

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PELOTON INTERACTIVE, INC.,
Petitioner,

v.

NEC CORPORATION,
Patent Owner.

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

**SUPPLEMENTAL DECLARATION OF DR. JEREMY COOPERSTOCK IN
SUPPORT OF PATENT OWNER'S SUR-REPLY**

TABLE OF CONTENTS

I.	Introduction.....	3
II.	Chou does not render obvious limitation 1(d) of the '101 Patent.....	4
A.	Chou does not disclose, teach, or suggest determining the code rate based on “available time.”.....	4
1.	The “gap $g(n-1)$ ” does not represent the number of bits in the decoder buffer for frame (n-1).....	6
2.	A POSITA would not modify Chou as Dr. Reader suggests.....	13
3.	A POSITA would not modify Equations 18-19 of Chou as Dr. Reader suggests	14
B.	Chou does not disclose, teach, or suggest determining the code rate based on “remaining time.”.....	15

I, Jeremy Cooperstock, Ph.D. declare as follows:

1. I make this Declaration based upon my own personal knowledge and, if called upon to testify, would testify competently to the matters contained herein.
2. I understand that Peloton Interactive, Inc. (“Petitioner” or “Peloton”) submitted a petition (“Petition”) for *inter partes* review, challenging Claims 1, 2, 4, 5, 7, 11, 12, 14-18, 20-23, and 25-29 (the “Challenged Claims”) of U.S. Patent No. 8,752,101 (the “’101 Patent”). I have been asked by counsel for Patent Owner, NEC Corporation (“Patent Owner” or “NEC”) to provide technical opinions in connection with the *inter partes* review.
3. I submit this Supplemental Declaration in support of Patent Owner’s Sur-Reply to address the new arguments presented by Petitioner in its Petitioner’s Reply and its associated supplemental declaration from Dr. Reader.

I. INTRODUCTION

4. I fully incorporate by reference in this supplemental declaration the analysis and opinions contained in my previous declaration (Ex. 2012) in this matter, dated May 31, 2024. I have reviewed Petitioner’s Reply, as well as the supporting supplemental declaration of Dr. Reader (Ex. 1040). I have been asked to provide this supplemental declaration in order to respond to the newly articulated arguments made by Petitioner and Dr. Reader regarding the Chou reference (“Chou”) (Ex. 1004) in the context of claim limitation 1(d) of the ’101 Patent (Ex. 1001).

II. CHOU DOES NOT RENDER OBVIOUS LIMITATION 1(D) OF THE '101 PATENT

A. Chou does not disclose, teach, or suggest determining the code rate based on “available time.”

5. At paragraph 5, Dr. Reader re-alleges that the “client buffer duration ($t_d(n) - t_a(n)$) corresponds to the “available time” and [that] Chou’s technique to determine code rate is based on the client buffer duration.” Ex. 1040, ¶ 5. However, Dr. Reader does not point to anything in Chou that supports his position or contradicts my explanation that the “client buffer duration ($t_d(n) - t_a(n)$)” is (1) not “based on the content data stored in the storage device,” as the claim recites and (2) not used in the process of determining the code rate in Chou.

6. Dr. Reader then relies on Equations 10-19 and Figures 2-4, to allege that “Equation 18 in Chou can be used to calculate the coding rate $r_c(n)$ for frame (n) based on the size of the decoder buffer (gap $g(n-1)$ either in bits or seconds).” Ex. 1040, ¶¶ 7-13. For the reasons below, I disagree with Dr. Reader’s analysis. Moreover, Dr. Reader takes the position that since the “playback deadline” and “target schedule” are introduced after Chou’s discussion of Equation 18, that “a POSITA would have understood that when the Equation 18 accounts for the ‘playback deadline’ and ‘target schedule’ of Figure 5, it would factor in the client buffer size according to this model.” Ex. 1040, ¶ 14. I disagree. Dr. Reader cites to paragraph 67 of Chou for allegedly supporting this statement, but paragraph 67 of

