809 (the ’809 patent”)
1011	Japanese Unexamined Patent Application Publication No. 2004-172830
1012	Certified English Translation of Japanese Unexamined Patent Application Publication No. 2004-172830
1013	Certified English Translation of Japanese Unexamined Patent Application Publication No. 2005-191786
1014	Certified English Translation of Japanese Unexamined Patent Application Publication No. 2007-13705
1015	Kozamernik, <i>Webcasting - The Broadcasters’ Perspective</i> , EBU Technical Review (Mar. 2000) (“Kozamernik”)
1016	U.S. Patent No. 7,606,928 (“Wang”)
1017	U.S. Patent No. US 6,637,031 B1 (“Chou ’031”)
1018	Cen, Pu, and Walpole, <i>Flow and Congestion Control for Internet Streaming Applications</i> , Multimedia Computing and Networking (1998)
1019	Lu, <i>Signal Processing for Internet Video Streaming: A Review</i> , Proceedings of SPIE, Vol. 3947, p. 246-259 (Jan. 2000) (“Lu”)
1020	Haskell, Puri, and Netravali, <i>Digital Video: An Introduction to MPEG-2</i> , ch. 8, pp. 156-182 (“Haskell”)
1021	ITU H.262 (07/95)

Exhibit No.	Description
1022	Ebrahimi, <i>MPEG-4 natural video coding – An overview</i> , Signal Processing: Image Communication, Vol. 15, Issue 4-5, pp. 365-385 (Jan. 2000) (“Ebrahimi”)
1023	ITU H.261 (03/93)
1024	ITU H.222.0 (07/95)
1025	Rui, <i>An Overview of MPEG-2</i> , The 1997 Digital Video Test Symposium
1026	Huang, Chou, Klemets, <i>Optimal Coding Rate Control for Scalable Streaming Media</i> (“Huang”)
1027	U.S. Publication No. US 2004/0128396 A1 (“Patrick”)
1028	U.S. Patent No. US 7,711,841 B2 (“Desphande”)
1029	van Beek and Demircin, <i>Delay-Constrained Rate Adaptation for Robust Video Transmission Over Home Networks</i> , IEEE International Conference, Vol. 2, (“van Beek”)
1030	U.S. Patent No. 5,822,524 (“Chen”)
1031	SEALED NEC’s Infringement Contentions served in <i>NEC Corp. v. Peloton Interactive, Inc.</i> , No. 1:22-cv-00987-CFC (D. Del., filed July 28, 2022)
1032	Reader C., <i>Intraframe and Interframe Adaptive Transform Coding</i> , SPIE Vol. 66 (1975)
1033	Reader C, “MPEG4: Coding for Content, Interactivity, and Universal Accessibility”, <i>Optical Engineering</i> , Vol. 35, No. 1, pp. 104-108 (Jan. 1996)
1034	“MPEG Video Compression Standard”, Edited by J. L. Mitchell et al, Chapter 16, “MPEG Patents”, pp.357-362
1035	Proposal and Consideration of Buffer Size, NTT, KDD, NEC, Fujitsu, CCITT SG IV Working Party XV/1 Specialist Group on coding for Visual Telephony, Doc. #287 (Jan. 1988)
1036	Scheduling Order filed on March 20, 2023 in <i>NEC Corp. v. Peloton Interactive, Inc.</i> , No. 1:22-cv-00987-CFC (D. Del., filed July 28, 2022)
1037	REDACTED PUBLIC version of revised Reader Declaration
1038	U.S. Patent No. 5,867,230 (“Wang ’230”)
1039	Declaration of Michael J. Hopkins (Oct. 31, 2023)
1040	Supplemental Declaration of Dr. Clifford Reader
1041	Jeremy Cooperstock Deposition Transcript (Aug. 22, 2024)
1042	Jeremy Cooperstock Deposition Transcript (Oct. 11, 2024)

Contrary to NEC's assertion, Peloton's Reply does not advance new theories. It is NEC's sur-reply that does so. Further, NEC arguments not only fail, but reflect fundamental misunderstandings of Chou's disclosures.