Chou only discusses the division of the y axis of Figure 4 by the code rate arrival rate r_a . Ex. 1004, ¶ 67. Chou does not here (or elsewhere) disclose, teach, or suggest that the “gap $g(n-1)$ ” (or gap $g(n)$) in Equation 18 or 19 corresponds to “client buffer duration $(t_d(n)-t_a(n))$ ” or that the “client buffer duration $(t_d(n)-t_a(n))$ ” is based on the “gap $g(n-1)$.” Instead, the gap “ $g(n)$ ” (or $g(n-1)$) and the “client buffer duration $(t_d(n)-t_a(n))$ ” are entirely different quantities, despite Chou referring to both of these quantities as “gaps”. Cf. Ex. 1004, Fig. 4 with Fig. 5. Thus, in my opinion, Equation 18 (or any other equation) does not “account” for the “playback deadline” and “target schedule” as Dr. Reader alleges in attempt to reconcile the new theory including the gap $g(n-1)$ with the previous theory that alleged the “client buffer duration $(t_d(n)-t_a(n))$ ” corresponds to the claimed “available time. See Ex. 1040, ¶ 14.

7. In summary, Dr. Reader alleges that the Chou discloses this limitation because (1) “gap $g(n-1)$ represents the number of bits in the decoder buffer for frame $(n-1)$ (Ex. 1040, ¶¶ 7-12); (2) that “a POSITA would be able to easily convert the buffer size in bits into the corresponding playback duration” (*Id.*, ¶ 13); and (3) that, at a later time, Equation 18 “accounts for” the “playback deadline” (*Id.*, ¶¶ 14) despite Equation 18 not including the “playback deadline” variable $(t_d(n))$ or the target schedule variable $(t_T(n))$. However, each of these assumptions is flawed for the reasons discussed below.

1. The “gap $g(n-1)$ ” does not represent the number of bits in the decoder buffer for frame $(n-1)$.

a) The “gap $g(n-1)$ ” and the “gap $g(n)$ ” are parameters of the encoder buffer (server side), not the decoder buffer (client side).

8. Chou discloses the “gap $g(n-1)$ ” and “gap $g(n)$ ” are *encoder* side variables, not *decoder* side variables. Ex. 1004, ¶¶ 61 (“A leaky bucket is a metaphor **for the encoder buffer**”), 69 (“Using Eq. (16), the **client can compute $t_b(n-1)$ from ... $g(n-1)$** (which can **be transmitted to the client** along with the data in frame $n-1$ or computed at the client as described in Section V-E).”); 113 (“Whenever a new coding bit rate starts, the **client receives the value $g(n)$ along with the new frame.**”). Chou’s “gap $g(n-1)$ ” is an “encoder side upper bound gap” that must be transmitted to or inferred by the client. Ex. 1004, ¶¶ 24, 25, 66 (“the gap $g(n)$ at frame n between the buffer tube upper bound and the coding schedule”), 69, 110 (“the server performs on-line leaky bucket simulations for each stream ... the sender computes $g(n)$... the **gap $g^{new}(n)$ for the new stream is transmitted to the client...**”), 112 (“If the sender finds a suitable coding bit rate and makes a switch at frame n , it **will transmit three values to the client** along with the frame: the new coding bit rate $r_c^{new}(n)$, **the current gap to the upper bound $g^{new}(n)$**”), 113 (“Whenever a new coding bit rate starts, the **client receives the value $g(n)$** along with the new frame. The values of $g(n)$ for successive frames can be then inferred by the client itself based on the coding bit rate $r_c(n)$ and the frame size $b(n)$ ”), 117

(“The client then employs the encoder buffer parameters and the **upper bound gap value provided by the server**”).

9. Chou expressly defines the upper bound gap $g(n)$ stating “[t]his upper bound gap is defined as the number of data bits that the server's encoder buffer can contain over the bits currently contained therein after a just-generated frame is fully input into the buffer.” Ex. 1004, ¶¶ 24, 66 (demonstrating connection between claimed “upper bound gap” and $g(n)$), 112 (same). Accordingly, $g(n-1)$ does not represent the number of bits in the decoder buffer; rather, – it is the number of bits that the encoder has available to be filled. Ex. 1004, ¶ 24.

10. Additionally, a value of “ $g(n)$ ” for the decoder buffer is never determined or transmitted from the client to the server, which affirms that $g(n)$ or $g(n-1)$ do not carry any information regarding the decoder buffer. Instead, Chou teaches only that $g(n-1)$ and $g(n)$ for the encoder buffer are determined and transmitted to the client (or the client can infer or estimate the encoder buffers $g(n)$). Ex. 1004, ¶¶ 24, 25, 66, 69, 110, 112, 113, 117, 126, 127.

11. Chou expressly teaches that “**encoder buffer parameters and upper bound gap**” are what are used to “**determine** on a frame-by-frame basis **the coding bit rate**.” Ex. 1004, ¶ 24. In contrast, the “available reproduction time” in the claims of the ’101 Patent is a time “based on the content data stored in the **storage device of the reception device**.” Ex. 1001. Cl 1.

b) The “gap $g(n-1)$ ” and the “gap $g(n)$ ” do not represent the “content data stored in the storage device of the reception device,” as claimed.

12. Dr. Reader ignores these teachings of Chou regarding the gap $g(n)$ and relies on Figures 3-4, and paragraphs 60 and 66 of Chou for his allegation that “gap $g(n-1)$ represents data in the decoder buffer for frame $(n-1)$.” Ex. 1040, ¶ 12. However, Dr. Reader leaves out important context to reach his conclusion. Particularly, at paragraph 66, Chou expressly states that the client buffer fullness is represented by the variable “ F^d ,” and not “ $g(n-1)$ ” or “ $g(n)$ ” alone. Ex. 1004, ¶ 66. Dr. Reader attempts to reconcile this by stating that “the decoder buffer fullness $F^d(n)$ is the sum of $b(n)$, i.e., “bits that comprise the frame” and $g(n)$, “bits that were present in the buffer $g(n)$ when frame (n) arrived.” Ex. 1040, ¶12. But, as an initial matter, $g(n)$ is not a buffer, thus it appears that Dr. Reader intended to say that “bits [$g(n)$] that were present in the buffer [] when frame (n) arrived.” Yet, as explained further below, that is a flawed understanding of Chou – $g(n)$ is not a variable that represents the size of the decoder buffer. Moreover, Dr. Reader also states in that same paragraph that “gap $g(n)$ represents data in the decoder buffer after frame (n) is *decoded* at the decoder.” *Id.* Thus, it is not clear if Dr. Reader is alleging that $g(n)$ is the bits present in the buffer after frame (n) has “arrived” (i.e., put into the buffer) or after frame (n) has been decoded (i.e., removed from the buffer), but both are incorrect for the reasons described below. Regardless, Dr. Reader’s conclusory allegation that “ $g(n)$ ”

corresponds to the bits that were present in the decoder buffer is false.

13. Chou teaches that the variable B is the total size of the encoder buffer, which can be represented by the size of the buffer tube. Ex. 1004, ¶¶ 60, 61 (“A leaky bucket is a metaphor for the encoder buffer.”), 62 (“A leaky bucket with size B , rate R , and initial fullness F^e ”). Chou derives the equations that “represent” $F^d(n)$ in paragraph 66 from the assumption that the encoder buffer and decoder buffer are “complementary,” meaning that Chou is making the assumption that at any given moment, the fullness of the encoder buffer F^e in addition to the fullness of the decoder buffer F^d is equal to the total size of the encoder buffer B . Ex. 1004, ¶¶ 60 66 (“ $F^d(n) = B - F^e(n)$ ”). Chou is only able to model the buffer fullness in this way by expressly stating that it is applying the assumptions, for purposes of explaining Figure 4, that the “leak rate R ” is equal to the average coding rate “ r_c ,” and that the frames are “instantaneously” encoded and decoded into and out of the encoder buffer and decoder buffer, respectively. *Id.* ¶ 60. Based on these assumptions, Chou arrives at equation (10). Ex. 1004, ¶¶ 65-66. In this way, Chou is stating that, if as soon as a frame is (instantaneously) loaded at the encoder, a POSITA may assume for purposes of discussion that “the number of data bits that the server's encoder buffer can contain over the bits currently contained therein” immediately appears in the decoder buffer thereby making the buffers “complementary.” Ex. 1004, ¶ 24. However, a POSITA would understand that “instantaneous” encoding and decoding

is inconsistent with the reality that these processes must take (non-zero) time to be executed.