I. The Petition Demonstrated That Chou Satisfies Limitation 1(D)

NEC admits that its argument that Chou fails to disclose Limitation 1(d) hinges on construing "'based on' to mean '*using as an input.*'" Sur-reply (SR) 2, 5. But that narrowing construction fails. Reply, 2-5; D.I. 41. Peloton showed that Chou's system dynamically adjusts code rate to control client buffer duration using feedback from the client to the server. Chou, ¶¶19, 81. Under the plain meaning of Limitation 1(d), this entails determining code rate *based on* "available time"—i.e., client buffer duration, which is the difference between playback deadline t_d and arrival time t_a . Reply, 6-9; Pet. 34-39. NEC dismisses Chou's disclosure in ¶19 as referring to *prior* systems, and maintains Chou teaches away from Limitation 1(d), SR 8-9, but that ignores Chou's express disclosure that "client buffer management is a *key element* affecting the performance," and that "*the present system* and process [entails] *controlling the size of the client buffer* over time." Chou, ¶20.

Peloton further demonstrated Chou also determines code rate based on "remaining time." Reply, 18-20; Pet. 37. NEC does not dispute Peloton's identified "remaining time," but instead proffers an overly narrow construction to argue that Chou's code rate determination is not *based on* remaining time. See SR 21-23.

II. Peloton’s Reply Does Not Advance a “New Theory”

While the Petition sufficiently demonstrated Limitation 1(d), Peloton’s Reply *also* showed that the gap “ $g(n-1)$ ” in Fig. 4—which Chou says “represents the size of the *decoder* buffer” (Chou, ¶60)—factors into Eq. 18. Reply, 11-16; Ex.1041, 100:18-102:21 (agreeing that ¶60 of Chou denotes decoder buffer size); *see also* Ex. 1017, 8:6-57, Fig. 4 (showing bits in encoder and decoder buffers, accounting for transmission delay). Chou further states that Eq. 18 (contrary to NEC’s suggestion) is not the ultimate code rate algorithm. To determine code rate, Chou’s client accounts for “network jitter, delays, and throughput changes” by adding the “playback deadline” and “target schedule”—i.e., “margin of safety”—to Eq. 18. Chou, ¶¶27, 70, 72. Even Dr. Cooperstock admitted that the playback deadline and target schedule are tracked *by the client*, not the server. Ex.1042, 80:24-81:6, 86:19-25. NEC itself agrees “a POSITA would recognize [such] real-world networking conditions (i.e., network latency, etc.).” SR 15. Yet such conditions could not possibly be accounted for by server-side “encoder parameters” alone; they surface at the *decoder (client-side)*, *after transmission*. Thus, Peloton did not offer a “new theory” as NEC suggests. SR 6-7, 11-13. Instead, its arguments are consistent with the Petition that Chou’s client buffer duration is the “available time.” Pet. 34-37 (citing Chou, ¶67). Peloton’s explanation of $g(n-1)$ was in response to NEC’s argument that Eq. 18 does not consider client buffer state, and is simply additional

proof that Chou determines code rate based on “available time.”

NEC’s other arguments lack merit. Its argument that Eq. 18 does not consider client buffer fullness, F^d , SR 16-17, fails because, Chou considers the data in the client buffer. Likewise, its argument that Peloton “proposed modification[s]” to meet limitation 1(d), SR 7, 11, is also wrong because Chou makes clear its determination process adds a “margin of safety” beyond Eq. 18. And while NEC complains that “ $g(n-1)$ is in bit and not time,” SR 16, Dr. Cooperstock agrees that “the translation between time and bits is simply a matter of dividing by rate.” Ex.1042, 74:11-12.

III. NEC’s “Encoder Parameters” Theory of Chou Fails for Other Reasons

NEC’s argument that Chou determines code rate using only “encoder buffer parameters...and not any decoder client buffer parameters,” SR 14, fails for other reasons, and reflects a broader misunderstanding of Chou’s disclosure.

A. Chou Uses “Encoder Parameters” *During Startup*, But Determines Code Rate Based on Decoder Status Thereafter

NEC fails to grasp that Chou’s usage of “encoder buffer parameters” to determine the code rate, Chou, ¶25, applies to code rate determination only during the *startup period*. *Id.*, ¶21 (“Initially, the servers chooses the coding bit rate during a startup period.”). “However, *after the startup period*, the client provides coding bit rate requests to the server” so that the “decoder buffer of the client [is] filled to a desired duration level”—i.e., based on decoder buffer conditions. *Id.* The client runs a “linear quadratic optical control technique” to update the code rate. *Id.*, ¶¶19, 22.

This is consistent with Chou’s disclosures of feedback, *id.*, ¶¶69, 81, and its goal of “controlling the size of the client buffer over time,” *id.*, ¶20. Thus, NEC’s treatment of Chou’s startup operating mode as its *only* mode of determining code rate is wrong.