14. In other words, Chou is asking the reader to ignore the realities of network behavior for purposes of explanation that the number of bits available to be loaded in (*i.e.*, the remaining capacity of) *the encoder buffer* (gap $g(n)$) in addition to the bits that have been added (frame size $b(n)$) to the leaky bucket can be used as a proxy for the decoder buffer fullness. *See* Ex. 1004, ¶ 65. That is, given this hypothetical, and the assumption that the fullness of the encoder buffer (F^e) added to the fullness of the decoder buffer (F^d) will be equal to the total amount of bits (B), then a POSITA can assume that $B - F^e$ is the fullness of the decoder buffer (F^d).

15. In reality, a POSITA would understand the sum of the buffer fullness of the encoder and the buffer fullness of the decoder would not be equal to the total size of the “complimentary” sized encoder (B). That is, a POSITA would understand that equation (10) would not be an accurate proxy for the decoder buffer fullness. In most cases, the sum of the buffer fullness may be larger or smaller than B because the actual number of bits in both buffers would depend on the arrival rate (which is going to directly depend on the “leak rate R ” and the network latency), the instantaneous coding rates of content, the playback speed, and other factors occurring at the client (*i.e.*, a pause, skip forward or back in playback, etc.). In other words, a POSITA would understand F^d could not be represented by the sum of $g(n-$

1) + $r_c(n)/f(n)$ as explained further below.

16. *First*, a POSITA would recognize that leak rate (R) and code rate (r_c) will vary, and at any particular time, will not necessarily be equal, as Chou assumes for equation (10). Moreover, a POSITA would recognize that the average code rate (r_c) is an average and the underlying content will have code rates that are sometimes greater or smaller than the leak rate, which means that equation (10) at any instant would likely not provide an accurate fullness of the buffer. Moreover, Chou itself expressly recognizes many scenarios where the leak rate (R) is not equal to the average code rate (r_c). Ex. 1004, ¶¶ 64 (“Leak rates R greater than r_c can also be employed.”), 90 (“The present system and process employs an alternative fast startup technique ... by choosing an initial coding bit rate r_c equal to half the arrival rate r_a .”). A POSITA would understand that equation (10) does not hold because depending on the leak rate (R) and the code rate (r_c) there will be a build-up of bits at either the encoder or decoder buffer, which means that the available number of bits to be dumped into the encoder (the gap $g(n)$) will not accurately represent a “fullness” of the decoder. In support of this conclusion, I explain in the following paragraph that Chou is teaching a POSITA that equation (10) is not an accurate representation of the buffer fullness in practice.

17. *Second*, Chou discloses that $g(n)$ is transmitted to the client when there is a change in code rates. Ex. 1004, ¶¶ 24, 25, 66, 69, 110, 112, 113, 117. Chou discloses

a technique for the “client” to “estimate” the gap $g(n)$ of the encoder buffer when “the server is unable to simulate the leaky buckets and cannot send $g^{new}(n)$ to the client.” Ex. 1004, ¶¶ 126-128. Accordingly, a POSITA would understand that if $g(n)$ was simply related to the “decoder buffer fullness,” Chou’s server would **not** need to transmit $g(n)$ to the client because the client would be able to ascertain its own fullness and thereby determine a value for $g(n)$. Thus, the fact that Chou’s server must send $g(n)$ demonstrates that there is no connection between $g(n)$ and the “decoder buffer fullness.” Moreover, a POSITA would understand that Chou’s simulation for estimating $g(n)$ (Ex. 1004, ¶¶ 126-128) based ***on the fullness of the encoder buffer*** would not be needed when the server is unable to transmit $g(n)$. Accordingly, a POSITA would understand that Chou is teaching that $g(n)$ is independent of the status of the decoder buffer including the fullness of the decoder buffer.

c) Chou does not disclose, teach, or suggest that the “decoder buffer fullness” F^d is used to determine the code rate.

18. Importantly, Chou fails to disclose or suggest that F^d is used in determining the code rate, which means that a POSITA would not be motivated to make necessary modifications to then use the “decoder buffer fullness” (in bits) to determine the “available time” and further use the available time to determine the code rate, as the claims require. Accordingly, Dr. Reader’s theory of “ $g(n-1)$ ” is

incorrect.

19. Moreover, Dr. Reader criticizes my table demonstrating the “available time.” Ex. 1040, ¶ 29. However, his criticisms there are based on the same assumptions that he made in this section. Accordingly, I disagree with Dr. Reader’s criticisms that my shown “available time” in the table is incorrect for at least the reasons described above.

2. A POSITA would not modify Chou as Dr. Reader suggests

20. As stated above, the fullness of the buffer F^d is never used in any of the equations. That is, F^d , that Chou discloses to be the decoder buffer fullness (in bits) is never used in the process of determining the code rate. Moreover, as also explained above, $g(n-1)$ is independent of the decoder fullness and therefore does not correspond to the “client buffer duration” (Pet., 39-40) or even the “the content data stored in the storage device of the client device” as required by the claim for the available time. Ex. 1001, Cl. 1.

21. Moreover, as indicated above, Dr. Reader admits that $g(n-1)$ is **in bits** and **not time**, which Dr. Reader states that “a POSITA would be able to easily convert” into time “providing the coding rate r_c is known.” Ex. 1040, ¶13. But Dr. Reader fails to reason that the “coding rate r_c ” for each frame in the buffer would need to be known in order to calculate this value. That is, the decoder buffer may have two or more frames that are each encoded at a different coding rate. The buffer duration is not

simply the total amount of bits divided by a coding rate, but instead a sum of the size of each frame in the decoder buffer divided by its respective coding rate. However, Chou does not suggest that $g(n-1)$ would convey such information for past frames in the buffer at least because it does not reflect the size of the decoder buffer for the reasons described above. Instead, as explained above, Chou only discloses sending $g(n-1)$ **from the server to the client**. Thus, Dr. Reader fails to explain (1) why a POSITA would modify Chou to have **the client** send $g(n-1)$ **to the server**; and (2) include information regarding the coding rate for each frame in the decoder buffer to ascertain the time available for reproducing content. A POSITA would not be motivated to do either of these modifications. As explained above, Chou teaches that the encoder buffer parameters control the coding rate and $g(n)$ is sent from the server to the client (meaning no information is transmitted to the server regarding the decoder buffer). Thus, in my opinion, even if a POSITA *could* modify Chou to monitor the frames in the buffer and their respective coding rates to determine a client buffer duration, the POSITA would not be motivated to because Chou never suggests that such a value is of importance.

3. *A POSITA would not modify Equations 18-19 of Chou as Dr. Reader suggests*

22. Dr. Reader's conclusion that "a POSITA would have understood that when the Equation 18 accounts for the 'playback deadline' and 'target schedule' of Figure 5, it would factor in the client buffer size according to this model" is not supported

by Chou itself. Dr. Reader cites to paragraph 67 of Chou for this conclusion, but that paragraph merely discusses another hypothetical to describe Figure 5. Particularly, Chou here is teaching that the network connection is no longer “isochronous” and asks the reader to “[a]ssume for the moment that bits arrive at the client at a constant rate r_a .” Ex. 1004, ¶67. Based on this new assumption, Chou teaches that the y-axis of Figure 4 of Chou can be divided by r_a to reach Figure 5. Ex. 1004, ¶67. Nowhere does Chou suggest here, or anywhere, that Equations 18 or 19 “factor in” or should be modified to “factor in” the buffer fullness. To the contrary, as explained above, Chou teaches determining the code rate based on the encoder buffer parameters. Ex. 1004, ¶¶ 24, 25, 69, 110, 113, 117. Accordingly, I disagree with Dr. Reader – a POSITA would not have understood that Equation (18) accounts for the playback deadline and target schedule. To the contrary, Equation (18) does not include any terms that could be considered analogous with the playback deadline and target schedule and a POSITA would not be motivated, or know how, to modify Equation 18 to “account” for the playback deadline ($t_d(n)$ for frame n) or the target schedule ($t_r(n)$ for frame n).