B. Chou Does Not Use “ $g^{new}(n)$ ” to Determine Code Rate

NEC and Dr. Cooperstock cite fact that Chou’s server transmits a gap value “ $g^{new}(n)$ ” to the client, Chou, ¶112, to argue that the decoder calculates code rate based on only encoder parameters. Ex.2016, ¶8. They are mistaken. Chou’s client determines the new code rate and communicates it to the server along with a maximum gap shift value, “ $\Delta^{max} g(n)$ ”. Chou, ¶¶97-100. If the new code rate is higher than the current one, the encoder determines whether the system can transition to the higher rate without causing underflow at the client, using $\Delta^{max} g(n)$ supplied by the client. *Id.*, ¶¶97-100, Fig. 10. The server then communicates the gap $g^{new}(n)$ and gap shift value $\Delta g(n-1)$ to the client, so the client can update the target schedule. Chou, ¶¶110, 112. Thus, $g^{new}(n)$ is employed by the server to transition to the new code rate *after* the code rate has been determined by the client. Ex.1042, 86:2-17 (admitting $g^{new}(n)$ is not used to update the code rate).

C. Chou’s Encoder *and* Decoder Both Simulate Leaky Bucket Models, and Their Buffers Each Have Their Own “Gaps”

Dr. Cooperstock’s assertion that Chou’s disclosure of various gaps “ $g(n)$ ” all pertain to the *encoder buffer*, Ex.2016, ¶8, is incorrect. His assertion stems from a mischaracterization of Chou’s ¶61 statement that “[a] leaky bucket model is a

metaphor for the encoder buffer.” Ex.2016, ¶¶8, 13. But Chou makes clear that *the decoder also simulates a leaky bucket model*, and the decoder buffer has its own set of gap values. Chou, ¶126 (“The client continues to update $g(n)$ for the new coding bit rate *by running its own leaky bucket simulator*.”). Chou’s encoder simulates its leaky bucket and conveys gap $g(n)$ to the client when the encoder switches to a new code rate. Chou, ¶113. Chou makes clear when a particular gap refers the status of the buffer at the decoder versus the encoder. For example, the gap “ $g(n-1)$ ” in Fig. 4, cited in the Reply, “represents the size of the *decoder buffer*.” Chou, ¶60; *see also id.*, ¶66. Elsewhere, Chou mentions that “the *sender* computes gap $g(n)$ ” for each virtual stream, and on a stream switch, “gap $g^{\text{new}}(n)$...is transmitted to the client.” Chou, ¶110; *see also id.*, ¶112. But gap $g^{\text{new}}(n)$, shown in Fig. 10, is a *different gap* from gap $g(n-1)$ in Fig. 4. Further, ¶69, cited by NEC, refers to §V.E. (¶113), which clarifies that it is $g(n)$ that is sent from server to client. In his declaration, Dr. Cooperstock improperly lumped *all* of Chou’s gaps together as “encoder-size variables.” Ex.2016, ¶8. But he later struggled to explain Fig. 10, admitting he had not studied that figure, *see* Ex.1042, 57:10-67:22, and he ultimately conceded that gaps in Fig. 4 and Fig. 10 are two “different things.” *Id.*, 69:2-70:12.

Date: October 25, 2024

Respectfully submitted,

By /Scott C. Weidenfeller/

Scott C. Weidenfeller

Registration No.: 54,531

Anupam Sharma

Registration No.: 55,609

Counsel for Petitioner

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6, I hereby certify that the foregoing Petitioner's Sur-Sur-Reply to Patent Owner's Sur-Reply was served by electronic mail today on the following counsel of record for Patent Owner:

Eliot Williams (eliot.williams@bakerbotts.com; dlbbnec@bakerbotts.com)
Robert Maier (robert.maier@bakerbotts.com)
Jennifer Tempesta (jennifer.tempesta@bakerbotts.com)
Michael Knierim (michael.knierim@bakerbotts.com)
Nicholas Palmieri (nicholas.palmieri@bakerbotts.com)
BAKER BOTTS L.L.P.

And with courtesy copy to:

Darren Franklin (dfranklin@sheppardmullin.com)
Daniel Yannuzzi (dyannuzzi@sheppardmullin.com)
Mark W. Catanese (mcatanese@sheppardmullin.com)
SHEPPARD, MULLIN, RICHTER & HAMPTON

Date: October 25, 2024

/Scott C. Weidenfeller/
Scott C. Weidenfeller
Registration No.: 54,531