B. Chou does not disclose, teach, or suggest determining the code rate based on “remaining time.”

23. Dr. Reader acknowledges that the gap defined by “ $t_d(n) - t_a(n-2)$ ” would be what corresponds to the claimed remaining time. However, Chou does not disclose that this is ever used to determine the code rate.

24. Dr. Reader brings forth three assumptions to allege that the gap defined by “ $t_d(n) - t_a(n-2)$ ” of Chou is used to determine the code rate. Particularly, Dr. Reader states that (1) “the upper bound $t_b(n)$ is based on arrival time”¹ (Ex. 1040, ¶ 22); (2) that Equation 18 “determines coding rate $r_c(n)$ for frame (n) when frame (n-2) has arrived at time $t_T(n)$,” (*Id.*, ¶ 23), and (3) Chou derives the target schedule $t_T(n)$ from the playback deadline (*Id.*, ¶ 24). However, these three arguments, together, do not teach or suggest determining a code rate based on the gap defined by “ $t_d(n) - t_a(n-2)$.”

25. *First*, Dr. Reader’s analysis does not explain how “ $t_a(n-2)$ ” or, alternatively, his newly defined variable of $t_{present}(n)$ is used by Equation 18. Dr. Reader states that “When Chou attempts to ascertain a value for the upper bound $t_b(n)$ from the present moment ($t_{present}$), Chou considers the time duration from the present moment ($t_{present}$) to the upper bound $t_b(n)$.” Ex. 1040, ¶ 23. Dr. Reader does not provide any support for his allegation. I disagree. In using Equation (18), any comparison between the upper bound $t_b(n)$ and the arrival time $t_a(n)$ is done for the same frame (n). That is, nowhere does Equation 18 suggest comparing the upper bound $t_b(n)$ with the arrival time two frames earlier $t_a(n-2)$. Dr. Reader fails to account for the temporal difference between the arrival time of frame (n-2) and the beginning of playback for

¹ Dr. Reader appears to mistakenly list the “arrival time” variable as “ $t_b(n)$,” but it should correctly be cited as “ $t_a(n)$.”

frame (n).

26. Dr. Reader may be implying that the fact that the calculation is *occurring* at time $t_a(n-2)$ is sufficient to meet the claims. In my opinion, it is not. A POSITA would not understand that a time the determining occurs means that the determining is “based on” that time. Instead, a POSITA would understand that a time an event occurs does not mean that the time is considered or that the event is based on the time. In contrast, a POSITA would understand that in order for the determining to be based on the “remaining time,” a duration of time between $t_d(n)$ and $t_a(n-2)$ must be used as an input, which Chou does not disclose, teach, or suggest.

27. *Second*, Dr. Reader’s reliance on Equation (20) (reproduced below) confirms Chou does not consider the arrival time of frame (n-2), $t_a(n-2)$, and the playback deadline of frame (n), $t_d(n)$. Ex. 1040, ¶22.

$$I = \sum_{n=0}^N \left((t_b(n) - t_T(n))^2 + \sigma \left(\frac{r_c(n+1) - r_c(n)}{\tilde{r}_\sigma(n)} \right)^2 \right), \quad (20)$$

Both the upper bound $t_b(n)$ and target schedule $t_T(n)$ in this equation correspond to the same frame (n), which does not address the issue that the “remaining time” corresponds to the arrival time of frame (n-2) and not frame (n). That is, even assuming Dr. Reader’s assumptions are correct, then Equation (20) does not account for the time shift in the variables, meaning that the remaining time is not used by

Chou, either directly or indirectly, in determining the code rate as Dr. Reader suggests.

28. The claim itself defines the “remaining time” as a duration of time between the reproduction start time and the present moment, but Dr. Reader’s theory fails to show that a duration denoted by “ $t_d(n) - t_d(n-2)$ ” is used in the determination of the code rate of future content data. Accordingly, a POSITA would understand that Chou fails to disclose, teach, or suggest determining the code rate using a “remaining time,” as claim 1 of the ’101 Patent recites.

I hereby declare under penalty of perjury under the laws of the United States of America that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine, imprisonment, or both, under 18 U.S.C. § 1001. I attest to having a final and complete copy of this Declaration before me at the time I add my signature.


Jeremy Cooperstock, Ph.D.

October 3, 2024
Date